МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО АТОМНОЙ ЭНЕРГИИ

РОССИЙСКАЯ АКАДЕМИЯ НАУК

РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

ИНСТИТУТ ОПТИКО-НЕЙРОННЫХ ТЕХНОЛОГИЙ РАН

НАУЧНАЯ СЕССИЯ МИФИ–2007

НЕЙРОИНФОРМАТИКА–2007

**IX ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ

Часть 1

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2007

УДК 001(06)+004.032.26 (06) Нейронные сети
ББК 72я5+32.818я5
М82

НАУЧНАЯ СЕССИЯ МИФИ–2007. VIII ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2007»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. Часть 1. – М.: МИФИ, 2007. – 178 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 24–26 января 2007 года в МИФИ в рамках IX Всероссийской конференции «Нейроинформатика–2007».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор
Ю. В. Тюменцев, кандидат технических наук

ISBN 5–7262–0708–4

Содержание

ПРЕДИСЛОВИЕ

1. В этой книге содержатся тексты ряда лекций, прочитанных на Седьмой Школе-семинаре «Современные проблемы нейроинформатики», проходившей 24–26 января 2007 года в МИФИ в рамках IX Всероссийской научно-технической конференции «Нейроинформатика–2007» (они включены в Часть 2 данного сборника), а также тексты трех лекций, представленных на Шестой Школе-семинаре, не вошедшие по ряду причин в состав сборника [9] (они включены в Часть 1).

При отборе и подготовке материалов для лекций авторы и редактор следовали принципам и подходам, сложившимся при проведении шести предыдущих Школ (см. [1–9]). А именно, основной целью Школы было, как всегда, рассказать слушателям о современном состоянии и перспективах развития важнейших направлений в теории и практике нейроинформатики, о ее применениях.

Основной задачей лекторов, приглашаемых из числа ведущих специалистов в области нейроинформатики и ее приложений, смежных областей науки, было дать живую картину современного состояния исследований и разработок, обрисовать перспективы развития нейроинформатики в ее взаимодействии с другими областями науки.

2. В Части 1 данного сборника публикуются тексты трех лекций из программы Шестой Школы-семинара, не вошедшие ранее в состав сборника [9]:

1. *А. А. Ежов.* Сознание, рефлексия и многоагентные системы.
2. *Н. Г. Макаренко.* Стохастическая динамика, марковские модели и прогноз.
3. *В. Кецман.* Новый SVM-алгоритм для сверхбольших наборов данных.

3. В программу Седьмой Школы-семинара «Современные проблемы нейроинформатики» на конференции «Нейроинформатика–2007» вошли следующие шесть лекций:

1. *В. Я. Сергин.* Биологически правдоподобная модель зрительного восприятия: Иерархия объемлющих сенсорных характеристик.
2. *А. А. Фролов.* Что такое интерфейс мозг-компьютер.
3. *С. А. Терехов.* Гениальные комитеты умных машин.

УДК 001(06)+004.032.26 (06) Нейронные сети

4. *Ю. Р. Цой.* Введение в нейроэволюционный подход: Основные концепции и приложения.
5. *Н. Г. Макаренко.* Топология изображений.
6. *В. Л. Введенский.* Построение смыслового пространства языка человека.

Четыре из перечисленных выше шести лекций публикуются в данном сборнике (в Части 2), две лекции (*А. А. Фролова* и *Н. Г. Макаренко*) в силу технических причин будут опубликованы в сборнике лекций следующей Школы-семинара.

4. В лекции **А. А. Ежова** «Сознание, рефлексия и многоагентные системы» предпринимается попытка нащупать пути решения проблемы сознания в рамках концепции многоагентных систем. Достаточно распространенной является точка зрения, согласно которой базой для решения данной проблемы должна служить квантовая механика. В лекции показано, что необходимо также использование аппарата статистической физики.

5. Лекция **Н. Г. Макаренко** «Стохастическая динамика, марковские модели и прогноз» продолжает серию выступлений автора, призванных обратить внимание нейроинформационного сообщества на ряд разделов математики, перспективных с точки зрения использования их при решении задач нейроинформатики. В данной лекции представлен новый метод марковского предсказания временных рядов, основанный на инвариантной мере случайной динамики, реализованной через сжимающие отображения, снабженные вероятностями (систему итеративных функций).

6. В лекции **В. Кецмана** «Новый SVM-алгоритм для сверхбольших наборов данных» рассматривается случай работы со сверхбольшими наборами данных (порядка нескольких миллионов обучающих пар). Дается сравнение нейронных сетей и машин опорных векторов с точки зрения решения задач классификации (распознавания образов) и регрессии (аппроксимации функций). Для решения задач рассматриваемого класса предлагается новый итерационный алгоритм, получивший наименование ISDA (Iterative Single Data Algorithm), основывающийся на последовательном использовании обучающих пар данных из имеющегося обучающего набора.

7. В лекции **В. Я. Сергина** «Биологически правдоподобная модель зрительного восприятия: Иерархия объемлющих сенсорных характеристик» вводится понятие объемлющей характеристики. Она представляет собой ответ данного перцептивного уровня на сенсорные признаки уровня, лежащего ниже, такие, что их специфическое сочетание составляет адаптивно

значимую целостность. Последовательность таких характеристик образует иерархию, от сенсорных признаков до целостных образов и сцен.

8. Лекция **С. А. Терехов** «Гениальные комитеты умных машин» посвящена актуальной проблеме повышения точности обучения машин путем объединения их в комитеты. Основная цель ее состоит в обсуждении алгоритмов для обработки очень больших наборов обучающих данных, пригодных к использованию на современных и перспективных многоядерных компьютерах.

9. В лекции **Ю. Р. Цоя** «Введение в нейроэволюционный подход: Основные концепции и приложения» дается обзор исследований, находящихся на стыке искусственных нейронных сетей и эволюционных вычислений. Проанализированы преимущества и недостатки нейроэволюционного подхода при решении задач эволюционной настройки весов и структуры нейросетей. Рассматривается применение нейроэволюционных алгоритмов при решении задач адаптивного управления, адаптивного поведения, многоагентных систем, эволюционной робототехники, поиска игровых стратегий и компьютерного творчества.

10. Лекция **В. Л. Введенского** «Построение смыслового пространства языка человека» посвящена изложен нетрадиционного подхода к проблеме представления человеческого языка в форме, приемлемой для компьютера. Объектом исследования является множество словарей разных языков. Было обнаружено, что устройство лексиконов разных языков подчиняется строгим математическим закономерностям, опираясь на которые, можно установить способ представления языка в мозге человека. Полученные результаты могут быть полезны для разработки систем речевого контакта с компьютером.

* * *

Для того, чтобы продолжить изучение вопросов, затронутых в лекциях, можно порекомендовать такой уникальный источник научных и научно-технических публикаций, как цифровая библиотека ResearchIndex (ее называют также CiteSeer, см. позицию [10] в списке литературы в конце предисловия). Эта библиотека, созданная и развиваемая отделением фирмы NEC в США, содержит уже около 800 тыс. публикаций, причем это число постоянно и быстро увеличивается за счет круглосуточной работы поисковой машины.

Каждый из хранимых источников (статьи, препринты, отчеты, диссертации и т. п.) доступен в полном объеме в нескольких форматах (PDF,

PostScript, DjVu и др.) и сопровождается очень подробным библиографическим описанием, включающим, помимо данных традиционного характера (авторы, заглавие, место публикации и/или хранения и др.), также и большое число ссылок-ассоциаций, позволяющих перейти из текущего библиографического описания к другим публикациям, «похожим» по теме на текущую просматриваемую работу. Это обстоятельство, в сочетании с весьма эффективным полнотекстовым поиском в базе документов по сформулированному пользователем поисковому запросу, делает библиотеку ResearchIndex незаменимым средством подбора материалов по требуемой теме.

Помимо библиотеки ResearchIndex, можно рекомендовать также богатый электронный архив публикаций [11], недавно открывшийся поисковый сервис Google Scholar [12], а также портал научных вычислений [13].

Перечень проблем нейроинформатики и смежных с ней областей, требующих привлечения внимания специалистов из нейросетевого и родственных с ним сообществ, далеко не исчерпывается, конечно, вопросами, рассмотренными в предлагаемом сборнике, а также в сборниках [1–9].

В дальнейшем предполагается расширение данного списка за счет рассмотрения насущных проблем собственно нейроинформатики, проблем «пограничного» характера, особенно относящихся к взаимодействию нейросетевой парадигмы с другими парадигмами, развиваемыми в рамках концепции мягких вычислений, проблем использования методов и средств нейроинформатики для решения различных классов прикладных задач. Не будут забыты и взаимодействия нейроинформатики с такими важнейшими ее «соседями», как нейробиология, нелинейная динамика, численный анализ и т. п.

Замечания, пожелания и предложения по содержанию и форме лекций, перечню рассматриваемых тем и т. п. просьба направлять электронной почтой по адресу tium@mai.ru *Тюменцеву Юрию Владимировичу*.

Литература

1. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // III Всероссийская научно-техническая конференция «Нейроинформатика-2001», 23–26 января 2001 г. / Отв. ред. *Ю. В. Тюменцев*. – М.: Изд-во МИФИ, 2001. – 212 с.

2. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 1. – М.: Изд-во МИФИ, 2002. – 164 с.

3. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 2. – М.: Изд-во МИФИ, 2002. – 172 с.

4. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 1. – М.: Изд-во МИФИ, 2003. – 188 с.

5. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 2. – М.: Изд-во МИФИ, 2003. – 180 с.

6. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VI Всероссийская научно-техническая конференция «Нейроинформатика-2004», 28–30 января 2004 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 1. – М.: Изд-во МИФИ, 2004. – 200 с.

7. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VI Всероссийская научно-техническая конференция «Нейроинформатика-2004», 28–30 января 2004 г. / Отв. ред. *Ю. В. Тюменцев*. Часть 2. – М.: Изд-во МИФИ, 2004. – 200 с.

8. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VII Всероссийская научно-техническая конференция «Нейроинформатика-2005», 26–28 января 2005 г. / Отв. ред. *Ю. В. Тюменцев*. – М.: Изд-во МИФИ, 2005. – 216 с.

9. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VIII Всероссийская научно-техническая конференция «Нейроинформатика-2006», 24–27 января 2006 г. / Отв. ред. *Ю. В. Тюменцев*. – М.: Изд-во МИФИ, 2006. – 244 с.

10. NEC Research Institute CiteSeer (also known as ResearchIndex) — Scientific Literature Digital Library.
URL: http://citeseer.ist.psu.edu/cs

11. The Archive arXiv.org e-Print archive — Physics, Mathematics, Nonlinear Sciences, Computer Science.
URL: http://arxiv.org/

12. Google Scholar.
URL: http://scholar.google.com/

13. Портал научных вычислений (Matlab, Fortran, C++ и т. п.)
URL: http://www.mathtools.net/

Редактор материалов выпуска,
кандидат технических наук *Ю. В. Тюменцев*
E-mail: tium@mai.ru

А. А. ЕЖОВ
Троицкий институт инновационных и термоядерных исследований,
г. Троицк, Московская область
E-mail: ezhov@triniti.ru

СОЗНАНИЕ, РЕФЛЕКСИЯ И МНОГОАГЕНТНЫЕ СИСТЕМЫ

Аннотация

Данная лекция, прочитанная на Школе-семинаре во время конференции «Нейроинформатика–2006», является результатом безуспешной попытки подготовить лекцию по многоагентным системам для студентов Экономико-аналитического института МИФИ и имеет отношение к одному из вопросов, обсуждавшихся тогда на Круглом столе по проблемам сознания, а именно: *«Можно ли объяснить сознание в рамках классической физики?»* В ней мы пытаемся показать, что изучение сознания может потребовать знания статистической физики (а не только квантовой механики, как принято считать физиками). Лекция может рассматриваться как расширенный комментарий к статье *A. A. Ezhov and A. Yu. Khrennikov*. Agents with left and right brain dominant hemispheres and quantum statistics // *Physical Review*, **E71**, 016138, 2005.

A. A. EZHOV
Troitsk Institute of Innovation and Fusion Research,
Troitsk, the Moscow Region
E-mail: ezhov@triniti.ru

CONSCIOUSNESS, REFLECTION AND MULTIAGENT SYSTEMS

Abstract

This Lecture given for the Tutorial Session at the Neuroinformatics-2006 Conference represents an unsuccessful attempt to prepare a lecture about multiagent systems for students of Economics and Analytics Institute at the Moscow Engineering and Physical Institute (MEPhI). The Lecture concerns one of questions discussed on the Round Table discussion held at the Conference and devoted to the consciousness problem. The question was formulated as: “May we explain the consciousness within the framework of the classical physics?” We try to demonstrate that statistical physics knowledge can be needed to study of the consciousness (not quantum mechanics only as physicists think usually). The Lecture can be considered as some extended comment for the article: *A. A. Ezhov and A. Yu. Khrennikov*. Agents with left and right brain dominant hemispheres and quantum statistics // *Physical Review*, **E71**, 016138, 2005.

УДК 001(06)+004.032.26 (06) Нейронные сети

Введение

Потерял сознание, очнулся —
Гиббс . . .

Из фольклора

Данная лекция не была бы произнесена устно без помощи *Андрея Юрьевича Хренникова, Андрея Григорьевича Хромова* и *Лидии Александровны Крайко*, и не была бы написана без поддержки *Юрия Владимировича Тюменцева*, которым автор выражает свою искреннюю благодарность.

В лекции обсуждаются следующие вопросы

- сознание и его дефицит;
- рефлексия по Лефевру;
- логика и физический мир;
- многоагентные системы и квантовые статистики.

Первый вопрос, на который однако надо ответить — а почему именно здесь и сейчас мы должны говорить о сознании? Может быть лучше поговорить о чем-то другом?

В 1999 году у нас была *«Дискуссия о нейрокомпьютерах — 10 лет спустя»* и в ее ходе *Сергей Александрович Шумский* резонно заметил ([1], с. 33):

> Десять лет назад была дискуссия и было решено, что главная проблема — это проблема внимания. Теперь опять говорим, что проблема внимания не решена. Значит она не главная, значит она не востребованная. Кто мне докажет, что понимание того, как человек мыслит и воспринимает мир, именно сейчас позарез нужно мировой экономике?

Вопрос правильный, и необходимость дискутировать *о сознании* здесь и сейчас надо тоже доказывать.

Кто только не изучает сознание! Неполный список включает философов, психологов, специалистов по когнитивным наукам, нейрофизиологов, физиков (некоторых), социологов, экономистов (немногих), медиков (психиатров) и всех, кому не лень (включая докладчика). Вопрос — зачем они это делают?

Вообще-то, для многих это вопрос *практический*. Например, для поступающих в аспирантуру. Всем аспирантам предстоит сдавать кандидатские

экзамены. Раньше сознание изучалось в курсе философии, и все соискатели знали, по крайней мере, что *«материя первична, а сознание вторично»*. Теперь философии нет, а есть *история и философия науки*. Изучение программы этого эпохального курса вдоль и поперек показывает, что о сознании там практически не упоминается, а если упоминается, то в специальном разделе *«философских проблем астрономии и космологии»*, где есть тема происхождения сознания, по-видимому, оно происходит где-то в космических глубинах. На Земле сознание, видимо, еще не произошло, или было частично потеряно (после 1991 года).

В качестве безусловно положительного нововведения программа устанавливает (наконец-то!) связь с тематикой конференции по нейроинформатике. В одном из рекомендуемых аспирантам пособий сказано:

> Искусственный нейропроцессор профессионала (профессора, доктора наук и т. д.) заменить может[1]. Однако ученого — не может.

Именно этот *«караул!»* автор и предлагает как одно из оправданий разговора о сознании.

Можно ли объяснить сознание и что понимать под его объяснением?

Так звучит первый вопрос нашего Круглого стола. Но, этот вопрос лучше оставить. Во-первых, как писал *Михаил Евграфович Салтыков-Щедрин* в *Истории одного города*:

> А так как вопрос этот длинный, а руки у них коротки, то очевидно, что существование вопроса только поколеблет их твердость в бедствиях, но в положении существенного улучшения все-таки не сделает.

Во-вторых, те кто изучал в аспирантуре *философию*, должны помнить о мысли одного ныне забытого немецкого мыслителя, что проблема не в том, чтобы *«объяснить мир, а в том, чтобы его изменить»*. К сознанию это относится в полной мере.

Итак, главное понять, как изменить сознание? Зачем это нужно? И тут мы подходим к такой волнующей теме, как анозогнозия.

[1]На одном из сайтов так и сказано: *Клямкин Александр Моисеевич*, доктор философских наук, процессор.

Анозогнозия: Теоретический аспект

> С некоторым трепетом я бы попросил моего терпеливого читателя ознакомиться в *«Справочнике по психическим заболеваниям»*, известном среди психиатров как DSM-III, с заболеванием, называемым *анозогнозией*. Этот синдром, которым в тот или иной период своей жизни страдает от 30 до 70% популяции. . . , в котором люди воспринимаются как мертвые или как машины. . . Я думаю, что читатель может допустить, что этим синдромом ежедневно страдали люди в период холодной войны, и, что более важно, многие из тех, о котором пишется в этой книге.
>
> *Филип Мировски*. Сны машин. Экономика становится киборг-наукой.

Анозогнозия (а + греч. nosos — болезнь, gnosis — знание). Этот синдром *неосознания болезни*, который наблюдается при некоторых психозах (например, при шизофрении) и органических поражениях головного мозга, был описан французским врачом *Бабинским* еще в 1914 году. Но только в 80-е годы он привлек пристальное внимание именно в связи с изучением сознания. Проявление анозогнозии легко описать на примере.

При поражении правого полушария головного мозга, часто развивается паралич левых конечностей (гемиплегия). У 50% таких больных наблюдается *анозогнозия при гемиплегии* — они не осознают, что парализованы.

На указание, что левая рука не движется больной говорит:

— Вот, пожалуйста — я ее поднял, а вот — опустил.

— А какая же рука лежит слева от Вас на кровати?

— Видимо ваша, доктор.

— Ну похлопайте двумя руками.

— Похлопал.

— А почему тихо?

— Потому что — я культурный человек.

— А пройтись не можете?

— Да только что ходил за газетами — вот прилег отдохнуть.

Больной анозогнозией способен сочинить любую фантастическую историю, доказывающую его полное здоровье. Правое полушарие у него страдает. Левое работает на полную катушку, то есть самозабвенно отрицает

действительность.

Индийский доктор *Вилаянур Рамачандран* объясняет анозогнозию тем, что схема нашего тела хранится в правом полушарии и при его поражении сохраняется, без модификаций. Поэтому больной ориентируется на устаревшую информацию. Однако, анозогнозия развивается и при других болезнях — при слепоте это так называемый синдром Антона, имеется также алкогольная анозогнозия.

Правое полушарие, как полагают, связано с осознанием нас как личности, и его поражение приводит к такому удивительному дефекту самосознания. Заметим, что есть данные о том что анозогнозия проявляется и при поражении левого полушария, но нарушения речи в этом случае затрудняют ее обнаружение.

Тем не менее, наиболее распространенным (и вопиющим) является связь анозогнозии с поражением именно правого полушария. В этом случае для лечения применяется метод калорической терапии, при которой больному в левое ухо вливается ледяная вода, что стимулирует правое полушарие головного мозга. На некоторое время к больному приходит осознание недуга, однако, он при этом не признает, что ранее отрицал его наличие у себя. Когда же лечебный эффект проходит, больной вновь утверждает, что совершенно здоров и отрицает, что ранее признавал болезнь.

Анозогнозия: Практический аспект

Здоровых нет — есть
необследованные.

Аксиома психиатрии.

Все это имело бы мало отношения к нашей жизни, если бы явления сходные с анозогнозией не наблюдались у доброй половины человечества в той или иной период жизни (смотри эпиграф к предыдущему разделу). Похоже, разбаланс полушарий, или, может быть, дефицит активности правого полушария, встречаются на каждом шагу без явных признаков органических повреждений или иных заболеваний.

Многие люди как бы не видят окружающих бед, отказываются их признавать и начинают городить изысканнейшую чушь, чтобы уйти от неприятных вопросов.

В связи с недавней оглушительно провальной попыткой экранизации *Мастера и Маргариты* на одном из интернет-форумов была приведена интересная мысль по поводу интерпретации замысла романа, говорящая о том, что врач *Булгаков* об анозогнозии знал, а может и страдал от нее.

> Вообще роман о том, как нежелание признать одну свою фатальную ошибку и бег от реальности приводят к совершенно безумным последствиям и концепциям и как человек способен выстроить сложнейшие теории и устроить вереницу жертвоприношений ради того, чтобы превратить свои преступления в свои достижения[2].

Переписывание истории также может быть объяснено заболеванием, хотя иногда в этом может проявляться простая корысть.

15 января 2006 года в передаче *В. Т. Третьякова «Что делать?»* несколько историков, жмурясь от предвкушения обещанного финансирования, рассказывали о заказе на переписывание истории России, при котором из нее исчезла бы Киевская Русь: «Ведь не может же история одного государства начинаться на территории другого государства!» Обсуждалось, с чего начать — с Новгорода, или уж прямо — с татаро-монгольского нашествия.

Все это, увы, имеет отношение и к экономике, и к политике. Для иллюстрации приведем еще один случай, который не нуждается в комментариях.

> **Случай с больным Г.**
> Г. рассердился и выпалил: Никакого роста смертности в России нет!. . . Вот у нас научный эксперт, он объяснит. Эксперт В. привел «научный» аргумент. . . РФ перешла на западную методику учета рождаемости. Раньше. . . младенцев, родившихся с весом менее 700 г. . . , не включали в статистику рождений, а теперь включают. А они, бедные, поголовно умирают, что и дает жуткий прирост смертности.
> (*С. Г. Кара-Мурза*. Идеология и мать ее наука // *Независимая газета*, 5 декабря 2005 года)

Низкую рождаемость в России и раннюю смертность мужчин Г. объяснил установкой СССР на равноправный доступ женщин к тяжелому физическому труду и тягой сильного пола к алкоголю.

[2]Конечно, на такую интерпретацию может всегда последовать ироничный комментарий: «Тема кровавой гэбни не раскрыта! Низачот!»

Самосознание

Анозогнозия выявляет важную роль,которую играет в самосознании правое полушарие головного мозга. И, хотя уже отмечалась возможное возникновение такого расстройства и при поражении левого полушария [52], существуют другие указания на основополагающую роль именно правой части мозга. Одно из таких свидетельств дает узнавание себя в зеркале.

Известно, что такой способностью, кроме людей, обладают лишь некоторые высшие приматы и дельфины.

Роль правого полушария в распознавании себя людьми была продемонстрирована *Кенаном* и сотрудниками [53]. В этих экспериментах больным с анестезией левого или правого полушария показывалась фотография, являющаяся химерой, составленой из фотографии испытуемого и известной личности (для женщин — *Мэрилин Монро*). Испытуемые, у которых было отключено правое полушарие сообщали впоследствии, что им показывали известную персону. И лишь отключение левого полушария (при активности правого) позволяло им сообщить о том, что они видели себя.

В экспериментах *Газзаниги* и сотрудников [54] аналогичные химерные изображения показывались больным с расщепленным мозгом. Изображения содержали смесь фотографии больного и известного ему человека (конкретно, *Газзаниги*), так что доля больного менялась от 100% до 0%. В этих экспериментах на вопрос «Это Вы?» правое полушарие отвечало утвердительно, только если доля себя в изображении была более 70%. А левое «узнавало» себя, когда эта доля была уже порядка 30%. На этом основании делался вывод, что, напротив, левое полушарие ответственно за узнавание себя.

Наоборот, правое полушарие узнавало в изображении другого, если его доля была невелика. А для левого полушария требовалось 70% другого, чтобы его распознать. Поэтому, правое полушарие признавалось ответственным для распознавания другого.

Однако, эти же результаты могут интерпретироваться и по-другому.

В случае изображения, содержащего лишь малую долю себя, левое полушарие может просто врать врачу, отвечая утвердительно на вопрос «Это Вы?» (вспомним анозогнозию).

В противоположном случае, когда доля чужого в изображении мала правое полушарие может испытывать затруднение при построении отрицания при формулировке ответа на вопрос: «Это Майк? (Газзанига)».

В таком случае вывод о роли левого полушария в узнавании себя может

оказаться преждевременным.

Если мы хотим, чтобы изучение сознания (или внимания) было бы *полезно для мировой экономики*, то мы могли бы постараться построить хотя бы примитивные модели сознательных экономических агентов.

Из вышесказанного следует, что правдоподобно сделать их двухполушарными и снабдить именно левое полушарие таких агентов способностью строить отрицания, лишив этой возможности их правое полушарие. В качестве основы таких моделей могут быть положены рефлексивные структуры.

Математическая психология и экономический агент

> Когда мы говорим «рефлексия», из глубин нашей немецкой памяти всплывает контекст, в котором подразумевается ни что иное, как сознание. Ну а если мы вспомним, что в рамках немецкого классического идеализма сознание суть всеохватная, все замыкающая на себя реальность, что человек как личность рефлексивно отождествляется с сознанием (как самосознанием), то мы поймем, что посягая на рассмотрение или даже на разгадывание того, что есть рефлексия, мы, даже перестав быть философами, тем не менее притязаем на охват этого пространства целиком.
>
> *В. Г. Гегель*

В двадцатом веке стало ясно, что, поскольку люди совершают все возможные иррациональные поступки и делают все возможные когнитивные ошибки, то рациональный агент не подходит для построения экономических моделей. Поэтому, именно когнитивные науки, частью которых является математическая психология, должны дать знания, необходимые при построении такой модели экономического агента, которая позволила бы описать реальную экономику. В частности, вклад в решение этой проблемы могла бы внести теория рефлексивных систем *Владимира Лефевра* [18]. Удивительным образом, эта теория, положительно воспринятая военными и дипломатами, и, возможно, сыгравшая определенную роль в трагической судьбе нашей страны [19], не была воспринята экономистами. Сам Лефевр говорил об этом так [20].

Первые реально работающие рефлексивные модели появились в конце 70-х годов. Их создание было активно поддержано военными и дипломатами. Однако экономисты встретили их достаточно холодно. Интерес военных и дипломатов стимулировался способностью рефлексивных моделей представлять сложные военно-политические коллизии, ранее находившиеся вне сферы научного рассмотрения. Реакция экономистов требует специального пояснения. В основе экономических моделей лежит представление о человеке как о рациональном существе, стремящемся максимизировать свою выгоду. Такой взгляд на человека уходит своими корнями в политическую экономию XVIII века. Рефлексивные модели внесли в научное представление о человеке новое измерение, связанное с такими категориями, как мораль, совесть и чувство справедливости. Они позволяют отражать ситуации, в которых люди не только стремятся получить материальный доход, но имеют и неутилитарные цели, совершают жертвенные поступки, стремятся выглядеть достойно и в своих собственных глазах, и в глазах других людей.

Центральное место в теории Лефевра играет аксиома, формальное выражение которой дается с помощью функции импликации:

$$c \to a = a + \bar{c}. \tag{1}$$

Экспоненциальное представление этой функции

$$a^c = c \to a \tag{2}$$

имеет замечательную интерпретацию, при которой основание (a) представляет собой предложение среды агенту сделать добрый ($a = 1$) или злой ($a = 0$) поступок, показатель (c) — оценку этого предложения как добро ($c = 1$) или зло ($c = 0$). Значение самой функции импликации определяет интенцию агента, которая может побуждать его сделать добро (1) или зло (0). Поскольку $0^0 = 1$, то аксиома алгебры Лефевра гласит, что зло осознанное как зло есть добро. Другие значения функции импликации приведены в табл. 1.

В теории Лефевра агенты могут взаимодействовать друг с другом и это взаимодействие может быть формально представлено выражениями типа

$$a^c * b^d, \tag{3}$$

Таблица 1. Значения функции импликации

a	c	a^c
1	1	1
1	0	1
0	1	0
0	0	1

где знак операции $*$ может означать как то, что агенты находятся в состоянии конфронтации, так и компромисса. В зависимости от того, какая из операций, сложение или умножение, используется для описания конфронтации (компромисса), получается два варианта теории, которые, как нашло подтверждение в ходе психологических экспериментов, описывают две этические системы, реализованные, в первом случае, в западном мире, а в другом — в СССР и некоторых восточных цивилизациях.

В первой (западной) этической системе, построенной на формальном запрете зла, агенты оценивают конфликт добра и зла позитивно, но, парадоксальным образом, для повышения своего этического статуса стремятся к компромиссу с противником. Во второй (советской) этической системе, построенной на неформальной декларации добра, агенты позитивно оценивают компромисс добра и зла, но, напротив, повышают свой этический статус, вступая в конфликт с противником. Некоторым аналогом представления о существовании двух этических систем может оказаться продемонстрированное *В. П. Масловым* существование двух характерных для капитализма и социализма математик [25]. Согласно Лефевру именно отсутствие автономного механизма разрешения конфликта в СССР (реализовавшегося ранее властью) привело к тому, что в России девяностых годов двадцатого века эту функцию взял на себя криминал, что предопределило негативное развитие общества. Сам Лефевр высоко оценивал собственную роль в окончании холодной войны и разрушении Советского Союза, так как, по его мнению, установленное им знание об особенностях советской этической системы помогло американцам научить советское руководство обманывать свой народ. Практическая рекомендация, вытекающая из метода контролируемой конфронтации, заключается в том, чтобы не требовать от советского правительства громогласного подписания компромисс-

ных документов, а представить ему возможность оформлять политические решения официально в одностороннем порядке. Эта история была описана Лефевром в статье [19]. Если Лефевр прав, то это означает, что мы действительно существенно отличаемся от западных людей, что, несомненно, должно было сказаться на результатах проводимых в России экономических экспериментов по стандартам иной этической системы.

Признания Лефевра вызвали острую реакцию *Глеба Павловского*, констатировавшего [21]:

> Да, мы превратились в «морально близких» Западу, но по случайности ли одновременно с этим мы стали братвой?

Нельзя ли все же использовать формализм Лефевра для построения интересных многоагентных экономических моделей с нетривиальным поведением, в которых агенты обязаны быть гетерогенными? Заметим, что у Лефевра агенты, принадлежащие разным этическим системам, могут сосуществовать в одном обществе, образуя именно такое гетерогенное сообщество. Но нужны ли экономические модели именно с такими агентами, как у Лефевра?

Наличие и роль ярко выраженной гетерогенности агентов в реальной экономике описано *Биллом Вильямсом* [22]. Вильямс отмечает, что 90% участников рынка — постоянных его неудачников — характеризует доминирование левого полушария головного мозга. Такие агенты рассматривают окружающий их мир как иерархическую систему с жесткой конкуренцией между участниками. Движущими силами поведения агентов с доминантным левым полушарием являются алчность и страх. Эти агенты стремятся достичь успеха, овладевая большим числом формальных теорий и инструментов анализа рынка, но парадоксальным образом это приводит их к поражению (согласно американскому экономисту *Полу Кругману* [23] этих людей, кроме жадности, характеризуют недальновидность, вера в то, что другие — еще глупее, стадное чувство, чрезмерная способность к обобщениям и пр.) Победителями на рынке оказываются, по Вильямсу, агенты с доминантным правым полушарием мозга. Такие агенты рассматривают мир как дружественную сетевую среду с горизонтальными связями (семья, друзья). Они не испытывают чувство страха и спокойно относятся к возможным потерям. Агенты с доминантным правым полушарием принимают решения, основываясь на своей интуиции и на неформальном, конкретном анализе ситуации. Интересно в этой связи отметить, что *Эдгар Петерс* [24] описал причину поражения «ленинского коммунизма» отсутствием чувства

алчности и страха — движущих сил рыночной экономики. Вспоминая Вильямса можно предположить, что эти чувства, действительно нужны рынку, но лишь для того, чтобы сформировать 90% людей, которые лишатся своих денег. По-видимому, на сохранение этого числа и направлены усилия масс-медиа, заполняющие эфир ужасами на все вкусы и играми на миллионы в любой валюте. Так или иначе, гетерогенность агентов, источником которой является межполушарная асимметрия и доминирование одного из полушарий, является плодотворной отправной точкой для построения моделей.

Этические системы и доминантность

> Время — это устройство, не позволяющее событиям заканчиваться сразу после их начала. . .
> Пространство — это устройство, препятствующее тому, чтобы все происходило в Кембридже.
>
> *Дхарма Кумар*

В статье [3] было выдвинуто предположение, что существование двух этических систем по Лефевру, может быть связано с доминированием у большинства агентов левого (в первой, западной этической системе) и правого (во второй, восточной этической системе) полушарий головного мозга. В поддержку такого соответствия было выдвинуто много аргументов. Рассмотрим еще один.

Связь двойственности смыслов операций сложения и умножения с доминантностью полушарий можно продемонстрировать на следующем примере. Учитывая, что левое полушарие головного мозга обрабатывает временную информацию, а правое — пространственную, рассмотрим два модельных мира, в первом из которых отсутствует пространство, а во втором — время.

В первом *внепространственном* мире (мире *левого* полушария мозга) все происходит в *единственной точке пространства*, но в разные моменты времени.

Во втором *вневременном* мире (мире *правого* полушария мозга) все происходит *в один момент времени*, но в разных его пространственных

точках.

Посмотрим, как можно проинтерпретировать в обоих этих мирах некоторое утверждение о физическом явлении, например такое: *«Вода кипит»* И *«Вода не кипит»*.

Во внепространственном мире такое утверждение будет противоречивым только если логическое умножение И понимать как И ОДНОВРЕМЕННО. Действительно, в таком мире вода не имеет протяженности и в данный момент времени может либо кипеть, либо не кипеть. Выражение для соответствующей истинности высказывания *«Вода кипит»* И ОДНОВРЕМЕННО *«Вода не кипит»*:

$$1 \cdot 0 = 0.$$

В то же время выражение *«Вода кипит»* И *«Вода не кипит»* не противоречиво если понимать И как И В ТОЙ ЖЕ ТОЧКЕ (а здесь весь мир — одна точка), поскольку она может кипеть в этом мире в один момент времени и не кипеть в другой. Таким образом, если И интерпретировать таким образом *«Вода кипит»* И В ТОЙ ЖЕ ТОЧКЕ *«Вода не кипит»*, то такой операции нужно сопоставит знак сложения:

$$1 + 0 = 1.$$

С другой стороны, во вневременном мире утверждение *«Вода кипит»* И *«Вода не кипит»* является противоречивым если только И будет означать И В ТОЙ ЖЕ ТОЧКЕ. То есть этой операции соответствует не сложение, как в первом случае, а умножение.

А так как во вневременном мире, в котором все происходит одновременно, высказывание *«Вода кипит»* И ОДНОВРЕМЕННО *«Вода не кипит»* не приводит ни к какому противоречию, так как она может кипеть в одной пространственной точке и не кипеть в другой, то И ОДНОВРЕМЕННО соответствует логическому сложению. Окончательно ситуацию иллюстрирует табл. 2.

Итак, как и в алгебре Лефевра, миры левого и правого полушария требуют дополнительного описания логических операций.

Гипотеза о связи этической системы с доминантностью позволяет попробовать, наконец, использовать теорию рефлексивных структур в социальных моделях вообще и в экономике, в частности. Удивительно, что при этом возникает прямая аналогия с физическими системами, описываемыми квантовой статистикой.

Таблица 2. Смысл логических операций во внепространственной и вневременной моделях мира левого и правого полушария головного мозга, соответственно

Модель мира	И В ТОЙ ЖЕ ТОЧКЕ	И ОДНОВРЕМЕННО
Внепространственный мир левого полушария	+	·
Вневременной мир правого полушария	·	+

Модель агентов, обладающих двумя ресурсами

Для неоклассической экономики характерно наличие у экономического агента единственного оптимизируемого ресурса — так называемой *полезности* (*utility*). В модели, излагаемой ниже, их два, что существенно. Мы представляем эту многоагентную модель, иллюстрирующую возникновение двух различных квантовых статистик — Бозе-Эйнштейна и Ферми-Дирака — в дружественной популяции индивидуумов, у которых доминирует правое полушарие мозга, и в конкурентной популяции индивидуумов, у которых доминирует левое полушарие мозга, соответственно. Для этого, мы приводим аргументы в пользу того, что лефевровская алгебра совести может быть использована естественным образом для описания стратегий принятия решения агентами, имитирующими людей, имеющих доминирование разных полушарий мозга. Можно предположить, что возникновение двух главных статистических распределений может иллюстрировать возникновение различным типов организации общества, и может быть также использовано при моделировании явлений на рынке и психических расстройств, в которых может быть задействовано переключение доминирования полушарий.

Изучение социальных и экономических процессов может основываться на использовании многоагентных моделей. В таких моделях каждый агент должен имитировать человека, который должен выживать в окружающей среде, используя для этого правильные стратегии принятия решения и взаимодействуя с другими агентами. Многие из подобных моделей используют нейронные сети для представления агента. Однако, для того чтобы описать

рыночные явления, необходимо сделать этих агентов достаточно разнообразными — гомогенная популяция агентов не имитирует правдоподобным образом поведения рынка. В результате, в популяцию агентов часто искусственным образом вводится гетерогенность. Однако, такая процедура часто носит *ad hoc* характер и в действительности не имеет отношения к знанию о реальной работе мозга и о когнитивных способностях человека. Говоря кратко, нейросетевые модели слишком примитивны для того, чтобы представлять работу мозга, поэтому необходимо искать модель агента более высокого уровня.

Описываемая далее модель естественным образом включает разнообразие стратегий принятия решения агентами, которое обусловлено доминированием различных полушарий мозга человека. Модель эта основывается на ряде предшествующих результатов.

Первый из них связан с пониманием глубинной сути рыночных явлений, описанным *Вильямсом*: можно обнаружить, что неудачники рынка (90% участников) являются лицами, имеющими выраженную доминантность левого полушария. Эти персоны ведомы страхом и алчностью (чувствами рассматриваемыми как необходимые ингредиенты рыночной экономики, например, *Петерсом* [24]) и они явно используют свои логические и математические способности (приписываемые левому полушарию) для принятия решений. Такие левополушарные люди имеют специфическую модель мира, рассматривая его как вертикальную иерархическую структуру, характеризуемую вечным соперничеством, внезапными падениями и трудным восхождением на верхние ступени социальной иерархии. С другой стороны, рыночные победители выглядят более правополушарными. Эти правополушарники не испытывают страха, полагаются на свою интуицию и рассматривают окружающую среду как дружелюбное место для кооперации и установления горизонтальных связей. Естественный вопрос, который мы хотим обсуждать далее, таков: можем ли мы имитировать такую разницу в картине мира, свойственную людям с доминантностью различных полушарий мозга, используя простые математические модели? Мы будем предполагать, что для этого будет естественно использовать многоагентную модель, в которой могут быть симулированы лево- и правополушарные агенты. Но как описать эти два типа агентов?

Здесь может оказаться полезным второй предшественник. В *«алгебре совести»*, развитой Лефевром, аргументируется возможность существования в действительности лишь двух типов этических систем. Можно предположить, что такая дихотомия может возникать вследствие асимметрии

мозга, и вследствие доминантности левого или правого полушария мозга.

Мы также покажем, что элегантный формализм Лефевра может быть естественным образом использован для разработки модели агентов, имеющих лево- и правополушарную доминантность. Более того, мы будем приводить аргументы в пользу того, что лишь две разумные стратегии принятия решений возникают в популяциях, в которых агенты стремятся сохранить свои физический и ментальный ресурс. По-видимому, наиболее интересной особенностью модели является та, что чистые сообщества, состоящие из лево- и правополушарных агентов с дружественными и конкурентными отношениями, соответственно, описываются известными квантовыми распределениями — Ферми-Дирака и Бозе-Эйнштейна.

Мы полагаем, что данное обстоятельство находится в согласии с нашей целью — имитировать наличие представления о конкурентной среде у левополушарных людей (выражаемое в эффективном отталкивании фермионов) и кооперации правополушарных (что можно соотнести с эффективным притяжением бозонов).

Нас не должно удивлять, что такие известные распределения возникают в нашей классической многоагентной модели. Существуют различные системы, как квантовые, так и классические, чье состояние равновесия описывается квантовыми статистическими распределениями.

Например, *Эванс* [31] обнаружил Бозе-Эйнштейновскую конденсацию при решении гетерогенной задачи перевозок (прыжков частиц). *Бьянкони* и *Барабаси* [41] показали, что статистика Бозе-Эйнштейна описывает растущую интернет-сеть (такая сеть постоянно растет путем добавления и удаления новых узлов и связей). *Сталиунас* [33] привел аргументы в пользу того, что Бозе-Эйнштейновская конденсация может возникать в классических системах, далеких от состояния и теплового равновесия, за счет когерентной динамики, или эквивалентной автокаталитической динамики в пространстве импульсов системы. Существенным условием для появления Бозе-Эйнштейновского распределения в данном случае является то, что случайная миграция частиц в пространстве импульсов зависит от степени занятости состояний этого пространства. Это типично для многих нелинейных систем. Поэтому, квантовый характер системы не является существенным ингредиентом Бозе-Эйнштейновской конденсации. Также, *Бьянкони* недавно обнаружила, что растущее дерево Кели, имеющее качественно разные узлы и тепловой шум, описывается статистикой Ферми-Дирака [34]. Ранее *Деррида* и *Лебовиц* обнаружили [35], как распределение Ферми-Дирака, так и Бозе-Эйнштейна при рассмотрении полностью асим-

метричных процессов исключения на кольце с N сайтами и p частицами. Мы покажем, что квантовые статистические распределения описывают также популяции агентов, обитающих в клеточной модели мира, описываемой в следующем разделе.

Клеточная модель мира

Предположим, что мир состоит из n клеток, в которых может находиться, в общем случае, произвольное число агентов $x^{(\alpha)}, \alpha = 1, \ldots, N$.

Каждый агент обладает двумя типами ресурсов: физическим и ментальным, которые характеризуются действительными значениями $p^{(\alpha)} \geqslant 0$ и $m^{(\alpha)} \geqslant 0$.

Агент $x^{(\alpha)} = \{p^{(\alpha)}, m^{(\alpha)}\}$ умирает, если любой из его ресурсов обнуляется. Таким образом, для выживания любой агент должен поддерживать положительными свой физический и ментальный ресурсы в любой момент времени $t \geqslant 0$: $p^{(\alpha)}(t) \geqslant 0$ and $m^{(\alpha)}(t) \geqslant 0$.

Предохранение обоих ресурсов от исчезновения представляет собой, в общем случае, противоречивую задачу.

Предположим, что каждый агент должен за время Δt затратить некоторый физический ресурс $\gamma \Delta t$ ($\gamma > 0$, $\tau > 0$) для сохранения своей физической структуры. Этот процесс сопровождается безусловным уменьшением физического ресурса агента.

К счастью, любой агент может потребить некоторую величину физического ресурса (еды) $h\Delta t$, которая возникает случайным образом в клетках мира. Для этого агент должен сменить клетку, в которой он находится, если такая еда возникает в другой клетке (он должен перейти в последнюю). Если же еда возникает в той клетке, в которой агент уже находится, то он может употребить ее тут же.

Мы будем предполагать, что одна и та же порция еды возникает в разных клетках мира с разной частотой $f_i, i = 1, \ldots, n$, отражающей привлекательность данной клетки для агента. Мы также предположим, что в интервал времени Δt агент может употребить $h\Delta t$ ($h > 0$) физического ресурса в любой клетке, где эта величина появится.

Предположим, что если агент *меняет свою клетку* для того чтобы употребить еду, его *ментальный* ресурс уменьшается на единицу. Мы будем интерпретировать такую ситуацию как если бы агент тратил свой ментальный ресурс на решение проблемы физического выживания. Мы также будем предполагать, что агент не может никоим образом увеличивать или

компенсировать свой ментальный ресурс. Конечно, если еда появляется в той клетке, в которой находится агент он может употребить ее без изменения ментального ресурса[3].

Мы будем интерпретировать появление еды в клетке, не занятой данным агентом, как *предложения окружающей среды изменить ментальность агента* или, что то же самое, заплатить единицей ментального ресурса за еду.

Пусть бинарная (булева) переменная a обозначает такое предложение, причем $a = 0$, если окружение предлагает сменить клетку.

Появление единицы физического ресурса в клетке, уже занятой данным агентом, может трактоваться как предложение ему *сохранить свою ментальность* и употребить еду бесплатно.

Пусть $a = 1$, если окружение предлагает агенту сохранить свою клетку.

Предположим, что любой агент может принять или отвергнуть такое предложение и что его решение является булевой функцией a, обозначаемой как $\psi(a)$. Пусть $\psi(a) = 0$ означает, что агент решил сменить клетку и занять другую, затрачивая свой ментальный ресурс, но употребляя еду, которая (к сожалению) предлагается в другой клетке. Аналогичным образом, $\psi(a) = 1$ означает, что агент решил остаться в своей старой клетке. Последнее решение сопровождается сохранением его ментального ресурса и уменьшением физического ресурса (если $a = 0$) или же бесплатным увеличением физического ресурса, если он, на счастье, возникает в той клетке, где агент уже находится (если $a = 1$).

[3]Необходимо уточнить далее наши концепции клеток мира и ментального выживания. Мировые клетки не следует рассматривать как ячейки некоторого физического пространства. Они не имеют соседних или далеких клеток, так что никакая метрика не вводится. Каждая из клеток может быть охарактеризована набором параметров, таких как (музыкант, юг) В таком случае для агента, занимающего такую клетку не принципиально, возникает ли пища в Миланском оркестре или в Барселонском оркестре. Он может менять свое реальное географическое положение, но будет оставаться в той же мировой клетке. Наоборот, если еда возникает в Стокгольмском оркестре, он должен сменить свою клетку, поскольку клетка с едой это теперь (музыкант, север). Агенту будет также необходимо изменить клетку для употребления еды, если от него потребуется стать ковбоем или поехать на Восток. Может показаться, что концепция мировой клетки довольно субъективна. Но мы будем предполагать, что все агенты имеют одинаковое представление о делении мира на такие клетки, так что эта клеточная структура может считаться объективной

Модель с отсутствием взаимодействий между агентами

Далее мы введем взаимодействие между агентами, но вначале рассмотрим модель без взаимодействий.

Давайте покажем, что существуют *только две различные стратегии* выживания невзаимодействующих объектов в описанном выше клеточном мире.

Мы уже предположили, что решение агента может быть описано булевой функцией одной переменной:

$$\psi = \psi(a). \tag{4}$$

Существуют четыре различных функции такого типа и мы обсудим их все. Первые две из них таковы:

- Если $\psi(a) \equiv 0$, то агент будет менять клетку, в которой он находится каждый раз при появлении еды, даже если она появится в клетке, где он уже находится. Это *совершенно неразумная стратегия*, так как она сопровождается неизбежным снижением ментального ресурса до фатального нулевого значения, т. е. до *ментальной гибели агента*.
- Если $\psi(a) = \bar{a}$, то агент будет всегда действовать вопреки предложению среды: Он будет менять свою клетку, если еда возникает как раз в ней, и будет оставаться в своей клетке, если еда будет возникать в другой клетке. Очевидно, что такой агент будет постепенно терять свой физический ресурс вплоть до своей *физической гибели*.

Заметим, что в обоих рассмотренных выше случаях новая клетка, в которую агент стремится перейти, когда окружение этого не требует, не определена. Таким образом, поведение агента в этих случаях будет выглядеть как случайное блуждание и динамика агента будет отчасти *стохастической*.

Рассмотрим теперь две *разумные стратегии*:

- Первая описывается булевой функцией

$$\psi(a) = 1. \tag{5}$$

  Агент сохраняет свой ментальный ресурс независимо от того, в какой клетке появляется еда. Так как эта еда может случайно возникнуть и в клетке, в которой данный агент уже находится, то он имеет также и шанс выжить физически, если такое счастливое событие происходит достаточно часто.

- Вторая разумная стратегия описывается функцией

$$\psi(a) = a. \tag{6}$$

  Используя эту стратегию, агент всегда следует предложению среды, увеличивая свой физический ресурс путем потребления еды. Он также имеет шанс сохранить свой ментальный ресурс в ситуациях, когда среда не требует от него сменить занимаемую клетку.

Дадим некоторую интерпретацию этих двух разумных стратегий. Для этого удобно представить соответствующие булевы функции в экспоненциальном виде:

$$\psi_R(a) = 1 \equiv a^a, \tag{7}$$

$$\psi_L(a) = a \equiv a^{\bar{a}}, \tag{8}$$

где

$$a^c = a + \bar{c} = c \to a \tag{9}$$

есть логическая импликация.

Мы будем называть эти стратегии *правополушарной* и *левополушарной*, соответственно. Мы также будем называть агентов, следующих этим стратегиям, *право*(*лево*)полушарными агентами. Можно привести некоторые предварительные аргументы в пользу такой интерпретации. Существуют некоторые экспериментальные свидетельства в пользу того, что правое полушарие не может строить логических отрицаний — все логические операции являются функциями левого полушария. Таким образом, принятие решения правополушарным агентом в случае $a = 0$ может быть проинтерпретировано в нашей модели так:

- Правополушарный агент *представляет*, что он *следует предложению среды* сменить клетку и уменьшает свой ментальный ресурс. Эта возможность потери себя при смене мировой клетки *ужасает его* и он отвергает такое предложение.

Напротив, левополушарный агент может создать ментальный образ, который соответствует *логической инверсии предложения среды*. Так что его принятие решения может быть описано так:

- Левополушарный агент *представляет*, что он *отвергает предложение среды* употребить еду в той или иной клетке. Эта возможность упустить шанс увеличить свой физический ресурс *ужасает его* и он следует предложению среды.

Мы увидим далее, что существуют также и другие аргументы в пользу такой интерпретации двух разумных стратегий [4].

Правополушарная стратегия

Пусть *взаимодействие агента со средой*, заключающееся в предложении агенту h единиц физического ресурса, имеет характерное время τ. Таким образом, вероятность агенту не получить предложения от среды убывает как $e^{-t/\tau}$.

Динамика популяции, состоящей только из правополушарных агентов, очень проста. Такие агенты не меняют своих клеток, а также не меняют своего ментального ресурса. Если число агентов в клетке i есть N_i, то

$$N_i(t) \equiv N_i(0), \tag{10}$$

$$m^{(\alpha)}(t) \equiv m^{(\alpha)}(0). \tag{11}$$

Их физический ресурс $p^{(\alpha)}(t)$, однако, изменяется во времени. Предположим, что за время τ агент тратит в среднем $\gamma\tau$ единиц физического ресурса. Тогда

$$p^{(\alpha)}(t+\Delta t) = p^{(\alpha)}(t) - \gamma\frac{\Delta t}{\tau} + f_i h\frac{\Delta t}{\tau}, \quad \alpha \in C_i, \tag{12}$$

[4]Один из таких аргументов был представлен *Ротенбергом* и *Аршавским*. Они предположили, что «. . . в наиболее общей форме различие между двумя стратегиями мышления сводится к противоположным способам организации смысловых связей между элементами информации. «Левополушарный способ» мышления так организует любой знаковый материал (как символический, так и иконический), чтобы получился строго упорядоченный и однозначно понимаемый контекст. Его формирование требует активного выбора, вне реальных и потенциальных связей между многоформными объектами и явлениями, небольшого числа определенных связей, которое не приведет к созданию внутренних противоречий (!) и облегчит упорядоченный анализ. . . Напротив, функция «правополушарного» «образного» мышления заключается в одномоментном схватывании бесконечного числа связей и формирование в результате такого схватывания интегрального, но *противоречивого контекста*. В таком контексте, целое не определяется своими частями, так как все специфические черты целого определяются только связями между этими частями. Наоборот, любой конкретный элемент такого контекста несет на себе печать целого. Новый опыт инкорпорируется в эту холистическую картину мира. Индивидуальные элементы образа взаимодействуют друг с другом во множестве семантических плоскостей одновременно. Примерам таких смысловых связей являются связи между образами в сновидениях, или в творчестве. Преимущества такой стратегии мышления проявляют себя только если информация является сложной, внутренне противоречивой и существенно не сводимой к непротиворечивому контексту» [30].

где $\alpha \in C_i$ означает, что α агент занимает клетку i.

Устремляя $\Delta t \to 0$, получим

$$\frac{d}{dt}p^{(\alpha)} = -\frac{1}{\tau}(\gamma - hf_i), \quad \alpha \in C_i, \tag{13}$$

Из (13) следует, что

$$p^{(\alpha)}(t) = p^{(\alpha)}(0) - \frac{1}{\tau}(\gamma - hf_i)t, \quad \alpha \in C_i. \tag{14}$$

Таким образом, правополушарный агент будет выживать в тех клетках, для которых $hf_i \geqslant \gamma$. Для клеток, у которых $hf_i < \gamma$, его физическая жизнь будет иметь продолжительность

$$T_{phys}^{(\alpha)} = \frac{\tau p^{(\alpha)}(0)}{\gamma - hf_i}. \quad \alpha \in C_i. \tag{15}$$

Следовательно, правополушарная стратегия является абсолютно пассивной и выживание агента зависит только от параметров среды, а также от шанса агенту занять изначально счастливую клетку.

Левополушарная стратегия: Распределение Гиббса

Левополушарные агенты меняют свои клетки для потребления пищи, которая им предлагается. Следовательно, среднее число частиц в клетке $\langle N_i \rangle$ становится функцией времени $\langle N_i(t) \rangle$. Конечно, нам приходится использовать средние величины, поскольку предложение пищи носит случайный характер. В интервале времени $[t, t + \Delta t]$ средние величины N_i будут возрастать вследствие прибытия агентов, принявших предложение употребить еду в i-й клетке и будут уменьшаться вследствие того, что некоторые агенты, которые вначале занимали клетку i, получают предложение пищи в других клетках. Соотношение баланса

$$\begin{aligned} \langle N_i(t + \Delta t) \rangle = \langle N_i(t) \rangle + \sum_{j \neq i} f_i\Big(\frac{\Delta t}{\tau}\Big)\langle N_j(t) \rangle - \\ - \sum_{j \neq i} f_j\Big(\frac{\Delta t}{\tau}\Big)\langle N_i(t) \rangle. \end{aligned} \tag{16}$$

В пределе $\Delta t \to 0$ получаем уравнение:

$$\frac{d}{dt}\langle N_i(t)\rangle = -\frac{1}{\tau}\langle N_i(t)\rangle + \frac{f_i N}{\tau}. \tag{17}$$

Его решение имеет вид:

$$\langle N_i(t)\rangle = (\langle N_i(0)\rangle - f_i N)e^{-t/\tau} + N f_i. \tag{18}$$

Очевидно, что асимптотически распределение средней занятости стремится к распределению частот представления пищи:

$$\lim_{t\to\infty}\langle N_i(t)\rangle = N f_i. \tag{19}$$

Следуя [32], введем энергию ячейки:

$$\epsilon_i = -\theta \log f_i, \tag{20}$$

где параметр θ характеризует температуру среды.

Тогда равновесное распределение (19) принимает форму распределения Гиббса

$$\lim_{t\to\infty}\langle N_i(t)\rangle = N e^{-\epsilon_i/\theta}. \tag{21}$$

Чтобы получить среднее время жизни левополушарных агентов, вспомним, что они постепенно теряют свой ментальный ресурс, меняя клетку, в которой они находятся.

По определению, средний ментальный ресурс для всей популяции составляет

$$\langle m(t)\rangle = \frac{1}{N}\sum_{i=1}^{n}\sum_{\alpha\in C_i} m_i^{(\alpha)}(t), \tag{22}$$

где C_i — это набор номеров агентов, занимающих i-ю клетку. Средний ментальный ресурс агентов, занимающих i-ю клетку, равен

$$\langle m_i(t)\rangle = \frac{1}{\langle N_i(t)\rangle}\sum_{\alpha\in C_i} m_i^{(\alpha)}(t). \tag{23}$$

Напишем балансное соотношение для интегрального ментального ресурса агентов, занимающих i-ю клетку, которое отражает утечку агентов с

предыдущим значением ресурса и также приток агентов из других клеток с ресурсами, уменьшившимися на единицу:

$$\begin{aligned}\langle N_i(t+\Delta t)\rangle\langle m_i(t+\Delta t)\rangle = \langle N_i(t)\rangle\langle m_i(t)\rangle - \\ - \sum_{j\neq i} f_j\Big(\frac{\Delta t}{\tau}\Big)\langle N_i(t)\rangle\langle m_i(t)\rangle + \\ + \sum_{j\neq i} f_i\Big(\frac{\Delta t}{\tau}\Big)\langle N_j(t)\rangle(\langle m_i(t)\rangle - 1),\end{aligned} \tag{24}$$

или

$$\begin{aligned}\langle N_i(t+\Delta t)\rangle\langle m_i(t+\Delta t)\rangle = \langle N_i(t)\rangle\langle m_i(t)\rangle \times \\ \times \Big(1 - \frac{\Delta t}{\tau}(1-f_i)\Big) + \\ + \sum_{j\neq i} f_i\Big(\frac{\Delta t}{\tau}\Big)\langle N_j(t)\rangle(\langle m_i(t)\rangle - 1).\end{aligned} \tag{25}$$

Переходя к пределу $\Delta \to 0$ получим:

$$\begin{aligned}\frac{d}{dt}\langle N_i\rangle\langle m_i\rangle = -\frac{1}{\tau}(1-f_i)\langle N_i\rangle\langle m_i\rangle + \\ + \frac{f_i}{\tau}\Big(\sum_{j\neq i}\langle N_j\rangle\langle m_j\rangle - \sum_{j\neq i}\langle N_j\rangle\Big).\end{aligned} \tag{26}$$

Так как

$$\sum_{j\neq i}\langle N_j\rangle = N - \langle N_i\rangle, \tag{27}$$

и

$$\sum_{j\neq i}\langle N_j\rangle\langle m_j\rangle = \langle Nm\rangle - \langle N_i\rangle\langle m_i\rangle = N\langle m\rangle - \langle N_i\rangle\langle m_i\rangle, \tag{28}$$

после простых алгебраических преобразований получаем

$$\frac{d}{dt}\langle N_i\rangle\langle m_i\rangle = -\frac{1}{\tau}\langle N_i\rangle\langle m_i\rangle + \frac{Nf_i}{\tau}\big(\langle m\rangle - 1\big) + \frac{\langle N_i\rangle f_i}{\tau}. \tag{29}$$

Суммируя (29) по $i = 1, \ldots, n$ и принимая во внимание что

$$\sum_{i=1}^{n}\langle N_i\rangle\langle m_i\rangle = N\langle m\rangle, \tag{30}$$

получаем

$$\frac{d}{dt}\langle m\rangle = -\frac{1}{N\tau}\sum_{i=1}^{n}(1-f_i)\langle N_i(t)\rangle. \tag{31}$$

Теперь, используя явную форму $\langle N_i(t)\rangle$ можно проинтегрировать (31):

$$\begin{aligned}\langle m(t)\rangle = \langle m(0)\rangle - \frac{t}{\tau}\sum_{i=1}^{n} f_i(1-f_i) - \\ - \frac{1}{N}\sum_{i=1}^{n}(\langle N_i(0)\rangle - f_i N)(1-f_i)\left(1-e^{-t/\tau}\right).\end{aligned} \tag{32}$$

Из этого выражения мы можем вывести трансцендентное уравнение для времени ментального выживания T_m, полагая

$$\langle m(T_m)\rangle = 0, \tag{33}$$

$$\begin{aligned}T_m = \frac{\tau\langle m(0)\rangle}{\sum_{i=1}^{n} f_i(1-f_i)} - \frac{\tau}{N\sum_{i=1}^{n} f_i(1-f_i)}\times \\ \times\sum_{i=1}^{n}(\langle N_i(0)\rangle - f_i N)(1-f_i)\left(1-e^{-T_m/\tau}\right).\end{aligned} \tag{34}$$

Если $\langle m(0)\rangle$ достаточно велико, то

$$T_m \cong \frac{\tau\langle m(0)\rangle}{\sum_{i=1}^{n} f_i(1-f_i)}. \tag{35}$$

Из (35) следует, что если распределение предложение еды f стремится к сосредоточенному в одной клетке: $f_i \to \delta_{ik}$, то $T_m \to \infty$. Это означает, что если еда предлагается в единственной клетке, то левополушарные агенты немедленно занимают ее и могут поддерживать свой ментальный ресурс навсегда не тронутым. Заметим, что общее выживание будет гарантировано, только если физическое выживание будет обеспечиваться предложением еды.

Взаимодействие агентов

Мы можем дальше развить модель, предполагая что агент может принять во внимание предложение другому агенту перед принятием решения. Это

может интерпретироваться, как если бы агент α «взаимодействовал мысленно» с другими агентами. Мы предположим, что такое взаимодействие является парным, так что любой агент может принять во внимание ситуацию только с одним другим случайно выбранным агентом, включая самого себя. Конкретно, мы будем предполагать, что если еда предлагается агенту α, то *этот агент воображает, что она также предлагается агенту* β — см. рис. 1. В согласии с Лефевром, мы будем также предполагать, что агент может рассматривать два типа отношений с другим агентом: дружественное и конкурентное.

Теперь решение агента α зависит как от предложения среды a ему, так и от предложения ее b агенту β — *с точки зрения агента* α *та же самая единица физического ресурса предлагается и агенту* β. Теперь намерение агента α становится функцией двух переменных (см. рис. 1):

$$\psi = \psi(a, b). \tag{36}$$

Мы будем предполагать, что функция, описывающая решение правополушарного агента, совпадает с той , что описывает интенцию человека, относимого Лефевром *ко второй этической системе* [29]:

$$\psi_R = a^{a * b}, \tag{37}$$

где $* = +$, если *правополушарный агент* полагает, что агент β его *друг* и $* = \cdot$, если он полагает, что агент β его *враг*.

Решение (намерение) левополушарного агента будет определено в соответствии со случаем отсутствия взаимодействий путем логического отрицания выражения в показателе экспоненты. Таким образом,

$$\psi_L = a^{\overline{a * b}}. \tag{38}$$

Здесь, в соответствии с лефевровским определением, относящимся к агентам, принадлежащим к *первой этической системе* (которая в нашей модели отождествляется с левополушарной доминантностью), мы полагаем что $* = \cdot$, если *левополушарный агент* α считает, что агент β его *друг*, и $* = +$, если он полагает, что агент β его *враг*.

Заметим, что если вид намерения правополушарного агента в нашей системе *совпадает* с таковым для индивидуума, принадлежащего второй этической системе, введенным Лефевром, то форма намерения левополушарного агента *отличается* от решения агента, принадлежащего первой

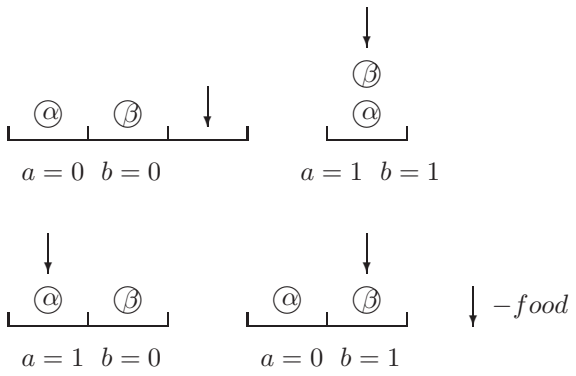

Рис. 1. Четыре различных случая предложения еды и соответственных параметров, описывающих предложения среды, a и b, агентам α и β, соответственно

этической системе. Это различие выражается в логическом отрицании выражения, находящегося в показателе экспоненты (38).

Рассмотрим намерения двух типов агентов как *функции двух переменных*, a и b.

Для правополушарного агента α, который рассматривает β как своего друга, мы получаем значения функции, приведенные в третьем столбце табл. 2. Посмотрим на вторую строку этого столбца: правополушарный агент следует предложению среды употребить еду в новой клетке со снижением ментального ресурса (действуя по сути как левополушарный агент), если только среда не требует сделать это его друга, с которым данный агент α ментально взаимодействует. То есть, правополушарный агент меняет свою клетку и идет в другую клетку (где ему предлагается еда), если в этой новой клетке уже находится его друг, с которым агент α ментально взаимодействует. Будем называть такую стратегию правополушарного агента *«идти к другу»*.

Для правополушарного агента, который рассматривает агента β как врага, мы получаем решения, представленные в четвертом столбце табл. 2.

Видно, что враги *совершенно не влияют* на намерение правополушарно-

Таблица 3. Решения правополушарного агента, который учитывает предложение другу (3 столбец) и врагу (4-й столбец)

a	b	a^{a+b}	$a^{a\cdot b}$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

Таблица 4. Решения левополушарного агента, который учитывает предложение другу (3 столбец) и врагу (4-й столбец)

a	b	a^{a+b}	$a^{a\cdot b}$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	1

го агента. Можно заключить, что в описываемой модели правополушарные агенты учитывают только ситуацию со своими друзьями.

Для левополушарного агента, который ментально взаимодействует с другом (с его точки зрения), мы получаем (третий столбец табл. 3), что друзья левополушарного агента не влияют на его решение. Наоборот, для левополушарного агента, который учитывает ситуацию с врагом, решения представлены в четвертом столбце табл. 3. Вновь, обращая внимание на вторую строку этого столбца, мы заключаем, что левополушарный агент действует как правополушарный, не следуя предложению среды употребить еду в другой клетке, если только среда не требует от его врага сменить его клетку (так что еда предлагается как раз в клетке, которую занимает случайно выбранный враг, с которым агент α ментально взаимодействует). Другими словами, левополушарный агент не идет в новую клетку за едой, если его враг уже в ней. Будем называть такую стратегию левополушарного агента *«не присоединяться к врагу»*. Можно заключить, что левополушар-

ные агенты принимают во внимание только ситуацию с врагами.

Интересно заметить, что решение левополушарного агента может быть представлено в более простой форме, которая не требует лефевровского использования различных операций для описания дружественных и конкурентных отношений в различных этических системах. Используя знаки $+$ и $\cdot$ для них (как для правополушарного агента), функция $a^{\overline{a\cdot b}}$ может быть заменена тождественной ей функцией $a^{\overline{a}+\overline{b}}$, а функция $a^{\overline{a+b}}$ — тождественной ей функцией $a^{\overline{a}\cdot\overline{b}}$. Эти новые формы могут быть легко интерпретированы как такие, в которых левополушарный агент представляет, что как он, так и агент, с которым он взаимодействует, отвергают предложения среды.

Важно отметить, что описанные правила ментального взаимодействия агентов могут быть подтверждены лишь при использовании данной модели для описания реальных явлений. Главное оправдание для них состоит в том, что эти правила соответствуют правилам, сформулированным Лефевром для второй этической системы (что нашло экспериментальные подтверждения [29]) и гипотезе о том, что только левое полушарие может осуществлять логические операции.

Самовоздействие агентов

Нам следует также рассмотреть ситуацию, когда агент взаимодействует сам с собой при принятии решения. Напомним, что функция $\psi(a,b) = a^{a+b}$ описывает ситуацию, когда правополушарный агент α представляет себе ситуацию, в которой он находится в дружественных отношениях с агентом β (ментально взаимодействует с ним). Если агент α выбирает $\beta = \alpha$, то он может представить ситуацию взаимодействия с собой. Поскольку трудно представить, что этот агент находится в дружественных или конкурентных отношениях с самим собой, то естественно рассмотреть в качестве такого взаимодействия *рефлексию* агента.

Для правополушарного агента такую рефлексию можно определить как

$$\psi = a^{a^a}. \tag{39}$$

Это выражение означает, что правополушарный агент воображает, что он представляет себя принимающим предложение среды. Очевидно, что для такого рефлексирующего (взаимодействующего с собой) агента

$$\psi = a + \overline{a^a} = a + \overline{a + \overline{a}} = a + \overline{1} \equiv a. \tag{40}$$

Следовательно, такой агент действует как левополушарный агент! Таким образом, *взаимодействие правополушарного агента с самим собой превращает его в левополушарного агента*. Мы будем считать это единственным возможным следствием рефлексии правополушарного агента, следующего из его неспособности строить логические отрицания.

Напротив, мы будем предполагать, что в отличие от правополушарного агента рефлексирующий левополушарный может *либо* оставаться таковым, *либо* вести себя как правополушарный, то есть в модели будет существовать очевидная асимметрия между рефлексирующими правополушарными и левополушарными агентами: рефлексирующий правополушарный агент действует как левополушарный, но рефлексирующий левополушарный остается левополушарным или становится правополушарным.

Чтобы оправдать такую асимметрию и двойственность самовоздействия левополушарного агента рассмотрим еще одну интерпретацию ключевой для нас функции импликации.

Об общем случае взаимодействия агентов

Ранее мы рассмотрели простейшие случаи динамики популяций невзаимодействующих агентов, имеющих различное доминирование полушарий. Для ментально взаимодействующих агентов аналогичное рассмотрение (относящееся, например, к выживанию агентов) является более сложным и мы планируем рассмотреть эту проблему в дальнейшем. Здесь же лишь отметим, что выживание агентов зависит от взаимодействия распределения агентов по клеткам и распределения предложения еды по клеткам. Оно также сильно зависит от конкретной структуры отношений между агентами (дружественных или конкурентных). Качественно, чем больше агентов рассматриваются данным правополушарным агентом как друзья, тем больше его мобильность в клеточном мире и больше шансов избегнуть физической гибели. С другой стороны, чем больше агентов рассматриваются данным левополушарным агентов в качестве врагов, тем больше вероятность отвергнуть предложение еды и больше шансов сохранить свой ментальный ресурс. В общем случае, при произвольных отношениях между агентами, многоагентная система может быть изучена главным образом путем компьютерного моделирования, а не аналитического рассмотрения.

Можно увидеть, как ментальное взаимодействие может усложнить свойства модели, рассматривая простейшие примеры, проиллюстрированные рис. 2. В первом случае два правополушарных агента, которые находятся

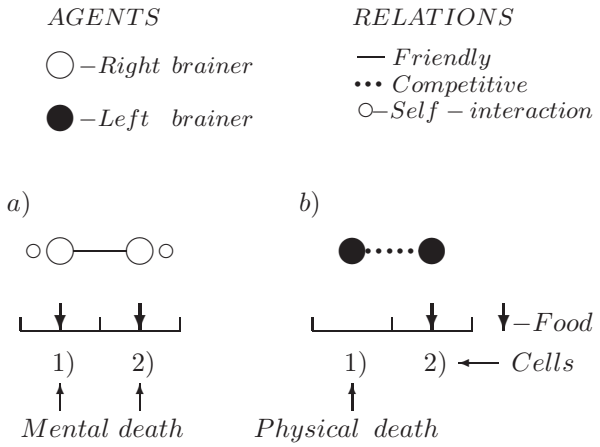

Рис. 2. **(a)** Два дружественных (оба агента рассматривают друг друга как друзей) рефлексивных правополушарных агента, живущих в двухклеточном мире, проявляют качественно поведение невзаимодействующих левополушарных агентов, если еда предъявляется равновероятно в обеих клетках. Эти агенты могут погибнуть вследствие недостатка ментального ресурса. **(b)** Левополушарный агент, занимающий первую клетку, в которой еда не предлагается, может погибнуть вследствие недостатка физического ресурса (как невзаимодействующие правополушарные агенты), если он находится в конкурентных отношениях с другим левополушарным агентом, занимающим вторую клетку, где еда предлагается. Заметим, что в этом случае мы исключаем самовоздействие агентов, так что агент из первой клетки не имеет шансов попасть во вторую.

в дружественных отношениях друг с другом, занимают две клетки (например, вначале один агент в клетке, как показано на рис. 2а). Если еда предлагается в обеих этих клетках, например, с одинаковой вероятностью, то поведение этих агентов будет качественно очень похоже на поведение невзаимодействующих левополушарных агентов. Таким образом, эти правополушарные агенты могут умереть вследствие истощения их ментальных ресурсов. В некотором смысле такое поведение выглядит неразумным, поскольку предложение пищи может дать им в некоторых случаях выжить физически без смены клетки пребывания. Во втором случае мы рассматриваем двух левополушарных агентов. Если агент, изначально занимающий первую клетку, находится в конкурирующих отношениях с другим левополушарным агентом, который занимает вторую клетку и еда предлагается только в этой последней клетке, то при отсутствии самовзаимодействия поведение первого агента будет аналогично поведению правополушарного агента в модели без взаимодействия. Например, агент, занимающий первую клетку, может погибнуть из-за нехватки физического ресурса. Заметим, что влияние самовоздействия (рефлексии), которое действительно дает агентом в популяции минимальную мобильность, уменьшается с ростом числа N агентов в популяции. Несмотря на сложность общей модели ментально взаимодействующих агентов, интересно рассмотреть ее предельные случаи, которые ведут нас к квантовой статистике.

Правополушарная стратегия: Распределение Бозе-Эйнштейна

Рассмотрим сообщество агентов с правополушарной доминантностью, которые находятся в дружественных отношениях друг с другом[5]. Мы уже рассматривали случай невзаимодействующих правополушарных агентов и показали, что они занимают свои первоначальные клетки в любой момент времени t. Появление взаимодействия позволяет правополушарному агенту сменить свою клетку, если:

- он ментально взаимодействует с другом, который уже занимает клетку, в которой появляется еда — *идти к другу*;
- он ментально взаимодействует сам с собой — это рефлексивное взаимодействие превращает его поведение в левополушарно-подобное и позволяет сменить первоначальную клетку.

[5]Мы уже отмечали, что такие отношения типичны для правополушарных агентов.

Предположим, что агент α занимает клетку j, но еда предлагается в клетке i. Пусть агент α ментально взаимодействует со случайно выбранным агентом (включая себя). Вероятность того, что он выберет агента, который занимает как раз клетку i составляет $\langle N_i \rangle / N$, в то время как вероятность того, что он выберет себя составляет $1/N$. Таким образом, полная вероятность для агента α перейти в клетку i составляет

$$p_{j \to i} = (\langle N_i \rangle + 1)/N. \tag{41}$$

Это в точности известная вероятность для занятия состояния i бозонами. Следовательно, ее применение ведет к Бозе-Эйнштейновскому распределению агентов в клеточном пространстве.

Действительно, так как вероятность появления еды в ячейке i составляет f_i, то вероятность агенту α занять новую клетку i в интервале времени $[t, t + \Delta t]$ пропорциональна $(\langle N_i(t) \rangle + 1) f_i$. После нормировки она может быть записана как

$$p_{j \to i} = \frac{(\langle N_i(t) \rangle + 1) f_i}{N}. \tag{42}$$

В состоянии равновесия, как это следует из *принципа детального баланса*, скорость обмена между двумя клетками i и j должна быть одинаковой:

$$\langle N_j \rangle (\langle N_i \rangle + 1) f_i = \langle N_i \rangle (\langle N_j \rangle + 1) f_j, \tag{43}$$

или, принимая во внимание (20),

$$\frac{\langle N_i \rangle}{\langle N_i \rangle + 1} e^{\epsilon_i / \theta} = \frac{\langle N_j \rangle}{\langle N_j \rangle + 1} e^{\epsilon_i / \theta}. \tag{44}$$

Так как последнее соотношение выполняется для любых i и j, то выражение в левой части является постоянной, $e^{\mu/\theta}$, где μ — это химический потенциал, то

$$\frac{\langle N_i \rangle}{\langle N_i \rangle + 1} e^{\epsilon_i / \theta} = e^{\mu/\theta}. \tag{45}$$

Из последнего выражения немедленно следует, что

$$\langle N_i(\epsilon_i) \rangle = \frac{1}{e^{(\epsilon_i - \mu)/\theta} - 1}. \tag{46}$$

Левополушарная стратегия: Распределение Ферми-Дирака

Рассмотрим теперь сообщество левополушарных агентов, которые находятся в конкурентных отношениях друг с другом. Согласно *Вильямсу* [22], именно *конкурентные* отношения типичны для левополушарных персон. С другой стороны, дружественные отношения, как мы говорили, не влияют на решения левополушарных агентов. Также, мы уже рассмотрели ранее случай невзаимодействующих левополушарных агентов и показали, что в равновесии они достигают распределения Гиббса. Появление ментального взаимодействия позволяет левополушарному агенту удержать свою клетку если он взаимодействует ментально с врагом, который уже занимает клетку, где предлагается еда (*не иду к врагу*).

Предположим, что агент α занимает клетку j, а еда предлагается в клетке i. Далее нужно будет рассмотреть два случая самовоздействия агента.

Вначале предположим, что такое самовоздействие оставляет левополушарного агента левополушарным. Пусть далее агент α случайно выбирает агента (включая себя) для ментального взаимодействия. Вероятность, что он выберет агента не занимающего клетку i составляет $(N - \langle N_i \rangle)/N$. Поэтому, вероятность агенту α перейти в клетку i есть также

$$p_{j\to i} = \frac{N - \langle N_i \rangle}{N}. \tag{47}$$

Заметим, что агент α может выбрать себя в качестве партнера для ментального взаимодействия. Но это, по предположению, не может изменить его намерения перейти в клетку i.

Вновь, так как вероятность появления еды в клетке i составляет f_i, то вероятность агенту α занять новую клетку i в интервале времени $[t, t+\Delta t]$ равна

$$p_{j\to i} = \frac{(N - \langle N_i(t) \rangle) f_i}{N}. \tag{48}$$

Используя вновь принцип детального баланса, мы учитываем, что в равновесии скорости обмена между двумя клетками, i и j, должны равняться:

$$\langle N_j \rangle (N - \langle N_i \rangle) f_i = \langle N_i \rangle (N - \langle N_j \rangle) f_j \tag{49}$$

или

$$\frac{\langle N_i \rangle}{N - \langle N_i \rangle} e^{\epsilon_i/\theta} = \frac{\langle N_j \rangle}{N - \langle N_j \rangle} e^{\epsilon_i/\theta}. \tag{50}$$

Как и ранее, поскольку последнее соотношение удовлетворяется для любых i и j, выражение в левой части постоянно, $e^{\mu/\theta}$. Таким образом,

$$\frac{\langle N_i \rangle}{N - \langle N_i \rangle} e^{\epsilon_i/\theta} = e^{\mu/\theta}. \tag{51}$$

Из последнего уравнения следует, что

$$\langle N_i(\epsilon_i) \rangle = \frac{N}{e^{(\epsilon_i - \mu)/\theta} + 1}. \tag{52}$$

Таким образом, в равновесии сообщество левополушарных агентов, которые находятся в конкурентных отношениях друг с другом, описывается обобщенным распределением Ферми-Дирака, в котором N фермионам дозволено оказаться в одном энергетическом состоянии. Можно считать это распределение обычным фермиевским с вырождением энергетического состояния $g = N$.

Если мы предположим, что самовоздействие левополушарного агента превращает его в правополушарного, то он не сможет переходить в иную клетку при таком самовоздействии. Это приведет к изменению выражения (47) на следующее

$$p_{j \to i} = \frac{N - (\langle N_i \rangle + 1)}{N} = \frac{(N-1) - \langle N_i \rangle}{N}. \tag{53}$$

Рассуждения, аналогичные только что приведенным, дают следующее распределение агентов в этом случае:

$$\langle N_i(\epsilon_i) \rangle = \frac{N-1}{e^{(\epsilon_i - \mu)/\theta} + 1}. \tag{54}$$

Это — распределение Ферми-Дирака, в котором $N-1$ фермионам дозволено оказаться в одном энергетическом состоянии. Можно считать, как и ранее, это распределение обычным фермиевским с вырождением энергетического состояния $g = N - 1$.

Такое тонкое различие двух распределений нам понадобится в дальнейшем.

Если термодинамическое равновесие может быть достигнуто за времена, за которые ни один агент не погибает вследствие истощения физического или ментального ресурса, то величина химического потенциала μ может быть определена при использовании нормировки, как для сообществ

левополушарных агентов, так и для ранее рассмотренного сообщества правополушарных агентов:

$$\sum_{i=1}^{n} \langle N_i(\epsilon_i) \rangle = N. \tag{55}$$

Это соотношение отражает сохранение числа агентов.

Для правополушарных агентов, описываемых Бозе-Эйнштейновским распределением, мы получаем соотношение, из которого может быть найдено значение химического потенциала

$$\sum_{i=1}^{n} \frac{1}{e^{(\epsilon_i - \mu)/\theta} - 1} = N. \tag{56}$$

Очевидно, что это значение, в общем случае, зависит от температуры.

Для левополушарных агентов, подчиняющихся статистике Ферми-Дирака, мы получаем в случае сохранения левополушарности при самовоздействии

$$\sum_{i=1}^{n} \frac{1}{e^{(\epsilon_i - \mu)/\theta} + 1} = 1. \tag{57}$$

Во втором случае выражение похоже.

Заметим, что принцип детального баланса утверждает, что если $\langle N_1 \rangle$ и $\langle N_2 \rangle$ представляют собой среднее число состояний, помеченных 1 и 2, то число переходов из 1 в 2 должно равняться числу переходов из 2 в 1. Этот принцип сильнее, чем условие равновесия само по себе и глубок связан с микроскопической обратимостью и взаимностью Онзагера. Он может использоваться не только для вывода распределения Ферми-Дирака для *фермионов* и распределения Бозе-Эйнштейна для *бозонов*, но также и для получения промежуточных квантовых статистических распределений для *анионов*. Замечательно, что статистические распределения для анионов могут быть получены без использования теоремы о связи спина со статистикой.

Переключение полушарий и промежуточная квантовая статистика

Следует отметить некоторые дальнейшие возможные приложения данной модели. Очевидно, что при приближении к критическим режимам, когда физический или ментальный ресурс исчезает, агент может изменить свою

стратегию выживания, изменяя доминантное полушарие[6]. В то же время, *этическая система* агента не меняется. В нашем подходе более плодотворно рассмотреть *переключение стратегии* при удержании природы связей между агентами. Поэтому, динамика переключения полушарий может быть естественным образом введена в нашу модель Хорошо известно, что нарушения циклов доминирования полушарий рассматривается некоторыми авторами как источник различных ментальных расстройств.

Например, *Петтигрю* предположил, что уменьшение скорости бинокулярной конкуренции — такая конкуренция как раз связана с переключением доминантности полушарий — является индикатором биполярного расстройства [37][7].

Следовательно, динамика переключения полушарий может быть использована для моделирования, например, маниакально-депрессивного синдрома, гипотетически вызванного взаимодействием агентов. Это открывает путь к учету социальных условий при развитии ментальных расстройств.

С этой точки зрения интересно изучить наиболее интересный общий случай популяции, состоящей из агентов с различной полушарной доминантностью и найти ее равновесные состояния. Представляется, что в этом случае будут адекватны более общие формы квантовой статистики. Например, как это было показано *Хуангом* [40], если допускается трансмутация бозонов и фермионов (в нашем случае она в точности соответствует переключению доминантности полушарий), то система состоящая из бозонов и фермионов будет иметь функцию статистического распределения анионов.

Промежуточные типы статистики могут быть найдены не только в системах квантовых квазичастиц. Например, *Бьянкони* представила случай негомогенной растущей *сложной сети* с различными свойствами узлов, демонстрирующей смешанную квантовую статистику [41].

[6]Заметим, что в лефевровской *Алгебре совести* изменение отношений агентов (от дружественного к конкурентному и наоборот) рассматривается как средство повышение его *этического статуса*.

[7]Свидетельство в пользу того, что межполушарное переключение связано с острой депрессией, и что это заболевание может *инициироваться* или *подавляться* электрической стимуляцией лишь одной половины мозга представлено в [38]. Недавно *Додсон* значительно уменьшил симптомы мании у пациента путем введения холодной воды в левое ухо [39]. В общем случае, левое полушарие чрезмерно активировано при мании, в сравнении с чрезмерной активацией правого полушария при депрессии. Так как тепловая вестибулярная стимуляция эффективно действует на одно полушарие, мания могла бы быть излечена тепловой стимуляцией слева, которая увеличила бы активность справа и, поэтому, уменьшила бы манию.

Критическая проблема введения механизма трансмутации должна быть рассмотрена и она представляется чрезвычайно важной, поскольку дает возможность найти долю лево- и правополушарно доминантных агентов в многоагентной модели с переключением полушарий.

Следует отметить другое возможное направление развития модели. Несмотря на возникновение квантовоподобных статистик в только что представленной модели, мы предполагаем тем не менее, что агенты действуют строго классическим образом: *либо* как правополушарные, *либо* как левополушарные (с возможным переключением доминантности). Так что любое наблюдение квантовоподобного статистического поведения популяции агентов не будет означать квантовый характер самих агентов. Тем не менее, возможно обобщить эту *классическую модель* на *квантовую область*, предполагая что агенты могут находиться в суперпозиции:

$$\psi = \beta|right\rangle + \gamma|left\rangle, \tag{58}$$

где β и γ являются комплексными амплитудами события, при котором агент действует как право- или левополушарная персона, соответственно.

Такая модель может иметь некоторую связь с *двусмысленной статистикой*, рассмотренной *Медведевым* [42]. В этом виде статистики все частицы имеют неизвестный тип. Это может быть следствием, например, осцилляции типа частицы, когда частицы являются в данный период бозонами, но затем превращаются в фермионы *и наоборот*[8]. В ходе парного взаимодействия частица распознает тип другой частицы (*и наоборот*) и взаимодействует с ней согласно ее выявленному типу.

Если вероятность того, что частица будет распознана как бозон (фермион) есть $p_b(p_f)$, то эти частицы будут подчиняться статистике анионов, которая может быть выведена с помощью деформированного коммутационного соотношения

$$a_i a_j^\dagger - q a_j^\dagger a_i = \delta_{ij}, \tag{59}$$

где $q = (p_b - p_f)/(p_b + p_f)$. Конечно, важно исследовать, обладают ли подобные обобщения представленной модели какими-либо новыми и *экспериментально проверяемыми* свойствами. Но этот вопрос выходит за рамки данной статьи.

[8]Таким образом, осцилляции доминирования полушарий могут быть описаны естественным образом.

Заключение

Итак, мы показали, как можно сформулировать многоагентную модель, описывающую популяции агентов с различной доминантностью полушарий мозга и аргументировали, что такие популяции могут подчиняться известным квантовым статистикам и могут также в потенциале описываться промежуточными квантовыми статистиками. Это указывает на возможность применения квантовой статистики к изучению социальных и экономических явлений.

Применение формализма Лефевра к построению многоагентных экономических агентов могло бы, как и отмечал Лефевр, внести в них реалистический элемент, связанный с наличием у агента не только экономических, но и внеэкономических мотивов поведения. Задача построения таких моделей не кажется невозможной и надуманной. Но пользу такого подхода можно будет оценить только в ходе их детальной разработки и применения.

Автор надеется, что это может явиться некоторым аргументом в пользу того, что построение систем рефлексирующих агентов полезно для мировой экономики.

Литература

1. Дискуссия о нейрокомпьютерах — 10 лет спустя. – М.: МИФИ, 2000.
2. *Ежов А. А.* Что такое эконофизика? // В сб.: *Физическая экономика + эконофизика = ЭкономИФИзика.* – М.: МИФИ, 2006, с. 16–24.
3. *Ezhov A. A., Khrennikov A. Yu.* Agents with left and right brain dominant hemispheres and quantum statistics // *Physical Review E*. – **71**, 016138, 2005.
4. Actes de la tables ronde *S.Carnot et l'essor de la termodynamique*. – Paris, 1976.
5. *Mantegna R. N., Stanley H. E.* An introduction to Econophysics. – Cambridge, 2000.
6. *Foley D.* The strange history of the economic agent // *The New School Economic Review*. – **1**, 2004.
7. *Mirowski F.* More heat than light. – Cambridge University Press, 1989.
8. *Zaostrovtsev A.* The principal conflict in contemporary Russian economic thought: Traditional approaches against economics. – HWWA *Discussion Paper 329*, Hamburg Institute of International Economics, 2005.
9. *Mirowski F.* Machine dreams. Economics becomes a cyborg science. – Cambridge University Press, 2001.
10. *Фейнман Р.* Характер физических законов. – М.: Мир, 1968.
11. *Keen S.* Debunking economics. A naked emperor of the social sciences. – Zed Books, 2002.

12. *Jinshan W., Zengru D., Yang Z. R.* Division of labor as the result of phase transition // *Physica A*. – **323**, pp. 663–676, 2003.

13. *Yasutomi A.* The emergence and collapse of money // *Physica D*. – **82**, pp. 180–194, 1995.

14. *Bouchad J.-P., Mézard M.* Wealth condensation in a simple model of economy // *Physica A*. – **282**, p. 536, 2000.

15. *Scafetta N., Picozzi S., West B. J.* An out of equilibrium model of the distributions of wealth // *Quantitative Finance*. – **4**, p. 353, 2004.

16. *Meyer D. A.* Quantum strategies // *Phys.Rev. Lett.* – **82**, pp. 1052–1055, 1999.

17. *Маслов В. П.* Квантовая экономика. – М.: Наука, 2005.

18. *Лефевр В. А.* Алгебра совести. – М.: Когито-Центр, 2002.

19. *Лефевр В. А.* Просчеты миротворчества // *Рефлексивные процессы и управление*. – том 2, № 2, 2002. – с. 48–51.

20. *Лефевр В. А.* Где искать истоки демографического кризиса? // *Независимая газета*, 22 ноября 2000 г.

21. *Павловский Г.* К теогонии братвы // *Русский журнал*, 10 ноября 1997 г.

22. *Вильямс Б.* Торговый хаос. – М.: Аналитика, 2005.

23. *Кругман П.* Великая ложь. – М.: АСТ, 2004.

24. *Петерс Э.* Хаос и порядок на рынках капитала. – М.: Мир, 2000.

25. *Маслов В. П.* Нелинейное среднее в экономике // *Математические заметки*. – **78**, pp. 377–395, 2005.

26. *Ezhov A. A., Khrennikov A. Yu.* On ultrametricity and symmetry between Bose-Einstein and Fermi-Dirac systems // *AIP Conf. Proc.* – **826**, issue 1, pp. 55–64, 2006.

27. *Epstein J. M., Axtell R. L.* Growing artificial societies — social science from the bottom up. – Brookings Institution Press, Washington; MIT Press, Cambridge, MA, 1992.

28. *Grothmann R.* Multi-agent market modeling based on neural networks. – PhD Thesis, University of Bremen, 2002.

29. *Lefebvre V. A.* Algebra of conscience. – Kluwer Academic Publisher, 2001.

30. *Rotenberg V. S., Arshavsky V. V.* // *Homeostasis*. – **38**, No. 2, p. 49 (1997).

31. *Evans M. R.* // *Europhys. Lett.* – **36**, p. 13 (1996).

32. *Bianconi G., Barabasi A.-L.* // *Phys. Rev. Lett.* – **86**, p. 5632 (2001).

33. *Staliunas K.* // e-print cond-mat/0001347 (2000).

34. *Bianconi G.* // *Phys. Rev. E*. – **66**, 036116, (2002).
35. *Derrida B., Lebowitz J. L.* // *Phys. Rev. Lett*. – **80**, 209 (1998).
36. *Acharya R., Narayana Swamy P.* // *J. Phys. A: Math. Gen.* – **37**, 2527 (2004).
37. *Pettigrew J. D., Miller S. M.* // *Proc. Roy. Soc*. – **B 265**: 2141A (1998).
38. *Bejjani B.-P. et al*. // *New England J. Medicine*. – **340**, 1476 (1999).
39. *Dodson M. J.* // *Neurol. Neurosurg. Psychiatry*. – **75**, 163 (2004).
40. *Wung-Hong Huang* // *Phys. Rev. E*. – **51**, 3729 (1995).
41. *Bianconi G.* // *Phys. Rev. E*. – **66**, 056123 (2002).
42. *Medvedev M. V.* // *Phys. Rev. Lett*. – **78**, 4147 (1997).
43. *Mézard M., Parisi G., Sourlas N., Toulouse G., Virasoro M. A.* // *Phys. Rev. Lett*. – **52**, 1156 (1984).
44. *Parisi G., Ricci-Tersenghi F.* // *J. Physics A*. – **33**, 113 (2000)
45. *Mantegna R. N.* // *European Physical Journal B*. – **11**, 193 (1999).
46. *Murtagh F.* Identifying the ultrametricity of time series.
47. *Rosu H. C., de la Cruz F. A.* // *Physica Scripta*. – **65**, 377 (2002)
48. *Rammal R., Toulouse G., Virasoro M. A.* // *Reviews of Modern Physics*. – **58**, 765 (1986).
49. *Lerman I. C.* Classification et analyse ordinale de donneés. – Paris: Dunod, 1981.
50. *Murtagh F.* // *J. of Classification*. – **21**, 167 (2004).
51. *Ezhov A. A., Khrennikov A. Yu.* On ultrametricity and a symmetry between Bose-Einstein and Fermi-Dirac systems // In: *A.Yu.Khrennikov, Z.Rakic,* and *I.V.Volovich*, Eds. *p-adic mathematical physics, AIP Conf. Proc.*, vol.826, pp. 55–64, 2006.
52. *Cocchini G., Della Sala S., Beschin N.* Assessment of anosognosia for hemiplegia.
53. *Keenan J. P., Nelson A., O'Connor M., Pascual-Leone A.* Self-recognition and the right hemisphere // *Nature*. – 409 (18), 305 (2001).
54. *Turk D. J., Heatherton T. F., Kelley W. M., Funnell M. G., Gazzaniga M. S., Macrae N. C.* Mike or me? Self-recognition in a split-brain patient // *Nature Neuroscience*. – **5** (9), pp. 841–842 (2002).

Александр Александрович ЕЖОВ, кандидат физико-математических наук, начальник лаборатории квантовых нейронных систем Троицкого института инновационных и термоядерных исследований (ТРИНИТИ). Области научных интересов — теория переноса нейтронов, нейронные сети, квантовые вычисления и эконофизика. Автор более 60 научных публикаций.

Н. Г. МАКАРЕНКО
Главная астрономическая обсерватория РАН, Санкт-Петербург;
Институт математики, Алма-Ата, Казахстан
E-mail: ng-makar@mail.ru, makarenko@math.kz

СТОХАСТИЧЕСКАЯ ДИНАМИКА, МАРКОВСКИЕ МОДЕЛИ И ПРОГНОЗ

Аннотация

Лекция представляет новый метод марковского предсказания временных рядов. Он основан на инвариантной мере случайной динамики, которая реализуется Системой Итеративных Функций (IFS) — сжимающих отображений, снабженных вероятностями. Параметры этой системы получаются в результате решения обратной задачи на основе оценки эмпирической меры из временных рядов.

N. G. MAKARENKO
Pulkovo Astronomical Observatory, St-Petersburg;
Institute of Mathematics, Kazakhstan, Alma-Ata
E-mail: ng-makar@mail.ru, makarenko@math.kz

STOCHASTIC DYNAMICS, MARKOVIAN MODELS AND FORECAST

Abstract

The lecture gives a new method of Markovian prediction of time series. It is based on invariant measure of stochastic dynamics, which is implemented as Probabilistic Iterated Function System (PIFS), i. e. a set of contractive maps with probabilities. Parameters of the system are obtained as a result of the inverse problem solution using the time series empirical measure.

УДК 001(06)+004.032.26 (06) Нейронные сети

Введение

> Сто тысяч миллионов до неба, умноженные на отсюда и до фига дают представление об идее, которую мы попытались до вас донести.
>
> *Дуглас Адамс*
> *«Автостопом по Галактике»*

В силу сложившихся традиций мы обычно пытаемся упорядочить свои впечатления о наблюдаемом Мире в форме моделей. Самые простые из них позволяют интерпретировать эксперимент в форме схемы, логической или математической, редуцируя наблюдаемое многообразие Фактов к возможно меньшему числу параметров порядка. Такие модели-интерпретаторы отвечают, или точнее, формализуют вопрос: *«Как это происходит?»* Более продвинутые модели пытаются ответить на вопрос: *«Почему это происходит именно так, а не иначе?»* В любом варианте мы хотим, чтобы модель была не только «объяснительной», но и позволяла бы увидеть чуть дальше «кончика носа» — интервала времени, по которому она строилась. Иными словами, модель должна предсказывать.

Предсказывать поведение динамической системы, вообще говоря, интересно как вперед (Prediction), так и назад во времени (Postdiction). Однако, в присутствии диссипации и конечной точности начальных данных реализовать Postdiction, по-видимому, гораздо сложнее, если это вообще можно сделать. Во-первых, существует слишком много возможных сценариев *Прошлого*, которые могли в принципе закончится существующим *Настоящим*, и все они для нас *равновозможны*. Во-вторых, инверсия даже линейного предиктора приводит к NP-сложным алгоритмам [1].

Современная техника детерминированного нелинейного Prediction временных рядов, о которой я подробно рассказывал в одной из своих прошлых лекций [2], основана на топологических моделях, которые получаются из наблюдений. Реконструкция представляет собой дифференцируемое вложение временного ряда в евклидово пространство подходящей размерности. При этом сам временной ряд рассматривается как *платоновская тень* фазовой траектории аттрактора системы на произвольное направление. Символы веры, необходимые для корректной реализации такой процедуры, сводятся к предположению о существовании абстрактной гладкой

динамической системы с компактным низкоразмерным аттрактором. Предиктор строится по полученной многомерной копии аттрактора.

Его локальный вариант называют *методом аналогов*. Этот метод формализует библейский тезис *«Что было то и будет»*[1] [2]. Другими словами, мы находим в *Прошлом* нашей реконструкции фрагменты траекторий, которые похожи (аналогичны) последнему известному нам фрагменту. После этого выписываем их динамические продолжения, т. е. их «будущее в прошлом». Локальное предсказание получается усреднением вариантов *déjà vu*[3].

Глобальный предиктор основан на всей непрерывной и известной истории ряда и представляет собой нелинейную авторегрессионную модель с памятью, равной размерности вложения. Фактически, мы здесь имеем дело с нелинейным отображением векторов реконструкции, известных из истории ряда, в их последующие по времени значения. Аппроксимация неизвестной нелинейной функции многих переменных может быть получена разными методами.

Искусственные нейронные сети, возможно, являются лучшим аппроксиматором, обученным такому отображению. Однако, в общем случае квадратичного функционала ошибок и конечной обучающей выборки задача аппроксимации не является корректной. Поэтому выбор варианта, претендующего на роль *реального* из множества альтернативных равновозможных решений, всегда является делом весьма интимным даже при наличии хорошего регуляризатора. Для выбора варианта нередко используют даже эстетический принцип: *«Возможно, это хорошо выглядит»*.

Совсем иная ситуация возникает, когда у нас нет оснований полагать, что динамика исходной системы является детерминированной. По разным причинам, возникают большие сомнения в выполнимости *Кредо идеального экспериментатора* [2], необходимого для топологической реконструкции. В этом случае приходится использовать стохастические модели и вероятностный прогноз.

Я позволю себе привести здесь цитату из замечательного эссе *Станислава Лема*, которая очень точно описывает двусмысленность недетерми-

[1]Если ссылка на примечание представляет собой число, заключенное в круглые скобки, например, (3), то она обозначает номер примечания, помещенного в конце данного текущего раздела. Ссылка в виде числа, не заключенного в скобки — обычное подстраничное примечание.

[2]Екл. 1:9.

[3]Уже виденное.

нированной ситуации[4]: *«К понятию вероятности мы прибегаем, когда не знаем чего-либо с полной уверенностью. Но неуверенность эта носит либо чисто субъективный характер (я не знаю, что произойдет, но кто-то другой, возможно, и знает), либо объективный (никто не знает и знать не может). Субъективная вероятность — это протез при информационном увечье; не зная, какая лошадь возьмет приз, и угадывая лишь по числу лошадей (если их четыре, то у каждой один шанс из четырех на победу), мы поступаем как слепой в комнате, заставленной мебелью. Вероятность подобна трости слепца, которой он нащупывает дорогу. Если бы он видел, то не нуждался бы в палке, а если бы я знал, какая лошадь резвее всех, то не нуждался бы в теории вероятностей»*.

К счастью, стохастические модели, основанные на *марковских* процессах, названных в честь А. А. Маркова (1), не зависят от того, что именно характеризует непредсказуемость будущих событий: наши знания о них или сами эти события. Марковский процесс строится на принципе: *будущее зависит от прошлого только через настоящее*.

На формальном языке условная вероятность наблюдать «завтра» некоторое значение x_{n+1}, при условии что ему предшествовала последовательность значений $x_n, x_{n-1}, \ldots, x_0$, редуцируется к условной вероятности, зависящей лишь от наблюдаемого «сегодняшнего» значения x_n:

$$p\big(x_{n+1}|x_n, x_{n-1}, \ldots, x_0\big) = p\big(x_{n+1}|x_n\big).$$

Простейший вариант марковского процесса реализуется AR-моделью, описывающей случайное смещение точки $Tx_n = (ax_n + \xi_n) \to x_{n+1}$, где случайная переменная ξ_n подчиняется гауссовскому распределению $\mathcal{N}(\xi, \sigma)$. Легко показать, что ее эволюция $x_n \to x_{n+1}$ полностью описывается условной (переходной) вероятностью

$$p\big(x_{n+1}|x_n\big) = p(x_{n+1}, x_n)/p(x_n),$$

где $p(x_{n+1}, x_n) = \mathcal{N}(x_{n-1} - ax_n, \sigma)$.

Рассмотренный пример иллюстрирует идею *случайной динамической системы*. Она получается, если вместо одного оператора T использовать случайно выбранные отображения из набора $T \equiv \{T_k\},\ k = 1, 2, \ldots, N$. Каждое $T_i : X \to X$ является непрерывным преобразованием на *компактном* подмножестве $X \in R^n$.

[4] *Ст. Лем.* О невозможности жизни. О невозможности прогнозирования / Вторжение с Альдебарана. – М.: Ин. Лит. 1960.

Случайная орбита такой системы — это просто последовательность точек

$$\{x_m\}_{m=0}^{\infty}: \quad x_m = T_{k_{m-1}} \circ \ldots \circ T_{k_1} \circ T_{k_0} x_0 \equiv T x_0,$$

полученных итеративным применением операторов из набора T. Каждый номер $k_i \in \{1, 2, \ldots, N\}$ может быть выбран независимо, либо с вероятностью, зависящей от номера k_{i-1} предшествующего отображения. В последнем случае мы получим как раз марковский процесс.

Разобьем фазовое пространство X для нашей случайной системы на конечное число клеток $\{A_1, \ldots, A_n\}$. Марковская модель задается $n \times n$ матрицей вероятностей

$$P_{ij} = \mathrm{Prob}_{\mu}\{Tx \in A_j | x \in A_i\},$$

описывающих переходы из клетки A_i в клетку A_j под действием итераций T. Основной проблемой является нахождение этих переходных вероятностей.

Вспомним, что вероятность имеет смысл только в контексте заданной вероятностной меры, которая помечена выше символом μ. К счастью оказывается, что для большинства орбит нашей случайной системы такая инвариантная мера существует! В частности, для каждой из клеток A_i она пропорциональна доле времени, которое проводит в клетке орбита, или, что то же самое, относительному числу точек орбиты, которые в ней «застряли».

Остается проблема поиска подходящих отображений T. Проще всего искать под фонарем, хотя иногда искомое может находиться под крышкой в кастрюле с супом! Такой кастрюлей в нашем случае является геометрия фракталов, о которой я рассказывал в своей лекции [3].

Вспомним, что фрактал является предельным образом рекуррентной динамики *Системы Итеративных Функций (`IFS`)* в пространстве компактов. Последовательные итерации сжатий приводят к единственной неподвижной точке — *фракталу* или *аттрактору `IFS`*. Если снабдить каждое из отображений вероятностями, случайная динамика `IFS` продуцирует единственную инвариантную (мультифрактальную) меру на аттракторе. Случайный алгоритм построения фракталов эквивалентен функционированию искусственной нейронной сети [4].

Предположим теперь, что нам задан аттрактор и мера на нем. Поиск `IFS` и их вероятностей составляет содержание *Обратной задачи* в теории фракталов. Она корректно поставлена и решается как оптимизационная

задача с ограничениями. Известные методы ее решения включают и генетический алгоритм.

Вот к такому необычному контексту приводит современный вариант марковского прогноза. Почти библейский случай: пошел *Саул, Кисов* сын, искать ослиц, а нашел *Царство*[5]. Попыткой понятного описания этого *Фрактального Царства* и является предлагаемая Лекция.

Я следовал форме моих прошлых лекций: каждый раздел сопровождается Примечаниями и Путеводителем по литературе. Последний раздел содержит, в качестве иллюстрации, практическое применение теории к предсказанию магнитных бурь. Эти результаты были получены совсем недавно моими коллегами *Л. М. Каримовой*, *О. А. Круглун* и *С. А. Мухамеджановой* и не были озвучены в устном варианте Лекции.

Я благодарен своей дочери и сотруднице *Ирине Макаренко*, без помощи которой текст вряд ли был бы когда-нибудь закончен.

Примечания

1. *Андрей Андреевич Марков-старший* (14.6.1856–20.7.1922) — выдающийся русский математик, отец математика *А. А. Маркова-младшего*. Родился в Рязани. В 1874 году поступил в Петербургский университет. В 1880 году защитил свою знаменитую магистерскую диссертацию «О бинарных квадратичных формах положительного определителя», а 1881 году — докторскую диссертацию. С 1880 года *А. А. Марков* — приват-доцент Петербургского университета. По предложению *Пафнутия Львовича Чебышева* избран адъюнктом (1886 г.), затем экстраординарным академиком (1890 г.)и ординарным академиком (1896 г.) В 1886 году А. А. Марков — экстраординарный профессор Петербургского университета, а в 1893 году — ординарный профессор. В 1905 году ушел в отставку в звании заслуженного профессора. В 1912 году по собственной просьбе был отлучен от церкви. В письме к Синоду он писал: «Я не усматриваю существенной разницы между иконами и идолами, которые, конечно, не боги, а их изображения».

[5]Книга царств 1:9.

Метрика и метрические пространства

МУ — единица расстояния в Индии. Означает предел слышимости мычания коровы.

Невероятно? Но факт!
Еш гвардия, Ташкент, 1971.

В этом разделе мы приведем некоторые общие определения, необходимые для работы в пространствах, которые мы будем использовать.

Метрикой в пространстве X называют функцию $d : X \times X \to R$. Эта запись означает, что каждой паре точек из $X \times X$ ставится в соответствие вещественное число d. Для всех $x, y \in X$ функция d удовлетворяет следующим требованиям:

1. $d(x, y) \geqslant 0$;
2. $d(x, y) = 0$ тогда и только тогда, когда $x = y$;
3. $d(x, y) = d(y, x)$ — свойство симметрии;
4. $d(x, z) \leqslant d(x, y) + d(y, z)$ — неравенство треугольника.

Если отменить условие (2), то мы получим *псевдометрику*, а если заменить свойство (4) на условие «остроугольности»

$$d(x, z) \leqslant \max\bigl(d(x, y), d(y, z)\bigr),$$

то получится *ультраметрика*. Она используется в некоторых задачах распознавания образов [5], когда пространства признаков имеют высокую размерность и приобретают необычные свойства (1).

Пара (X, d) называется *метрическим пространством*. Последовательность точек $x_1, x_2, \ldots, x_k, \ldots$ в (X, d) называют *фундаментальной*, если расстояния между точками уменьшаются, т. е. $d(x_i, x_j) \to 0$, когда $i, j \to \infty$. Если любая такая последовательность сходится к некоторой точке $x \in X$, т. е. $d(x_k, x) \to 0$, когда $k \to \infty$, то пространство (X, d) называют *полным*.

Полные метрические пространства имеют очень важное для нас свойство, которое выражается *теоремой Банаха* (2) или *Принципом сжимающих*

отображений: в полном метрическом пространстве любое сжимающее отображение имеет единственную неподвижную точку.

Напомним, что отображение $F : X \to R$ называют *сжимающим*, если существует такое число $r \in (0,1)$, что для любых двух точек $x_1, x_2 \in X$

$$d\big(F(x_1), F(x_2)\big) \leqslant r\, d(x_1, x_2).$$

Эквивалентным является следующее определение. Пусть (X, d) — полное метрическое пространство. Отображение $S : X \to X$ удовлетворяет *условию Липшица*, если для всех $x, y \in X$ существует такое число $c \geqslant 0$, что

$$d\big(S(x), S(y)\big) \leqslant c d(x, y).$$

Постоянной Липшица $Lip\, S$ для S называют минимальное c, для которого выполняется условие Липшица. Отображение S называют сжатием, если $Lip\, S < 1$.

Неподвижная точка a отображения F — это решение уравнения $F(x) = x$. Неподвижная точка притягивает последовательность точек, полученных итерациями F, т. е. начиная с любой начальной точки, последовательность

$$x,\ F(x),\ F^{\circ 2}(x) = F\big(F(x)\big),\ F^{\circ 3}(x) = F\Big(F\big(F(x)\big)\Big),\ \ldots$$

сходится к a. Используя неравенство треугольника (4), можно оценить скорость сходимости таких итераций к неподвижной точке.

Пусть $d(x, F(x)) \leqslant \varepsilon$ и r — коэффициент сжатия для F. Тогда

$$\begin{aligned} & d(x, F^{\circ k}(x)) \leqslant \\ & \leqslant d\big(x, F(x)\big) + d\big(F(x), F^{\circ 2}(x)\big) + \ldots + d\big(F^{\circ k-1}(x), F^{\circ k}(x)\big) \leqslant \\ & \leqslant d\big(x, F(x)\big)(1 + r + \ldots + r^{k-1}) \leqslant \varepsilon(1 - r^{k-1})/(1 - r). \end{aligned}$$

При $k \to \infty$ из последнего неравенства получаем оценку:

$$d(x,\ a) \leqslant \frac{\varepsilon}{1 - r}.$$

В стандартных курсах анализа теорема Банаха получается из утверждения, что график любой непрерывной функции, определенной в замкнутом интервале $[a, b]$ для всех значений $x \in [a, b]$, по меньшей мере один раз пересекает диагональ $x = g(x)$.

Действительно, график такой функции не может начинаться в a и заканчиваться в b, в противном случае они уже будут являться его неподвижными точками, и тогда уже нечего доказывать. Поэтому, полагая, например, что $g(a) \geqslant a$, $g(b) \leqslant b$, мы получим, что непрерывная функция $g(x) - x$ удовлетворяет на границах интервала неравенствам: $g(a) - a \geqslant 0$, $g(b) - b \leqslant 0$. Но тогда, по теореме о промежуточном значении, всегда найдется точка $c \in [a, b]$, такая что $g(c) - c = 0$. Если g — линейная функция с $|g'| < 1$, т. е. является сжатием, ее графиком является прямая, которая пересекает диагональ только в одной точке! Заметим, что в нашем примере метрическое пространство является полным — это замкнутый интервал. Рассмотрим теперь другие метрические пространства и Принцип сжимающих отображений в них.

Примечания

1. Объем n-мерного шара равен $V_n(r) = C_n r^n$. В этой формуле $C_n = \pi^{n/2}/\Gamma(n/2+1)$ и Γ — гамма-функция. Если взять например $n = 2k$, то получим $V_{2k}(r) = (\pi r^2)^k (k!)^{-1}$. Легко видеть, что при $k > \pi r^2$ увеличение k (т. е. размерности n) приводит к уменьшению объема! Доля объема в шаровом ε-слое будет равна

$$\frac{\textit{объем слоя}}{\textit{объем шара}} = \frac{C_n r^n - C_m (r-\varepsilon)^n}{C_n r^n} = 1 - \left(1 - \frac{\varepsilon}{r}\right)^n \to 1,$$

   т. е. стремится к 1 при возрастании n. Поэтому практически весь объем шара большой размерности сосредоточен вблизи его поверхности. Так что относительно любой точки внутри шара почти все пары точек выборки образуют остроугольные треугольники.

2. *Банах Стефан*, он же *Степан Степанович Гречек* (30.03.1892–31.08.1945) Родился в Кракове. Член Польской АН и член-корреспондент АН УССР, профессор Львовского университета. Один из создателей современного функционального анализа. Во время оккупации находился в нацистском институте, где разрабатывалась противотифозная сыворотка, а ученого использовали для кормления вшей. *Банах* один из авторов так называемой *Шкотской книги* — реликвии математического мира. Рукопись, написанная в пивной, содержала 193 важных математических проблемы, придуманных *Банахом*, *Мазуром*, *Штейнхаусом* и *Уламом* на встречах в *Шкотской кавьярне*

(г. Львов). Многие решения утеряны: авторы записывали их сперва химическим карандашом на мраморных столиках. Призами, в зависимости от сложности задачи, были: пять малых кружек пива, вино, ужин и живой гусь. Книга была переведена на английский язык и издана *Уламом* в 1957 году.

Путеводитель по литературе. Приведенные определения можно найти в любом курсе функционального анализа. Популярное введение в теорему Банаха и ее варианты содержит книга [6]. Пример парадокса с многомерным объемом взят из замечательной книги *Хемминга* [7]. Использование ультраметрики в различных задачах описано в статье *Муртага* [5]; другие работы на эту тему есть на его страничке.

Пространство компактов и IFS

> Ничто не строится на камне, все на песке, но долг человеческий строить, как если бы камнем был песок.
>
> *Х. Л. Борхес*
> *«Фрагменты апокрифического Евангелия»*

Ограниченные и замкнутые множества называются *компактными* или просто *компактами*. Ограниченность означает, что такое множество можно поместить внутрь шара конечного радиуса, а замкнутое множество содержит в себе все свой предельные (граничные) точки. Компактами, например, являются замкнутые интервалы вида $[a, b]$ на вещественной оси, но не полуоткрытые и открытые интервалы, такие как $[a, b)$, $(a, b]$, (a, b).

Рассмотрим пространство H, точками которого являются компакты из R^n. Если использовать такие множества из R^2, то точками в H будут сегменты, квадраты, круги, треугольники и т. д. Расстояние в H измеряется с помощью *метрики Хаусдорфа* (1).

Рассмотрим два множества A и B. Обозначим через $d(x, B)$ расстояние самой удаленной от B точки $x \in A$, где удаленность понимается в смысле обычного евклидова расстояния:

$$d(x, B) = \sup_{x \in A} \inf_{y \in B} d(x, y).$$

Это выражение описывает две операции. Первая состоит в том, что для произвольной *фиксированной* точки $x \in A$ мы находим расстояния до всех точек $y \in B$ и выбираем среди них кратчайшее. Символически это записывается как $\inf\limits_{y \in B} d(x, y)$. Затем переходим в найденную, самую близкую точку множества $y \in B$ и ищем наиболее удаленную от нее точку $x \in A$, т. е. берем верхнюю грань $\sup\limits_{x \in A} d(x, y)$ расстояний для всех точек, в предположении, что точка y фиксирована. Аналогично, обозначим через $d(y, A)$ расстояние самой удаленной от A точки $y \in B$ (рис. 1). Оно получается из выражения: $d(y, A) = \sup\limits_{y \in B} \inf\limits_{x \in A} d(x, y)$. В результате мы получим два, вообще говоря, разных числа. Расстояние Хаусдорфа можно определить как максимальное из них:

$$d_H(A, B) = \max\Big\{d(x, B),\ d(y, A)\big|\ x \in A,\ y \in B\Big\}$$

или как сумму этих двух чисел:

$$d_H(A, B) = \sup_{x \in A} \inf_{y \in B} d(x, y) + \sup_{y \in B} \inf_{x \in A} d(x, y).$$

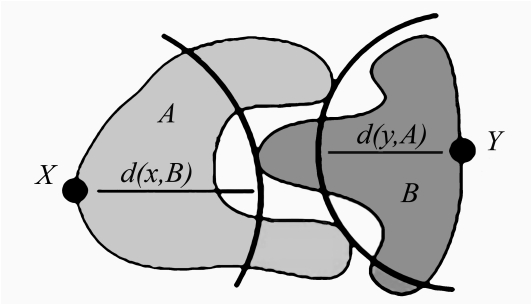

Рис. 1. Определение метрики Хаусдорфа

Существует и геометрический эквивалент этого определения. Покроем каждую точку множества A диском с радиусом ε. Возьмем объединение всех этих дисков. Те, что декорируют внутренние точки A совпадут с этим множеством. Диски с центрами на граничных точках A увеличат исходное множество на величину ε. Для того, чтобы уяснить это, представьте себе диск, центр которого движется по границе A и окрашивает

своей поверхностью полоску шириной ε, за счет которой и увеличивается площадь множества. Формально, то что получилось, записывают как множество точек удаленных от A на расстояние, не превышающее ε, т. е. $A_\varepsilon = \{x \mid d(x, A) \leqslant \varepsilon\}$. Множество A_ε называют *параллельным телом* для A или его *дилатацией*.

Будем увеличивать радиус диска до тех пор, пока дилатация A_ε не поглотит множество B, т. е. $B \subseteq A_\varepsilon$. Эта ситуация возникнет при некотором радиусе ε_1. Теперь вернемся к исходным множествам и, оставив A в покое, дилатируем B так, чтобы $A \subseteq B_\varepsilon$. Это произойдет при некотором другом радиусе ε_2. Тогда *метрикой Хаусдорфа* называют минимальный из полученных радиусов:

$$d_H(A, B) = \min\Big\{\varepsilon : A \subseteq B_\varepsilon \ \text{ и } \ B \subseteq A_\varepsilon\Big\}.$$

Все приведенные определения удовлетворяют свойствам метрики (1–4). Метрика $d_H(A, B)$ имеет два важных свойства. Пусть каждое из множеств A и B составлено из объединения конечного числа компонентов. Тогда расстояние между двумя такими кластерами не превышает максимального расстояния между их компонентами:

$$d_H\Big(\bigcup_{i=1}^{N} A_i, \quad \bigcup_{i=1}^{N} B_i\Big) \leqslant \max_{1 \leqslant i \leqslant N} d_H\big(A_i, B_i\big),$$

и в этом состоит первое свойство.

Прежде чем пояснить второе свойство, определим действие сжимающего отображения F с коэффициентом r на компакт. Запись $F(A)$, где A — компактное множество, мы будем понимать в смысле действия F на каждую точку компакта, т. е. $F(A) \equiv \Big\{F(x) \;\Big|\; \forall x \in A\Big\}$. Так вот, второе свойство метрики Хаусдорфа заключается в том, что d_H сохраняет свойство сжатия в пространстве H:

$$d_H\Big(F\big(A\big), F\big(B\big)\Big) \leqslant r\, d_H\big(A, B\big).$$

Пара (H, d_H) является полным метрическим пространством. Следовательно, любое сжимающее отображение будет иметь в этом пространстве единственную неподвижную точку. Все это выглядит слишком тривиальным.

Для того чтобы прийти к более интересным ситуациям, вместо одного сжатия рассмотрим их конечный набор. Пусть, например, таким набором будут $(W_1, W_2, \ldots, W_N)$, где каждое из W_i имеет свой собственный коэффициент сжатия r_i. Такой набор называют гиперболической[6] *Системой Итеративных Функций (Iterated Function System — IFS и* `IFS`*)*.

Определим их коллективное действие на компакт A *оператором Хатчинсона*

$$\boldsymbol{W}(A) = \bigcup_{i=1}^{N} W_i(A).$$

Эта запись означает, что на A независимо действует каждый из W_i, а затем берется объединение их образов. Последние являются просто компактными множествами и, следовательно, используя два описанных выше свойства метрики Хаусдорфа, легко убедиться, что $\boldsymbol{W}$ является сжатием в H с коэффициентом $r = \max\{r_i\}$. Действительно, для любых $A_1, A_2 \in H$

$$d_H\big(\boldsymbol{W}(A_1), \boldsymbol{W}(A_2)\big) = d_H\Big(\bigcup_{i=1}^{N} W_i(A_1),\ \bigcup_{i=1}^{N} W_i(A_2)\Big) \leqslant$$
$$\leqslant \max_{1\leqslant i\leqslant N} d_H\big(W_i(A_1), W_i(A_2)\big) \leqslant r\, d_H(A_1, A_2).$$

Используя *Принцип сжимающих отображений*, мы тотчас получаем, что существует единственная неподвижная точка в H, т. е. непустое компактное множество, которое называется *аттрактором и* `IFS`, такое что

$$A = \boldsymbol{W}(A) = \bigcup_{i=1}^{N} W_i(A) \quad \text{и} \quad \forall B \in H \lim_{k\to\infty} \boldsymbol{W}^{\circ k}(B) = A.$$

Первое выражение примечательно тем, что выражает неподвижную точку — компактное множество A — как объединение своих собственных уменьшенных копий. Иначе говоря, A является *коллажем*. Например, для $N = 2$ мы получим

$$A = W_1(A) \cup W_2(A) = W_1(W_1(A) \cup W_2(A)) \cup W_2(W_1(A) \cup W_2(A)) =$$
$$= W_{11}(A) \cup W_{12}(A) \cup W_{21}(A) \cup W_{22}(A),$$

[6]Здесь «гиперболическая» — синоним «сжимающей».

где $W_{ij} = W_i \circ W_j$. Повторяя это разложение k раз, получим

$$A = \bigcup_{1 \leqslant \sigma_1 \ldots \sigma_k \leqslant 2} W_{\sigma_1 \ldots \sigma_k}(A),$$

где $W_{\sigma_1 \ldots \sigma_k} = W_{\sigma_1} \circ \ldots \circ W_{\sigma_k}$.

Рассмотрим простой пример. Пусть $\boldsymbol{W} = W_1 \cup W_2$, где $W_1 = x/3$, $W_2 = x/3+2/3$. Выберем в качестве B единичный интервал $B = I = [0,1]$. Первая итерация дает $\boldsymbol{W}(I) = \big\{[0,1/3] \cup [2/3,1]\big\}$. Вычислим вторую итерацию $\boldsymbol{W}^{\circ 2}(I)$. Применение сжатия W_1 к двум полученным интервалам дает $W_1\big([0,1/3]\big) = [0,1/9]$ и $W_1\big([2/3,1]\big) = [2/9,1/3]$. Независимо применяя к этим же интервалам W_2 получаем $W_2\big([0,1/3]\big) = [2/3,7/9]$ и $W_1([2/3,1]) = [8/9,1]$. Таким образом, $\boldsymbol{W}^{\circ 2}(I) = \big\{[0,1/9] \cup [2/9,1/3] \cup [2/3,7/9] \cup [8/9,1]\big\}$. Нетрудно догадаться, что продолжая эту процедуру *ad infinitum*, мы получим классический фрактал — множество Кантора. Оно и является неподвижной точкой выбранного оператора!

Большую роль в теории и `IFS` играет *Теорема о коллаже*, которая следует из неравенства треугольника для метрики Хаусдорфа. Пусть $B \in H$ и $\{W_i\}$, $i = \overline{1,N}$ — `IFS` с максимальной константой Липшица $c = \max\{c_i\}$, которая имеет аттрактор $\mathcal{A} = \mathbf{W}(\mathcal{A})$. Тогда

$$d_H(\mathcal{A}, B) \leqslant \frac{d_H\big(B, \mathbf{W}(B)\big)}{1-c}.$$

Иными словами, чем меньше расстояние (его называют *коллаж-расстоянием*) между произвольным начальным множеством B и его образом $\mathbf{W}(B)$, тем ближе B к аттрактору $\mathcal{A}$.

Примечания

1. *Феликс Хаусдорф* (*Felix Hausdorff*, 8.11.1868–26.01.1942) — немецкий математик, один из основоположников современной топологии. Вывел и исследовал такие понятия, как хаусдорфовы пространства и хаусдорфова размерность. Внес большой вклад в теорию множеств, функциональный анализ, теорию топологических групп и теорию чисел. Выступал как писатель под псевдонимом *Поль Монгре*. Профессор университетов в Лейпциге, Грейфсвальде и Бонне. В 1942 году, когда отправка его и его семьи в концлагерь стала неизбежной, покончил жизнь самоубийством вместе с женой и ее сестрой.

Путеводитель по литературе. Лучшее ведение в метрику Хаусдорфа можно найти в статьях *Джона Хатчинсона*, доступных на его страничке [8], в замечательной книге *Михаила Барнсли* [9], а также в монографиях *Фальконера* [10, 11], сканированные копии которых можно разыскать в библиотеках электронных книг. Популярное изложение содержится в моих лекциях [3, 12].

Пространство кодов

> Я тоже играю символами, но я играю, не забывая, что речь-то идет лишь об игре.
>
> *Иоганн Кеплер*

Рассмотрим слова, образованные из символов или букв некоторого алфавита $\mathcal{A}$, емкость которого равна L. Когда алфавит цифровой — $\mathcal{A} = \{0, 1, 2, \ldots, L-1\}$, словами являются конечные или бесконечные числовые последовательности. Например, если $\mathcal{A} = \{0, 1\}$, то слова имеют вид $s = 01000110\ldots$. Если алфавит составлен из букв $\mathcal{A} = \{\sigma_1, \sigma_2, \ldots, \sigma_{L-1}\}$, то словами являются бесконтекстные (т. е. не имеющие семантического смысла) буквенные последовательности $s = \sigma_{i_1}, \sigma_{i_2}, \ldots$, где $i_k \in \{1, 2, \ldots, L\}$.

Мы часто будем рассматривать слова одинаковой длины K. В этом случае, число всех возможных слов составляет величину L^K. Обозначим через Σ пространство, точками которого являются описанные выше слова. Такие пространства часто используют на практике. В генетике, например, ДНК представляет собой текст, состоящий из нуклеотидов, каждый из которых содержит одну из четырех букв — азотистых оснований: A (аденин), G (гуанин), T (тимин), C (цитозин).

Известно несколько способов получения символической последовательности из временных рядов. Можно, например, использовать *поворотные точки* ряда, т. е. максимумы и минимумы его графика. Пометив минимумы символом 0, а максимумы символом 1, мы получим бинарную последовательность.

Другой способ берет начало из символической динамики [13]. Проще всего преобразовать временной ряд $\{x_i\}$ в бинарный «текст», состоящий из слов $S = s_1, s_2, \ldots, s_k$, где $s_i \in \{0, 1\}$, хотя обобщение на $L > 2$ тривиально. Выберем для этого некоторое пороговое значение ординаты

графика временного ряда $x_i = h$ и трансформируем все отсчеты ряда в символы по правилу: $s_i = 0$, если $x_i < h$, и $s_i = 1$, если $x_i \geqslant h$. Чтобы получить отдельные слова, прочитаем полученный текст с помощью шаблона, содержащего K окон, сдвигая его вдоль текста на один символ. Например, для текста $011010010110\ldots$ и $K = 3$ мы получим слова 011, 110, 101 и т. д.

Можно изучить статистику полученных слов, построив гистограмму их распределения. Разделим единичный отрезок $[0, 1)$ на 2^K интервалов, и каждому слову S из 2^K возможных слов длины K поставим в соответствие точку (адрес слова) из $[0, 1)$ по правилу

$$x(S) = \frac{s_1}{2} + \frac{s_2}{2^2} + \ldots + \frac{s_K}{2^K}.$$

Например, слово 101 является *адресом* точки $x = 1/2 + 0/2^2 + 1/2^3 = 5/8$.

Распределение адресов дает нам гистограмму встречаемости слов. Понятно, что в случае алфавита, содержащего 3 символа, следует использовать троичное разложение и т. п. Если полученная гистограмма *стационарна*, т. е. не меняет своей формы для различных фрагментов текста, ее можно использовать как оценку *эмпирической меры* случайной динамической системы, которая получается следующим образом.

Рассмотрим `IFS` с аттрактором $I = [0, 1]$: $W_1(x) = x/2$, $W_2(x) = x/2 + 1/2$. Возьмем произвольную точку $x_0 \in [0, 1]$. Напомним, что это адрес некоторого слова. Будем выбирать случайно каждое из двух отображений W_i, $i = 1, 2$ с вероятностями p_1 и p_2, соответственно. Заметим, что адрес, начинающийся с нуля, соответствует точке из левой половины I: $0s_2 \ldots s_K \in [0, 1/2]$, тогда как слово, начинающееся с единицы, из правой половины: $1s_2 \ldots s_K \in [1/2, 1]$. Следовательно, применение `IFS` индуцирует появление суффикса $s_K = 1$ с вероятностью p_1, и суффикса $s_K = 0$ с вероятностью p_2.

Для следующего двоичного разряда, который уточняет адрес в каждом из 2^2 интервалов, мы получим вероятности $\{p_1p_1;\ p_1p_2;\ p_2p_1;\ p_2p_2\}$. Продолжая этот процесс, мы придем, как мы убедимся позже, к единственной гистограмме, которую называют *биноминальной мерой*. Она может рассматриваться как *теоретическая* относительно эмпирической оценки, полученной выше, если конечно вероятности выбраны правильно.

Рассмотрим теперь некоторые способы определения расстояний в пространстве кодов.

Метрика Хемминга (1) между словами равной длины определяется как

число позиций, для которых соответствующие символы различны. Например, расстояние Хемминга между словами 1011101 и 1001001 равно двум, а расстояние между словами 2143896 и 2233796 равно трем.

Метрика Левенштейна (2), которую называют еще *дистанцией редактирования*, определяется как минимальное число элементарных преобразований (удаление, вставка и замена) символов, необходимых для преобразования одного слова в другое. Так, например, для того чтобы трансформировать слово МАША в САРАЙ необходимо заменить М на С, Ш на Р и вставить Й. Следовательно, расстояние Левенштейна в данном случае равно трем.

Метрика d_Σ определяется только для пар разных $(s_1 \neq s_2)$ бесконечных слов выражением $d_\Sigma(s_1, s_2) = 1/L^k$. Здесь k — индекс первого символа, по которому различаются два слова. Например, слова $s_1 = 110010100100001\ldots$ и $s_2 = 110011011011101\ldots$ впервые различаются для $k = 6$, так что $d_\Sigma(s_1, s_2) = 1/2^6$. Можно показать, что (Σ, d_Σ) является компактным метрическим пространством.

Наша метрика обладает еще одним приятным свойством. Определим преобразование сдвига $s_m : \Sigma \to \Sigma$ на словах следующим образом. Для каждого $m = \{0, 1, 2, 3, \ldots, L-1\}$ положим $s_m(\sigma_1, \sigma_2, \ldots) = m, \sigma_1, \sigma_2, \ldots$. Иными словами, сдвиг — это добавление к слову в качестве префикса одной из букв алфавита.

Рассмотрим в качестве примера два бинарных слова $s_1 = 01001\ldots$ и $01100\ldots$, так что $d_\Sigma(s_1, s_2) = 1/4$. Добавим к каждому слову префикс, скажем 0. Расстояние между полученными новыми словами очевидно уменьшится $d_\Sigma(s_1, s_2) = 1/8$. Так что наш сдвиг в пространстве (Σ, d_Σ) является сжатием! К этому результату мы еще вернемся.

Метрика d_k, учитывающая память в словах, определяется для конечных L–слов выражением

$$d_k(s_1, s_2) = \sum_{i=1}^{L} k^{L-i+1} \delta(\sigma_i(s_1), \sigma_i(s_2)),$$

где величина $k \leqslant 1/2$; $\delta(i, j) = 0$ при $i = j$ и $\delta(i, j) = 1$ при $i \neq j$. Например, для $s_1 = 1100$, $s_2 = 1110$ и $k = 1/2$

$$d_k(s_1, s_2) = 1/4.$$

С другой стороны, расстояние между $s_1 = 1100$ и $s_2 = 1101$, которые

различаются последним символом, равно

$$d_k(s_1, s_2) = 1/2.$$

Расстояние между двумя тождественными бинарными L–словами в этой метрике будет, очевидно, равно нулю.

Примечания

1. *Ричард Весли Хэмминг* (11.02.1915–07.01.1998) — почетный член IEEE и член Национальной Академии инженерных наук, лауреат премий Тьюринга, Пиорее, Пендера и Рейма. Родился в Чикаго. В 1945 году участвовал в знаменитом Манхэттенском проекте. Затем, в течение 30 лет конструировал компьютеры в лаборатории Белла, где работал и *Клод Шеннон.* Хэмминга называют гением одной идеи. Она была изложена в 1950 году, в его единственной научной статье, посвященной кодам, и содержала конструкцию блочного кода, корректирующего одиночные ошибки, которые возникают при передаче сообщений. Эта работа Хэмминга была отмечена многими наградами. Существует медаль IEEE в честь *Р. Хемминга.*

2. *Владимир Иосифович Левенштейн* (род. в 1935 г.) — российский ученый, доктор физико-математических наук, работает в Институте прикладной математики им. М. В. Келдыша. Метрика, названная его именем, введена в 1965 году. Она с успехом применяется, например, в сейсмологии для идентификации коллективных движений литосферных блоков по синхронным записям различных сейсмических станций.

Путеводитель по литературе. Метрика Левенштейна была введена в работе [14], а ее применение в сейсмологии описано в статье [15]. Метрику d_Σ использовал *М. Барнсли* [9]. Метрика с памятью используется в серии работ *Петера Тино* [16, 17]; на его страничке можно найти также применение сжимающих отображений к проблемам математической лингвистики и анализу ДНК [18].

Меры

Бесконечность больше, чем самое большое из всего, что есть, и еще чуть-чуть.

Дуглас А.
«Автостопом по Галактике»

Абстрактные меры. Мера в определенном смысле обобщает понятие числа: это функция, которая ставит в соответствие множеству его «размер». Представим себе бесконечную массу, равномерно распределенную в евклидовом пространстве R^n. Количество массы внутри единичного куба соответствует его (*лебеговой*) мере $0 \leqslant \mu \leqslant \infty$. *Меры Радона*, к которым относится и лебегова мера, приписывают каждому ограниченному множеству *конечную* меру. Наиболее общими являются *борелевы регулярные меры*; к последним относятся k–мерные меры *Хаусдорфа*, которые не являются радоновыми мерами при $k < n$.

Мы начнем с наиболее популярной меры Лебега. Для прямоугольного множества $R \subset R^n$: $R = [a_1, b_1] \times \ldots \times [a_n, b_n]$ определим меру Лебега как его евклидов объем: $m(R) = [a_1 - b_1] \times \ldots \times [a_n - b_n]$. Тогда, для любого множества $A \subset R^n$ мера определяется как нижняя граница мер объединения прямоугольных множеств, содержащих A:

$$\mu(A) = \inf_{A \in \bigcup R_i} \sum m(R_i). \tag{1}$$

Мера Лебега имеет следующие полезные свойства:

1) $\mu(\varnothing) = 0$ для пустого множества;
2) если A является объединением счетного числа множеств $A \subset \bigcup\limits_{i=1}^{\infty} A_i$, то $\mu(A) \leqslant \sum\limits_{i=1}^{\infty} \mu(A_i)$, где равенство достигается если множества A_i попарно не пересекаются.

В качестве примера рассмотрим множество нулевой лебеговой меры. Традиционный прием построения фрактального множества Кантора заключается в рекуррентном удалении средней трети из единичного интервала $I = [0, 1]$, затем двух фрагментов по $1/9$ из двух получившихся фрагментов

и так далее, *ad infinitum*:

$$\begin{aligned}
&K_0 = [0,1];\\
&K_1 = [0,1/3] \cup [2/3,1];\\
&K_2 = [0,1/9] \cup [2/9,1/3] \cup [2/3,7/9] \cup [8/9,1]; \ldots
\end{aligned}$$

Предельное множество $\mathcal{K} = \bigcap\limits_{n=0}^{\infty} K_n$ состоит из общей части всех точек K_n и называется *множеством Кантора*. На i-ом шаге каждое из множеств содержится внутри 2^i прямоугольников размером 3^{-i}. Поэтому, согласно (1), $\mu(\mathcal{K}) \leqslant 2^i \times 3^{-i}$ для всех i, так что при $i \to \infty$ $\mu(\mathcal{K}) \to 0$.

В общем случае, если в пространстве выделен класс подмножеств $\mathcal{F}$, замкнутый относительно операций конечных пересечений и объединений, то функция $\mu : \mathcal{F} \to [0,\infty]$ называется *конечно-аддитивной мерой*, при условиях

1) $\mu(\varnothing) = 0$;
2) если $\{E_i\}_{i=1}^N \subset \mathcal{F}$ — конечное семейство попарно непересекающихся множеств, т. е. $E_i \cap E_j = \varnothing$, $i \neq j$, то $\mu\Big(\bigcup\limits_{i=1}^{N} E_i\Big) = \sum\limits_{i=1}^{N} \mu(E_i)$.

Если условия (1)–(2) выполняются для *счетного* семейства $\mathcal{F}$, меру называют *счетно-аддитивной*.

Множество A называют *μ–измеримым в X*, если меру любого множества B можно представить в виде

$$\mu(B) = \mu(B \cap A) + \mu(B \cap A^C),$$

где $A^C = X \setminus A$ — дополнение множества A в X. Иными словами, измеримое множество разбивается на части, так что его можно представить в виде пересечения с любым другим множеством и его дополнением. Класс μ-измеримых множеств образует так называемую *σ-алгебру*, т. е. класс, замкнутый относительно операций взятия дополнения, счетных пересечений и объединений.

Итак, основная идея введения меры в пространстве X состоит в выделении некоторого класса $\mathcal{A}$ элементарных подмножеств, например, интервалов для меры Лебега. Пустое множество, само пространство X, а также объединение и пересечение конечного (счетного) числа множеств принадлежит этому же классу, представители которого объявляются *измеримыми множествами*. Пару $(X, \mathcal{A})$ называют *измеримым* пространством. Тогда

мера $\mu : \mathcal{A} \to [0, \infty]$ — это функция, которая ставит в соответствие каждому множеству $\mathcal{A} \subset X$ неотрицательное число из интервала $[0, \infty]$. Мера должна удовлетворять, по меньшей мере, условию конечной (счетной) аддитивности [(1)]. Тройку $(X, \mathcal{A}, \mu)$ называют *пространством с мерой*.

Динамические меры. До сих пор мы рассматривали меры как «длины». Однако, в общем случае, мера вовсе не обязана иметь геометрические атрибуты. *Борелеву меру* можно ввести, используя динамику некоторого непрерывного отображения единичного отрезка $f : [0, 1] \to [0, 1]$. *Орбитой* отображения f длины $n + 1$ называют последовательность точек $x_0, x_1, x_2, \ldots, x_n$, где каждая последующая точка — результат итерации предыдущей:

$$x_n = f^{\circ n}(x_0) = f\big(f^{\circ(n-1)}(x_0)\big).$$

Разобьем единичный интервал $I = [0, 1]$ на N сегментов

$$I_k = [(k-1)/N,\, k/N), \quad k = \overline{1, N}.$$

В результате итераций $f^{\circ n}(x_0)$ нашей функции некоторое число точек из орбиты длиной $n + 1$ попадает в I_k. Обозначим это число как

$$n_k = I_k \cap x_0,\, x_1,\, x_2, \ldots,\, x_n.$$

Его можно интерпретировать как время пребывания орбиты в I_k, если каждой итерации поставить в соответствие единицу времени. Можно надеяться, что при большом n и хороших[(2)] начальных значениях x_0 числа $p_k(n, x_0) = n_k/(n + 1)$ не зависят от n и выбора x_0. В этом случае можно доказать[(3)], что *почти для всех* орбит существует *инвариантная вероятностная мера* $\mu_k(I_k)$:

$$\mu_k = \lim_{n\to\infty} \frac{|I_k \cap \{x_0,\, x_1,\, x_2, \ldots,\, x_n\}|}{n}, \tag{2}$$

такая что $p_k(n, x_0) \approx \mu_k$ и $\sum\limits_k \mu_k = 1$. Выражение *почти для всех* понимается в том смысле, что исключения составляют множество нулевой лебеговой меры.

Заметим, что точка $y = f(x)$ для нашей функции $f : [0, 1] \to [0, 1]$ принадлежит некоторому объединению интервалов $U = I_m \cup I_p \cup \ldots \cup I_s$ тогда и только тогда, если $x \in f^{-1}(U)$. Это означает, что число точек последовательности $x_1, \ldots, x_{n+1}$, которые попали в U, совпадает с числом

тех точек орбиты $x_0, x_1, \ldots, x_n$, которые принадлежали $f^{-1}(U)$. Числа μ_k не должны зависеть от сдвига последовательности, так что выражение $\mu(U) = \mu\big(f^{-1}(U)\big)$ утверждает инвариантность меры.

Пусть $m(x_k)$ — мера, определенная в точке x_k. Тогда интеграл $\int g\,\mu$ от произвольной функции $g(x)$ по мере μ понимается как сумма [4]:

$$\int g\,\mu = \sum_{x_k} g(x_k)\ m(x_k), \tag{3}$$

где ряд сходится в абсолютном смысле.

Действие функций на меру. Пусть y_l — значения функции $y = f(x_k)$; таких точек может быть несколько[5] для каждой точки x_k. Тогда меру точек y_l естественно определить как меру их прообразов x_k — другого пути просто нет:

$$m(y_l) = \sum_{x_k = f^{-1}(y_l)} m(x_k). \tag{4}$$

Разумеется, это выражение имеет смысл, если число точек $f^{-1}(y_l)$ не бесконечно велико. Приведенные рассуждения помогают понять как функция действует на меру. Символически это действие называется [6] *push forward map* и записывается как $f \circ \mu = \mu(f^{-1})$, где $(\circ)$ — знак композиции. Иначе говоря, действие функции f на меру — это просто мера тех точек x, которые отобразились с помощью f в точки $y = f(x)$.

Для приложений полезно рассмотреть коллективное действие набора функций на меру. Рассмотрим, например, и `IFS`, снабженную вероятностями:

$$\{W_i; p_i\}, \quad i = \overline{1, N}, \quad p_1 + \ldots + p_N = 1.$$

Их коллективное действие на меру μ описывается *марковским оператором*

$$\mathbf{M}(\mu) = \sum_{i=1}^{N} p_i\,\mu \circ W_i^{-1}. \tag{5}$$

Для любой непрерывной функции $f : X \to R$ справедливо очень важ-

ное для нас выражение:

$$\int_X f\,d(M(\mu)) = \sum_{i=1}^{N} p_i \int_X f\,d(\mu \circ W_i^{-1}) =$$
$$= \sum_{i=1}^{N} p_i \int_{W_i(X)} f(x)\,d(\mu \circ W_i^{-1}) = \sum_{i=1}^{N} p_i \int_X f \circ W_i(x)\,d\mu \tag{6}$$

Поясним, как оно получается. Заметим для этого, что $\mu \circ W_i^{-1}$ — мера тех точек $x \in X$, которые генерируют образы $x \to W_i(x)$. Это обстоятельство можно учесть, используя дельта-функцию и запись $\delta(x - W_i(x))$. Тогда и получается замена $f(x) \to f \circ W_i(x)$ в последнем члене (6).

Мера называется *самоподобной*, если ее можно представить как линейную взвешенную комбинацию ее самой:

$$\mu = p_1\,\mu \circ W_1^{-1} + \ldots + p_N\,\mu \circ W_N^{-1}. \tag{7}$$

Поясним, что это означает. Представим себе один грамм массы $\mu = 1$, равномерно распределенной на интервале $I = [0, 1]$. Пусть и `IFS`

$$W_1 = x/2, \quad W_2 = x/2 + 1/2$$

выбираются с вероятностями p_1, p_2. После действия W_1 мы получим $W_1(I) = [0, 1/2]$, на котором соберется вся масса, т. е. $\mu[0, 1/2] = \mu \circ W_1^{-1}$. Независимое действие второго сжатия соберет всю массу на второй половине интервала: $W_2(I) = [1/2, 1] \to \mu[1/2, 1] = \mu \circ W_2^{-1}$. Поскольку масса должна сохраняться, мы распределяем ее в соответствии с выбранными весами, т. е. вероятностями по двум интервалам:

$$\mu = p_1 \mu \circ W_1^{-1} + p_2 \mu \circ W_2^{-1}.$$

Вторая итерация выбранной пары сжатий приведет к «перевзвешиванию» меры на каждом из четырех интервалов: $\mu[0, 1/4] = p_1\,p_1$; $\mu[1/4, 1/2] = p_1\,p_2$; $\mu[1/2, 3/4] = p_2\,p_1$; $\mu[0, 1/4] = p_2\,p_2$. Продолжая этот процесс, мы будем получать все более тонкую структуру самоподобной меры.

В общем случае обозначим для краткости $\mu \circ W_1^{-1} = S_1$ и $\mu \circ W_2^{-1} = S_2$. Тогда

$$\begin{aligned}\mu = p_1 S_1(\mu) + p_2 S_2(\mu) = \\ = p_1 S_1(p_1 S_1(\mu) + p_2 S_2(\mu)) + p_2 S_2(p_1 S_1(\mu) + p_2 S_2(\mu)) = \\ = p_{11} S_{11}(\mu) + p_{12} S_{12}(\mu) + p_{21} S_{21}(\mu) + p_{22} S_{22}(\mu),\end{aligned}$$

или, продолжая этот процесс, получим

$$\mu = \sum_{1 \leqslant \sigma_1, \ldots, \sigma_k \leqslant 2} p_{\sigma_1 \ldots \sigma_k} S_{\sigma_1 \ldots \sigma_k}(\mu),$$

где $p_{\sigma_1 \ldots \sigma_k} = p_{\sigma_1} \cdot \ldots \cdot p_{\sigma_k}$ и $S_{\sigma_1 \ldots \sigma_k} = S_{\sigma_1} \circ \ldots \circ p_{\sigma_k}$.

Очевидно, что самоподобная мера (7) возникает как неподвижная точка марковского оператора (5), т. е. $\mathrm{M}(\mu) = \mu$. Как мы уже знаем, это следствие сжимающих свойств оператора в соответствующем пространстве мер, которым мы и займемся в следующем разделе.

Примечания

1. Обычно на меру накладывают дополнительно условия инвариантности относительно группы движений. Так привычные лебеговы меры из евклидовой геометрии (длина, площадь, объем) инвариантны относительно группы твердотельных движений в R^3 — поворотов и сдвигов. В этом случае группа *просто транзитивна*: существует лишь одно преобразование $g \in G : y = g(x)$, которое преобразует x в y. Такая группа гарантирует существование единственной (с точностью до мультипликативного множителя) инвариантной меры. Если группа *сильно транзитивна*, то существуют, например, $g_1, g_2 \in G$, такие что $y = g_1(x) = g_2(x)$. В этом случае инвариантной меры может не существовать вообще. Примером может служить «резиновая» плоскость: преобразование $x \to y$ можно выполнить на ней как сдвигом, так и растяжением.

2. Речь идет о типичных начальных значениях. Так, точка $x_0 = 0$ для $f(x_n) = x_n(1 + x_n)$ — «плохая», поскольку она не генерирует орбиту. Таких точек может быть бесконечно много, но их множество должно иметь нулевую лебегову меру.

3. Представим себе путника, который блуждает по храму, состоящему из некоторого числа залов разного размера. Выбор зала случаен, но попадая в него, возможно неоднократно, путник посещает все места помещения. Тогда относительная доля времени, которую путник проводит в каждом помещении, почти для всех вариантов блуждания пропорциональна его размеру. Это утверждение связано с эргодической теоремой, которую в 1987 году доказал *Джон Элтон*. Пусть $\{w_i\}_{i=1}^N$ — набор сжимающих отображений на компактном метрическом пространстве X, снабженных вероятностями $\{p_i\}_{i=1}^N$. Тогда для всех непрерывных функций $f : X \to R$, почти всех начальных точек $x_0 \in X$ и почти всех орбит справедливо выражение:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i\leqslant n} f(x_n) = \int_X f(x)\, d\mu.$$

4. $m(x_k)$ можно рассматривать как количество «массы», содержащейся в точке, т. е. как меру Радона. Традиционная запись $\int g\, d\mu$ не имеет особого смысла для дискретного случая, когда не определен дифференциал меры. Интеграл понимается поэтому как сумма произведений $g(x_k)\,\mu(x_k)$ значений функции на значения распределения «массы», т. е. гистограммы, которая задает меру.

5. Иными словами, функция f — *сюръективна*, т. е. для каждого x существует *по меньшей мере* один y. Представьте себе, что наше отображение — это соответствие между точками на нити электрической лампочки в фонаре и точками светового пятна на стене. Очевидно, что для каждой точки пятна можно найти один или даже более прообраз на нити.

6. В общем случае, для гладкого отображения $F : M \to N$ между *любыми* дифференцируемыми многообразиями *push forward* в точке p является лучшей линейной аппроксимацией F вблизи p. Это может быть, например, линейное отображение (дифференциал dF) между касательными пространствами $F_* : T_pM \to T_{F(p)}N$. Для пары измеримых пространств $(X, \mathcal{A})$ и $(X, \mathcal{B})$ функция $f : X \to Y$ называется *измеримой*, если прообразом любого измеримого множества в Y является измеримое множество в X, т. е. $B \in \mathcal{B} \Longrightarrow f^{-1}(B) \in \mathcal{A}$. Если μ — мера в $(X, \mathcal{A})$, то *push forward* с помощью измеримой функции

$f : X \to Y$ — это мера $f_*\mu$ в $(X, \mathcal{B})$, определенная выражением: $f_*\mu(B) = \mu(f^{-1}(B))$.

Путеводитель по литературе. Изложение теории меры можно найти практически в каждом учебнике по функциональному анализу, смотри, например, [19, 20]. Краткое и очень понятное введение в этот предмет содержится в небольшой книжке *Брудно* [21] и лекциях *Джона Хатчинсона* [22]. Отличное введение в меры на фракталах содержится в главе 9 книги *Барнсли* [9], которая, к сожалению, малодоступна. Кратко необходимый аппарат изложен в монографии *Фальконера* [11], которую можно попытаться скачать с известного сайта www.lib.org.by. Для лучшего понимания действия сжатия на меру следует посмотреть замечательные работы *Эдварда Врская* [23,24], доступные на его web-страничке. О самоподобных мерах можно прочесть в статье [8] и в других публикациях *Хатчинсона*, которые можно найти на его сайте.

Пространство мер: метрика Монжа-Канторовича

> Расстояние между задачей и целью должно меряться не прямизной соединяющей линии, а кривизной нейтрализации замысловатости.
>
> *П. Таранов*
> *«Маневры общения»*

В замечательной книге *Дагласа Хофштадтера*[7] рассматривается следующая задача, которая называется «Собака и кость». На глазах собаки хозяин бросает кость через забор в соседний двор. Кость видна в щели забора, в котором, на довольно значительном расстоянии, есть открытая калитка. Некоторые собаки подбегают к забору и начинают лаять, глядя на кость. Другие, более умные, бегут к калитке. Им становится ясно, что кажущееся увеличение *физического расстояния* между начальным и желанным положением — путь через калитку — на самом деле уменьшает *расстояние*

[7] *Хофштадтер Д.* Гёдель, Эшер, Бах: Эта бесконечная гирлянда. – Самара: Барах–М, 2001.

проблемы! Очень многие задачи, в некотором смысле, являются вариантами задачи «Собака и кость», потому что проблемы обычно формулируют не в физическом, а в концептуальном пространстве. Поэтому для преодоления абстрактных «заборов» очень важно найти внутреннюю интерпретацию задачи с подходящей метрикой между проблемой и целью. Одной из таких метрик и является *расстояние Монжа–Канторовича*.

Эту метрику открывали по меньшей мере трижды. Первым был французский геометр и общественный деятель *Гаспар Монж*. Он занимался, в частности, устройством крепостей и задачами оптимального перемещения грунта из одного места в другой. Идеи решения этой задачи были основаны на мере, близкой к той, которую впоследствии использовал для транспортных задач *Леонид Витальевич Канторович*(2), называя ее *метрикой Монжа*. Наконец, ее независимо ввел *Джон Хатчинсон* (1981 г.) в своей основной работе «Фракталы и самоподобие», после чего в ряде зарубежных работ ее стали называть *метрикой Хатчинсона*.

Начнем с формального определения. Рассмотрим пространство борелевых мер $\mathcal{M} = \mathcal{M}(\mathcal{B}(X))$. Если этот термин вызывает неприятные ощущения, представьте себе пространство, точками которого, являются различные распределения масс на компактных носителях, т. е. гистограммы.

Пусть $g : X \to R$ — функции с $Lip\ g \leqslant 1$. Метрика *Монжа–Канторовича* в $\mathcal{M}$ определяет расстояние между двумя мерами μ и ν как

$$d_M(\mu, \nu) = \sup\left\{\left|\int g\, d\mu - \int g\, d\nu\right|\right\}, \tag{8}$$

где $|g(x) - g(y)| \leqslant |x - y|$, $\forall x, y \in X$.

Согласно определению, для того чтобы вычислить расстояние между двумя мерами, необходимо располагать множеством и `IFS`. Подставляя каждую из них в выражение (8), следует найти верхнюю границу разности двух интегралов. Она и будет искомым расстоянием.

Позже мы рассмотрим способ практического вычисления этой метрики, а пока читателю придется поверить, что пара $(\mathcal{M}, d_M)$ образует полное метрическое пространство. Более того, для любых $\mu, \nu \in \mathcal{M}$ марковский оператор $\mathbf{M}$ (5) определяет сжатие в $(\mathcal{M}, d_M)$, т. е.

$$d_M(\mathbf{M}(\nu_1), \mathbf{M}(\nu_2)) \leqslant c\, d_M(\nu_1, \nu_2). \tag{9}$$

Но в этом случае из *Принципа сжимающих отображений* следует, что для семейства и `IFS` с вероятностями $\{W_i; p_i\}$, $\sum_{i=1} p_i = 1$, существует

единственная борелева вероятностная мера μ, такая что для любого подмножества $A \in \mathcal{B}(X)$

$$\mu(A) = \mathbf{M}\,\mu(A) \equiv \sum_{i=1}^{N} p_i\,\mu \circ W_i^{-1}(A), \tag{10}$$

и для любой меры $\nu_0 \in \mathcal{M}$ последовательность

$$\nu_0,\, \mathbf{M}\nu_0,\, \mathbf{M}^{\circ 2}\nu_0, \ldots, \mathbf{M}^{\circ n}\nu_0$$

сходится к μ в метрике d_M, когда $n \to \infty$. Носителем инвариантной меры является аттрактор оператора Хатчинсона $\mathcal{A} = W(\mathcal{A}) = \bigcup_{i=1} W_i(\mathcal{A})$. Из выражения (10) следует, что мера μ инвариантна относительно действия $\mathbf{M}$, т. е. является его неподвижной точкой.

Earth-Mover's Distance. Численный вариант метрики Монжа–Канторовича известен как EMD–расстояние (**E**arth-**M**over's **D**istance)[8]. Идея его вычисления заключается в следующем. Аппроксимируем две меры μ, ν зеркальными гистограммами, имеющими общее основание. Пусть верхняя из них представляет собой кучки грунта μ, насыпанного в каждый бин гистограммы. Бины второй, «опрокинутой» гистограммы, моделируют емкость ям, т. е. меру ν. Пусть площади гистограмм совпадают, т. е. ямы ν способны вместить весь грунт из кучек μ. Задача заключается в перемещении грунта из верхней гистограммы в ямы нижней с *минимальными затратами*. Последние зависят от выбранного «плана» перемещения и его «стоимости», которая определяется расстоянием[9] d_{ij} между i-м бином, где находится грунт, и j-м бином ямы. Таким образом, вычисление EMD сводится к следующей стандартной транспортной задаче линейного программирования. Необходимо минимизировать транспортные расходы:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \cdot d_{ij} \to \min;$$

при следующих ограничениях:

$$\sum_{j=1}^{n} x_{ij} = a_i; \quad \sum_{i=1}^{m} f_{ij} = b_j; \quad f_{ij} \geqslant 0, \quad i = \overline{1,m}, \quad j = \overline{1,n}\,.$$

[8]Earth mover — это экскаватор, так что EMD — «экскаваторное расстояние».
[9]d_{ij} называют *ground distance*.

Здесь d_{ij} — стоимость перевозки единицы продукции из пункта i в j; f_{ij} — план перевозок $i \to j$; b_j — потребности в продукте в пункте j; a_i — запасы в пункте i. Для модели *закрытого* типа

$$\sum_{j=1}^{m} b_j = \sum_{i=1}^{n} a_i.$$

Основной трудностью решения задачи является большое число переменных и ограничений. Поэтому используются специальные алгоритмы, адаптированные, в частности, для вычисления EMD–расстояния между изображениями.

Примечания

1. *Монж Гаспар* (10.05.1746–28.07.1818) — академик, творец начертательной геометрии и один из организаторов Политехнической школы в Париже. Занимался математическим анализом, химией, метеорологией, практической механикой. В период Французской революции был членом комиссии по системе мер и весов, а также морским министром. Будучи одним из организаторов национальной обороны Республики, он описал способы извлечения селитры, необходимой для выделки пороха, из земли в хлевах, погребах и кладбищах. Устроил множество литейных пушечных заводов. Во время Директории *Монж* — участник похода *Наполеона* в Египет. Во времена Империи он сенатор, граф Пелузский и кавалер ордена Почетного легиона. *Монж* сделал ряд важных открытий в дифференциальной геометрии, создал свою теорию интегрирования уравнений с частными производными и нашел свое решение задачи о колебании струны.

2. *Леонид Витальевич Канторович* (19.01.1912–07.04.1986) — Нобелевский лауреат, экономист, академик, лауреат Сталинской (1949 г.) и Ленинской (1965 г.) премий. Родился в Санкт-Петербурге. В 14 лет поступил в Ленинградский университет. В 1938 году профессор *Л. В. Канторович* был назначен консультантом в лабораторию фанерной фабрики. Для решения проблем, связанных с производством фанеры, он разработал метод максимизации линейной функции при большом числе ограничителей, известный теперь как линейное программирование. В работе «Математические методы организации и планирования производства» (1939 г.) *Л. В. Канторович* показал, что такой

подход универсален для экономических проблем. Нобелевскую премию получил в 1975 г. совместно с *Тьяллингом Чарлзом Купмансом* за «вклад в теорию оптимального распределения ресурсов».

Путеводитель по литературе. Метрика Монжа–Канторовича–Хатчинсона подробно описана в монографиях [9, 11]. EMD–расстоянию, его свойствам и применению к анализу изображений посвящены работы [25, 26] и глава в диссертации [27]. В статье [28] описан улучшенный алгоритм вычисления EMD между гистограммами. О транспортных задачах и алгоритмах для их решения можно прочесть в великолепных лекциях *Б. Лемешко* [29]. Метод вычисления расстояния Монжа–Канторовича с помощью искусственных нейронных сетей предложен в статье *Ярослава Штарка* [30].

Обратная задача и IFS

... иногда чувствуешь, что Идея,
которая казалась Вдумчивой,
при ближайшем рассмотрении,
когда выходит наружу, и дает на
себя посмотреть со стороны,
оказывается совсем Другим
Делом.

А. Милн
«Винни-Пух»

Мы описали выше три ипостаси$^{(1)}$ *Теоремы о сжимающем отображении*. В евклидовом метрическом пространстве (X, d) сжатием является отображение Липшица с $Lip < 1$, которое имеет единственную неподвижную точку. В пространстве компактов (H, d_H) сжатием в метрике Хаусдорфа d_H является оператор Хатчинсона с неподвижной точкой — аттрактором IFS. Наконец, в пространстве борелевых мер $(\mathcal{M}, d_M)$ сжатие в метрике Монжа–Канторовича задает оператор Маркова для IFS с вероятностями. Неподвижной точкой является здесь единственная инвариантная мера, носитель которой — аттрактор IFS.

Предположим, что мы знаем носитель, т. е. аттрактор $\mathcal{A}$ и его «раскраску», т. е. меру μ на нем. Мы знаем, на основе вышеизложенной теории, что аттрактор и мера на нем *однозначно* определяются IFS с вероятностями.

Обратная задача теории Систем Итеративных Функций как раз и состоит в нахождении подходящей IFS $\{W_i\}$, $W_i = c_i x + a_i$, $i = \overline{1, N}$, и соответствующих вероятностей $\{p_i\}$ или переходных вероятностей P_{ij} в марковском случае.

Рассмотрим сперва случай, когда выбор отображений в динамике случайных IFS производится независимо, с помощью *испытаний Бернулли* (2), и опишем схему решения обратной задачи традиционным методом моментов, предложенным в пионерской работе *М. Барнсли* и др. [31].

Исходным является выражение (6) для инвариантной меры, которое мы перепишем в виде

$$\int g(x)\, d\mu(x) = \sum_{i=1}^{N} p_i \int g \circ W_i(x)\ d\mu(x). \tag{11}$$

Для случайной переменной $x \in [0, 1]$ определим набор статистических моментов:

$$\mu_k = \int x^k\, d\mu, \quad k = 0, 1, 2, \dots, \tag{12}$$

где интеграл понимается в смысле формулы (4). Для вероятностной меры, очевидно, $\mu_0 = 1$. В уравнении (11) выберем $g(x) = x^k$. Тогда,

$$\big(g \circ w_i\big)(x) = (c_i x + a_i)^k = \sum_{j=o}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) c_i^{k-j}\, x^{k-j}\, a_i^k. \tag{13}$$

После этого уравнение (11) принимает вид

$$\mu_k = \int g(x)\, d\mu(x) = \sum_{i=1}^{N} p_i \sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) c_i^{k-j}\, a_i^k\, \mu_{k-j}. \tag{14}$$

Последнее выражение приводит к рекурсивной формуле для моментов [10]:

$$\Big(1 - \sum_{i=1}^{N} p_i\, c_i^k\Big)\mu_k = \sum_{j=0}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) \mu_{k-j} \Big(\sum_{i=1}^{N} p_i\, c_i^{k-j}\, a_i^j\Big). \tag{15}$$

Эта формула содержит и коэффициенты IFS и вероятности. Однако, чтобы использовать уравнения (15) нужны значения теоретических моментов.

[10] Для марковской схемы соответствующие формулы можно найти в работе [32].

Поэтому функционал обратной задачи основан на сравнении этих неизвестных моментов с эмпирическими моментами $\hat{\mu}_k$, вычисленными по эмпирической оценке мере $\hat{\mu}$ для $k = 1, 2, \ldots, M$. О том, как можно получить такие оценки из временного ряда методами символической динамики, рассказывалось в разделе «Пространство кодов».

Для численного решения (3) используют стандартные оптимизационные пакеты или генетический алгоритм [33]. Критерием оптимального решения служит либо разность моментов (эмпирических и модельных), либо разности гистограмм эмпирической и модельной меры.

Для большинства практических задач проблема существенно упрощается. Так, например, если речь идет о предсказании пороговых значений некоторой величины, естественно использовать бинарный алфавит $\{0, 1\}$. Следовательно, в уравнениях (11)–(15) достаточно ограничиться $N = 2$. Эмпирическая мера, полученная представлением слов в двоичном коде, имеет в качестве носителя единичный интервал. Поэтому можно выбрать `IFS` с фиксированными коэффициентами в виде: $W_1(x) = x/2$, $W_1(x) = x/2 + 1/2$ и аттрактором $[0, 1]$. Обратная задача сводится тогда либо к нахождению вероятностей p_1, p_2, либо, в марковском варианте, вероятностей p_{11} и p_{22} переходов символов $0 \to 0$ и $1 \to 1$, соответственно. Сопряженная пара вероятностей вычисляется из соотношений $p_{12} = 1 - p_{11}$ и $p_{21} = 1 - p_{22}$.

Примечания

1. Кроме того, существует еще сжатие в пространстве кодов (Σ, d_Σ), о котором мы уже упоминали. Пусть $\mathcal{A}$ — аттрактор `IFS` с вероятностями. Тогда существует непрерывное отображение $F : \Sigma \to \mathcal{A}$ из пространства кодов Σ, определенное для всех последовательностей $\sigma_1, \sigma_2, \ldots \in \Sigma$ как [34]

$$F(\sigma_1, \sigma_2, \ldots) = \lim_{k \to \infty} w_{\sigma_1} \circ w_{\sigma_2} \circ \ldots \circ w_{\sigma_k}(x). \tag{16}$$

Сдвиги в пространстве (Σ, d_Σ), определенные для каждого m, принадлежащего алфавиту $\{1, 2, \ldots, M\}$, как

$$s_m(\sigma_1 \sigma_2 \sigma_3 \ldots) = m \sigma_1 \sigma_2 \sigma_3 \ldots$$

являются сжатиями и для `IFS`:

$$S = \{\Sigma, s_1, \ldots, s_M;\ p_1, \ldots, s_M\}.$$

Ее аттрактором является Σ. Единственная вероятностная мера на аттракторе π определяется как вероятность того, что в слове $\omega_1\omega_2\omega_3\ldots$ символы принимают значения $\omega_1 = \sigma_1, \ldots \omega_k = \sigma_k$. Мера π с мерой μ на аттракторе связаны отображением (16): $\mu = F(\pi)$.

2. Возьмите урну содержащую N_1 черных и N_2 белых шара. Выберите случайно один шар. Предположим, что он оказался черным. Если вы вернете его назад в урну, то вероятности извлечения шаров при следующим испытании останутся неизменными. Это и есть *испытание Бернулли*. При марковском процессе извлеченный шар не возвращается и вероятность достать вторично черный шар становится равной $(N_1 - 1)/(N_1 - 1 + N_2)$.

3. Вопросы существования и сходимости решения подробно обсуждаются в статьях [35]– [37]. В случае линейной оптимизационной задачи для функционала можно использовать L_1–метрику:

$$\sum_{i=1}^{M} |\mu_i - \hat{\mu}_i| \to \min,$$

L_2–метрику, коллаж-расстояние или *EMD*. Задачу можно свести даже к квадратичной, используя «расстояние в шаре» [38]:

$$\sum_{k=1}^{M} (\mu_k - \hat{\mu}_k)^2/k \to \min\,.$$

Предсказание магнитных бурь

По сие время не можно было человеку столько проникнуть в уставы естества, чтобы с вероятностью можно было сказать, от каких причин погода переменяется.

С. Я. Румовский
«Рассуждения о предсказании погод»

Для иллюстрации изложенной техники полезно привести практический пример. Прогноз магнитных бурь является достойной проблемой для марковского прогноза: слишком много случайных факторов *Космической погоды* должны сложится вместе, чтобы итогом стала наша головная боль.

Геомагнитные возмущения отслеживаются различными локальными и планетарными геомагнитными индексами. Причинами наиболее сильных из них — магнитных бурь — являются геоэффективные межпланетные возмущения солнечного происхождения: солнечные вспышки, выбросы корональных масс и межпланетные направленные взрывы, связанные с инверсиями межпланетного магнитного поля. Главная фаза геомагнитной бури представляет собой уменьшение H–компоненты поля от 50 до 400 nT (*нанотеслов*) и может продолжаться от нескольких часов до суток и даже более того [(1)].

Таким образом, магнитная буря — это сложный динамические процесс. Однако, ради простоты, при построении предиктора под «бурей» понимают лишь экстремальное событие — выброс индекса за фиксированный уровень, который принимают за «штатный» режим.

Одним из наиболее популярных индексов при исследовании бурь является среднесуточный D_{st}–индекс, который является мерой интенсивности кольцевого тока (рис. 2). Он вычисляется как усредненная величина возмущений, отсчитываемых от спокойного уровня по данным четырех магнитных обсерваторий, расположенных приблизительно вдоль магнитного экватора [11]. В магнитоспокойные дни величина D_{st} лежит в пределах ± 20 nT. Обычно, магнитной бурей считают значение $D_{st} \geqslant |-30|$ nT; для сильных бурь $D_{st} \geqslant |-50|$ nT и очень сильных $D_{st} \geqslant |-100|$ nT. Например, во время сильной бури 11 февраля 1958 года индекс D_{st} достиг величины -409 nT!

На рис. 3 приведена оценка эмпирической меры, полученная по 16801 значениям временного ряда среднесуточных значений D_{st}-индекса для бинарного алфавита при пороге -30 nT) и длине слова $K = 12$. Буре соответствовал символ 1, а ее отсутствию символ 0. Полученная гистограмма оказалась практически стационарной — она не меняет своей формы независимо от того, по какой половине ряда мы ее построим. Следовательно, в предположении эргодичности, ее можно считать оценкой инвариантной меры.

[11] D_{st}–индекс доступен на сайте: http://swdcdb.kugi.kyoto-u.ac.jp/dstdirV

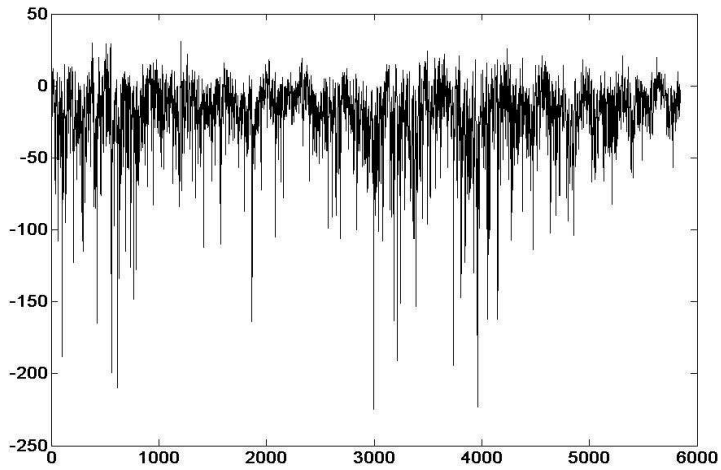

Рис. 2. График среднесуточных значений D_{st}–индекса (1957–2002 гг.). По вертикали — напряженность поля в nT

Для проверки самоподобия меры в пакете `FracLab`[12] были получены мультифрактальные спектры больших отклонений (2) этой меры для двух различных интервалов времени 1981–1991 гг. и 1992–2002 гг. Они фактически совпадают (рис. 4). При построении модели использовался фрагмент ряда с 1981 г. по 1996 г. (длина ряда — 5844 значения).

Для бинарного алфавита достаточно выбрать всего два сжимающих отображения, аттрактор которых — единичный отрезок. Так что проблема сводилась лишь к нахождению переходных вероятностей. Обратная задача для `IFS` решалась в среде `MATLAB 7.0` методом моментов (3) с использованием не менее 15 моментов с помощью `Pattern Search Tool`. Этот пакет содержит 5 методов поиска минимума оптимизационной задачи, включая генетический алгоритм. Все методы дали оценки переходных вероятностей, совпадающие с точностью до 5 знаков после запятой. Для моделирования меры использовались 2×10^5 рекуррентных точек с полученными вероятностями.

[12]Этот пакет доступен на сайте http://www.irccyn.ec-nantes.fr/hebergement/FracLab/

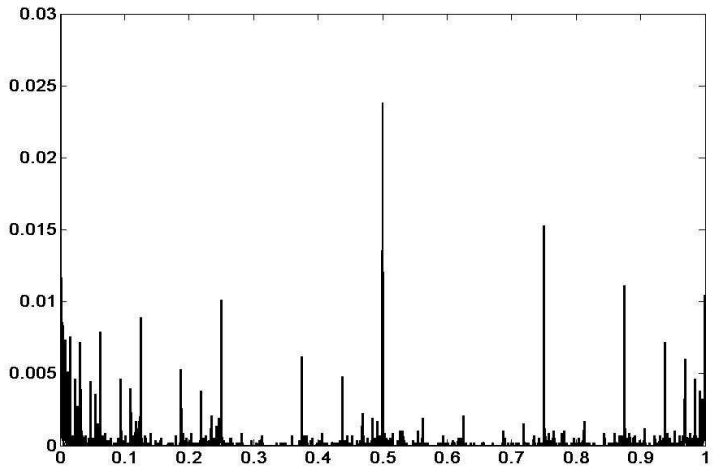

Рис. 3. Гистограмма распределения бинарных слов $K = 12$ для ряда D_{st}–индекса (1957–2002 гг.) с порогом -30 nT

В качестве альтернативы для решения обратной задачи использовалась *Теорема о Коллаже* в следующей модифицированной форме. Оценка эмпирической меры $\hat{\mu}$ рассматривалась как неподвижная точка марковского оператора. Переходные вероятности находились минимизацией функционала $d(\hat{\mu}, \mathbf{M}^{\circ n}\hat{\mu}) \to \min$, где расстоянием между двумя гистограммами $H = \{h_i\}$ и $K = \{k_i\}$ служила L_1–метрика:

$$d(H, K) = \sum_i |h_i - k_i|.$$

Сравнение модели с эмпирической мерой приведено на рис. 5, где показаны графики соответствующих кумулятивных гистограмм: $F = \sum_i |h - \overline{h}|$, построенных по аналогии с моделью случайного блуждания.

Для тестирования модели был получен *эпигноз* $^{(3)}$ 1095 значений нулей и единиц, на временном интервале 1997–1999 гг.

С практической точки зрения, ошибки в предсказании бури и ее отсутствия не равнозначны. Поэтому, для оценки качества прогноза использовался коэффициент $r = n_1/n_2$, предложенный в работе [32]. Он равен

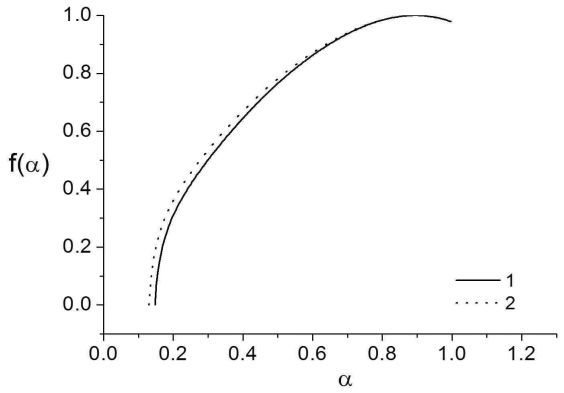

Рис. 4. Мультифрактальные спектры больших отклонений для эмпирической меры D_{st}–индекса 1981–1991 гг. (кривая 1) и 1992–2002 гг. (кривая 2).

отношению n_1 — числа *правильно* предсказанных суффиксов, при условии что соответствующие реальные значения содержали единицу, к общему количеству прогнозов слов n_2, реально содержащих единицу. Слова, содержащие одни нули, в статистике не участвовали. Метод моментов[13] дал следующие оценки эффективности предсказания (в скобках, для сравнения, указаны результаты работы [32]): на 1 день — $70.00(67.24)\%$, на 2 дня — $34.55(34.89)\%$ и на 3 дня — $19.63(19.47)\%$. Использование *Теоремы о Коллаже* дает эффективность: на 1 день — 70.00%, на 2 дня — 35.08% и на 3 дня — 19.63%.

Приведенные оценки показывают, что результаты вероятностного предсказания воспроизводимы и не слишком зависят от метода решения обратной задачи. В случае однодневного предсказания результаты вполне пригодны для практических задач, и даже для предсказания на 2 дня они лучше, чем случайный прогноз, который составляет лишь 25%.

[13] Максимальное число моментов было равно 30.

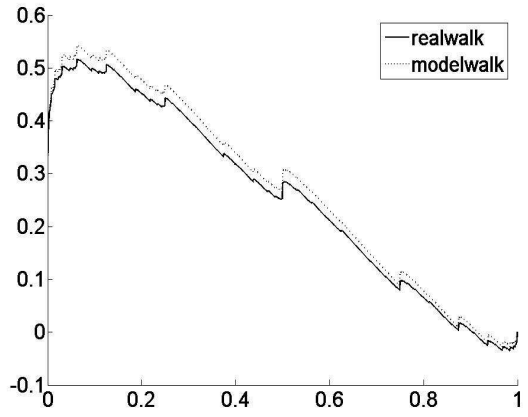

Рис. 5. Кумулятивные гистограммы для эмпирической меры и и IFS–модели (кружки — эмпирическая мера, сплошная линия — модель)

Примечания

1. Депрессия вызвана пересоединением силовых линий межпланетного и геомагнитного полей и флуктуациями размеров магнитосферы под действием потоков солнечного ветра. Эти процессы приводят к ускорению имеющейся в ней плазмы до энергий порядка нескольких тысяч электронвольт, формируя, таким образом, *кольцевой ток* на расстоянии от 3–5 радиусов Земли. Он течет в западном направлении (в северном полушарии) в окрестности Земли и порождает магнитное поле, направленное противоположно геомагнитному, и, следовательно, ослабляет его [39, 40].

2. Мультифрактальные свойства этого индекса с помощью вычислений *размерностей Реньи* были обнаружены *Ванлиссом, Аном и др.* [41].

3. Впервые метод марковского предсказания, основанный на IFS, для магнитных бурь применили *Ан, Ю и Ванлисс* [32]. Они использовали метод моментов. Мы воспроизвели их результаты и сравнили их с прогнозом, полученным с помощью *Теоремы о Коллаже*. Совсем недавно теми же авторами был предложен вариант оценки эмпириче-

ской меры, основанной на *Chaos Game* [42]. Он позволяет исключить зависимость модели от длины слова.

4. В футурологии существует следующая необщепринятая терминология. Термин *прогноз* используется в количественном контексте, скажем, когда требуется указать момент времени и значение некоторой величины. Его синонимом является обычно *предсказание*, хотя последний иногда используют в более общей ситуации. Например, прогноз землетрясений требует одновременного ответа на три вопроса: *где, когда, как сильно*? Но когда пытаются ответить на один или два вопроса, говорят о предсказании. *Эпигнозом* называют предсказание уже известных значений в предположении, что вы их не знаете. *Палегнозом* (*Postdiction*) называют прогноз неизвестного прошлого для временного ряда.

Путеводитель по литературе. Подробнее о физике магнитных бурь и геомагнитных индексах можно прочесть в монографиях [39, 40]. Разные подходы к предсказанию магнитных бурь описаны в статьях [42]– [45].

Заключение

Многое мы знали бы лучше,
если бы не стремились узнать
столь точно.

Иоганн Вольфганг Гёте
«Годы странствий Вильгельма
Мейстера, или Отрекающиеся»

Резюмируем схему марковского предсказания. Предположим, что мы встретились с ситуацией, когда нет оснований полагать, что динамическая система, которая продуцирует временной ряд, допускает гладкое детерминированное объяснение. Более того, есть основания полагать, что мы имеем дело со стохастическим сценарием, когда значения временного ряда являются результатом суперпозиции множества независимых факторов.

Пусть статистика отсчетов демонстрирует долговременные зависимости, или, что то же самое, подчиняется распределению с «тяжелыми» хвостами.

Чтобы формализовать ситуацию, мы вынуждены снизить требования к точности идентификации фазовых точек, т. е. перейти к «крупнозернистому» пространству состояний. Это можно сделать разными способами, и один из самых удачных заключается в использовании символической динамики.

Предположим, что реальная динамика по меньшей мере *рекуррентна*, т. е. ее случайные траектории достаточно часто возвращаются и пересекают некоторую плоскость — сечение Пуанкаре, транверсальное к фазовому потоку.

Маркируем ячейки крупнозернистого разбиения этой плоскости буквами или символами некоторого алфавита Σ с объемом $|\Sigma|$; в простейшем случае символов может быть всего два: 0 и 1. В наблюдаемом временном ряде эта процедура соответствует выбору нескольких или всего одного уровня. Рекуррентные траектории «напечатают» для нас некоторый текст.

Любой осмысленный текст должен содержать слова. Мы получим их, выбрав шаблон фиксированной длины K, посредством которого прочтем множество $|\Sigma|^K$ слов, последовательно сдвигая шаблон вдоль текста на один символ. В бинарном варианте число возможных слов составит величину 2^K.

Используя $|\Sigma|$–ичное представление числа, выбранное в соответствие с объемом алфавита, поставим в соответствие каждому слову точку на единичном отрезке. В итоге получится гистограмма встречаемости слов. Предположим, что такие гистограммы, построенные для всего ряда и для его двух половин, практически совпадают. Иными словами, гистограмма стационарна. Нормируем ее и получим частоту встречаемости (вероятность) для каждого слова из нашего текста.

Почему бы не использовать ее для предсказания K–го символа (суффикса) в последнем известном $(K-1)$–буквенном слове? Казалось бы проблема выбора возможной альтернативы для последней буквы легко решается с помощью полученной гистограммы! Однако, представьте себе реальную лингвистическую ситуацию. Мы имеем в своем распоряжении фрагмент какого-либо текста из сборника, содержащего, например, произведения *Александра Сергеевича Пушкина* и *Велимира Хлебникова*. Составим частотный словарь встречаемости слов, состоящих, скажем из двух букв, используя пушкинский текст. С помощью этой гистограммы мы вряд ли сможем предсказать удивительные дифтонги из текста *Велимира Хлебникова*!

С аналогичной ситуацией мы встречаемся для временных рядов. Фраг-

мент ряда, который мы использовали для построения гистограммы, может содержать чрезмерно малую долю некоторых слов лишь потому, что их было мало в том динамическом сценарии, который мы использовали. Однако в будущем такие слова вполне могут появиться! Вот поэтому и нужна модель, которая способна генерировать статистически содержательную выборку слов, сохранив масштабные свойства эмпирической гистограммы.

Честно говоря, я не могу привести корректные доводы, почему модель, основанная на IFS с вероятностями, является лучшим из возможных вариантов. Однако эвристические резоны в ее пользу существуют. Предположим, что мы проверили скейлинговые свойства гистограммы, оценив лежандровский или какой-либо другой мультифрактальный спектр, и нашли, к своему удовольствию, что эмпирическая гистограмма статистически самоподобна. Таким образом, мы можем считать, что располагаем оценкой стационарной меры, а имея в виду эргодичность, и инвариантной меры с носителем на аттракторе — единичном интервале.

С другой стороны, мы знаем, что такой аттрактор можно получить с помощью пары сжимающих отображений, которые являются *линейным* аналогом диссипативной системы; кстати, недавно *Брумхед* и др. [46] указали на интригующие связи между IFS и нелинейными дифференциальными уравнениями. Мы знаем также, что единственную инвариантную меру на единичном интервале можно получить, используя случайную динамику и IFS с вероятностями.

Вот пожалуй и все, потому что обратная задача — это уже дело техники. Итак, выбираем число отображений N, входящих в IFS, равным объему алфавита: $N = |\Sigma|$. При фиксированных коэффициентах IFS в методе моментов свободным параметром является номер M максимального момента. Увеличение M уменьшает ошибку функционала. Однако слишком большие M могут приводить к неустойчивости численного алгоритма.

Результатом решения является матрица переходных вероятностей. Генерируем модельную меру с помощью рекуррентной последовательности IFS с полученными значениями переходных вероятностей, пока не наберем объем, превышающий объем эмпирической меры.

Модельная мера, на которой основано предсказание, должна хорошо представлять все возможные слова. В случае бинарного алфавита, успешное предсказание получается тогда, когда гистограмма имеет форму «корыта», приводящего к сравнимым вероятностям переходов $0 \to 0$ и $1 \to 1$. Для прогноза на один символ альтернатива 0 или 1 выбирается на основе модельной меры.

Схема выглядит очень красивой и математически безупречной: прогноз делается на твердой основе — инвариантной вероятностной мере. Ведь *Будущее* генетически связано с *Вероятностью*. Как говорил профессор *Бенедикт Коуска* [14]: «*То, что случается, если на самом деле случается, — то и случается. Вероятность появляется лишь там, где нечто еще не успело случиться. Так утверждает наука*».

Литература

1. *Alefeld G., Koshelev M., Mayer G.* Why it is computationally harder to reconstruct the Past than to predict the Future // *Intern. J. Theor. Phys.*. – 1997. – v. 36. – pp. 1683–1689.
2. *Макаренко Н. Г.* Эмбедология и нейропрогноз // Сб. научн. тр. Всероссийской научно-техн. конференции *«Нейроинформатика–2003»*. Часть 1. – М.: Изд-во МИФИ, 2003. – с. 86–148.
3. *Макаренко Н. Г.* Фракталы, аттракторы, нейронные сети и все такое // Сб. научн. тр. Всероссийской научно-техн. конференции *«Нейроинформатика–2002»*. Часть 2. – М.: Изд-во МИФИ, 2002. – с. 121–169.
4. *Stark J.* Iterated Function Systems as Neural Networks // *Neural Networks*. – 1991. – v. 4. – pp. 679–690.
5. *Murtagh F.* Identifying the ultrametricity of time series // *European Physical Journal B*. – 2005. – v. 43 — pp. 573–579.
URL: http://www.cs.rhul.ac.uk/home/fionn/papers/
6. *Шашкин Ю. А.* Неподвижные точки. – Серия «Популярные лекции по математике», Вып. 60. – М.: Наука, 1989. – 77 с.
7. *Хемминг Р. В.* Теория кодирования и теория информации. Пер. с англ. – М.: Радио и связь, 1983. – 176 с.
8. *Hutchinson J.* Fractals: a mathematical framework. Deterministic and random fractals.
URL: http://www.maths.anu.edu.au/∼john/publications.html
9. *Barnsley M.* Fractals everywhere. – N.Y.: Academic Press, 1988. – 531 p.
10. *Falconer K.* Fractal geometry: Mathematical foundations and applications. – Wiley, 2003. – 337 p.
11. *Falconer K.* Techniques in fractal geometry. – John Wiley & Sons, 1997. – 256 p.

[14] *Станислав Лем.* «О невозможности жизни, О невозможности прогнозирования».

12. *Макаренко Н. Г.* Фракталы, мультифрактальные меры и аттракторы // *Нелинейные волны'2002*. – Нижний Новгород, 2003. – с. 381.

13. *Daw C. S., Finney C. E. A., Tracy E. R.* A review of symbolic analysis of experimental data // *Rev. of Scientific Instruments*. – 2003. – v. 74 — pp. 916–930.

14. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // ДАН СССР, 1965. Вып. 163. Т. 4. – с. 845–848.

15. *Гвишиани А. Д., Жижин М. Н., Иваненко Т. И.* Синтаксический анализ записей сильных движений // *Компьютерный анализ геофизических полей (Вычислительная сейсмология)*. 1990, Вып. 23. – с. 235–253.

16. *Tino P.* Spatial Representation of Symbolic Sequences through Iterative Function Systems // *IEEE Transactions on Systems, Man, and Cybernetics*. Part A: Systems and Humans, 1999. – **29**(4) – p. 386–392.

17. *Tino P., Dorffner G.* Predicting the future of discrete sequences from fractal representations of the past // *Machine Learning*, 2001. – **45** (2) – pp. 187–218.

18. *Tino P.* Multifractal properties of Hao's geometric representations of DNA sequences // *Physica A: Statistical Mechanics and its Applications*, 2002. – **304** (3–4) – pp. 480–494.
URL: http://www.cs.bham.ac.uk/∼pxt/my.publ.html

19. *Колмогоров А. Н., Фомин С. В.* Элементы теории функций и функционального анализа. – М: Наука, 1989. – 496 с.

20. *Титчмарш Е.* Теория функций. – М: Наука, 1980. – 454 с.

21. *Брудно А. Л.* Теория функций действительного переменного. – М: Наука, 1971. – 120 с.

22. *Hutchinson J.* Measure theory. – 1995.
URL: http://www.maths.anu.edu.au/∼john/lecture_notes.html

23. *Vrscay E. R.* From fractal image compression to fractal-based methods in mathematics // *Fractals in Multimedia*, ed. by *M. F. Barnsley*, *D. Saupe* and *E. R. Vrscay*. – Springer-Verlag, N. Y. – 2002.
URL: http://links.uwaterloo.ca/person.ed.html

24. *Vrscay E. R.* A Hitchhiker's guide to "fractal-based" function approximation and image compression.
URL: http://links.uwaterloo.ca/person.ed.html

25. *Rubner Y., Tomasi C., Guibas L. J.* The Earth mover's distance as a metric for image retrieval. – Technical Rep. STAN-CS-TN-98-86.

26. *Kaijser T.* Computing the Kantorovich distance for images // *J. Mathematical Imaging and Vision.* – 1998. – v. 9. – pp. 173–191.

27. *Wadstromer N.* Coding of fractal binary images with contractive set mappings composed of affine transformations. – PhD Thesis. – Linkopings Univer, 2001.

28. *Ling H.. Okada K.* EMD-L1: An efficient and robust algorithm for comparing histogram-based descriptors // *European Conference on Computer Vision 2006*. URL: http://www.cs.umd.edu/∼hbling/main.htm

29. *Лемешко Б. Ю.* Методы оптимизации. Конспект лекций. URL: http://www.ami.nstu.ru/∼headrd/

30. *Stark J.* A neural network to compute the Hutchinson metric in fractal image processing // *IEEE Trans. Neural Networks.* – 1991. – v. 2. – pp. 56–158.

31. *Barnsley M. F., Ervin V., Hardin D., Lancaster J.* Solution of an inverse problem for fractals and other sets // *Proc. Natl. Acad. Sci. USA*. – 1986. – v. 83. – pp. 1975–1977.

32. *Ahn V. V., Yu Z. G., Wanliss J. A., Watson S. M.* Prediction of magnetic storm events using the D_{st} index // *Nonlinear Processes in Geophysics.* – 2005. – v. 12. – pp. 799-806.

33. *Lutton E., Levy-Vehel J., Cretin G., Glevarec Ph., Roll C.* Mixed IFS: Resolution of the inverse problem using genetic programming // *Complex Systems.* – 1995. – v. 9. – pp. 375–398.

34. *Barnsley M., Hutchinson J., Stenflo O.* A fractal valued random iteration algorithm and fractal hierarchy. URL: http://wwwmaths.anu.edu.au/∼barnsley/pages/publication_list.htm

35. *Forte B., Vrscay E. R.* Solving the inverse problem for function/image approximations using iterated function systems, I. Theoretical basis; II. Algorithm and computations // *Fractals.* – 1994. – v. 2, 3. – pp. 325–346. URL: http://citeseer.ist.psu.edu/forte94solving.html

36. *Handy C. R., Mantica G.* Inverse problems in fractal construction: moment method solution // *Phys. D.* – 1990. – v. 43. – pp. 17–36.

37. *Abendat S., Demko S., Turchetti G.* Local moments and inverse problem for fractal measures // *Inverse Problems.* – 1992. – v. 8. – pp. 739–750.

38. *Iacus St. M., Torre D. L.* Approximating distribution functions by iterated function systems // *Departemental Working Papers 2002–03*, Department of Economics University of Milan Italy. URL: http://ideas.repec.org/e/pla155.html

39. *Яновский Б. М.* Земной магнетизм. – Ленинград: ЛГУ, 1978. – 592 с.

40. *Пудовкин М. И., Распопов О. М., Клейменова Н. Т.* Возмущения электромагнитного поля Земли. – Ленинград: ЛГУ, 1976. – 247 с.

41. *Wanliss J. A., Ahn V. V., Yu Z. G., Watson S.* Multifractal modeling of magnetic storms via symbolic dynamics analysis // *J. Geopys. Res.* – 2005. – v. 110. – p. 814.

42. *Yu Z. G., Ahn V.V., Wanliss J.A., Watson S.M.* Chaos game representation of the Dst index and prediction of geomagnetic storm events // *Chaos, Solitons and Fractals.* – 2006. – v. 31. – pp. 736–746.

43. *Watanabe Sh., Sagawa E., Ohtaka K., Shimazu H.* Prediction of the D_st index from solar wind parameters by a neural network method // *Earth Planets Space.* – 2002. – v. 54. – pp. 1263–1275.

44. *Stepanova M., Antonova E., Troshichev O.* Prediction of Dst variations from Polar Cap indices using time-delay neural network // *J. Atmosph. and Solar-Terrestrial Phys.* – 2005. – v. 67. – pp. 1658–1664.

45. *Strivastava N.* A logistic regression model for predicting the occurrence of intense geomagnetic storms // *Ann. Geophys.* – 2005. – v. 23. – pp. 2969–2974.

46. *Broomhead D. S., Huke J. P., Muldoon M. R., Stark J.* Iterated function system models of digital channels. // Proc. Roy. Soc. London, Ser. A. –2004. – v. 460. – pp. 3123-3142.

Николай Григорьевич МАКАРЕНКО, доктор технических наук, главный научный сотрудник Лаборатории компьютерного моделирования (Институт математики, Алма–Ата, Казахстан); доктор физико-математических наук и ведущий научный сотрудник Главной астрономической обсерватории РАН (Пулково, Санкт-Петербург, Россия). Области научных интересов: фрактальная геометрия, вычислительная топология, алгоритмическое моделирование, детерминированный хаос, нейронные сети, физика Солнца. Имеет более 90 научных публикаций.

В. КЕЦМАН
Университет Окленда,
Новая Зеландия
E-mail: v.kecman@auckland.ac.nz
http://www.support-vector.ws

НОВЫЙ SVM-АЛГОРИТМ ДЛЯ СВЕРХБОЛЬШИХ НАБОРОВ ДАННЫХ

Аннотация

В лекции рассматривается новейший алгоритм обучения для машин опорных векторов (SVM), известных также под наименованием «машины ядерных функций» (kernel machines), при работе со сверхбольшими наборами данных (например, содержащими несколько миллионов обучающих пар). Вначале сравниваются нейронные сети и машины опорных векторов с точки зрения решения с их помощью задач классификации (распознавания образов) и регрессии (аппроксимации функций), после чего вводится алгоритм обучения для машин опорных векторов (SVM-алгоритм). Показано, что выбор значений весов в SVM-алгоритме приводит к задаче квадратичного программирования с ограничениями. В отличие от классических задач квадратичного программирования, матрицы Гессе, с которыми приходится иметь дело при реализации SVM-алгоритмов, обычно очень плотно заполнены ненулевыми элементами. Кроме, того размерность для таких матриц меняется с изменением числа обучающих пар данных. Это приводит к появлению матриц Гессе сверхбольших размерностей, что затрудняет решение задачи обучения. Для решения такого рода сверхбольших задач предлагается новый итерационный алгоритм, получивший наименование ISDA (Iterative Single Data Algorithm), основывающийся на последовательном использовании обучающих пар данных из имеющегося обучающего набора. Дается доказательство сходимости этого алгоритма, которое базируется на сходимости итеративного метода Гаусса–Зейделя для решения систем линейных уравнений. Приводятся результаты проверки работоспособности предлагаемого алгоритма на тестовых наборах данных. Алгоритм ISDA показал себя наиболее быстрым среди существующих в настоящее время при точном решении задач классификации и регрессии для сверхбольших наборов обучающих данных.

VOJISLAV KECMAN
School of Engineering,
The University of Auckland
Private Bag 92019, Auckland, New Zealand
E-mail: v.kecman@auckland.ac.nz
http://www.support-vector.ws

NEW SUPPORT VECTOR MACHINES ALGORITHM FOR HUGE DATA SETS

Abstract

The seminar presents the newest learning algorithm for support vector machines (SVMs), a.k.a., kernel machines, when faced with huge data sets (say, millions of training data pairs). It starts with comparisons between the so-called neural networks (NNs) and SVMs for solving classification (pattern recognition) and regression (function approximation) tasks and then it introduces the SVMs learning algorithm. It shows how comes that learning the SVMs' weights means solving the quadratic programming (QP) problem with constraints. Unlike in classic QP problems, Hessian matrices involved in SVMs are usually extremely dense. In addition, their sizes scale with the number of training data pairs. This leads to huge Hessian matrices, and to intractable solutions. The new iterative single data algorithm (ISDA) is proposed for solving such huge problems. The proof of its convergence, based on the convergence of Gauss-Seidel iterative method for solving linear systems of equations, is given. Some comparisons on benchmark data sets are presented. ISDA implementation seems to be the quickest software for exact solving classification and regression tasks from huge training data sets problems at the moment.

Introduction

Today, we are surrounded by an ocean of all kind of experimental data (i. e., examples, samples, measurements, records, patterns, pictures, tunes, observations etc) produced by various sensors, cameras, microphones, pieces of software and/or other human made devices. The amount of data produced is enormous and ever increasing. The first obvious consequence of such a fact is — humans can't handle such massive quantity of data which are usually appearing in the numeric shape as the huge (rectangular or square) matrices. Typically, the number of their rows tells about the number of data pairs collected, and the number of columns represents the dimensionality of data. Thus, faced with the Giga- and Terabyte sized data files one has to develop new approaches, algorithms

and procedures. Few techniques for coping with huge data size problems are presented here. This explains the appearance of a wording "*huge data sets*" in the title of the seminar.

The direct consequence is that (instead of attempting to dive into the sea of hundreds of thousands or millions of high-dimensional data pairs) we have to develop other "machines" i. e., "devices" for analyzing, recognition in, and/or learning from, such huge data sets. The so-called "learning machine" is predominantly a piece of software that implements both the learning algorithm and the function (network, model) which parameters has to be determined by the learning part of the software. Next, it is worth of clarifying the fact that many authors tend to label similar (or even same) models, approaches and algorithms by different names. One is just destine to cope with concepts of data mining, knowledge discovery, neural networks, Bayesian networks, machine learning, pattern recognition, classification, regression, statistical learning, decision trees, decision making etc. All of them usually have a lot in common, and often they use same sets of techniques for adjusting, tuning, training or learning the parameters defining the models. The common object for all of them is a training data set. All the various approaches mentioned start with a set of data pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ represents the input variables (causes, observations, records) and y_i denote the measured outputs (responses, labels).

This is a seminar on (machine) learning from empirical data by applying support vector machines (SVMs). We first show some similarities and differences between the SVMs and NNs and then we introduce the QP based learning for SVMs. The SVMs fall into the big group of supervised learning algorithms. This means that for each input vector (measurements) $\mathbf{x}_i$ there is always a known output value (label) y_i. Here, we do not present neither the semi-supervised learning algorithms (when only smaller part of inputs have corresponding labeled outputs), nor unsupervised methods (such as PCA or ICA, when there are no labeled desired outputs at all). The basic aim of this chapter is to give, as far as possible, a condensed (but systematic) presentation of a novel learning paradigm embodied in SVMs. Our focus will be on the *constructive part* of the SVMs' learning algorithms for both the classification (pattern recognition) and regression (function approximation) problems. Consequently, we will not go into all the subtleties and details of the statistical learning theory (SLT) and structural risk minimization (SRM) which are the theoretical foundations for learning algorithms presented below. The approach here seems more appropriate for the application oriented readers. The theoretically minded and interested reader may find an extensive presentation of both the SLT and

SRM in (*Vapnik* and *Chervonenkis*, 1989; *Vapnik*, 1995, 1998; *Cherkassky* and *Mulier*, 1998; *Cristianini* and *Shawe-Taylor*, 2001; *Kecman*, 2001; *Schölkopf* and *Smola* 2002). Instead of diving into such statistically colored theory, a quadratic programming based learning (leading to parsimonious SVMs) will be presented in a gentle way — starting with linear separable problems, through the classification tasks having overlapped classes but still a linear separation boundary, beyond the linearity assumptions to the nonlinear separation boundary, and finally to the linear and nonlinear regression problems. Here, the adjective "parsimonious" denotes a SVM with a small number of support vectors ("hidden layer neurons"). The scarcity of the model results from a sophisticated, QP based, learning that matches the model capacity to data complexity ensuring a good generalization, i. e., a good performance of SVM on the future, previously, during the training unseen, data.

Same as the neural networks (or similarly to them), SVMs possess the well-known ability of being universal approximators of any multivariate function to any desired degree of accuracy. Consequently, they are of particular interest for modeling the unknown, or partially known, highly nonlinear, complex systems, plants or processes. Also, at the very beginning, and just to be sure what the whole chapter is about, we should state clearly when there is no need for an application of SVMs' model-building techniques.

In short, whenever there exists an analytical closed-form model or, there is a knowledge making it possible to devise one, there is no need to resort to the learning from empirical data by SVMs (or by any other type of a learning machine).

Basics of learning from data by NNs and SVMs

SVMs have been developed in the reverse order to the development of neural networks. SVMs evolved from the sound theory to the implementation and experiments, while the NNs followed more heuristic path, from applications and extensive experimentation to the theory. It is interesting to note that the very strong theoretical background of SVMs did not make them widely appreciated at the beginning. The publication of the first papers by *Vapnik* and *Chervonenkis* (Vapnik and Chervonenkis, 1964, 1968) went largely unnoticed till 1992. This was due to a widespread belief in the statistical and/or machine learning community that, despite being theoretically appealing, SVMs are neither suitable nor relevant for practical applications. They were taken seriously

only when excellent results on practical learning benchmarks were achieved (in numeral recognition, computer vision and text categorization). Today, SVMs show better results than (or comparable outcomes to) NNs and other statistical models, on the most popular benchmark problems.

The learning problem setting for SVMs is as follows: there is some unknown nonlinear dependency (mapping, function) $y = f(\mathbf{x})$ between some high-dimensional input vector $\mathbf{x}$ and the scalar output y (or the vector output $\mathbf{y}$ as in the case of multiclass SVMs). There is no information about the underlying joint probability functions here. Thus, one must perform a *distribution-free learning*. The only information available is a training data set $D = \{(\mathbf{x}_i, y_i) \in X \times Y\}$, $i = 1, \ldots, l$, where l stands for the number of the training data pairs and is therefore equal to the size of the training data set D. Often, y_i is denoted as d_i (i.e., t_i), where $d(t)$ stands for a desired (target) value. Hence, SVMs belong to the supervised learning techniques.

Note that this problem is similar to the classic statistical inference. However, there are several very important differences between the approaches and assumptions in training SVMs and the ones in classic statistics and/or NNs modeling. Classic statistical inference is based on the following three fundamental assumptions:

1. Data can be modeled by a set of linear in parameter functions; this is a foundation of a parametric paradigm in learning from experimental data.
2. In the most of real-life problems, a stochastic component of data is the normal probability distribution law, that is, the underlying joint probability distribution is a Gaussian distribution.
3. Because of the second assumption, the induction paradigm for parameter estimation is the maximum likelihood method, which is reduced to the minimization of the sum-of-errors-squares cost function in most engineering applications.

All three assumptions on which the classic statistical paradigm relied turned out to be inappropriate for many contemporary real-life problems (*Vapnik*, 1998) because of the following facts:

1. Modern problems are high-dimensional, and if the underlying mapping is not very smooth the linear paradigm needs an exponentially increasing number of terms with an increasing dimensionality of the input space X (an increasing number of independent variables). This is known as “the curse of dimensionality”.

2. The underlying real-life data generation laws may typically be very far from the normal distribution and a model-builder must consider this difference in order to construct an effective learning algorithm.
3. From the first two points it follows that the maximum likelihood estimator (and consequently the sum-of-error-squares cost function) should be replaced by a new induction paradigm that is uniformly better, in order to model non-Gaussian distributions.

In addition to the three basic objectives above, the novel SVMs' problem setting and inductive principle have been developed for standard contemporary data sets which are typically high-dimensional and sparse (meaning, the data sets contain small number of the training data pairs).

SVMs are the so-called "nonparametric" models. "Nonparametric" does not mean that the SVMs' models do not have parameters at all. On the contrary, their "learning" (selection, identification, estimation, training or tuning) is the crucial issue here. However, unlike in classic statistical inference, the parameters are not predefined and their number depends on the training data used. In other words, parameters that define the capacity of the model are data-driven in such a way as to match the model capacity to data complexity. This is a basic paradigm of the structural risk minimization (SRM) introduced by *Vapnik* and *Chervonenkis* and their coworkers that led to the new learning algorithm. Namely, there are two basic constructive approaches possible in designing a model that will have a good generalization property:

1. Choose an appropriate structure of the model (order of polynomials, number of HL neurons, number of rules in the fuzzy logic model) and, keeping the estimation error (a.k.a. confidence interval, a.k.a. variance of the model) fixed in this way, minimize the training error (i.e., empirical risk), or
2. Keep the value of the training error (a.k.a. an approximation error, a.k.a. an empirical risk) fixed (equal to zero or equal to some acceptable level), and minimize the confidence interval.

Classic NNs implement the first approach (or some of its sophisticated variants) and SVMs implement the second strategy. In both cases the resulting model should resolve the trade-off between under-fitting and over-fitting the training data. The final model structure (its order) should ideally *match the learning machines capacity with training data complexity.* This important difference in two learning approaches comes from the minimization of different

cost (error, loss) functionals. Table 1 tabulates the basic risk functionals applied in developing the three contemporary statistical models.

Table 1. Basic Models and Their Error (Risk) Functionals

Multilayer perceptron (NN)	Regularization Network (Radial Basis Functions Network)	Support Vector Machine
$R=\sum_{i=1}^{l}\underbrace{(d_i-f(\mathbf{x}_i,\mathbf{w}))^2}_{\text{Closeness to data}}$	$R=\sum_{i=1}^{l}\underbrace{(d_i-f(\mathbf{x}_i,\mathbf{w}))^2}_{\text{Closeness to data}}+\lambda\underbrace{\|\mathbf{P}f\|^2}_{\text{Smoothness}}$	$R=\sum_{i=1}^{l}\underbrace{L_\varepsilon}_{\text{Closeness to data}}+\underbrace{\Omega(l,h)}_{\text{Capacity of a machine}}$

Closeness to data ≈= training error, a.k.a. empirical risk

In Table 1, d_i stands for desired values, $\mathbf{w}$ is the weight vector subject to training, λ is a regularization parameter, $\mathbf{P}$ is a smoothness operator, L_ε is a SVMs' loss function, h is a VC dimension and Ω is a function bounding the capacity of the learning machine. In *classification problems* L_ε is typically 0–1 loss function, and in regression problems L_ε is the so-called Vapnik's ε-insensitivity loss (error) function (see more about it in the section "Regression by Support Vector Machines")

$$L_\varepsilon = |y - f(\mathbf{x},\mathbf{w})|_\varepsilon = \begin{cases} 0, & \text{if } |y - f(\mathbf{x},\mathbf{w})| \leqslant \varepsilon, \\ |y - f(\mathbf{x},\mathbf{w})| - \varepsilon, & \text{otherwise,} \end{cases} \tag{1}$$

where ε is a radius of a "tube" within which the regression function must lie after the successful learning. (Note that for $\varepsilon = 0$, the interpolation of training data will be performed). It is interesting to note that (*Girosi*, 1997) has shown that under some constraints the SV machine can also be derived from the framework of regularization theory rather than SLT and SRM. Thus, *unlike the classic adaptation algorithms present in NNs (that work in the L_2 norm) SV machines represent novel learning techniques which perform SRM*. In this way, the SV machine creates a model with minimized VC dimension and when the VC dimension of the model is low, the expected probability of error is low as well. This means; it's very likely that they will have good performance on previously unseen data, i.e. a good generalization. This property is of particular

interest because the model that generalizes well is a good model and not the model that performs well on training data pairs. Very good performance on training data usually leads to an extremely undesirable overfitting.

As it will be shown below, in the "simplest" pattern recognition tasks, support vector machines use a linear separating hyperplane to create a *classifier with a maximal margin*. In order to do that, the learning problem for the SV machine will be cast as a *constrained nonlinear optimization* problem. In this setting the cost function will be quadratic and the constraints linear (i.e., one will have to solve a classic *quadratic programming problem*).

In cases when given classes cannot be linearly separated in the original input space, the SV machine first (non-linearly) transforms the original input space into a higher dimensional *feature space*. This transformation can be achieved by using various nonlinear mappings; polynomial, sigmoid as in multilayer perceptrons, RBF mappings having as the basis functions radially symmetric functions such as Gaussians, or multiquadrics or different spline functions. After this nonlinear transformation step, the task of a SV machine in finding the linear optimal separating hyperplane in this feature space is "relatively trivial". Namely, the optimization problem to solve in a feature space will be of the same kind as the calculation of a maximal margin separating hyperplane in an original input space for linearly separable classes. How, after the specific nonlinear transformation, nonlinearly separable problems in input space can become linearly separable problems in a feature space will be shown later.

In a probabilistic setting, there are three basic components in all supervised learning from data tasks: a *generator* of random inputs $\mathbf{x}$, a *system* whose *training responses* y (i.e., d) are used for training the learning machine, and a *learning machine* which, by using inputs $\mathbf{x}_i$ and system's responses y_i, should learn (estimate, model) the unknown dependency between these two sets of variables (namely, $\mathbf{x}_i$ and y_i) defined by the weight vector $\mathbf{w}$(Fig. 1).

The figure shows the most common learning setting that some readers may have already seen in various other fields — notably in statistics, NNs, control system identification and/or in signal processing. During the (successful) training phase a learning machine should be able to find the relationship between an input space X and an output space Y, by using data D in regression tasks (or to find a function that separates data within the input space, in classification ones). The result of a learning process is an "approximating function" $f_a(\mathbf{x}, \mathbf{w})$, which in statistical literature is also known as, a *hypothesis* $f_a(\mathbf{x}, \mathbf{w})$. This function approximates the underlying (or true) dependency between the input and output in the case of regression, and the decision boundary, i.e., separation function,

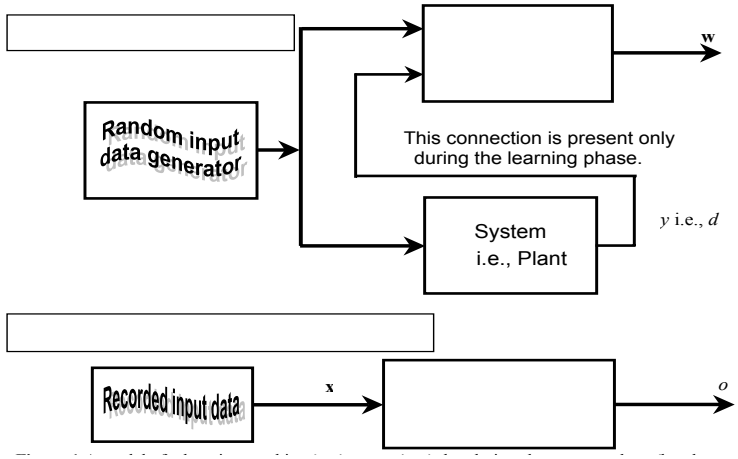

Figure 1. A model of a learning machine *(top)* $\mathbf{w} = w(\mathbf{x}, y)$ that during *the training phase* (by observing inputs $\mathbf{x}_i$ to, and outputs y_i from, the system) estimates (learns, adjusts, trains, tunes) its parameters (weights) $\mathbf{w}$, and in this way learns mapping $y = f(\mathbf{x}, \mathbf{w})$ performed by the system. The use of $f_a(\mathbf{x}, \mathbf{w}) \sim y$ denotes that *we will rarely* try to *interpolate* training data pairs. We would rather seek an *approximating function* that can generalize well. After the training, at the *generalization* or *test phase*, the output from a machine $o = f_a(\mathbf{x}, \mathbf{w})$ is expected to be "a good" estimate of a system's true response y.

in a classification. The chosen hypothesis $f_a(\mathbf{x}, \mathbf{w})$ belongs to a *hypothesis space of functions* $H(f_a \in H)$, and it is a function that minimizes some *risk functional* $R(\mathbf{w})$.

It may be practical to remind the reader that under the general name "approximating function" we understand any mathematical structure that maps inputs $\mathbf{x}$ into outputs y. Hence, an "approximating function" may be: a multilayer perceptron NN, RBF network, SV machine, fuzzy model, Fourier truncated series or polynomial approximating function. Here we discuss SVMs. A set of parameters $\mathbf{w}$ is the very subject of learning and generally these parameters are called *weights*. These parameters may have different geometrical and/or physical meanings. Depending upon the hypothesis space of functions H we are working with the parameters $\mathbf{w}$ are usually:

- the hidden and the output layer weights in multilayer perceptrons;
- the rules and the parameters (for the positions and shapes) of fuzzy subsets;
- the coefficients of a polynomial or Fourier series;
- the centers and (co)variances of Gaussian basis functions as well as the output layer weights of this RBF network;
- the support vector weights in SVMs.

There is another important class of functions in learning from examples tasks. A learning machine tries to capture an unknown *target function* $f_o(\mathbf{x})$ that is believed to belong to some target space T, or to a class T, that is also called a *concept class*. Note that we rarely know the target space T and that our learning machine generally does not belong to the same class of functions as an unknown target function $f_o(\mathbf{x})$. Typical examples of target spaces are continuous functions with s continuous derivatives in n variables; Sobolev spaces (comprising square integrable functions in n variables with s square integrable derivatives), band-limited functions, functions with integrable Fourier transforms, Boolean functions, etc. In the following, we will assume that the target space T is a space of differentiable functions. The basic problem we are facing stems from the fact that we know very little about the possible underlying function between the input and the output variables. All we have at our disposal is a training data set of labeled examples drawn by independently sampling $a(X \times Y)$ space according to some unknown probability distribution.

Now, we stop with general issues and concepts and we present the learning algorithms for SVMs, or we show how, from a training data, they can learn the unknown dependency.

Support Vector Machines in Classification and Regression

Below, we focus on the algorithm for implementing the SRM induction principle on the given set of functions. It implements the strategy mentioned previously — it keeps the training error fixed and minimizes the confidence interval. We first consider a "simple" example of linear decision rules (i.e., the separating functions will be hyperplanes) for binary classification (dichotomization) of linearly separable data. In such a problem, we are able to perfectly classify data pairs, meaning that an empirical risk can be set to zero. It is the easiest classification problem and yet an excellent introduction of all relevant and important ideas underlying the SLT, SRM and SVM.

Our presentation will gradually increase in complexity. It will begin with a *Linear Maximal Margin Classifier for Linearly Separable Data* where there is no sample overlapping. Afterwards, we will allow some degree of overlapping of training data pairs. However, we will still try to separate classes by using linear hyperplanes. This will lead to the *Linear Soft Margin Classifier for Overlapping Classes*. In problems when linear decision hyperplanes are no longer feasible, the mapping of an input space into the so-called feature space (that "corresponds" to the HL in NN models) will take place resulting in the *Nonlinear Classifier.* Finally, in the subsection on *Regression by SV Machines* we introduce same approaches and techniques for solving regression (i.e., function approximation) problems.

Linear Maximal Margin Classifier for Linearly Separable Data

Consider the problem of binary classification or dichotomization. Training data are given as

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l), \ \mathbf{x} \in \Re^n, \ y \in \{+1, -1\}. \tag{2}$$

For reasons of visualization only, we will consider the case of a two-dimensional input space, i.e., $\mathbf{x} \in \Re^2$. Data are linearly separable and there are many different hyperplanes that can perform separation (Fig. 2)[1]. How to find "the best" one? The difficult part is that all we have at our disposal are sparse training data. There are many functions that can solve given pattern recognition (or functional approximation) tasks. In such a problem setting, the SLT (developed in the 1960s by *Vapnik* and *Chervonenkis*) shows that it is crucial to restrict the class of functions implemented by a learning machine to one with a complexity that is suitable for the amount of available training data.

In the case of a classification of linearly separable data, this idea is transformed into the following approach — among all the hyperplanes that minimize the training error (i.e., empirical risk) find the one with the largest margin. This is an intuitively acceptable approach. Just by looking at Fig. 2 we will find that the dashed separation line shown in the *right graph* seems to promise *probably* good classification while facing previously unseen data (meaning, in the generalization, i.e. test, phase). Or, at least, it seems to probably be better in

[1]Actually, for $\mathbf{x} \in \Re^2$, the separation is performed by "planes" $w_1x_1 + w_2x_2 + b = o$. In other words, the decision boundary, i.e., the separation line in input space is defined by the equation $w_1x_1 + w_2x_2 + b = 0$.

generalization than the dashed decision boundary having smaller margin shown in the left graph. This can also be expressed as that a classifier with smaller margin will have higher expected risk.

By using given training examples, during the learning stage, our machine finds parameters $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T$ and b of a discriminant or decision function $d(\mathbf{x}, \mathbf{w}, b)$ given as

$$d(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{n} w_i x_i + b, \tag{3}$$

where $\mathbf{x}, \mathbf{w} \in \Re^n$, and the scalar b is called *a bias*[2]. After the successful training stage, by using the weights obtained, the learning machine, given previously unseen pattern $\mathbf{x}_p$, produces output o according to an *indicator function* given as

$$i_F = o = \mathrm{sign}(d(\mathbf{x}_p, \mathbf{w}, b)), \tag{4}$$

where o is the standard notation for the *output* from the learning machine. In other words, *the decision rule is*:

if $d(\mathbf{x}_p, \mathbf{w}, b) > 0$, the pattern $\mathbf{x}_p$ belongs to a class 1 (i.e., $o = y_1 = +1$), and

if $d(\mathbf{x}_p, \mathbf{w}, b) < 0$ the pattern $\mathbf{x}_p$ belongs to a class 2 (i.e., $o = y_2 = -1$).

The *indicator function* i_F given by (4) is a step-wise (i.e., a stairs-wise) function (see Figs 3 and 4). At the same time, the decision (or discriminant) function $d(\mathbf{x}_p, \mathbf{w}, b)$ is a hyperplane. Note also that both a decision hyperplane d and the indicator function i_F live in an $n+1$-dimensional space or they lie "over" a training pattern's n-dimensional input space. There is one more mathematical object in classification problems called a *separation boundary* that lives in the same n-dimensional space of input vectors $\mathbf{x}$. Separation boundary separates vectors $\mathbf{x}$ into two classes. Here, in cases of linearly separable data, the boundary is also a (separating) hyperplane but of a lower order than $d(\mathbf{x}_p, \mathbf{w}, b)$.

The decision (separation) *boundary* is an intersection of a decision *function* $d(\mathbf{x}, \mathbf{w}, b)$ and a space of input features. It is given by

$$d(\mathbf{x}, \mathbf{w}, b) = 0. \tag{5}$$

All these functions and relationships can be followed, for two-dimensional inputs $\mathbf{x}$, in Fig. 6. In this particular case, the decision boundary i.e., separating

[2]Note that the dashed separation lines in Fig. 2 represent the line that follows from $d(\mathbf{x}, \mathbf{w}, b) = 0$.

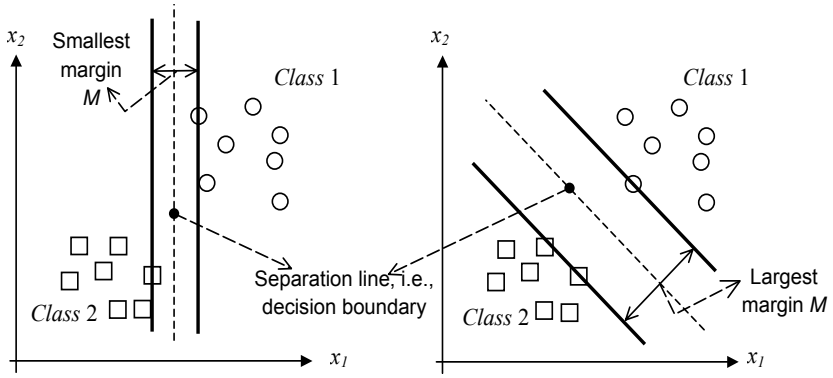

Figure 2. Two-out-of-many separating lines: a good one with a large margin *(right)* and a less acceptable separating line with a small margin, *(left)*.

(hyper)plane is actually a separating line in a $x_1 - x_2$ plane and, a decision function $d(\mathbf{x}, \mathbf{w}, b)$ is a plane over the 2-dimensional space of features, i.e., over a $x_1 - x_2$ plane.

In the case of 1-dimensional training patterns x (i.e., for 1-dimensional inputs x to the learning machine), decision function $d(\mathbf{x}, \mathbf{w}, b)$ is a straight line in an $x - y$ plane. An intersection of this line with an x-axis defines a point that is a separation boundary between two classes. This can be followed in Fig. 4. Before attempting to find an optimal separating hyperplane having the largest margin, we introduce the concept of the *canonical hyperplane.* We depict this concept with the help of the 1-dimensional example shown in Fig. 4. Not quite incidentally, the decision plane $d(\mathbf{x}, \mathbf{w}, b)$ shown in Fig. 6 is also a *canonical* plane. Namely, the values of d and of i_F are the same and both are equal to $|1|$ for the support vectors depicted by stars. At the same time, for all other training patterns $|d| > |i_F|$. In order to present a notion of this new concept of the canonical plane, first note that there are many hyperplanes that can correctly separate data. In Fig. 4 three different decision functions $d(\mathbf{x}, \mathbf{w}, b)$ are shown. There are infinitely many more. In fact, given $d(\mathbf{x}, \mathbf{w}, b)$, all functions $d(\mathbf{x}, k\mathbf{w}, kb)$, where k is a positive scalar, are correct decision functions too. Because parameters $(\mathbf{w}, b)$ describe the same separation hyperplane as parameters $(k\mathbf{w}, kb)$ there is a need to introduce the notion of a *canonical hyperplane*.

A hyperplane is in the canonical form with respect to training data $\mathbf{x} \in X$ if

$$\min_{x_i \in X} |\mathbf{w}^T \mathbf{x}_i + b| = 1. \tag{6}$$

The solid line $d(\mathbf{x}, \mathbf{w}, b) = -2x + 5$ in Fig. 4 fulfills (6) because *its minimal absolute value for the given six training patterns* belonging to two classes is 1. It achieves this value for two patterns, chosen as support vectors, namely for $x_3 = 2$, and $x_4 = 3$. For all other patterns, $|d| > 1$.

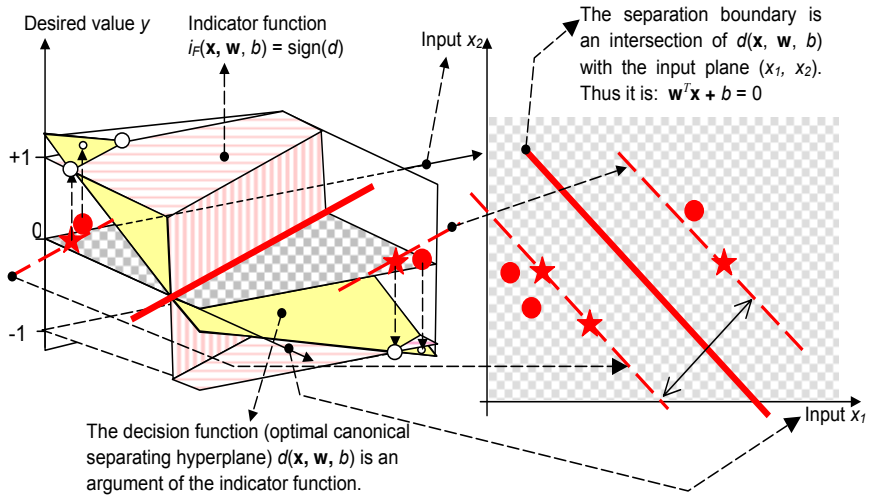

Figure 3. The definition of a decision (discriminant) *function* or hyperplane $d(\mathbf{x}, \mathbf{w}, b)$, a decision (separating) *boundary* $d(\mathbf{x}, \mathbf{w}, b) = 0$ and an indicator function $i_F = \text{sign}(d(\mathbf{x}_p, \mathbf{w}, b))$ which value represents a learning, or SV, machine's output o.

Note an interesting detail regarding the notion of a canonical hyperplane that is easily checked. There are many different hyperplanes (planes and straight lines for 2–D and 1–D problems in Figs 3 and 4 respectively) that have the same separation boundary (solid line and a dot in Figs 3 (right) and 4 respectively). At the same time there are far fewer hyperplanes that can be defined as canonical ones fulfilling (6). In Fig. 4, i.e., for a 1-dimensional input vector x, the canonical hyperplane is unique. This is not the case for training patterns of

higher dimension. Depending upon the configuration of class' elements, various canonical hyperplanes are possible.

Therefore, there is a need to define an *optimal* canonical hyperplane (OCSH) as a canonical hyperplane having a *maximal margin*. This search for a separating, maximal margin, canonical hyperplane is the ultimate learning goal in statistical learning theory underlying SV machines. Carefully note the adjectives used in the previous sentence. The hyperplane obtained from a limited training data must have a *maximal margin* because it will *probably* better classify new data. It must be in *canonical* form because this will ease the quest for significant patterns, here called support vectors. The canonical form of the hyperplane will also simplify the calculations. Finally, the resulting hyperplane must ultimately *separate* training patterns.

We avoid the derivation of an expression for the calculation of a distance (margin M) between the closest members from two classes for its simplicity here. The margin M can be derived by both the geometric and algebraic argument and is given as

$$M = \frac{2}{\|\mathbf{w}\|}. \tag{7}$$

This important result will have a great consequence for the constructive (i.e., learning) algorithm in a design of a maximal margin classifier. It will lead to solving a quadratic programming (QP) problem which will be shown shortly. Hence, the "good old" gradient learning in NNs will be replaced by solution of the QP problem here. This is the next important difference between the NNs and SVMs and follows from the implementation of SRM in designing SVMs, instead of a minimization of the sum of error squares, which is a standard cost function for NNs.

Equation (7) is a very interesting result showing that minimization of a norm of a hyperplane normal weight vector $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}} = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$ leads to a maximization of a margin M. Because a minimization of $\sqrt{f}$ is equivalent to the minimization of f, the minimization of a norm $||\mathbf{w}||$ equals a minimization of $\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{n} w_i^2 = w_1^2 + w_2^2 + \cdots + w_n^2$, and this leads to a maximization of a margin M. Hence, the learning problem is

$$\text{minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w}, \tag{8a}$$

subject to constraints introduced and given in (8b) below[3].

[3] A multiplication of $\mathbf{w}^T\mathbf{w}$ by 0.5 is for numerical convenience only, and it doesn't change the solution.

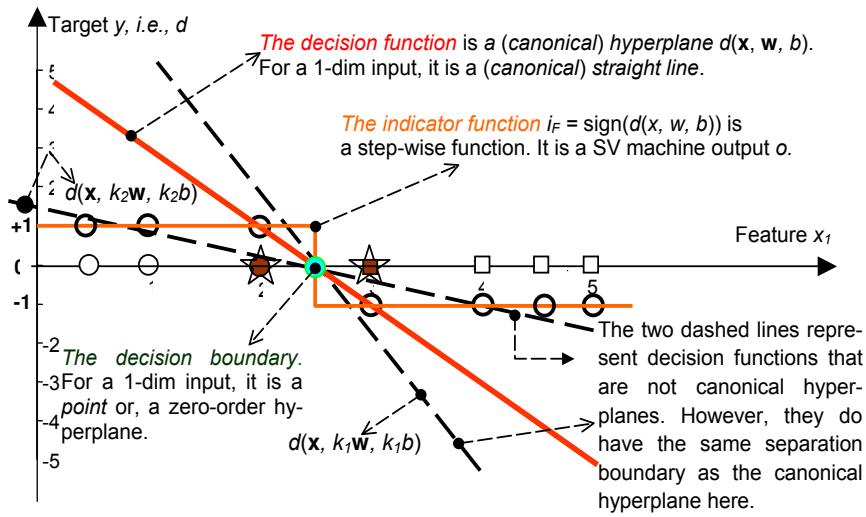

Figure 4. SV classification for 1-dimensional inputs by the linear decision function. Graphical presentation of a *canonical hyperplane*. For 1-dimensional inputs, it is actually a canonical straight line (depicted as a thick straight solid line) that passes through points $(+2, +1)$ and $(+3, -1)$ defined as the support vectors (stars). The two dashed lines are the two other decision hyperplanes (i.e., straight lines). The training input patterns $\{x_1 = 0.5,\ x_2 = 1,\ x_3 = 2\} \in Class\,1$ have a desired or target value (label) $y_1 = +1$. The inputs $\{x_4 = 3,\ x_5 = 4,\ x_6 = 4.5,\ x_7 = 5\} \in Class\,2$ have the label $y_2 = -1$.

Note that in the case of linearly separable classes empirical error equals zero ($R_{emp} = 0$ in (2a)) and minimization of $\mathbf{w}^T\mathbf{w}$ corresponds to a minimization of a confidence term Ω. The OCSH, i.e., a separating hyperplane with the largest margin defined by $M = 2/||\mathbf{w}||$, specifies *support vectors*, i.e., training data points closest to it, which satisfy $y_j[\mathbf{w}^T\mathbf{x}_j + b] \equiv 1,\ j = 1, N_{SV}$. For all the other (non-SVs data points) the OCSH satisfies inequalities $y_j[\mathbf{w}^T\mathbf{x}_j + b] > 1$. In other words, for all the data, OCSH should satisfy the following constraints

$$y_j[\mathbf{w}^T\mathbf{x}_j + b] > 1, \quad i = 1, l \tag{8b}$$

where l denotes a number of training data points, and N_{SV} stands for a number

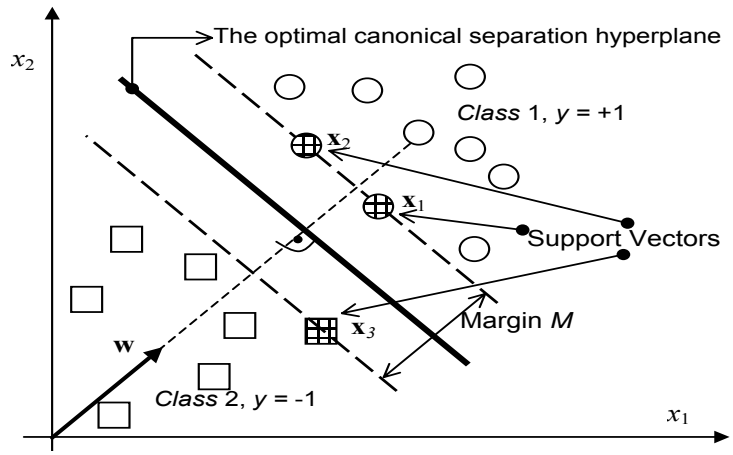

Figure 5. The optimal canonical separating hyperplane (OCSH) with the largest margin intersects halfway between the two classes. The points closest to it (satisfying $y_j|\mathbf{w}^T\mathbf{x}_j + b| = 1$, $j = 1, N_{SV}$) are *support vectors* and the OCSH satisfies $y_j|\mathbf{w}^T\mathbf{x}_j + b| \geqslant 1$, $j = 1, l$ (where l denotes the number of training data and N_{SV} stands for the number of SV). Three support vectors ($\mathbf{x}_1$ and $\mathbf{x}_2$ from class 1, and $\mathbf{x}_3$ from class 2) are the textured training data.

of SVs. The last equation can be easily checked visually in Figs 3 and 4 for 2-dimensional and 1-dimensional input vectors $\mathbf{x}$ respectively. Thus, in order to find the OCSH having a maximal margin, a learning machine should minimize $||\mathbf{w}||^2$ subject to the inequality constraints (8b). This is a *classic quadratic optimization problem with inequality constraints*. Such an optimization problem is solved by the *saddle point* of the Lagrange functional (Lagrangian)[4]

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l} \alpha_i\{y_i[\mathbf{w}^T\mathbf{x}_i + b] - 1\} \tag{9}$$

where the α_i are Lagrange multipliers. The search for an optimal *saddle point*

[4]In forming the Lagrangian, for constraints of the form $f_i > 0$, the inequality constraints equations are multiplied by *nonnegative* Lagrange multipliers (i.e., $\alpha_i \geqslant 0$) and *subtracted* from the objective function.

$(\mathbf{w}_o, b_o, \boldsymbol{\alpha}_o)$ is necessary because Lagrangian L must be *minimized* with respect to $\mathbf{w}$ and b, and has to be *maximized* with respect to nonnegative α_i (i.e., $\alpha_i \geqslant 0$ should be found). This problem can be solved either in a *primal space* (which is the space of parameters $\mathbf{w}$ and b) or in a *dual space* (which is the space of Lagrange multipliers α_i). The second approach gives insightful results and we will consider the solution in a dual space below. In order to do that, we use Karush-Kuhn-Tucker (KKT) conditions for the optimum of a constrained function. In our case, both the objective function (9) and constraints (8b) are *convex* and KKT conditions are *necessary* and *sufficient* conditions for a maximum of (9). These conditions are: at the saddle point $(\mathbf{w}_o, b_o, \boldsymbol{\alpha}_o)$, derivatives of Lagrangian L with respect to primal variables should vanish which leads to,

$$\frac{\partial L}{\partial \mathbf{w}_o} = 0, \text{ i.e.,} \qquad \mathbf{w}_o = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \tag{10}$$

$$\frac{\partial L}{\partial b_o} = 0, \text{ i.e.,} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{11}$$

and the KKT complementarity conditions below (stating that at the solution point the products between dual variables and constraints equals zero) must also be satisfied,

$$\alpha_i \{ y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1 \} = 0, \quad i = 1, l. \tag{12}$$

Substituting (10) and (11) into a *primal variables Lagrangian* $L(\mathbf{w}, b, \boldsymbol{\alpha})$ (9), we change to the *dual variables Lagrangian* $L_d(\alpha)$

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j. \tag{13}$$

In order to find the optimal hyperplane, a dual Lagrangian $L_d(\boldsymbol{\alpha})$ has to be *maximized* with respect to nonnegative α_i (i.e., α_i must be in the nonnegative quadrant) and with respect to the equality constraint as follows

$$\alpha_i \geqslant 0, \quad i = 1, l, \tag{14a}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0. \tag{14b}$$

Note that the dual Lagrangian $L_d(\boldsymbol{\alpha})$ is expressed in terms of training data and depends *only* on the *scalar products* of input patterns $(\mathbf{x}_i^T\mathbf{x}_j)$. The dependency of $L_d(\boldsymbol{\alpha})$ on a scalar product of inputs will be very handy later when analyzing nonlinear decision boundaries and for general nonlinear regression. Note also that the number of unknown variables equals the number of training data l. After learning, the number of free parameters is equal to the number of SVs but it does not depend on the dimensionality of input space. Such a *standard quadratic optimization problem* can be expressed in a *matrix notation* and formulated as follows:

Maximize

$$L_d(\boldsymbol{\alpha}) = -0.5\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} + \mathbf{f}^T\boldsymbol{\alpha}, \tag{15a}$$

subject to

$$\mathbf{y}^T\boldsymbol{\alpha} = 0, \tag{15b}$$

$$\alpha_i \geqslant 0, \quad i = 1, l \tag{15c}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T$, $\mathbf{H}$ denotes the Hessian matrix ($H_{ij} = y_i y_j(\mathbf{x}_i\mathbf{x}_j) = y_i y_j \mathbf{x}_i^T\mathbf{x}_j$) of this problem[5], and $\mathbf{f}$ is an $(l, 1)$ unit vector $\mathbf{f} = \mathbf{1} = [1\,1\,\ldots\,1]^T$. Solutions α_{oi} of the dual optimization problem above determine the parameters $\mathbf{w}_o$ and b_o of the optimal hyperplane according to (10) and (12) as follows

$$\mathbf{w}_o = \sum_{i=1}^{l} \alpha_{oi} y_i \mathbf{x}_i, \tag{16a}$$

$$b_o = \frac{1}{N_{SV}} \sum_{s=1}^{N_{SV}} \left(\frac{1}{y_s} - \mathbf{x}_s^T\mathbf{w}_o\right) = \frac{1}{N_{SV}} \sum_{s=1}^{N_{SV}} (y_s - \mathbf{x}_s^T\mathbf{w}_o), \quad s = 1, N_{SV}. \tag{16b}$$

In deriving (16b) we used the fact that y can be either +1 or -1, and $1/y = y$. N_{SV} denotes the number of support vectors. There are two important observations about the calculation of $\mathbf{w}_o$. First, an optimal weight vector $\mathbf{w}_o$, is obtained in (16a) as a linear combination of the training data points and second, $\mathbf{w}_o$ (same as the bias term b_0) is calculated by using only the selected data points called *support vectors* (SVs). The fact that the summations in (16a) goes over all training data patterns (i.e., from 1 to l) is irrelevant because the Lagrange multipliers for all non-support vectors equal zero ($\alpha_{oi} = 0, \quad i = N_{SV} + 1, l$).

[5]Note that maximization of (15a) equals a minimization of $L_d(\boldsymbol{\alpha}) = 0.5\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} - \mathbf{f}^T\boldsymbol{\alpha}$, subject to the same constraints.

Finally, having calculated $\mathbf{w}_o$ and b_o we obtain a decision hyperplane $d(\mathbf{x})$ and an indicator function $i_F = o = \mathrm{sign}(d(\mathbf{x}))$ as given below

$$d(\mathbf{x}) = \sum_{i=1}^{l} w_{oi} x_i + b_o = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}^T \mathbf{x}_i + b_o\,, \quad i_F = o = \mathrm{sign}(d(\mathbf{x})). \quad (17)$$

Training data patterns having non-zero Lagrange multipliers are called *support vectors*. For linearly separable training data, all support vectors lie on the margin and they are generally just a small portion of all training data (typically, $N_{SV} \ll l$). Figs 3, 4 and 5 show the geometry of standard results for non-overlapping classes.

Before presenting applications of OCSH for both overlapping classes and classes having nonlinear decision boundaries, we will comment only on whether and how SV based linear classifiers actually implement the SRM principle. The more detailed presentation of this important property can be found in (*Kecman*, 2001; *Schölkopf* and *Smola* 2002)). First, it can be shown that an increase in margin reduces the number of points that can be shattered i.e., the increase in margin reduces the VC dimension, and this leads to the decrease of the SVM capacity. In short, by minimizing $||\mathbf{w}||$ (i.e., maximizing the margin) the SV machine training actually minimizes the VC dimension and consequently a generalization error (expected risk) at the same time. This is achieved by imposing a structure on the set of canonical hyperplanes and then, during the training, by choosing the one with a minimal VC dimension. A structure on the set of canonical hyperplanes is introduced by considering various hyperplanes having different $||\mathbf{w}||$. In other words, we analyze sets S_A such that $||\mathbf{w}|| \leqslant A$. Then, if $A_1 \leqslant A_2 \leqslant A_3 \leqslant \ldots \leqslant A_n$, we introduced a nested set $S_{A1} \subset S_{A2} \subset S_{A3} \subset \ldots \subset S_{An}$. Thus, if we impose the constraint $||\mathbf{w}|| \leqslant A$, then the canonical hyperplane cannot be closer than $1/A$ to any of the training points $\mathbf{x}_i$. *Vapnik* in (Vapnik, 1995) states that the VC dimension h of a set of canonical hyperplanes in $\Re^n$ such that $||\mathbf{w}|| \leqslant A$ is

$$H \leqslant \min[R^2 A^2, n] + 1, \quad (18)$$

where all the training data points (vectors) are enclosed by a sphere of the smallest radius R. Therefore, a small $||\mathbf{w}||$ results in a small h, and minimization of $||\mathbf{w}||$ is an implementation of the SRM principle. In other words, a minimization of the canonical hyperplane weight norm $||\mathbf{w}||$ minimizes the VC dimension according to (18). See also Fig. 4 that shows how the estimation

error, meaning the expected risk (because the empirical risk, due to the linear separability, equals zero) decreases with a decrease of a VC dimension. Finally, there is an interesting, simple and powerful result (*Vapnik*, 1995) connecting the generalization ability of learning machines and the number of support vectors. Once the support vectors have been found, we can calculate the bound on the expected probability of committing an error on a test example as follows

$$E_l[P(\text{error})] \leqslant \frac{E[\text{number of support vectors}]}{l}, \tag{19}$$

where E_l denotes expectation over all training data sets of size l. Note how easy it is to estimate this bound that is independent of the dimensionality of the input space. Therefore, an SV machine having a small number of support vectors will have good generalization ability even in a very high-dimensional space.

Example below shows the SVM's learning of the weights for a simple separable data problem in both the primal and the dual domain. The small number and low dimensionality of data pairs is used in order to show the optimization steps analytically and graphically. The same reasoning will be in the case of high dimensional and large training data sets but for them, one has to rely on computers and the insight in solution steps is necessarily lost.

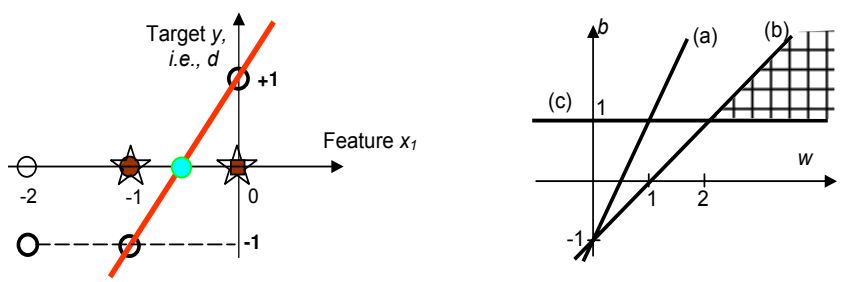

Figure 6. *Left:* Solving SVM classifier for 3 data shown. SVs are star data. *Right:* Solution space $w - b$.

Example*: Design of an SVM classifier for 3 data shown in Fig. 6 above.*
First we solve the problem in the *primal domain*: From the constraints (8b) it

follows

$$2w - 1 \geqslant b, \qquad (a)$$

$$w - 1 \geqslant b, \qquad (b)$$

$$b \geqslant 1. \qquad (c)$$

The three straight lines corresponding to the equalities above are shown in Fig. 6 right. The textured area is a feasible domain for the weight w and bias b. Note that the area is not defined by the inequality (a), thus pointing to the fact that the point -1 is not a support vector. Points -1 and 0 define the textured area and they will be the supporting data for our decision function. The task is to minimize (8a), and this will be achieved by taking the value $w = 2$. Then, from (b), it follows that $b = 1$. Note that (a) must not be used for the calculation of the bias term b. Because both the cost function (8a) and the constraints (8b) are convex, the primal and the dual solution must produce same w and b. Dual solution follows from maximizing (13) subject to (14) as follows

$$L_d = \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}[\,\alpha_1\ \alpha_2\ \alpha_3]\begin{bmatrix} 4 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix},$$

such that

$$-\alpha_1 - \alpha_2 + \alpha_3 = 0,$$

$$\alpha_1 \geqslant 0, \quad \alpha_2 \geqslant 0, \quad \alpha_3 \geqslant 0,$$

The dual Lagrangian is obtained in terms of α_1 and α_2 after expressing α_3 from the equality constraint and it is given as $L_d = 2\alpha_1 + 2\alpha_2 - 0.5(4\alpha_1^2 + 4\alpha_1\alpha_2 + \alpha_2^2)$. L_d will have maximum for $\alpha_1 = 0$, and it follows that we have to find the maximum of $L_d = 2\alpha_2 - 0.5\alpha_2^2$ which will be at $\alpha_2 = 2$. Note that the Hessian matrix is extremely bad conditioned and if the QP problem is to be solved by computer $\mathbf{H}$ should be regularized first. From the equality constraint it follows that $\alpha_3 = 2$ too. Now, we can calculate the weight vector w and the bias b from (16a) and (16b) as follows,

$$w = \sum_{i=1}^{3} \alpha_i y_i \mathbf{x}_i = 0(-1)(-2) + 2(-1)(-1) + 2(1)0 = 2\,.$$

The bias can be calculated by using SVs only, meaning from either point -1 or point 0. Both result in same value as shown below

$$b = -1 - 2(-1) = 1, \ \text{ or } b = 1 - 2(0) = 1.$$

Linear Soft Margin Classifier for Overlapping Classes

The learning procedure presented above is valid for linearly separable data, meaning for training data sets without overlapping. Such problems are rare in practice. At the same time, there are many instances when linear separating hyperplanes can be good solutions even when data are overlapped (e.g., normally distributed classes having the same covariance matrices have a linear separation boundary). However, quadratic programming solutions as given above cannot be used in the case of overlapping because the constraints $y_i[\mathbf{w}^T\mathbf{x}_i + b] \geqslant 1$, $i = 1, l$ given by (8b) cannot be satisfied. In the case of an overlapping (see Fig 7), the overlapped data points cannot be correctly classified and for any misclassified training data point $\mathbf{x}_i$, the corresponding α_i will tend to infinity. This particular data point (by increasing the corresponding α_i value) attempts to exert a stronger influence on the decision boundary in order to be classified correctly. When the α_i value reaches the maximal bound, it can no longer increase its effect, and the corresponding point will stay misclassified. In such a situation, the algorithm introduced above chooses (almost) all training data points as support vectors. To find a classifier with a maximal margin, the algorithm presented in the section "Linear Maximal Margin Classifier for Linearly Separable Data" above, must be changed allowing some data to be unclassified. Better to say, we must leave some data on the "wrong" side of a decision boundary. In practice, we allow a *soft* margin and all data inside this margin (whether on the correct side of the separating line or on the wrong one) are neglected. The width of a soft margin can be controlled by a corresponding penalty parameter C (introduced below) that determines the trade-off between the training error and VC dimension of the model.

The question now is how to measure the degree of misclassification and how to incorporate such a measure into the hard margin learning algorithm given by equations (8). The simplest method would be to form the following learning problem

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C(\text{number of misclassified data}), \tag{20}$$

where C is a penalty parameter, trading off the margin size (defined by $||\mathbf{w}||$, i.e., by $\mathbf{w}^T\mathbf{w}$) for the number of misclassified data points. Large C leads to small number of misclassifications, bigger $\mathbf{w}^T\mathbf{w}$ and consequently to the smaller margin and vice versa. Obviously taking $C = \infty$ requires that the number of misclassified data is zero and, in the case of an overlapping this is

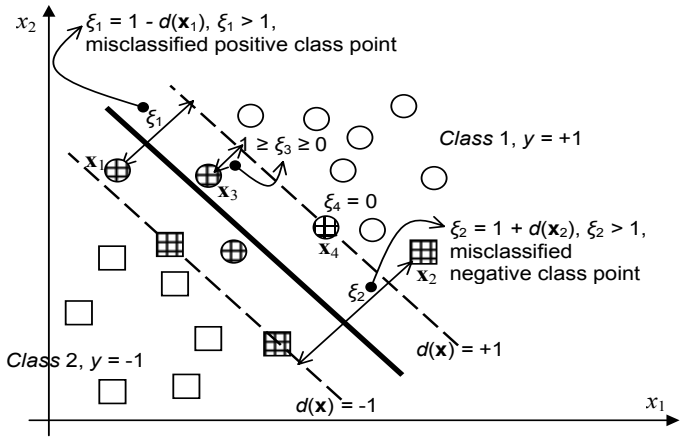

Figure 7. The soft decision boundary for a dichotomization problem with data overlapping. Separation line (*solid*), margins (*dashed*) and support vectors (textured training data points). 4 SVs in positive class (*circles*) and 3 SVs in negative class (*squares*). 2 misclassifications for positive class and 1 misclassification for negative class.

not possible. Hence, the problem may be feasible only for some value $C < \infty$.

However, the serious problem with (20) is that the error's counting can't be accommodated within the handy (meaning reliable, well understood and well developed) quadratic programming approach. Also, the counting only can't distinguish between huge (or disastrous) errors and close misses! The possible solution is to measure the distances ξ_i of the points crossing the margin from the corresponding margin and trade their sum for the margin size as given below

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C(\text{sum of distances of the wrong side points}). \qquad (21)$$

In fact this is exactly how the problem of the data overlapping was solved in (*Cortes*, 1995; *Cortes* and *Vapnik*, 1995) — by generalizing the optimal "hard" margin algorithm. They introduced the nonnegative *slack variables* $\xi_i (i = 1, l)$ in the statement of the optimization problem for the overlapped data points.

Now, instead of fulfilling (8a) and (8b), the separating hyperplane must satisfy

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i, \tag{22a}$$

subject to

$$y_i[\mathbf{w}^T\mathbf{x}_i + b] \geqslant 1 - \xi_i, \quad i = 1, l, \ \xi_i \geqslant 0, \tag{22b}$$

i.e., subject to

$$\mathbf{w}^T\mathbf{x}_i + b \geqslant +1 - \xi_i, \quad \text{for } y_i = +1, \ \xi_i \geqslant 0, \tag{22c}$$

$$\mathbf{w}^T\mathbf{x}_i + b \geqslant -1 + \xi_i, \quad \text{for } y_i = -1, \ \xi_i \geqslant 0. \tag{22d}$$

Hence, for such a *generalized* optimal separating hyperplane, the functional to be minimized comprises an extra term accounting the cost of overlapping errors. In fact the cost function (22a) can be even more general as given below

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i^k, \tag{22e}$$

subject to same constraints. This is a convex programming problem that is usually solved only for $k = 1$ or $k = 2$, and such soft margin SVMs are dubbed *L1* and *L2* SVMs respectively. By choosing exponent $k = 1$, neither slack variables ξ_i nor their Lagrange multipliers β_i appear in a dual Lagrangian L_d. Same as for a linearly separable problem presented previously, for *L1* SVMs ($k = 1$) here, the solution to a quadratic programming problem (22), is given by the saddle point of the primal Lagrangian $L_p(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ shown below

$$\begin{gathered} L_p(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i - \\ -\sum_{i=1}^{l}\alpha_i\{y_i[\mathbf{w}^T\mathbf{x}_i + b] - 1 + \xi_i\} - \sum_{i=1}^{l}\beta_i x_i, \quad \text{for } L1 \text{ SVM} \end{gathered} \tag{23}$$

where α_i and β_i are the Lagrange multipliers. Again, we should find an *optimal* saddle point $(\mathbf{w}_o, b_o, \boldsymbol{\xi}_o, \boldsymbol{\alpha}_o, \boldsymbol{\beta}_o)$ because the Lagrangian L_p has to be *minimized* with respect to $\mathbf{w}$, b and $\boldsymbol{\xi}$, and *maximized* with respect to nonnegative α_i and β_i. As before, this problem can be solved in either a *primal space* or *dual space* (which is the space of Lagrange multipliers α_i and β_i). Again, we consider a solution in a dual space as given below by using

- standard conditions for an optimum of a constrained function

$$\frac{\partial L}{\partial \mathbf{w}_o} = 0, \text{ i.e.,} \qquad \mathbf{w}_o = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \tag{24}$$

$$\frac{\partial L}{\partial b_o} = 0, \text{ i.e.,} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{25}$$

$$\frac{\partial L}{\partial \xi_{io}} = 0, \text{ i.e.,} \qquad \alpha_i + \beta_i = C, \tag{26}$$

- and the KKT complementarity conditions below,

$$\alpha_i \{ y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1 + \xi_i \} = 0, \quad i = 1, l, \tag{27a}$$

$$\beta_i \xi_i = (C - \alpha_i) \xi_i = 0, \quad i = 1, l. \tag{27b}$$

At the optimal solution, due to the KKT conditions (27), the last two terms in the primal Lagrangian L_p given by (23) vanish and the *dual variables Lagrangian* $L_d(\boldsymbol{\alpha})$, for *L1* SVM, is not a function of β_i. In fact, it is same as the hard margin classifier's L_d given before and repeated here for the soft margin one,

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j. \tag{28}$$

In order to find the optimal hyperplane, a dual Lagrangian $L_d(\boldsymbol{\alpha})$ has to be *maximized* with respect to nonnegative and (unlike before) smaller than or equal to C, α_i. In other words with

$$C \geqslant \alpha_i \geqslant 0, \quad i = 1, l, \tag{29a}$$

and under the constraint (25), i.e., under

$$\sum_{i=1}^{l} \alpha_i y_i = 0. \tag{29b}$$

Thus, the final quadratic optimization problem is practically same as for the separable case the only difference being in the modified bounds of the Lagrange multipliers α_i. The penalty parameter C, which is now the upper bound on α_i,

is determined by the user. The selection of a "good" or "proper" C is always done experimentally by using some cross-validation technique. Note that in the previous linearly separable case, without data overlapping, this upper bound $C = \infty$. We can also readily change to the matrix notation of the problem above as in equations (15). Most important of all is that the learning problem is expressed only in terms of unknown Lagrange multipliers α_i, and known inputs and outputs. Furthermore, optimization does not solely depend upon inputs $\mathbf{x}_i$ which can be of a very high (inclusive of an infinite) dimension, but it depends upon a scalar product of input vectors $\mathbf{x}_i$. It is this property we will use in the next section where we design SV machines that can create nonlinear separation boundaries. Finally, expressions for both a *decision function* $d(\mathbf{x})$ and an *indicator function* $i_F = \mathrm{sign}(d(\mathbf{x}))$ for a soft margin classifier are same as for linearly separable classes and are also given by (17).

From (27) follows that there are only three possible solutions for α_i (see Fig. 7)

1. $\alpha_i = 0, \;\; \xi_i = 0 \rightarrow$ data point $\mathbf{x}_i$ is correctly classified.
2. $C > \alpha_i > 0 \rightarrow$ then, the two complementarity conditions must result result in $y_i[\mathbf{w}^T\mathbf{x}_i + b] - 1 + \xi_i = 0$, and $\xi_i = 0$. Thus, $y_i[\mathbf{w}^T\mathbf{x}_i + b] = 1$ and $\mathbf{x}_i$ is a support vector. The support vectors with $C \geqslant \alpha_i \geqslant 0$ are called *unbounded* or *free support vectors*. They lie on the two margins.
3. $\alpha_i = C \rightarrow$ then, $y_i[\mathbf{w}^T\mathbf{x}_i + b] - 1 + \xi_i = 0$, and $\xi_i \geqslant 0$, and $\mathbf{x}_i$ is a support vector. The support vectors with $\alpha_i = C$ are called *bounded support vectors*. They lie on the "wrong" side of the margin. For $1 > \xi_i \geqslant 0$, $\mathbf{x}_i$ is still correctly classified, and if $\xi_i \geqslant 1$, $\mathbf{x}_i$ is misclassified.

For <u>*L2* SVM</u> the second term in the cost function (22e) is quadratic, i.e., $C\sum_{i=1}^{l}\xi_i^2$, and this leads to changes in a dual optimization problem which is now,

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \Big(\mathbf{x}_i^T\mathbf{x}_j + \frac{\delta_{ij}}{C}\Big), \tag{30}$$

subject to

$$\alpha_i \geqslant 0, \quad i = 1, l, \tag{31a}$$

$$\sum_{i=1}^{l}\alpha_i y_i = 0. \tag{31b}$$

where, $\delta_{ij} = 1$ for $i = j$, and it is zero otherwise. Note the change in Hessian matrix elements given by second terms in (30), as well as that there is no upper

bound on α_i. The detailed analysis and comparisons of the $L1$ and $L2$ SVMs is presented in (*Abe*, 2004). We use the most popular $L1$ SVMs here, because they usually produce more sparse solutions, i.e., they create a decision function by using less SVs than the $L2$ SVMs.

The Nonlinear Classifier

The linear classifiers presented in two previous sections are very limited. Mostly, classes are not only overlapped but the genuine separation functions are nonlinear hypersurfaces. A nice and strong characteristic of the approach presented above is that it can be easily (and in a relatively straightforward manner) extended to create nonlinear decision boundaries. The motivation for such an extension is that an SV machine that can create a nonlinear decision hypersurface will be able to classify nonlinearly separable data. This will be achieved by considering a linear classifier in the so-called *feature space* that will be introduced shortly. A very simple example of a need for designing nonlinear models is given in Fig. 8 where the true separation boundary is quadratic. It is obvious that no errorless linear separating hyperplane can be found now. The best linear separation function shown as a dashed straight line would make six misclassifications (textured data points; 4 in the negative class and 2 in the positive one). Yet, if we use the nonlinear separation boundary we are able to separate two classes without any error. Generally, for n-dimensional input patterns, instead of a nonlinear curve, an SV machine will create a nonlinear separating hypersurface.

The basic idea in designing nonlinear SV machines is to map input vectors $\mathbf{x} \in \Re^n$ into vectors $\Phi(\mathbf{x})$ of a higher dimensional *feature space* F (where Φ represents mapping: $\Re^n \to \Re^f$), and to solve a linear classification problem in this feature space

$$\mathbf{x} \in \Re^n \to \Phi(\mathbf{x}) = [\phi_1(\mathbf{x})\ \phi_2(\mathbf{x})\ \ldots\ \phi_n(\mathbf{x})]^T \in \Re^f. \tag{32}$$

A mapping $\Phi(\mathbf{x})$ is chosen in advance, i.e., it is a fixed function. Note that an input space ($\mathbf{x}$-space) is spanned by components x_i of an input vector $\mathbf{x}$ and a feature space F (Φ-space) is spanned by components $\phi_i(\mathbf{x})$ of a vector $\Phi(\mathbf{x})$. By performing such a mapping, we hope that in a Φ-space, our learning algorithm will be able to linearly separate images of $\mathbf{x}$ by applying the linear SVM formulation presented above. (In fact, it can be shown that for a whole class of mappings the linear separation in a feature space is always possible. Such

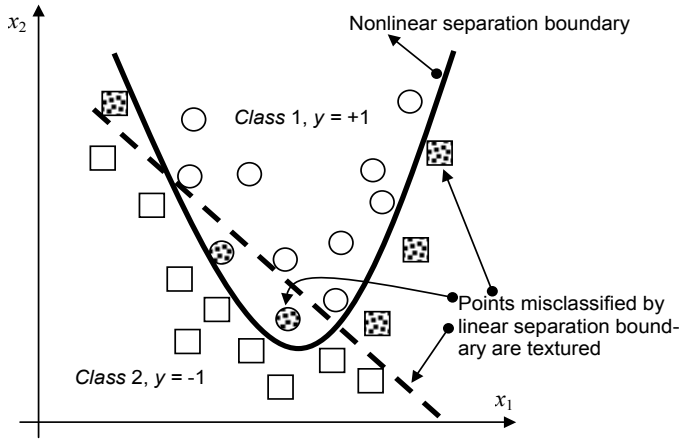

Figure 8. A nonlinear SVM without data overlapping. A true separation is a quadratic curve. The nonlinear separation line (*solid*), the linear one (*dashed*) and data points misclassified by the linear separation line (*the textured training data points*) are shown. There are 4 misclassified negative data and 2 misclassified positive ones. SVs are not shown.

mappings will correspond to the positive definite kernels that will be shown shortly). We also expect this approach to again lead to solving a quadratic optimization problem with similar constraints in a Φ-space. The solution for an indicator function

$$i_F(\mathbf{x}) = \operatorname{sign}(\mathbf{w}^T\Phi(\mathbf{x}) + b) = \operatorname{sign}\Big(\sum_{i=1}^{l} y_i\alpha_i\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) + b\Big),$$

which is a linear classifier in a feature space, will create a nonlinear separating hypersurface in the original input space given by (33) below. (Compare this solution with (17) and note the appearances of scalar products in both the original X-space and in the feature space F).

The equation for an $i_F(\mathbf{x})$ just given above can be rewritten in a "neural

networks" form as follows

$$i_F(\mathbf{x}) = \operatorname{sign}\Big(\sum_{i=1}^{l} y_i\alpha_i\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) + b\Big) = \\ = \operatorname{sign}\Big(\sum_{i=1}^{l} y_i\alpha_i k(\mathbf{x}_i,\mathbf{x}) + b\Big) = \operatorname{sign}\Big(\sum_{i=1}^{l} \nu_i k(\mathbf{x}_i,\mathbf{x}) + b\Big) \tag{33}$$

where v_i corresponds to the output layer weights of the "SVM's network" and $k(\mathbf{x}_i, \mathbf{x})$ denotes the value of the kernel function that will be introduced shortly[6]. Note the difference between the weight vector $\mathbf{w}$ which norm should be minimized and which is the vector of the same dimension as the feature space vector $\Phi(\mathbf{x})$ and the weightings $v_i = \alpha_i y_i$ that are scalar values composing the weight vector $\mathbf{v}$ which dimension equals the number of training data points l. The $(l - N_{\mathrm{SVs}})$ of v_i components are equal to zero, and only N_{SVs} entries of $\mathbf{v}$ are nonzero elements.

A simple example below (Fig 9) should exemplify the idea of a nonlinear mapping to (usually) higher dimensional space and how it happens that the data become linearly separable in the F-space.

Consider solving the simplest 1-D classification problem given the input and the output (desired) values as follows: $\mathbf{x} = [-1\ 0\ 1]^T$ and $\mathbf{d} = \mathbf{y} = [-1\ 1\ -1]^T$. Here we choose the following mapping to the feature space: $\Phi(\mathbf{x}) = [\varphi_1(\mathbf{x})\ \ \varphi_2(\mathbf{x})\ \ \varphi_3(\mathbf{x})]^T = [x^2\ \ \sqrt{2}x\ \ 1]^T)$

The mapping produces the following three points in the feature space (shown as the *rows* of the matrix $\mathbf{F}$ (F standing for features))

$$\mathbf{F} = \begin{bmatrix} 1 & -\sqrt{2} & 1 \\ 0 & 0 & 1 \\ 1 & \sqrt{2} & 1 \end{bmatrix}^T .$$

These three points are linearly separable by the plane $\varphi_3(\mathbf{x}) = 2\varphi_1(\mathbf{x})$ in a feature space as shown in Fig. 10. It is easy to show that the mapping obtained by $\Phi(\mathbf{x}) = [x^2\ \ \sqrt{2}x\ \ 1]^T$ is a scalar product implementation of a quadratic kernel function $(\mathbf{x}_i^T\mathbf{x}_j + 1)^2 = k(\mathbf{x}_i, \mathbf{x}_j)$. In other words, $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$. This equality will be introduced shortly.

There are two basic problems when mapping an input $\mathbf{x}$-space into higher order F-space:

[6] v_i equals $y_i\alpha_i$ in the classification case presented above and it is equal to $(\alpha_i - \alpha_i^*)$ in the regression problems.

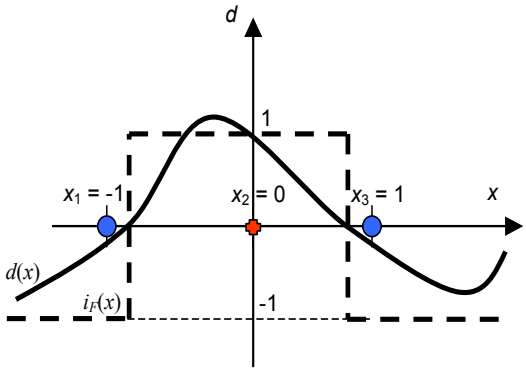

Figure 9. A nonlinear 1-dimensional classification problem. One possible solution is given by the decision function $d(x)$ (*solid curve*) i.e., by the corresponding indicator function defined as $i_F = \mathrm{sign}(d(x))$ (*dashed stepwise function*).

1) the choice of mapping $\Phi(\mathbf{x})$ that should result in a "rich" class of decision hypersurfaces,
2) the calculation of the scalar product $\Phi^T(\mathbf{x})\Phi(\mathbf{x})$ that can be computationally very discouraging if the number of features f (i.e., dimensionality f of a feature space) is very large.

The second problem is connected with a phenomenon called the *"curse of dimensionality"*. For example, to construct a decision surface corresponding to a polynomial of degree *two* in an n-D input space, a dimensionality of a feature space $f = n(n+3)/2$. In other words, a feature space is spanned by f coordinates of the form

$$\begin{aligned} z_1 = x_1, \ldots, z_n = x_n \quad & (n \text{ coordinates}), \\ z_{n+1} = (x_1)^2, \ldots, z_{2n} = (x_n)^2 \quad & (\text{next } n \text{ coordinates}), \\ z_{2n+1} = x_1 x_2, \ldots, z_f = x_n x_{n-1} \quad & (n(n-1)/2 \text{ coordinates}), \end{aligned}$$

and the separating hyperplane created in this space, is a second-degree polynomial in the input space (*Vapnik*, 1998). Thus, constructing a polynomial of degree two only, in a 256-dimensional input space, leads to a dimensionality of

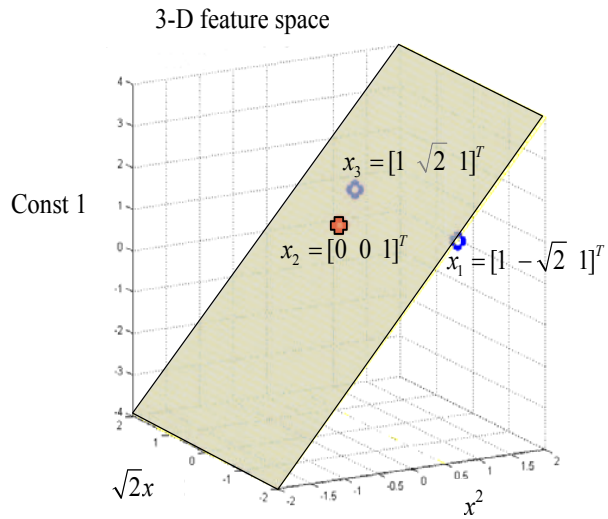

Figure 10. The three data points of a problem in Fig. 9 are linearly separable in the feature space (obtained by the mapping $\Phi(\mathbf{x}) = [\varphi_1(\mathbf{x}) \quad \varphi_2(\mathbf{x}) \quad \varphi_3(\mathbf{x})]^T = [x^2 \quad \sqrt{2}x \quad 1]^T$). The separation boundary is given as the plane $\varphi_3(\mathbf{x}) = 2\varphi_1(\mathbf{x})$ shown in the figure.

a feature space $f = 33,152$. Performing a scalar product operation with vectors of such, or higher, dimensions, is not a cheap computational task. The problems become serious (and fortunately only seemingly unsolvable) if we want to construct a polynomial of degree 4 or 5 in the same 256-dimensional space leading to the construction of a decision hyperplane in a billion-dimensional feature space.

This explosion in dimensionality can be avoided by noticing that in the quadratic optimization problem given by (13) and (28), as well as in the final expression for a classifier, *training data only appear in the form of scalar products* $\mathbf{x}_i^T\mathbf{x}_j$. These products will be replaced by scalar products

$$\Phi^T(\mathbf{x})\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_n(\mathbf{x})]^T[\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_n(\mathbf{x})]$$

in a feature space F, and the latter can be and will be expressed by using the *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$.

Note that a *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$ is a function in input space. Thus, the basic advantage in using kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is in avoiding performing a mapping $\Phi(\mathbf{x})$ at all. Instead, the required scalar products in a feature space $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$, are calculated directly by computing kernels $K(\mathbf{x}_i, \mathbf{x}_j)$ for given training data vectors in an input space. In this way, we bypass a possibly extremely high dimensionality of a feature space F. Thus, by using the chosen kernel $K(\mathbf{x}_i, \mathbf{x}_j)$, we can construct an SVM that operates in an infinite dimensional space (such a kernel function is a Gaussian kernel function given in table 2 below). In addition, as will be shown below, by applying kernels we do not even have to know what the actual mapping $\Phi(\mathbf{x})$ is. A kernel is a function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j). \tag{34}$$

There are many possible kernels, and the most popular ones are given in Table 2. All of them should fulfill the so-called Mercer's conditions. The Mercer's kernels belong to a set of *reproducing kernels.* For further details see (*Mercer*, 1909; *Aizerman* et al, 1964; *Smola* and *Schölkopf*, 1997; *Vapnik*, 1998; *Kecman*, 2001).

The simplest is a linear kernel defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$. Below we show a few more kernels.

Table 2. Popular Admissible Kernels

Kernel functions	**Type of classifier**
$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T\mathbf{x}_i)$	Linear, dot product, kernel, CPD
$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}^T\mathbf{x}_i) + 1]^d$	Complete polynomial of degree *d,* PD
$K(\mathbf{x}, \mathbf{x}_i) = e^{-\frac{1}{2}[(\mathbf{x}-\mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x}-\mathbf{x}_i)]}$	Gaussian RBF, PD
$K(\mathbf{x}, \mathbf{x}_i) = \tanh[(\mathbf{x}^T\mathbf{x}_i) + b]$*	Multilayer perceptron, CPD
$K(\mathbf{x}, \mathbf{x}_i) = \dfrac{1}{\sqrt{\lVert \mathbf{x} - \mathbf{x}_i \rVert^2 + \beta}}$	Inverse multiquadric function, PD

*only for certain values of *b,* (C)PD = (conditionally) positive definite

POLYNOMIAL KERNELS

Let $\mathbf{x} \in \Re^2$ i.e., $\mathbf{x} = [x_1 \; x_2]^T$, and if we choose $\Phi(\mathbf{x}) = [x_1^2 \;\; \sqrt{2}x_1x_2 \;\; x_2^2]^T$ (i.e., there is an $\Re^2 \to \Re^3$ mapping), then the dot product

$$\begin{aligned}\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j) &= [x_{i1}^2 \;\; \sqrt{2}\,x_{i1}x_{i2} \;\; x_{i2}^2][x_{j1}^2 \;\; \sqrt{2}\,x_{j1}x_{j2} \;\; x_{j2}^2]^T = \\ &= [x_{i1}^2x_{j1}^2 \;\; 2\,x_{i1}x_2x_{j1}x_{j2} \;\; x_{i2}^2x_{j2}^2] = \\ &= (\mathbf{x}_i^T\mathbf{x}_j)^2 = K(\mathbf{x}_i, \mathbf{x}_j)\end{aligned}$$

or

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^2 = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j).$$

Note that in order to calculate the scalar product in a feature space $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$, we do not need to perform the mapping $\Phi(\mathbf{x}) = [x_1^2 \;\; \sqrt{2}\,x_1x_2 \;\; x_2^2\,]^T$ at all. Instead, we calculate this product directly in the input space by computing $(\mathbf{x}_i^T\mathbf{x}_j)^2$. This is very well known under the popular name of *the kernel trick.* Interestingly, note also that other mappings such as an

$\Re^2 \to \Re^3$ mapping given by $\Phi(\mathbf{x}) = [x_1^2 - x_2^2 \;\; 2x_1x_2 \;\; x_1^2 + x_2^2]$, or an
$\Re^2 \to \Re^4$ mapping given by $\Phi(\mathbf{x}) = [x_1^2 \;\; x_1x_2 \;\; x_1x_2 \;\; x_2^2]$,

also accomplish the same task as $(\mathbf{x}_i^T\mathbf{x}_j)^2$.

Now, assume the following mapping

$$\Phi(\mathbf{x}) = [1 \quad \sqrt{2}\,x_1 \quad \sqrt{2}\,x_2 \quad \sqrt{2}\,x_1x_2 \quad x_1^2 \quad x_1^2],$$

i.e., there is an $\Re^2 \to \Re^3$ mapping plus bias term as the constant 6th dimension's value. Then the dot product in a feature space F is given as

$$\begin{aligned}\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j) &= 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{i2}x_{j1}x_{j2} + x_{i1}^2x_{j1}^2 + x_{i2}^2x_{j2}^2 = \\ &= 1 + 2(\mathbf{x}_i^T\mathbf{x}_j) + (\mathbf{x}_i^T\mathbf{x}_j)^2 = \\ &= (\mathbf{x}_i^T\mathbf{x}_j + 1)^2 = K(\mathbf{x}_i, \mathbf{x}_j)\end{aligned}$$

or

$$(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + 1)^2 = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j).$$

Thus, the last mapping leads to the second order *complete* polynomial.

Many candidate functions can be applied to a convolution of an inner product (i.e., for kernel functions) $K(\mathbf{x}, \mathbf{x}_i)$ in an SV machine. Each of these functions constructs a different nonlinear decision hypersurface in an input space. In the first three rows, the table 2 shows the three most popular kernels in SVMs'

in use today, and the inverse multiquadrics one as an interesting and powerful kernel to be proven yet.

The positive definite (PD) kernels are the kernels which Gramm matrix $\mathbf{G}$ (a.k.a. Grammian, or a design matrix) calculated by using all the l training data points is positive definite (meaning all its eigenvalues are strictly positive, i.e., $\lambda_i > 0,\ i = 1, l$)

$$\mathbf{G} = \mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_l) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_l) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathbf{x}_l, \mathbf{x}_1) & k(\mathbf{x}_l, \mathbf{x}_2) & \cdots & k(\mathbf{x}_l, \mathbf{x}_l) \end{bmatrix} \tag{35}$$

The kernel matrix $\mathbf{G}$ is a symmetric one. Even more, any symmetric positive definite matrix can be regarded as a kernel matrix, that is — as an inner product matrix in some space.

Finally, we arrive at the point of presenting the learning in nonlinear classifiers (in which we are ultimately interested here). The learning algorithm for a nonlinear SV machine (classifier) follows from the design of an *optimal separating hyperplane* in a *feature space*. This is the same procedure as the construction of a "hard" (13) and "soft" (28) margin classifiers in an $\mathbf{x}$-space previously. In a $\Phi(\mathbf{x})$-space, the dual Lagrangian, given previously by (13) and (28), is now

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \, \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_j, \tag{36}$$

and, according to (34), by using chosen kernels, we should maximize the following dual Lagrangian

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \, K(\mathbf{x}, \mathbf{x}_i), \tag{37}$$

subject to

$$\alpha_i \geqslant 0, \quad i = 1, l, \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{37a}$$

In a more general case, because of a noise or due to generic class' features, there will be an overlapping of training data points. Nothing but constraints for

α_i change. Thus, the nonlinear "soft" margin classifier will be the solution of the quadratic optimization problem given by (37) subject to constraints

$$C \geqslant \alpha_i \geqslant 0, \quad i = 1, l \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{37b}$$

Again, the only difference to the separable nonlinear classifier is the upper bound C on the Lagrange multipliers α_i. In this way, we limit the influence of training data points that will remain on the "wrong" side of a separating nonlinear hypersurface. After the dual variables are calculated, the decision hypersurface $d(\mathbf{x})$ is determined by

$$d(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b = \sum_{i=1}^{l} \nu_i K(\mathbf{x}, \mathbf{x}_i) + b, \tag{38}$$

and the indicator function is

$$i_F(\mathbf{x}) = \text{sign}[d(\mathbf{x})] = \text{sign}\Big[\sum_{i=1}^{l} \nu_i K(\mathbf{x}, \mathbf{x}_i) + b\Big].$$

Note that the summation is not actually performed over all training data but rather over the support vectors, because only for them do the Lagrange multipliers differ from zero. The existence and calculation of a bias b is now not a direct procedure as it is for a linear hyperplane. Depending upon the applied kernel, the bias b can be implicitly part of the kernel function. If, for example, Gaussian RBF is chosen as a kernel, it can use a bias term as the $(f+1)^{st}$ feature in F-space with a constant output $= +1$, but not necessarily. In short, all PD kernels do not necessarily need an explicit bias term b, but b can be used. In section "On the Equality of Kernel AdaTron and Sequential Minimal Optimization and Alike Algorithms for Kernel Machines" we will develop new iterative learning algorithm for models having a bias term b[7]. Same as for the linear SVM, (37) can be written in a matrix notation as

maximize

$$L_d(\boldsymbol{\alpha}) = -0.5\boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} + \mathbf{f}^T \boldsymbol{\alpha}, \tag{39a}$$

subject to

$$\mathbf{y}^T \boldsymbol{\alpha} = 0,$$

[7]More on this can be found in (*Kecman, Huang*, and *Vogt*, 2005) as well as in the (*Vogt* and *Kecman*, 2005).

and

$$C \geqslant \alpha_i \geqslant 0, \quad i = 1, l. \tag{39c}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T$, denotes the Hessian matrix

$$H_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

of this problem and $\mathbf{f}$ is an $(l, 1)$ unit vector $\mathbf{f} = \mathbf{1} = [1 \;\; 1 \;\; \ldots \;\; 1]^T$. Note that if $K(\mathbf{x}_i, \mathbf{x}_j)$ is the positive definite matrix, then so is the matrix $y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ too.

The following 1-D example (just for the sake of graphical presentation) will show the creation of a linear decision function in a feature space and a corresponding nonlinear (quadratic) decision function in an input space.

Suppose we have 4 1-D data points given as $x_1 = 1$, $x_2 = 2$, $x_3 = 5$, $x_4 = 6$, with data at 1, 2, and 6 as class 1 and the data point at 5 as class 2, i.e., $y_1 = -1$, $y_2 = -1$, $y_3 = 1$, $y_4 = -1$. We use the polynomial kernel of degree 2, $K(x, y) = (xy + 1)^2$. C is set to 50, which is of lesser importance because the constraints will be not imposed in this example for maximal value for the dual variables alpha will be smaller than $C = 50$.

Case 1: Working with a bias term b as given in (38).

We first find $\alpha_i (i = 1, \ldots, 4)$ by solving dual problem (39) having a Hessian matrix

$$\mathbf{H} = \begin{bmatrix} 4 & 9 & -36 & 49 \\ 9 & 25 & -121 & 169 \\ -36 & -121 & 676 & -961 \\ 49 & 169 & -961 & 1369 \end{bmatrix}.$$

Alphas are $\alpha_1 = 0$, $\alpha_2 = 2.499999$, $\alpha_3 = 7.333333$, $\alpha_4 = 4.833333$ and the bias b will be found by using (16b), or by fulfilling the requirements that the values of a decision function at the support vectors should be the given y_i. The model (decision function) is given by

$$d(x) = \sum_{i=1}^{4} y_i \alpha_i K(x, x_i) + b = \sum_{i=1}^{4} \nu_i (x x_i + 1)^2 + b,$$

or by

$$d(x) = 2.499999(-1)(2x+1)^2 + 7.333333(1)(5x+1)^2 + 4.833333(-1)(6x+1)^2 + b,$$

$$d(x) = -0.666667x^2 + 5.333333x + b.$$

Bias b is determined from the requirement that at the SV points 2, 5 and 6, the outputs must be -1, 1 and -1 respectively. Hence, $b = -9$, resulting in the decision function

$$d(x) = -0.666667x^2 + 5.333333x + b.$$

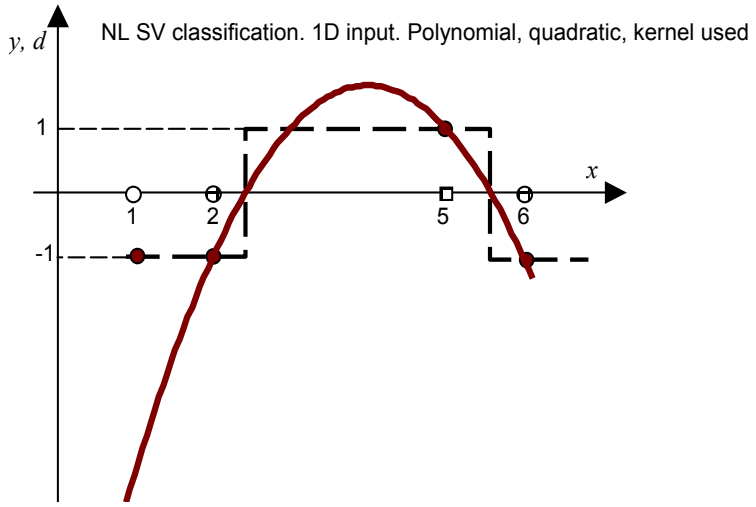

Figure 11. The nonlinear decision function (*solid*) and the indicator function (*dashed*) for 1-D overlapping data. By using a complete second order polynomial the model with and without a bias term b are same.

The nonlinear (quadratic) decision function and the indicator one are shown in Fig. 11. Note that in calculations above 6 decimal places have been used for alpha values. The calculation is numerically very sensitive, and working with fewer decimals can give very approximate or wrong results.

The complete polynomial kernel as used in the case 1, is *positive definite* and there is no need to use an explicit bias term b as presented above. Thus, one can use the same second order polynomial model without the bias term b. Note that in this particular case there is no equality constraint equation that originates from an equalization of the primal Lagrangian derivative in respect

to the bias term b to zero. Hence, we do not use (39b) while using a positive definite kernel without bias as it will be shown below in the case 2.

Case 2: Working without a bias term b.

Because we use the same second order polynomial kernel, the Hessian matrix $\mathbf{H}$ is same as in the case 1. The solution without the equality constraint for alphas is: $\alpha_1 = 0$, $\alpha_2 = 24.999999$, $\alpha_3 = 43.333333$, $\alpha_4 = 27.333333$. The model (decision function) is given by

$$d(x) = \sum_{i=1}^{4} y_i \alpha_i K(x, x_i) = \sum_{i=1}^{4} \nu_i (x x_i + 1)^2,$$

or by

$$d(x) = 2.499999(-1)(2x+1)^2 + 7.333333(1)(5x+1)^2 + 4.833333(-1)(6x+1)^2,$$

$$d(x) = -0.666667x^2 + 5.333333x + b.$$

Thus the nonlinear (quadratic) decision function and consequently the indicator function in the two particular cases are equal.

Example: *Nonlinear classifier for an Exclusive-Or (XOR) problem*

In the *next example* shown by Figs 12 and 13 we present all the important mathematical objects of a nonlinear SV classifier by using a classic XOR (*exclusive-or*) problem. The graphs show all the mathematical functions (objects) involved in a nonlinear classification. Namely, the nonlinear decision function $d(\mathbf{x})$, the NL indicator function $i_F(\mathbf{x})$, training data $(\mathbf{x}_i)$, support vectors $(\mathbf{x}_{SV})_i$ and separation boundaries.

The same objects will be created in the cases when the input vector $\mathbf{x}$ is of a dimensionality $n > 2$, but the visualization in these cases is not possible. In such cases one talks about the decision hyperfunction (hypersurface) $d(\mathbf{x})$, indicator hyperfunction (hypersurface) $i_F(\mathbf{x})$, training data $(\mathbf{x}_i)$, support vectors $(\mathbf{x}_{SV})_i$ and separation hyperboundaries (hypersurfaces).

Note the different character of a $d(\mathbf{x})$, $i_F(\mathbf{x})$ and separation boundaries in the two graphs given below. However, in both graphs all the data are correctly classified. Fig. 12 shows the resulting functions for the Gaussian kernel functions, while Fig. 13 presents the solution for a complete second order polynomial kernel. Below, we present the analytical derivation of the (saddle like) decision function in the later (polynomial kernel) case.

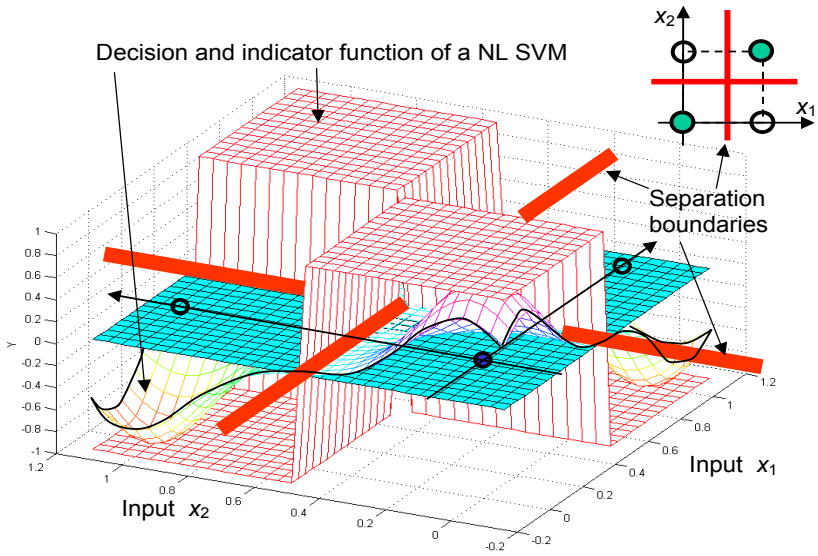

Figure 12. XOR problem. Kernel functions are the *2-D Gaussians* and they are not shown here. The nonlinear decision function, the nonlinear indicator function and the separation boundaries are shown. All four data are chosen as support vectors.

The analytic solution to the Fig. 13 for the *second order polynomial kernel* (i.e., for $(\mathbf{x}_i^T\mathbf{x}_j+1)^2 = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$, where

$$\Phi(\mathbf{x}) = [1\sqrt{2}\,x_1 \quad \sqrt{2}\,x_2 \quad \sqrt{2}\,x_1x_2 \quad x_1^2 \quad x_2^2],$$

no explicit bias and $C=\infty$) goes as follows. Inputs and desired outputs are,

$$\mathbf{x} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}^T, \qquad \mathbf{y} = \mathbf{d} = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}^T.$$

The dual Lagrangian (37) has the Hessian matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 9 & -4 & -4 \\ -1 & -4 & 4 & 1 \\ -1 & -4 & 1 & 4 \end{bmatrix}.$$

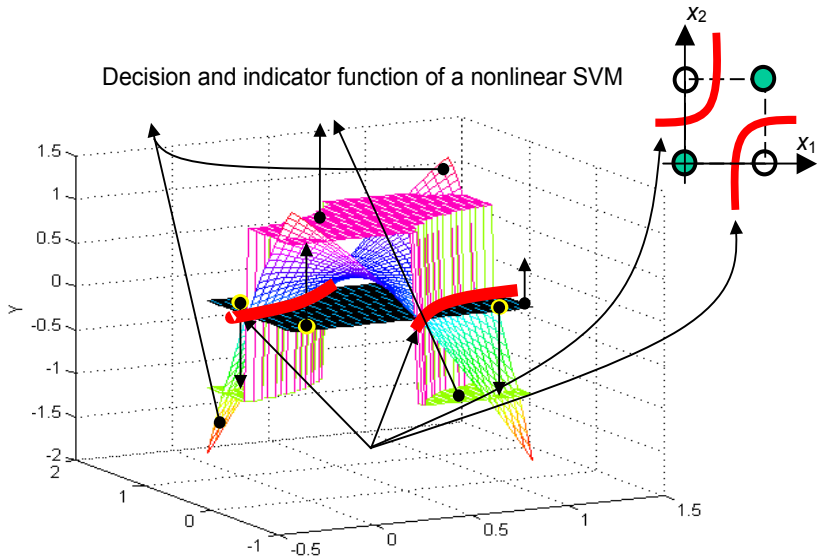

Figure 13. XOR problem. Kernel function is a *2-D polinomial*. The nonlinear decision function, the nonlinear indicator function and the separation boundaries are shown. All four data are support vectors.

The optimal solution can be obtained by taking the derivative of L_d with respect to dual variables $\alpha_i (i = 1, 4)$ and by solving the resulting linear system of equations taking into account the constraints. The solution to

$$\begin{array}{rcrcrcrcl} \alpha_1 & + & \alpha_2 & - & \alpha_3 & - & \alpha_4 & = & 1, \\ \alpha_1 & + & 9\alpha_2 & - & 4\alpha_3 & - & 4\alpha_4 & = & 1, \\ -\alpha_1 & - & 4\alpha_2 & + & 4\alpha_3 & + & \alpha_4 & = & 1, \\ -\alpha_1 & - & 4\alpha_2 & + & \alpha_3 & + & 4\alpha_4 & = & 1, \end{array}$$

subject to $\alpha_i > 0$, $(i = 1, 4)$, is $\alpha_1 = 4.3333$, $\alpha_2 = 2.0000$, $\alpha_3 = 2.6667$ and

$\alpha_4 = 2.6667$. The decision function in a 3-D space is

$$d(\mathbf{x}) = \sum_{i=1}^{4} y_i \alpha_i \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) =$$
$$= \big(4.3333 \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} + 2 \begin{bmatrix}1 & \sqrt{2} & \sqrt{2} & \sqrt{2} & 1 & 1\end{bmatrix} -$$
$$-2.6667 \begin{bmatrix}1 & \sqrt{2} & 0 & 0 & 1 & 0\end{bmatrix} - 2.6667 \begin{bmatrix}1 & 0 & \sqrt{2} & 0 & 0 & 1\end{bmatrix}\big)\Phi(\mathbf{x}) =$$
$$= \begin{bmatrix}1 & -0.9429 & -0.9429 & 2.8284 & -0.6667 & -0.6667\end{bmatrix} \times$$
$$\times \begin{bmatrix}1 & \sqrt{2}\,x_1 & \sqrt{2}\,x_2 & \sqrt{2}\,x_1x_2 & x_1^2 & x_2^2\end{bmatrix}^T,$$

and finally

$$d(\mathbf{x}) = 1 - 1.3335x_1 - 1.3335x_2 + 4x_1x_2 - 0.6667x_1^2 - 0.6667x_2^2.$$

It is easy to check that the values of $d(\mathbf{x})$ for all the training inputs in $\mathbf{x}$ equal the desired values in $\mathbf{d}$. The $d(\mathbf{x})$ is the saddle-like function shown in Fig. 13.

Here we have shown the derivation of an expression for $d(\mathbf{x})$ by using explicitly a mapping Φ. Again, we do not have to know what mapping Φ is at all. By using *kernels in input space*, we calculate a *scalar product* required in a *(possibly high dimensional) feature space* and we avoid mapping $\Phi(\mathbf{x})$. This is known as kernel "trick". It can also be useful to remember that the way in which the kernel "trick" was applied in designing an SVM can be utilized in all other algorithms that depend on the scalar product (e.g., in principal component analysis or in the nearest neighbor procedure).

Regression by Support Vector Machines

In the regression, we estimate the functional dependence of the dependent (output) variable $y \in \Re$ on an n-dimensional input variable $\mathbf{x}$. Thus, unlike in pattern recognition problems (where the desired outputs y_i are discrete values e.g., Boolean) we deal with *real valued* functions and we model an $\Re^n$ to $\Re^1$ mapping here. Same as in the case of classification, this will be achieved by training the SVM model on a training data set first. Interestingly and importantly, a learning stage will end in the same shape of a dual Lagrangian as in classification, only difference being in a dimensionalities of the Hessian matrix and corresponding vectors which are of a double size now e.g., $\mathbf{H}$ is a $(2l, 2l)$ matrix.

Initially developed for solving classification problems, SV techniques can be successfully applied in regression, i.e., for a functional approximation problems (*Drucker* et al, (1997), *Vapnik* et al, (1997)). The general regression learning problem is set as follows — the learning machine is given l training data from which it attempts to learn the input-output relationship (dependency, mapping or function) $f(\mathbf{x})$. A training data set

$$D = \{[\mathbf{x}(i), y(i)] \in \Re^n \times \Re, \quad i = 1, \ldots, l\}$$

consists of l pairs $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$, where the inputs $\mathbf{x}$ are n-dimensional vectors $\mathbf{x} \in \Re^n$ and system responses $y \in \Re$, are continuous values.

We introduce all the relevant and necessary concepts of SVMs' regression in a gentle way starting again with a *linear regression hyperplane* $f(\mathbf{x}, \mathbf{w})$ given as

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b. \tag{40}$$

In the case of SVM's regression, we measure the *error of approximation* instead of the margin used in classification. The most important difference in respect to classic regression is that we use a novel loss (error) functions here. This is the Vapnik's *linear loss function* with ε-insensitivity zone defined as

$$E(\mathbf{x}, y, f) = |y - f(\mathbf{x}, \mathbf{w})|_\varepsilon = \begin{cases} 0, & \text{if } |y - f(\mathbf{x}, \mathbf{w})| \leqslant \varepsilon, \\ |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon, & \text{otherwise}, \end{cases} \tag{41a}$$

or as,

$$e(\mathbf{x}, y, f) = \max(0, |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon). \tag{41b}$$

Thus, the loss is equal to 0 if the difference between the predicted $f(\mathbf{x}_i, \mathbf{w})$ and the measured value y_i is less than ε. Vapnik's ε-insensitivity loss function (41) defines an ε tube (Fig. 15). If the predicted value is within the tube the loss (error or cost) is zero. For all other predicted points outside the tube, the loss equals the magnitude of the difference between the predicted value and the radius ε of the tube.

The two classic error functions are: a square error, i.e., L_2 norm $(y - f)^2$, as well as an absolute error, i.e., L_1 norm, least modulus $|y - f|$ introduced by Yugoslav scientist *Rudjer Boskovic* in 1755 (*Eisenhart*, 1962). The latter error function is related to Huber's error function. An application of Huber's

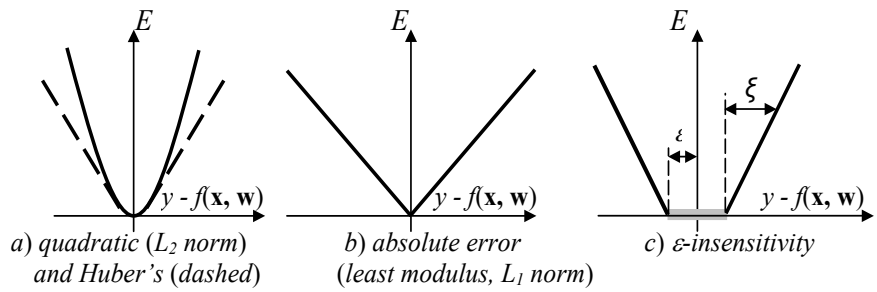

Figure 14. Loss (error) functions

error function results in a *robust regression*. It is the most reliable technique if nothing specific is known about the model of a noise. We do no present Huber's loss function here in analytic form. Instead, we show it by a dashed curve in Fig. 14a. In addition, Fig. 14 shows typical shapes of all mentioned error (loss) functions above.

Note that for $\varepsilon = 0$, Vapnik's loss function equals a least modulus function. Typical graph of a (nonlinear) regression problem as well as all relevant mathematical variables and objects required in, or resulted from, a learning unknown coefficients w_i are shown in Fig. 15.

We will formulate an SVM regression's algorithm for the linear case first and then, for the sake of a NL model design, we will apply mapping to a feature space, utilize the kernel "trick" and construct a nonlinear regression hypersurface. This is actually the same order of presentation as in classification tasks. Here, for the regression, we 'measure' the empirical error term R_{emp} by Vapnik's ε-insensitivity loss function given by (41) and shown in Fig. 14c (while the minimization of the confidence term Ω will be realized through a minimization of $\mathbf{w}^T\mathbf{w}$ again). The empirical risk is given as

$$R^{\varepsilon}_{emp}(\mathbf{w}, b) = \frac{1}{l}\sum_{i=1}^{l}\left|y_i - \mathbf{w}^T\mathbf{x}_i - b\right|_{\varepsilon}, \tag{42}$$

Fig. 16 shows two linear approximating functions as dashed lines inside an ε-tube having the same empirical risk R^{ε}_{emp} as the regression function $f(\mathbf{x}, \mathbf{w})$ on the training data.

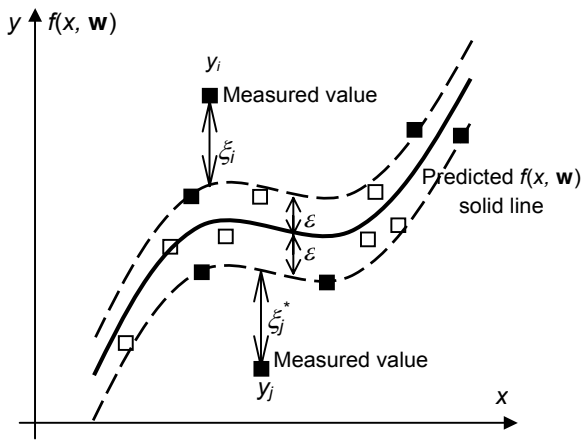

Figure 15. The parameters used in (1-D) support vector regression. Filled squares data ■ are support vectors, and the empty ones □ are not. Hence, SVs can appear only on the tube boundary or outside the tube.

As in classification, we try to minimize both the empirical risk R^{ε}_{emp} and $||\mathbf{w}||^2$ simultaneously. Thus, we construct a linear regression hyperplane

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x} + b$$

by minimizing

$$R = \frac{1}{2}\,||\mathbf{w}||^2 + C\sum_{i=1}^{l}\left|y_i - f(\mathbf{x}_i, \mathbf{w})\right|_{\varepsilon}, \tag{43}$$

Note that the last expression resembles the ridge regression scheme. However, we use Vapnik's ε-insensitivity loss function instead of a squared error now. From (41) and Fig. 15 it follows that for all training data outside an ε-tube,

$|y_i - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi$ for data "above" an ε-tube, or
$|y_i - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi^*$ for data "below" an ε-tube .

Thus, minimizing the risk R above equals the minimization of the following

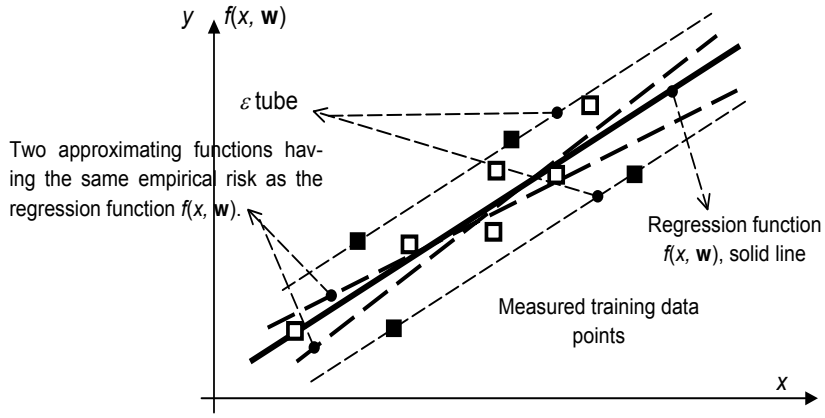

Figure 16. Two linear approximations inside an ε tube (*dashed lines*) have the same empirical risk R^{ε}_{emp} on the training data as the regression function (*solid line*).

risk

$$R_{\mathbf{w},\xi,\xi^*} = \Big[\frac{1}{2}\,||\mathbf{w}||^2 + C\Big(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i^*\Big)\Big], \tag{44}$$

under constraints

$$y_i - \mathbf{w}^T\mathbf{x}_i - b \leqslant \varepsilon + \xi_i, \quad i = 1, l, \tag{45a}$$

$$\mathbf{w}^T\mathbf{x}_i + b - y_i \leqslant \varepsilon + \xi_i^*, \quad i = 1, l, \tag{45b}$$

$$\xi_i \geqslant 0, \quad \xi_i^* \geqslant 0, \quad i = 1, l, \tag{45c}$$

where ξ_i and ξ_i^* are slack variables shown in Fig. 15 for measurements “above” and “below” an ε-tube respectively. Both slack variables are positive values. Lagrange multipliers α_i and α_i^* (that will be introduced during the minimization below) related to the first two sets of inequalities above, will be nonzero values for training points “above” and “below” an ε-tube respectively. Because no training data can be on both sides of the tube, either α_i or α_i^* will be nonzero. For data points inside the tube, both multipliers will be equal to zero. Thus $\alpha_i\alpha_i^* = 0$.

Note also that the constant C that influences a trade-off between an approximation error and the weight vector norm $||\mathbf{w}||$ is a design parameter that is chosen by the user. An increase in C penalizes larger errors i.e., it forces ξ_i and ξ_i^* to be small. This leads to an approximation error decrease which is achieved only by increasing the weight vector norm $||\mathbf{w}||$. However, an increase in $||\mathbf{w}||$ increases the confidence term Ω and does not guarantee a small generalization performance of a model. Another design parameter which is chosen by the user is the required precision embodied in an ε value that defines the size of an ε-tube. The choice of ε value is easier than the choice of C and it is given as either maximally allowed or some given or desired percentage of the output values y_i (say, $\varepsilon = 0.1$ of the mean value of $\mathbf{y}$).

Similar to procedures applied in the SV classifiers' design, we solve the constrained optimization problem above by forming a *primal variables Lagrangian* as follows,

$$\begin{aligned} &L_p(\mathbf{w}, b, \xi_i, \xi_i^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) = \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\big(\xi_i + \xi_i^*\big) - \sum_{i=1}^{l}\big(\beta_i^*\xi_i^* + \beta_i\xi_i\big) - \\ &\quad - \sum_{i=1}^{l}\alpha_i\big[\mathbf{w}^T\mathbf{x}_i + b - y_i + \varepsilon + \xi_i\big] - \\ &\quad - \sum_{i=1}^{l}\alpha_i^*\big[\mathbf{w}^T\mathbf{x}_i + b - y_i + \varepsilon + \xi_i\big]. \end{aligned} \tag{46}$$

A primal variables Lagrangian $L_p(\mathbf{w}, b, \xi_i, \xi_i^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)$ has to be *minimized* with respect to primal variables $\mathbf{w}$, b, ξ_i and ξ_i^* and *maximized* with respect to nonnegative Lagrange multipliers α_i, α_i^*, β_i and β_i^*. Hence, the function has the saddle point at the optimal solution ($\mathbf{w}_o$, b_o, ξ_{io}, ξ_{io}^*) to the original problem. At the optimal solution the partial derivatives of L_p in respect to primal variables vanishes. Namely,

$$\frac{\partial L_p(\mathbf{w}_o, b_o, \xi_{io}, \xi_{io}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial \mathbf{w}} = \mathbf{w}_o - \sum_{i=1}^{l}\big(\alpha_i - \alpha_i^*\big)\mathbf{x}_i = 0, \tag{47}$$

$$\frac{\partial L_p(\mathbf{w}_o, b_o, \xi_{io}, \xi_{io}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial b} = \sum_{i=1}^{l}\big(\alpha_i - \alpha_i^*\big) = 0, \tag{48}$$

$$\frac{\partial L_p(\mathbf{w}_o, b_o, \xi_{io}, \xi^*_{io}, \alpha_i, \alpha^*_i, \beta_i, \beta^*_i)}{\partial \xi_i} = C - \alpha_i, \beta_i = 0, \tag{49}$$

$$\frac{\partial L_p(\mathbf{w}_o, b_o, \xi_{io}, \xi^*_{io}, \alpha_i, \alpha^*_i, \beta_i, \beta^*_i)}{\partial \xi^*_i} = C - \alpha^*_i - \beta^*_i = 0, \tag{50}$$

Substituting the KKT above into the primal L_p given in (46), we arrive at the problem of the *maximization of a dual variables Lagrangian* $L_d(\alpha, \alpha^*)$ below,

$$\begin{aligned} L_d(\alpha, \alpha^*) = &-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha^*_i)(\alpha_j - \alpha^*_j)\mathbf{x}_i^T\mathbf{x}_j - \\ &- \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha^*_i) + \sum_{i,j=1}^{l}(\alpha_i - \alpha^*_i)y_i = \\ = &-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha^*_i)(\alpha_j - \alpha^*_j)\mathbf{x}_i^T\mathbf{x}_j - \\ &- \sum_{i=1}^{l}(\varepsilon - y_i)\alpha_i - \sum_{i=1}^{l}(\varepsilon + y_i)\alpha^*_i \end{aligned} \tag{51}$$

subject to constraints

$$\sum_{i=1}^{l}\alpha^*_i = \sum_{i=1}^{l}\alpha_i \quad \text{or} \quad \sum_{i=1}^{l}\big(\alpha_i - \alpha^*_i\big) = 0, \tag{52a}$$

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, l\,, \tag{52b}$$

$$0 \leqslant \alpha^*_i \leqslant C, \quad i = 1, l\,, \tag{52c}$$

Note that the dual variables Lagrangian $L_d(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$ is expressed in terms of Lagrange multipliers α_i and α^*_i only. However, the size of the problem, with respect to the size of an SV classifier design task, is doubled now. There are $2l$ unknown dual variables ($l \quad \alpha_i - s$ and $l \quad \alpha^*_i - s$) for a linear regression and the Hessian matrix $\mathbf{H}$ of the quadratic optimization problem in the case of regression is a $(2l, 2l)$ matrix. The *standard quadratic optimization problem* above can be expressed in a *matrix notation* and formulated as follows:

$$\text{minimize} \quad L_d(\boldsymbol{\alpha}) = 0.5\,\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} + \mathbf{f}^T\boldsymbol{\alpha}, \tag{53}$$

subject to (52) where

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l, \alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*]^T,$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{G} & -\mathbf{G} \\ -\mathbf{G} & \mathbf{G} \end{bmatrix},$$

$\mathbf{G}$ is an (l, l) matrix with entries $G_{ij} = [\mathbf{x}_i^T \mathbf{x}_j]$ for a linear regression[8], and

$$\mathbf{f} = [\varepsilon - y_1, \varepsilon - y_2, \ldots, \varepsilon - y_l, \varepsilon + y_1, \varepsilon + y_2, \ldots, \varepsilon + y_l]^T.$$

Again, (53) is written in a form of some standard optimization routine that typically *minimizes* given objective function subject to same constraints (52).

The learning stage results in l Lagrange multiplier *pairs* (α_i, α_i^*). After the learning, the number of nonzero parameters α_i or α_i^* is equal to the number of SVs. However, this number does not depend on the dimensionality of input space and this is particularly important when working in very high dimensional spaces. Because at least one element of each pair (α_i, α_i^*), $i = 1, l$, is zero, the product of α_i and α_i^* is always zero, i.e., $\alpha_i \alpha_i^* = 0$.

At the optimal solution the following *KKT complementarity conditions* must be fulfilled

$$\alpha_i \big(\mathbf{w}^T \mathbf{x}_i + b - y_i + \varepsilon + \xi_i\big) = 0, \tag{54}$$

$$\alpha_i^* \big(-\mathbf{w}^T \mathbf{x}_i - b + y_i + \varepsilon + \xi_i^*\big) = 0, \tag{55}$$

$$\beta_i \xi_i = (C - \alpha_i)\xi_i = 0, \tag{56}$$

$$\beta_i^* \xi_i^* = (C - \alpha_i^*)\xi_i^* = 0, \tag{57}$$

(56) states that for $0 < \alpha_i < C$, $\xi_i = 0$ holds. Similarly, from (57) follows that for $0 < \alpha_i^* < C$, $\xi_i^* = 0$ and, for $0 < \alpha_i < C$, $\alpha_i^* < C$, from (54) and (55) follows,

$$\mathbf{w}^T \mathbf{x}_i + b - y_i + \varepsilon = 0, \tag{58}$$

$$-\mathbf{w}^T \mathbf{x}_i - b + y_i + \varepsilon = 0. \tag{59}$$

Thus, for all the data points fulfilling $y - f(\mathbf{x}) = +\varepsilon$, dual variables α_i must be between 0 and C, or $0 < \alpha_i < C$, and for the ones satisfying $y - f(\mathbf{x}) = -\varepsilon$, α_i^* take on values $0 < \alpha_i^* < C$. These data points are called the *free* (or

[8]Note that G_{ij}, as given above, is a badly conditioned matrix and we rather use $G_{ij} = [\mathbf{x}_i^T \mathbf{x}_j + 1]$ instead.

unbounded) support vectors. They allow computing the value of the bias term b as given below

$$b = y_i - \mathbf{w}^T\mathbf{x}_i - \varepsilon = 0, \quad \text{for } \ 0 < \alpha_i < C, \tag{60a}$$

$$b = y_i - \mathbf{w}^T\mathbf{x}_i + \varepsilon = 0, \quad \text{for } \ 0 < \alpha_i^* < C. \tag{60b}$$

The calculation of a bias term b is numerically very sensitive, and it is better to compute the bias b by averaging over all the *free* support vector data points.

The final observation follows from (56) and (57) and it tells that for all the data points outside the ε-tube, i.e., when both $\xi_i > 0$ and $\xi_i^* > 0$, both α_i and α_i^* equal C, i.e., $\alpha_i = C$ for the points above the tube and $\alpha_i^* = C$ for the points below it. These data are the so-called *bounded support vectors.* Also, for all the training data points within the tube, or when $|y - f(\mathbf{x})| < \varepsilon$, both α_i and α_i^* equal zero and they are neither the support vectors nor do they construct the decision function $f(\mathbf{x})$.

After calculation of Lagrange multipliers α_i and α_i^*, using (47) we can find an *optimal* (desired) weight vector of the *regression hyperplane* as

$$\mathbf{w}_o = \sum_{i=1}^{l} \big(\alpha_i - \alpha_i^*\big)\mathbf{x}_i. \tag{61}$$

The best regression hyperplane obtained is given by

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}_o^T\mathbf{x} + b = \sum_{i=1}^{l} \big(\alpha_i - \alpha_i^*\big)\mathbf{x}_i^T\mathbf{x} + b. \tag{62}$$

More interesting, more common and the most challenging problem is to aim at solving the *nonlinear regression tasks*. A generalization to nonlinear regression is performed in the same way the nonlinear classifier is developed from the linear one, i.e., by carrying the mapping to the feature space, or by using kernel functions instead of performing the complete mapping which is usually of extremely high dimension (possibly even of an infinite dimension as it is the case with a Gaussian kernel function). Thus, the nonlinear regression function in an input space will be devised by considering a linear regression hyperplane in the *feature space*.

We use the same basic idea in designing SV machines for creating a *nonlinear regression function*. First, a mapping of input vectors $\mathbf{x} \in \Re^n$ into vectors $\Phi(\mathbf{x})$ of a higher dimensional *feature space* F (where Φ represents mapping:

$\Re^n \to \Re^f$) takes place and then, we solve a linear regression problem in this feature space. A mapping $\Phi(\mathbf{x})$ is again the chosen in advance, or fixed, function. Note that an input space ($\mathbf{x}$-space) is spanned by components x_i of an input vector $\mathbf{x}$ and a feature space F (Φ-space) is spanned by components $\phi_i(\mathbf{x})$ of a vector $\Phi(\mathbf{x})$. By performing such a mapping, we hope that in a Φ-space, our learning algorithm will be able to perform a linear regression hyperplane by applying the linear regression SVM formulation presented above. We also expect this approach to again lead to solving a quadratic optimization problem with inequality constraints in the feature space. The (linear in a feature space F) solution for the regression hyperplane $f = \mathbf{w}^T\Phi(\mathbf{x}) + b$, will create a nonlinear regressing hypersurface in the original input space. The most popular kernel functions are *polynomials* and *RBF* with *Gaussian kernels.* Both kernels are given in Table 2.

In the case of the *nonlinear regression*, the learning problem is again formulated as the maximization of a dual Lagrangian (53) with the Hessian matrix $\mathbf{H}$ structured in the same way as in a linear case, i.e. $\mathbf{H} = [\mathbf{G} \quad -\mathbf{G}; -\mathbf{G} \quad \mathbf{G}]$ but with the changed Grammian matrix $\mathbf{G}$ that is now given as

$$\mathbf{G} = \begin{bmatrix} G_{11} & \cdots & G_{1l} \\ \vdots & G_{ii} & \vdots \\ G_{l1} & \cdots & G_{ll} \end{bmatrix} \tag{63}$$

where the entries $G_{ij} = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j),\ i, j = 1, l.$

After calculating Lagrange multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, we can find an optimal weighting vector of the *kernels expansion* as

$$\mathbf{v}_o = \boldsymbol{\alpha} - \boldsymbol{\alpha}^*. \tag{64}$$

Note however the difference in respect to the linear regression where the expansion of a regression function is expressed by using the optimal weight vector $\mathbf{w}_o$. Here, in a NL SVMs' regression, the optimal weight vector $\mathbf{w}_o$ could be of infinite dimension (which is the case if the Gaussian kernel is used). Consequently, we neither calculate $\mathbf{w}_o$ nor we have to express it in a closed form at all. Instead, we create the best nonlinear regression function by using the weighting vector $\mathbf{v}_o$ and the kernel (Grammian) matrix $\mathbf{G}$ as follows,

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{G}\mathbf{v}_o + b. \tag{65}$$

In fact, the last result follows from the very setting of *the learning (optimizing) stage in a feature space* where, in all the equations above from (45) to (62),

we replace $\mathbf{x}_i$ by the corresponding feature vector $\Phi(\mathbf{x}_i)$. This leads to the following changes:

- instead $G_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ we get $G_{ij} = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$ and, by using the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$, it follows that $G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$;
- similarly, (61) and (62) change as follows:

$$\mathbf{w}_o = \sum_{i=1}^{l} \big(\alpha_i - \alpha_i^*\big)\Phi(\mathbf{x}_i), \tag{66}$$

and,

$$\begin{aligned} f(\mathbf{x}, \mathbf{w}) = \mathbf{w}_o^T \Phi(\mathbf{x}) + b = \sum_{i=1}^{l} \big(\alpha_i - \alpha_i^*\big)\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) + b = \\ = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i \mathbf{x} + b. \end{aligned} \tag{67}$$

If the bias term b is explicitly used as in (65) then, for a NL SVMs' regression, it can be calculated from the upper SVs as,

$$\begin{aligned} b = y_i - \sum\nolimits_{j=1}^{N\ free\ upper\ SVs} \big(\alpha_i - \alpha_i^*\big)\Phi^T(\mathbf{x}_j)\Phi(\mathbf{x}_i) - \varepsilon = \\ = y_i - \sum\nolimits_{j=1}^{N\ free\ upper\ SVs} \big(\alpha_i - \alpha_i^*\big) K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon, \\ \text{for } \ 0 < \alpha_i < C, \end{aligned} \tag{68a}$$

or from the lower ones as,

$$\begin{aligned} b = y_i - \sum\nolimits_{j=1}^{N\ free\ lower\ SVs} \big(\alpha_i - \alpha_i^*\big)\Phi^T(\mathbf{x}_j)\Phi(\mathbf{x}_i) + \varepsilon = \\ = y_i - \sum\nolimits_{j=1}^{N\ free\ lower\ SVs} \big(\alpha_i - \alpha_i^*\big) K(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon, \\ \text{for } \ 0 < \alpha_i^* < C, \end{aligned} \tag{68b}$$

Note that $\alpha_j^* = 0$ in (68a) and so is $\alpha_j = 0$ in (68b). Again, it is much better to calculate the bias term b by an averaging *over all* the *free* support vector data points.

There are a few learning parameters in constructing SV machines for regression. The three most relevant are the insensitivity zone ε, the penalty parameter

C (that determines the trade-off between the training error and VC dimension of the model), and the shape parameters of the kernel function (variances of a Gaussian kernel, order of the polynomial, or the shape parameters of the inverse multiquadrics kernel function). All three parameters' sets should be selected by the user. To this end, the most popular selection method is a cross-validation. Unlike in a classification, for not too noisy data (primarily without huge outliers), the penalty parameter C could be set to infinity and the modeling can be controlled by changing the insensitivity zone ε and shape parameters only.

The *example* below shows how an increase in an insensitivity zone ε has smoothing effects on modeling highly noise polluted data. Increase in ε means a reduction in requirements on the accuracy of an approximation. It decreases the number of SVs leading to higher data compression too. This can be readily followed in the lines and Fig. 17 below.

Example: *Nonlinear regression by SVMs*

The task here is to construct an SV machine for modeling measured data pairs. The underlying function (known to us but, not to the SVM) is a sinus function multiplied by the square one (i.e., $f(x) = x^2 \sin x$) and it is corrupted by 25% of normally distributed noise with a zero mean. Analyze the influence of an insensitivity zone ε on modeling quality and on a compression of data, meaning on the number of SVs.

Fig. 17 shows that for a very noisy data a decrease of an insensitivity zone ε (i.e., shrinking of the tube shown by dashed line) approximates the noisy data points more closely. The related more and more wiggly shape of the regression function can be achieved only by including more and more support vectors. However, being good on the noisy training data points easily leads to an overfitting. The cross-validation should help in finding correct ε value, resulting in a regression function that filters the noise out but not the true dependency and which, consequently, approximate the underlying function as close as possible.

The approximation functions shown in Fig. 17 are created by 9 and 18 weighted Gaussian basis functions for $\varepsilon = 1$ and $\varepsilon = 0.75$ respectively. These supporting functions are not shown in the figure. However, the way how the learning algorithm selects SVs is an interesting property of support vector machines and in Fig. 18 we also present the supporting Gaussian functions.

Note that the selected Gaussians lie in the dynamic area of the function in Fig. 18. Here, these areas are close to both the left hand and the right hand boundary. In the middle, the original function is pretty flat and there is no need to cover this part by supporting Gaussians. The learning algorithm realizes this

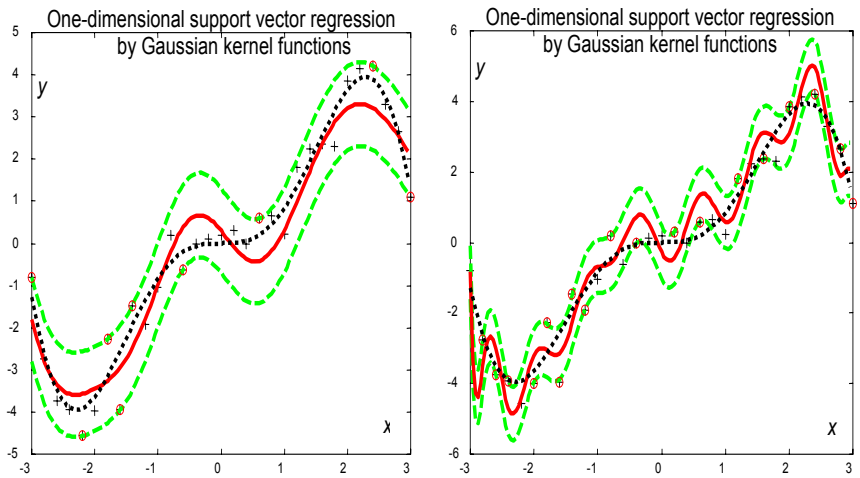

Figure 17. The influence of an insensitivity zone ε on the model performance. A nonlinear SVM creates a regression function f with Gaussian kernels and models a highly polluted (25% noise) function $x^2 sin(x)$ (*dotted*). 31 training data points (*plus signs*) are used. **Left:** $\varepsilon = 1$; 9 SVs are chosen (*encircled plus signs*). **Right:** $\varepsilon = 0.75$; the 18 chosen SVs produced a better approximation to noisy data and, consequently, there is the tendency of overfitting.

fact and simply, it does not select any training data point in this area as a support vector. Note also that the Gaussians are not weighted in Fig 18, and they all have the peak value of 1. The standard deviation of Gaussians is chosen in order to see Gaussian supporting functions better. Here, in Fig. 18, $\sigma = 0.6$. Such a choice is due the fact that for the larger σ values the basis functions are rather broad and flat above the domain shown. Thus, the supporting Gaussian functions are covering the whole domain as the broad umbrellas. For very big variances one wouldn't be able to distinguish them visually.

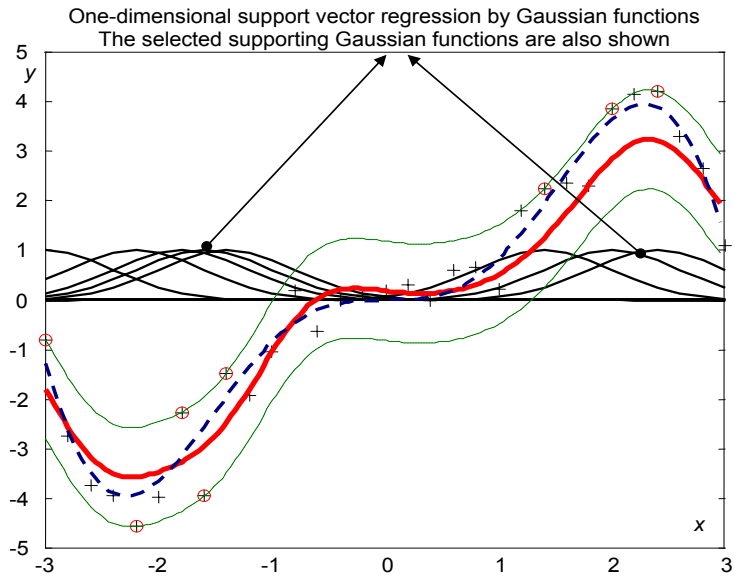

Figure 18. Regression function f created as the sum of 8 weighted Gaussian kernels. A standard deviation of Gaussian bells $\sigma = 0.6$. Original function (*dashed line*) is $x^2 \sin x$ and it is corrupted by 0.25% noise. 31 training data points are shown as plus signs. Data points selected as the SVs are encircled. The 8 selected supporting Gaussian functions are centered at these data points.

Implementation Issues

In both the classification and the regression the learning problem boils down to solving the QP problem subject to the so-called 'box-constraints and to the equality constraint in the case that a model with a bias term b is used. The SV training works almost perfectly for not too large data basis. However, when the number of data points is large (say $l > 2,000$) the QP problem becomes extremely difficult to solve with standard QP solvers and methods. For example, a classification training set of 50,000 examples amounts to a Hessian matrix $\mathbf{H}$ with 2.5×10^9 (2.5 billion) elements. Using an 8-byte floating-point representation we need 20,000 Megabytes = 20 Gigabytes of memory (*Osuna* et al, 1997).

This cannot be easily fit into memory of present standard computers, and this is the single basic disadvantage of the SVM method. There are three approaches that resolve the QP for large data sets. *Vapnik* in (*Vapnik*, 1995) proposed the *chunking method* that is the decomposition approach. Another *decomposition* approach is suggested in (*Osuna* et al, 1997). The sequential minimal optimization (*Platt*, 1997) algorithm is of different character and it seems to be an "error back propagation" for SVM learning. A systematic exposition of these various techniques is not given here, as all three would require a lot of space. However, the interested reader can find a description and discussion about the novel algorithms in (*Kecman, Huang*, and *Vogt*, 2005; *Vogt* and *Kecman*, 2005). The *Vogt* and *Kecman*'s chapter discusses the application of an *active set* algorithm in solving small to medium sized QP problems. For such data sets and when the high precision is required the active set approach in solving QP problems seems to be superior to other approaches (notably the interior point methods and SMO algorithm). The *Kecman, Huang,* and *Vogt*'s chapter introduces the efficient *iterative single data algorithm* (*ISDA*) for solving huge data sets (say more than 100,000 or 500,000 or over 1 million training data pairs). It seems that ISDA is the fastest algorithm at the moment for such large data sets (see the comparisons with SMO in (*Kecman, Huang* and *Vogt*, 2005)) still ensuring the convergence to the global minimum. This means that the ISDA provides the exact, and not the approximate, solution to the original dual problem. In the next section we will introduce the ISDA algorithm. As for now, let us conclude the presentation of the classic SVMs part by summarizing the basic constructive steps that lead to the SV machine.

A training and design of a support vector machine is an *iterative* algorithm and it involves the following steps:

a) define your problem as the classification or as the regression one;
b) preprocess your input data: select the most relevant features, scale the data between $[-1, 1]$, or to the ones having zero mean and variances equal to one, check for possible outliers (strange data points);
c) select the kernel function that determines the hypothesis space of the decision and regression function in the classification and regression problems respectively;
d) select the "shape", i.e., "smoothing" parameter of the kernel function (for example, polynomial degree for polynomials and variances of the Gaussian RBF kernels respectively);
e) choose the penalty factor C and, in the regression, select the desired accuracy by defining the insensitivity zone ε too;

f) solve the QP problem in l and $2l$ variables in the case of classification and regression problems respectively;
g) validate the model obtained on some previously (i.e., during the training) unseen test data, and if not pleased iterate between steps `d` (or, eventually `c`) and `g`.

The optimizing part `(f)` is computationally extremely demanding. First, the Hessian matrix $\mathbf{H}$ scales with the size of a data set — it is an (l, l) and an $(2l, 2l)$ matrix in classification and regression respectively. Second, unlike in classic original QP problems $\mathbf{H}$ is very dense matrix and it is usually badly conditioned requiring regularization before any numeric operation. Regularization means an addition of a small number to the diagonal elements of $\mathbf{H}$. Luckily, there are many reliable and fast QP solvers. A simple internet search will reveal many of them. Particularly, in addition to the classic ones such as MINOS or LOQO for example, there are many more free QP solvers designed specially for the SVMs. The most popular ones are — the LIBSVM, SVMlight, SVM Torch, mySVM and SVM Fu. All of them can be downloaded from their corresponding sites. Good educational software in MATLAB named LEARNSC, with very good graphic presentations of all relevant objects in a SVM modeling, can be downloaded from the author's book site [9] too.

Finally we mention that there are many alternative formulations and approaches to the QP based SVMs described above. Notably, they are the linear programming SVMs (*Mangasarian*, 1965; *Friess* and *Harrison*, 1998; *Smola*, et al, 1998; *Hadzic* and *Kecman*, 1999; *Graepel* et al, 1999; *Kecman* and *Hadzic*, 2000; *Kecman*, 2001; *Kecman, Arthanari, Hadzic*, 2001), υ-SVMs (*Schölkopf* and *Smola*, 2002) and least squares support vector machines (*Suykens* et al, 2002). Their description is far beyond this chapter and the curious readers are referred to references given above.

Below we introduce a novel Iterative Single Data Algorithm (ISDA) for resolving the problems coming from huge Hessian matrices in training SVMs. First, we show the equality of various approaches in learning from data and afterwards we present two variants of ISDA, namely one with the bias term b and the other without it.

The lines below are the shortened versions of two papers presented at the ESANN 2003 and 2004 which can be downloaded (together with few more contributions of the author) from the author's site [10].

[9] URL: www.support-vector.ws
[10] URL: http://www.support-vector.ws/html/publications.html

On the Equality of Kernel AdaTron and Sequential Minimal Optimization and Alike Algorithms for Kernel Machines

This section presents the equality of a kernel AdaTron (KA) method (originating from a gradient ascent learning approach) and sequential minimal optimization (SMO) learning algorithm (based on an analytic quadratic programming step) in designing the support vector machines (SVMs) having *positive definite kernels*. The conditions of the equality of two methods are established. The equality is valid for both the nonlinear classification and the nonlinear regression tasks, and it sheds a new light to these seemingly different learning approaches. The section also introduces other learning techniques related to the two mentioned approaches, such as the nonnegative conjugate gradient, classic Gauss-Seidel (GS) coordinate ascent procedure and its derivative known as the successive over-relaxation (SOR) algorithm as a viable and usually faster training algorithms for performing nonlinear classification and regression tasks. The convergence theorem for these related iterative algorithms is proven. Due to restricted space, the presentation is scarce, giving only the final expressions. More detailed derivations can be found in some papers and book chapter from the site given at the end of the previous section above. In addition, the ISDA software for solving huge SVMs' learning problems can be downloaded from appropriate site [11]. The site is accompanying the *Huang*, *Kecman* and *Kopriva*'s book which is in print at Springer Verlag.

Introduction

One of the mainstream research fields in learning from empirical data by support vector machines, and solving both the classification and the regression problems, is an implementation of the incremental learning schemes when the training data set is huge. Among several candidates that avoid the use of standard quadratic programming (QP) solvers, the two learning approaches which have recently got the attention are the KA (*Anlauf, Biehl*, 1989; *Friess, Cristianini, Campbell*, 1998; *Veropoulos*, 2001) and the SMO (*Platt*, 1998, 1999; *Vogt*, 2002). Due to its analytical foundation the SMO approach is particularly popular and at the moment the widest used, analyzed and still heavily developing algorithm. At the same time, the KA although providing similar results in solving classification problems (in terms of both the accuracy and the training

[11] URL: www.learning-from-data.com

computation time required) did not attract that many devotees. There are two basic reasons for that. First, until recently (*Veropoulos*, 2001) the KA seemed to be restricted to the classification problems only and second, it "lacked" the fleur of the strong theory (despite its beautiful "simplicity" and strong convergence proofs). The KA is based on a gradient ascent technique and this fact might have also distracted some researchers being aware of problems with gradient ascent approaches faced with possibly ill-conditioned kernel matrix.

Here we show when and why the recently developed algorithms for SMO using positive definite kernels or models *without a bias term* (Vogt, 2002), and the KA for both *classification* (*Friess, Cristianini, Campbell*, 1998) and *regression* (*Veropoulos*, 2001) are identical. Both the KA and the SMO algorithm attempt to solve the QP problem in the case of ***classification*** by *maximizing* the dual Lagrangian under the constraints as given in equations (37).

In the case of the ***nonlinear regression*** the learning problem is the maximization of a dual Lagrangian as given in equations (51) and (52). Note that (51) is given for a linear regression hypersurface. For the nonlinear regression the scalar product $\mathbf{x}_i^T \mathbf{x}_j$ must be replaced by the kernel function value $K(\mathbf{x}_i, \mathbf{x}_j)$.

The KA and SMO learning algorithms without-bias-term

It is known that *positive definite kernels* (such as the most popular and the most widely used RBF Gaussian kernels as well as the complete polynomial ones) do not require bias term (*Evgeniou, Pontil, Poggio*, 2000). Below, the KA and the SMO algorithms will be presented for such a fixed (i.e., no-) bias design problem and compared for the classification and regression cases. The equality of two learning schemes and resulting models will be established. Originally, in (*Platt*, 1998, 1999), the SMO *classification* algorithm was developed for solving the problem (1) including the constraints related to the bias b. In these early publications the case when bias b is fixed variable was also mentioned but the detailed analysis of a fixed bias update was not accomplished.

Incremental Learning in Classification

(a) Kernel AdaTron in classification. The classic AdaTron algorithm as given in (*Anlauf* and *Biehl*, 1989) is developed for linear classifier. The KA is a variant of the classic AdaTron algorithm in the feature space of SVMs (*Friess* et al., 1998). The KA algorithm solves the maximization of the dual

Lagrangian (37) by implementing the gradient ascent algorithm. The update $\Delta\alpha_i$ of the dual variables α_i is given as

$$\Delta\alpha_i = \eta\frac{\partial L_d}{\partial \alpha_i} = \eta\Big(1 - y_i\sum_{j=1}^{l}\alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\Big) = \eta\big(1 - y_i f_i\big), \tag{69a}$$

where f_i is the value of the decision function f at the point $\mathbf{x}_i$, i.e.,

$$f_i = \sum_{j=1}^{l}\alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j),$$

and y_i denotes the value of the desired target (or the class' label) which is either $+1$ or -1. The update of the dual variables α_i is given as

$$\alpha_i \leftarrow \min(\max(0, \alpha_i + \Delta\alpha_i), C), \quad (i = 1, \ldots, l). \tag{69b}$$

In other words, the dual variables α_i are clipped to zero if $(\alpha_i + \Delta\alpha_i) < 0$. In the case of the soft nonlinear classifier $(C < \infty)$, α_i are clipped between zero and C, $(0 \leqslant \alpha_i \leqslant C)$. The algorithm converges from any initial setting for the Lagrange multipliers α_i.

(b) SMO without-bias-term in classification. Recently (*Vogt*, 2002) derived the update rule for multipliers α_i that includes a detailed analysis of the Karush-Kuhn-Tucker (KKT) conditions for checking the optimality of the solution [12]. The following update rule for α_i for a no-bias SMO algorithm was proposed

$$\Delta\alpha_i = -\frac{y_i E_i}{K(\mathbf{x}_i, \mathbf{x}_i)} = -\frac{y_i f_i - 1}{K(\mathbf{x}_i, \mathbf{x}_i)} = \frac{1 - y_i f_i}{K(\mathbf{x}_i, \mathbf{x}_i)}, \tag{70}$$

where $E_i = f_i - y_i$ denotes the difference between the value of the decision function f at the point $\mathbf{x}_i$ and the desired target (label) y_i. Note the equality of (69a) and (70) when the learning rate in (69a) is chosen to be $\eta_i = 1/K(\mathbf{x}_i, \mathbf{x}_i)$. The important part of the SMO algorithm is to check the KKT conditions with precision τ (e.g., $\tau = 10^{-3}$) in each step. An update is performed only if

$$\begin{aligned} &\alpha_i < C \ \wedge\ y_i E_i < -\tau, \quad \text{or} \\ &\alpha_i > 0 \ \wedge\ y_i E_i > \tau. \end{aligned} \tag{70a}$$

[12] As referred above, a fixed bias update was only mentioned in Platt's papers.

After an update, the same clipping operation as in (69b) is performed

$$\alpha_i \leftarrow \min(\max(0, \alpha_i + \Delta\alpha_i), C), \quad (i = 1, \ldots, l). \tag{70b}$$

It is the nonlinear clipping operation in (69b) and in (70b) that strictly equals the KA and the SMO without-bias-term algorithm in solving nonlinear classification problems. This fact sheds new light on both algorithms. This equality is not that obvious in the case of a "classic" SMO algorithm with bias term due to the heuristics involved in the selection of active points which should ensure the largest increase of the dual Lagrangian L_d during the iterative optimization steps.

Incremental Learning in Regression. Similarly to the case of classification, there is a strict equality between the KA and the SMO algorithm when positive definite kernels are used for nonlinear regression.

(a) Kernel AdaTron in regression. The first extension of the Kernel AdaTron algorithm for regression is presented in (*Veropoulos*, 2001) as the following gradient ascent update rules for α_i and α_i^*

$$\begin{aligned} \Delta\alpha_i = \eta_i \frac{\partial L_d}{\partial \alpha_i} &= \eta_i \Big(y_i - \varepsilon - \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) \Big) = \\ &= \eta_i (y_i - \varepsilon - f_i) = -\eta_i (E_i + \varepsilon), \end{aligned} \tag{71a}$$

$$\begin{aligned} \Delta\alpha_i^* = \eta_i \frac{\partial L_d}{\partial \alpha_i^*} &= \eta_i \Big(-y_i - \varepsilon + \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}_i) \Big) = \\ &= \eta_i (y_i - \varepsilon + f_i) = \eta_i (E_i - \varepsilon), \end{aligned} \tag{71b}$$

where y_i is the measured value for the input $\mathbf{x}_i$, ε is the prescribed insensitivity zone, and $E_i = f_i - y_i$ stands for the difference (an error) between the regression function f at the point $\mathbf{x}_i$ and the desired target value y_i at this point. The calculation of the gradient above does not take into account the geometric reality that no training data can be on both sides of the tube. In other words, it does not use the fact that either α_i or α_i^* or both will be nonzero, i.e., that $\alpha_i \alpha_i^* = 0$ must be fulfilled in each iteration step. Below we derive the gradients of the

dual Lagrangian L_d accounting for geometry. This new formulation of the KA algorithm strictly equals the SMO method and it is given as

$$\begin{aligned}\frac{\partial L_d}{\partial \alpha_i^*} &= -K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i - \sum_{j=1,j\neq i}^{l}\big(\alpha_j-\alpha_j^*\big)K(\mathbf{x}_j,\mathbf{x}_i) + y_i - \varepsilon + \\ &\quad + K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i - K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* = \\ &= K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* - \big(\alpha_i-\alpha_i^*\big)K(\mathbf{x}_i,\mathbf{x}_i) - \\ &\quad - \sum_{j=1,j\neq i}^{l}\big(\alpha_j-\alpha_j^*\big)K(\mathbf{x}_j,\mathbf{x}_i) + y_i - \varepsilon = \\ &= -K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* + y_i - \varepsilon - f_i = -\big(K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* + E_i + \varepsilon\big).\end{aligned} \tag{72a}$$

For the α_i^* multipliers, the value of the gradient is

$$\frac{\partial L_d}{\partial \alpha_i^*} = -K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i + E_i - \varepsilon. \tag{72b}$$

The update value for α_i is now

$$\Delta\alpha_i = \eta_i\frac{\partial L_d}{\partial \alpha_i} = -\eta_i\big(K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* + E_i + \varepsilon\big), \tag{73a}$$

$$\alpha_i \leftarrow \alpha_i + \Delta\alpha_i = \alpha_i + \eta_i\frac{\partial L_d}{\partial \alpha_i} = \alpha_i - \eta_i\big(K(\mathbf{x}_i,\mathbf{x}_i)\,\alpha_i^* + E_i + \varepsilon\big). \tag{73b}$$

For the learning rate $\eta_i = 1/K(\mathbf{x}_i,\mathbf{x}_i)$ the gradient ascent learning KA is defined as,

$$\alpha_i \leftarrow \alpha_i - \alpha_i^* - \frac{E_i+\varepsilon}{K(\mathbf{x}_i,\mathbf{x}_i)}, \tag{74a}$$

Similarly, the update rule for α_i^* is

$$\alpha_i^* \leftarrow \alpha_i^* - \alpha_i + \frac{E_i-\varepsilon}{K(\mathbf{x}_i,\mathbf{x}_i)}. \tag{74b}$$

Same as in the classification, α_i and α_i^* are clipped between zero and C,

$$\alpha_i \leftarrow \min(\max(0,\alpha_i),C), \quad i=1,\ldots,l, \tag{75a}$$

$$\alpha_i^* \leftarrow \min(\max(0,\alpha_i^*),C), \quad i=1,\ldots,l. \tag{75b}$$

(b) SMO without-bias-term in regression. The first algorithm for the SMO without-bias-term in regression (together with a detailed analysis of the KKT conditions for checking the optimality of the solution) is derived in (*Vogt*, 2002). The following learning rules for the Lagrange multipliers α_i and α_i^* updates were proposed

$$\alpha_i \leftarrow \alpha_i - \alpha_i^* - \frac{E_i + \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)}, \tag{76a}$$

$$\alpha_i^* \leftarrow \alpha_i^* - \alpha_i + \frac{E_i - \varepsilon}{K(\mathbf{x}_i, \mathbf{x}_i)}. \tag{76b}$$

The equality of equations (74a, b) and (76a, b) is obvious when the learning rate, as presented above in (74a, b), is chosen to be $\eta_i = 1/K(\mathbf{x}_i, \mathbf{x}_i)$. Thus, in both the classification and the regression, the optimal learning rate is not necessarily equal for all training data pairs. For a Gaussian kernel, $\eta = 1$ is same for all data points, and for a complete n^{th} order polynomial each data point has different learning rate $\eta_i = 1/(\mathbf{x}_i^T \mathbf{x}_i + 1)^n$. Similar to classification, a joint update of α_i and α_i^* is performed only if the KKT conditions are violated by at least τ , i.e. if

$$\begin{aligned} &\alpha_i < C \ \wedge \ \varepsilon + E_i < -\tau, \quad \text{or} \\ &\alpha_i > 0 \ \wedge \ \varepsilon + E_i > \tau, \quad \text{or} \\ &\alpha_i^* < C \ \wedge \ \varepsilon - E_i < -\tau, \quad \text{or} \\ &\alpha_i^* > 0 \ \wedge \ \varepsilon - E_i > \tau. \end{aligned} \tag{77}$$

After the changes, the same clipping operations as defined in (11) are performed

$$\alpha_i \leftarrow \min(\max(0, \alpha_i), C), \quad i = 1, \ldots, l, \tag{78a}$$

$$\alpha_i^* \leftarrow \min(\max(0, \alpha_i^*), C), \quad i = 1, \ldots, l. \tag{78b}$$

The KA learning as formulated in this section and the SMO algorithm without-bias-term for solving regression tasks are strictly equal in terms of both the number of iterations required and the final values of the Lagrange multipliers. The equality is strict despite the fact that the implementation is slightly different. In every iteration step, namely, the KA algorithm updates both weights α_i and α_i^* without any checking whether the KKT conditions are fulfilled or not, while the SMO performs an update according to equations (77).

The Coordinate Ascent Based Learning for Nonlinear Classification and Regression Tasks — The Gauss-Seidel Algorithm

When positive definite kernels are used, the learning problem for both tasks is same. In a vector-matrix notation, in a dual space, the learning is defined as:

maximize

$$L_d(\alpha) = -0.5\alpha^T \mathbf{K}\alpha + \mathbf{f}^T\alpha, \tag{79}$$

such that

$$0 \leqslant \alpha_i \leqslant C, \quad (i = 1, \ldots, n), \tag{80}$$

where, in the classification $n = l$ and the matrix $\mathbf{K}$ is an (l, l) symmetric positive definite matrix, while in regression $n = 2l$ and $\mathbf{K}$ is a $(2l, 2l)$ symmetric semi-positive definite one. Note that the constraints (80) define a convex subspace over which the convex dual Lagrangian should be maximized. It is very well known that the vector $\boldsymbol{\alpha}$ may be looked at as the solution of a system of linear equations

$$\mathbf{K}\boldsymbol{\alpha} = \mathbf{f} \tag{81}$$

subject to the same constraints as given by (80).

Thus, it may seem natural to solve (81), subject to (80), by applying some of the well known and established techniques for solving a general linear system of equations. The size of training data set and the constraints (80) eliminate direct techniques. Hence, one has to resort to the *iterative approaches* in solving the problems above. There are three possible iterative avenues that can be followed. They are; the use of the Non-Negative Least Squares (NNLS) technique (*Lawson* and *Hanson*, 1974), application of the Non-Negative Conjugate Gradient (NNCG) method (*Hestenes*, 1980) and the implementation of Gauss-Seidel (GS) i.e., the related Successive Over-Relaxation technique (SOR). The first two methods, in their "classic" appearance, solve for the non-negative constraints only. Thus, they are not suitable in solving "soft" tasks, when penalty parameter $C < \infty$ is used, i.e., when there is an upper bound on maximal value of α_i. In the case of nonlinear regression, one can apply NNLS and NNCG by taking $C = \infty$ and compensating (i.e. smoothing or "softening" the solution) by increasing the sensitivity zone ε. The two methods (namely NNLS and NNCG) are not suitable for solving soft margin ($C < \infty$) classification problems in their present form, because there is no other parameter that can be used in "softening" the margin. Recently, the NNCG algorithm for solving (81) with box-constraints (80) is developed and presented in (*Huang, Kecman* and

Kopriva, 2006). However, due to the usually bad conditioned Hessian matrix $\mathbf{K}$, NNCG will not be used in the lines below.

Here we show how to extend the application of GS and SOR to both the nonlinear classification and to the nonlinear regression tasks. The Gauss-Seidel method solves (81) by using the ith equation to update the ith unknown doing it iteratively, i.e., starting in the kth step with the first equation to compute the α_1^{k+1}, then the second equation is used to calculate the α_2^{k+1} by using new α_1^{k+1} and $\alpha_i^k \quad (i > 2)$ and so on. The iterative learning takes the following form,

$$\begin{aligned}\alpha_i^{k+1} &= \Big(f_i - \sum_{j=1}^{i-1} K_{ij}\alpha_j^{k+1} - \sum_{j=i+1}^{n} K_{ij}\alpha_j^k\Big)/K_{ii} = \\ &= \alpha_i^k - \frac{1}{K_{ii}}\Big(\sum_{j=1}^{i-1} K_{ij}\alpha_j^{k+1} + \sum_{j=i}^{n} K_{ij}\alpha_j^k - f_i\Big) = \\ &= \alpha_i^k + \frac{1}{K_{ii}}\left.\frac{\partial L_d}{\partial \alpha_i}\right|_{k+1},\end{aligned} \tag{82}$$

where we use the fact that the term within a second bracket (called the residual r_i in mathematics' references) is the ith element of the gradient of a dual Lagrangian L_d given in (79) at the $(k+1)^{th}$ iteration step. The equation (82) above shows that GS method is a *coordinate* gradient ascent procedure as the KA and the SMO are. *The KA and SMO for positive definite kernels equal the GS!* Note that the optimal learning rate used in both the KA algorithm and in the SMO without-bias-term approach is exactly equal to the coefficient $1/K_{ii}$ in a GS method. Based on this equality, the convergence theorem for the KA, SMO and GS (i.e., SOR) in solving (81) subject to constraints (80) can be stated and proved as follows:

Theorem: For SVMs with positive definite kernels (while using them without the bias term b) the iterative learning algorithms KA i.e., SMO i.e., GS i.e., SOR, in solving nonlinear classification and regression tasks (81) subject to constraints (80), converge starting from any initial choice of $\boldsymbol{\alpha}_0$.

Proof: The proof is based on the very well known theorem of convergence of the GS method for symmetric positive definite matrices in solving (81) without constraints (*Ostrowski*, 1966). First note that for positive definite kernels, the matrix $\mathbf{K}$ created by terms $y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ in the second sum in (1), and involved in solving classification problem, is also positive definite. In regression

tasks $\mathbf{K}$ is a symmetric positive semi-definite (meaning still convex) matrix, which after a mild regularization given as ($\mathbf{K} \leftarrow \mathbf{K} + \lambda\mathbf{I}$, $\lambda \sim 1e-12$) becomes positive definite one. (Note that the proof in the case of regression does not need regularization at all, but there is no space here to go into these details). Hence, the learning without constraints (80) converges, starting from any initial point $\boldsymbol{\alpha}_0$, and each point in an n-dimensional search space for multipliers α_i is a viable starting point ensuring a convergence of the algorithm to the maximum of a dual Lagrangian L_d. This, naturally, includes all the (starting) points within, or on a boundary of, any convex subspace of a search space ensuring the convergence of the algorithm to the maximum of a dual Lagrangian L_d over the given subspace. The constraints imposed by (80) preventing variables α_i to be negative or bigger than C, and implemented by the clipping operators above, define such a convex subspace. Thus, each "clipped" multiplier value α_i defines a new starting point of the algorithm guaranteeing the convergence to the maximum of L_d over the subspace defined by (80). For a convex constraining subspace such a constrained maximum is unique. ***Q.E.D.***

Due to the lack of the space we do not go into the discussion on the convergence rate here and we leave it to some other occasion. It should be only mentioned that both KA and SMO (i.e. GS and SOR) for positive definite kernels have been successfully applied for many problems (see references given here, as well as many other, benchmarking the mentioned methods on various data sets). Finally, let us just mention that the standard extension of the GS method is the method of successive over-relaxation that can reduce the number of iterations required by proper choice of relaxation parameter ω significantly. The SOR method uses the following updating rule

$$\begin{aligned} \alpha_i^{k+1} &= \alpha_i^k - \omega \frac{1}{K_{ii}} \Big(\sum_{j=1}^{i-1} K_{ij}\alpha_j^{k+1} + \sum_{j=i}^{n} K_{ij}\alpha_j^k - f_i \Big) = \\ &= \alpha_i^k + \omega \frac{1}{K_{ii}} \left. \frac{\partial L_d}{\partial \alpha_i} \right|_{k+1}, \end{aligned} \tag{83}$$

and similarly to the KA, SMO, and GS its convergence is guaranteed for $0 < \omega < 2$.

SVMs with a Bias Term b

Now, we discuss and present the use and calculation of the explicit bias term b in the support vector machines within the Iterative Single training Data learning Algorithm. The approach with a bias b can also be used for both nonlinear classification and nonlinear regression tasks. It is well known that for positive definite kernels there is no need for bias b (*Kecman*, 2001). We used this fact while developing ISDA in previous section. However, one can use the bias term b and this means implementing a different kernel. There is a report and a paper where this issue has been discussed. In (*Poggio* et al., 2001)

$$f(\mathbf{x}) = \sum_{j=1}^{l} w_j K(\mathbf{x}, \mathbf{x}_j) + b$$

and it was shown that $f(\mathbf{x})$ is a function resulting from a minimization of the functional shown below

$$I[f] = \sum_{j=1}^{l} V(y_j, f(\mathbf{x}_j)) + \lambda \big|\big| f \big|\big|^2_{K^*}, \tag{84}$$

where $K^* = K - a$ (for an appropriate constant a) and K is an original kernel function (more details can be found in the mentioned report). This means that by adding a constant term to a positive definite kernel function K, one obtains the solution to the functional $I[f]$ where K^* is a conditionally positive definite kernel. Interestingly, similar type of model was also presented in (*Mangasarian* and *Musicant*, 1999). However, their formulation is done for the classification problems only. They reformulated the optimization by adding the $b^2/2$ term to the cost function $||\mathbf{w}||^2/2$. This is equivalent to an addition of 1 to the original kernel matrix $\mathbf{K}$. As a result, they changed the original classification dual problem to the optimization of the following one

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (K(\mathbf{x}_i, \mathbf{x}_j) + 1). \tag{85}$$

Iterative Single Data Algorithm for SVMs with Bias

In section “On the Equality of Kernel AdaTron and Sequential Minimal Optimization and Alike Algorithms for Kernel Machines” and for the SVMs models

when positive definite kernels are used without a bias term b, the learning algorithms for classification and regression (in a dual domain) were solved with box constraints only, originating from minimization of a primal Lagrangian in respect to the weights w_i. However, there remains an open question — how to apply the proposed ISDA for the SVMs that do use explicit bias term b. Such general nonlinear SVMs in classification and regression tasks are given below,

$$f(\mathbf{x}_i) = \sum_{j=1}^{l} y_j \alpha_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) + b = \sum_{j=1}^{l} w_j K(\mathbf{x}_i, \mathbf{x}_j) + b, \tag{86a}$$

$$f(\mathbf{x}_i) = \sum_{j=1}^{l} (\alpha_j^* - \alpha_j) \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) + b = \sum_{j=1}^{l} w_j K(\mathbf{x}_i, \mathbf{x}_j) + b, \tag{86b}$$

where $\Phi(\mathbf{x}_i)$ is the m-dimensional vector that maps n-dimensional input vector $\mathbf{x}$ into the feature space[13]. For each SVMs' model in (86), there is also one *equality constraint* originating from a minimization of the primal objective function in respect to the bias b as given below,

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{87a}$$

in a classification, and

$$\sum_{i=1}^{l} \alpha_i^* = \sum_{i=1}^{l} \alpha_i \tag{87b}$$

in a regression.

The motivation for developing the ISDA for the SVMs with an explicit bias term b originates from the fact that the use of an explicit bias term b *seems to lead to the SVMs with less support vectors*. This fact can often be very useful for both the data (information) compression and the speed of learning. Below, we present an iterative learning algorithm for the classification SVMs (86a) with an explicit bias b, subjected to the equality constraint (87a)[14]. The problem to

[13]Note that for a classification model in (86a), we usually take the sign of $f(\mathbf{x})$ but this is of lesser importance now.

[14]The same procedure is developed for the regression SVMs but due to the space constraints we do not go into these details here. However we give some relevant hints for the regression SVMs with bias b.

solve is,

$$\min \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{88a}$$

such that

$$y_i[\mathbf{w}^T\Phi(\mathbf{x}_i) + b] \geqslant 1, \quad i = 1, \ldots, l\,, \tag{88b}$$

which can be transformed into its dual form by minimizing the primal Lagrangian

$$L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l} \alpha_i \big\{ y_i \big[\mathbf{w}^T\Phi(\mathbf{x}_i) + b\big] - 1 \big\} \tag{89}$$

in respect to $\mathbf{w}$ and b by using $\partial L_p/\partial \mathbf{w}_o = 0$ and $\partial L_p/\partial b = 0$, i.e. by exploiting

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i) \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{90}$$

The standard change to a dual problem is to substitute $\mathbf{w}$ from (90) into the primal Lagrangian and this leads to a dual Lagrangian problem below,

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i y_i b \tag{91}$$

subject to the box constraints (92) and, in a standard SVMs formulation, also to the equality constraint (93) as given below

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, l \tag{92}$$

and

$$\sum_{i=1}^{l} \alpha_i y_i = 0\,. \tag{93}$$

There are *three major avenues* (procedures, algorithms) possible in solving the dual problem (91), (92) and (93).

The *first one* is the standard SVMs algorithm which imposes the equality constraint (93) during the optimization and in this way ensures that the solution never leaves a feasible region. In this case the last term in (91) vanishes. (Note that in a standard SMO iterative scheme for training SVMs the minimal number of training data points enforcing (93) and ensuring staying in a feasible region

is two). After the dual problem is solved, the bias term is calculated by using *unbounded* Lagrange multipliers α_i (*Kecman*, 2001; *Schölkopf*, *Smola*, 2002) as follows

$$b = \frac{1}{\#UnboundSVecs}\Big(\sum\nolimits_{i=1}^{\#UnboundSVecs}\big(y_i - \mathbf{w}^T\Phi(\mathbf{x}_i)\big)\Big). \tag{94}$$

Below, we show *two more possible ways* how the ISDA works for the SVMs containing an explicit bias term too. In *the second method*, the cost function (88a) is augmented with the term $0.5kb^2$ (where $k \geqslant 0$). Note that this step is related to solving the dual problem by penalty method where a decrease in k leads to the stronger imposing of an equality constraint (see comments below). After forming the primal Lagrangian as well as using

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \Phi(\mathbf{x}_i) \quad \text{and} \quad b = \frac{1}{k}\sum_{i=1}^{l}\alpha_i y_i$$

(coming from $\partial L_p/\partial \mathbf{w}_o = 0$ and $\partial L_p/\partial b = 0$) one arrives at the dual problem not containing the explicit bias term b. Actually, the optimization of a dual Lagrangian is reformulated for the SVMs with a bias term b by applying "tiny" change only to the original matrix $\mathbf{K}$. For the *nonlinear classification* problems ISDA stands for an iterative solving of the following linear system

$$\mathbf{K}_k\,\boldsymbol{\alpha} = \mathbf{1}_l \tag{95a}$$

such that

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, l\,, \tag{95b}$$

where

$$K_k(\mathbf{x}_i, \mathbf{x}_j) = y_i y_j\Big(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{k}\Big),$$

$\mathbf{1}_l$ is an l-dimensional unity vector and C is a penalty factor equal to infinity for a hard margin classifier. Note that during the updates of α_i, the bias term b must not be used because it is implicitly incorporated within the $\mathbf{K}_k$ matrix. Only after the solution vector α in (95) is found, the bias b should be calculated either by using *unbounded* Lagrange multipliers α_i as given in (94), or by implementing the equality constraint from $\partial L_p/\partial b = 0$ and given as

$$b = \frac{1}{k}\sum\nolimits_{j=1}^{\#SVecs}\alpha_i y_i. \tag{96}$$

Note, however, that all the Lagrange multipliers, meaning both bounded (clipped to C) and unbounded (smaller than C) must be used in (96). Both equations, (94) and (96), result in the same value for the bias b. Thus, using the SVMs with an explicit bias term means that in the ISDA proposed above original kernel is changed, i.e., another kernel function is used. This means that the alpha values will be different for each k chosen, and so will be the value for b. However, the final SVM as given in (86) is produced by original kernels. Namely, $f(\mathbf{x})$ is obtained by adding the sum of weighted original kernel values and corresponding bias term b.

The second method presented above and aimed at an extending of the ISDA to the SVMs with a bias term b is related to the classic (quadratic) penalty methods for solving optimization problems with an equality constraint. Namely, the addition of $0.5kb^2$ to (88a) changes the last term of (91) to

$$\frac{1}{2k}\Big|\Big|\sum_{i=1}^{l}\alpha_i y_i\Big|\Big|_2^2$$

which is equivalent to applying a penalty parameter of $1/k$ to the L_2 norm of the equality constraint (93). As a result, for a large value of $1/k$, the solution will have a small L_2 norm of (93). In other words, as k approaches zero a bias b converges to the solution of the standard QP method that enforces the equality constraint. However, we do not use the ISDA with small parameter k values here, because the condition number of the matrix $\mathbf{K}_k$ increases as $1/k$ rises. Furthermore, the strict fulfilment of (93) may not be needed in obtaining a good SVM. Here, in classifying the MNIST data with Gaussian kernels, the value $k = 10$ proved to be a very good one justifying all the reasons for its introduction (fast learning, small number of support vectors and good generalization).

The third method in implementing the ISDA for SVMs with the bias term b is to work with original cost function (88a) and keep imposing the equality constraint during the iterations as suggested in (*Veropoulos*, 2001). The learning starts with $b = 0$ and after each epoch the bias b is updated by applying a secant method as follows

$$b^k = b^{k-1} - \omega^{k-1}\frac{b^{k-1} - b^{k-2}}{\omega^{k-1} - \omega^{k-2}} \tag{97}$$

where $\omega = \sum_{i=1}^{l}\alpha_i y_i$ represents the value of an equality constraint after each epoch. In the case of the regression SVMs, equation (97) is used by implement-

ing the corresponding regression's equality constraint, namely $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)$. This is different from (*Veropoulos*, 2001) where an iterative update after each data pair is proposed. In our SVMs regression experiments such an updating led to an unstable learning. Also, in an addition to changing expression for ω, both the $\mathbf{K}$ matrix, which is now $(2l, 2l)$ matrix, and the right hand side of (95a) which becomes $(2l, 1)$ vector, should be changed too and formed as given in (*Kecman, Vogt, Huang*, 2003).

Performance of an ISD Learning Algorithm and Comparisons

To measure the relative performance of different ISDAs, we ran all the algorithms with RBF Gaussian kernels on a MNIST dataset with 576-dimensional inputs (*Dong* et al, 2003), and compared the performance of our ISD algorithm with LIBSVM V2.4 (*Chang* et al, 2003) which is one of the fastest and the most popular SVM solvers at the moment based on the SMO type of an algorithm. The MNIST dataset consists of 60,000 training and 10,000 test data pairs. To make sure that the comparison is based purely on the nature of the algorithm rather than on the differences in implementation, our encoding of the algorithms are the same as LIBSVM's one in terms of caching strategy (LRU–Least Recent Used), data structure, heuristics for shrinking and stopping criterions. The only significant difference is that instead of two heuristic rules for selecting and updating two data points at each iteration step aiming at the maximal improvement of the dual objective function, our ISDA selects the worse KKT violator only and updates its α_i at each step.

Also, in order to speed up the LIBSVM's training process, we modified the original LIBSVM routine to perform faster by reducing the numbers of complete KKT checking without any deterioration of accuracy. All the routines were written and compiled in Visual C++ 6.0, and all simulations were run on a 2.4 GHz P4 processor PC with 1.5 Gigabyte of memory under the operating system Windows XP Professional. The shape parameter σ^2 of an RBF Gaussian kernel and the penalty factor C are set to be 0.3 and 10 (*Dong J.X.* et al, 2003). The stopping criterion τ and the size of the cache used are 0.01 and 250 Megabytes. The simulation results of different ISDA against both LIBSVM are presented in Tables 3 and 4, and in a Fig. 19. The first and the second column of the tables show the performance of the original and modified LIBSVM respectively. The last three columns show the results for single data point learning algorithms with various values of constant $1/k$ added to the kernel matrix in (95a). For $k = \infty$, ISDA is equivalent to the SVMs without bias term, and for $k = 1$,

Table 3. Simulation time for different algorithms

	LIBSVM original	**LIBSVM modified**	**Iterative single data algorithm (ISDA)** $k = 1$	$k = 10$	$k = \infty$
Class	Time(sec)	Time(sec)	Time(sec)	Time(sec)	Time(sec)
0	1606	885	800	**794**	1004
1	740	**465**	490	491	855
2	2377	1311	1398	**1181**	1296
3	2321	1307	1318	**1160**	1513
4	1997	1125	1206	**1028**	1235
5	2311	1289	1295	**1143**	1328
6	1474	818	808	**754**	1045
7	2027	1156	2137	**1026**	1250
8	2591	1499	1631	**1321**	1764
9	2255	1266	1410	**1185**	1651
Time Increase	**+95.3%**	**+10.3%**	**+23.9%**	**0**	**+28.3%**

Table 4. Number of support vectors for each algorithm

	LIBSVM original	**LIBSVM modified**	**Iterative single data algorithm (ISDA)** $k = 1$	$k = 10$	$k = \infty$
Class	# SV (BSV)	# SV (BSV)	# SV (BSV)	# SV (BSV)	# SV (BSV)
0	2172 (0)	2172 (0)	2162 (0)	2132 (0)	**2682** (0)
1	1440 (4)	1440 (4)	1429 (4)	1453 (4)	**2373** (4)
2	3055 (0)	3055 (0)	3047 (0)	3017 (0)	**3327** (0)
3	2902 (0)	2902 (0)	2888 (0)	2897 (0)	**3723** (0)
4	2641 (0)	2641 (0)	2623 (0)	2601 (0)	**3096** (0)
5	2900 (0)	2900 (0)	2884 (0)	2856 (0)	**3275** (0)
6	2055 (0)	2055 (0)	2042 (0)	2037 (0)	**2761** (0)
7	2651 (4)	2651 (4)	3315 (4)	2609 (4)	**3139** (4)
8	3222 (0)	3222 (0)	3267 (0)	3226 (0)	**4224** (0)
9	2702 (2)	2702 (2)	2733 (2)	2756 (2)	**3914** (2)
Average # of SV	**2574**	**2574**	**2639**	**2558**	**3151**

BSV = Bounded SVs

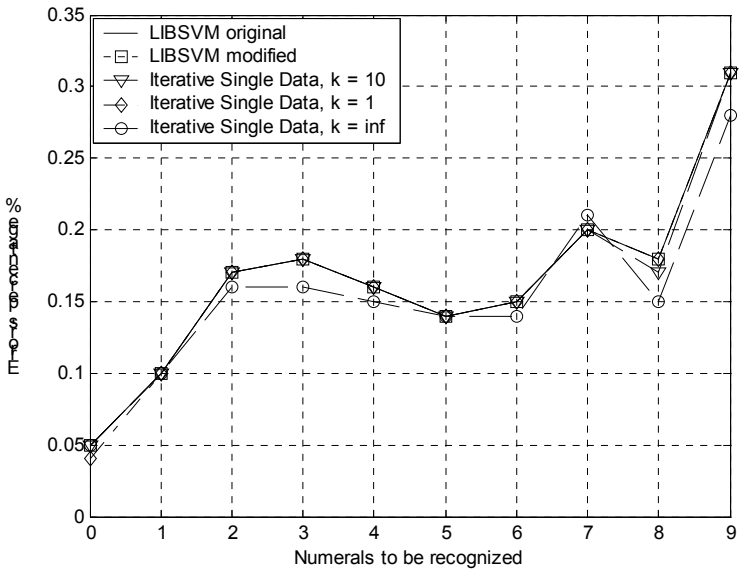

Figure 19. The percentage of errors on the test data

it is the same as the classification formulation proposed in (*Mangasarian* and *Musicant*, 1999).

Table 3 illustrates the running time for each algorithm. The ISDA with $k = 10$ was the quickest and required the shortest average time (T_{10}) to complete the training. The average time needed for the original LIBSVM is almost $2T_{10}$ and the average time for a modified version of LIBSVM is 10.3% bigger than T_{10}. This is contributed mostly to the simplicity of the ISD algorithm. One may think that the improvement achieved is minor, but it is important to consider the fact that approximately more than 50% of the CPU time is spent on the final checking of the KKT conditions in all simulations. During the checking, the algorithm must calculate the output of the model at each datum in order to evaluate the KKT violations. This process is unavoidable if one wants to ensure the solution's global convergence, i.e. that *all the data* do satisfy the KKT conditions with precision τ indeed. Therefore, the reduction of time spent

on iterations is approximately double the figures shown. Note that the ISDA slows down for $k < 10$ here. This is a consequence of the fact that with a decrease in k there is an increase of the condition number of a matrix $\mathbf{K}_k$, which leads to more iterations in solving (95). At the same time, implementing the no-bias SVMs, i.e., working with $k = \infty$, also slows the learning down due to an increase in the number of support vectors needed when working without bias b.

Table 4 presents the numbers of support vectors selected. For the ISDAs, the numbers reduce significantly when the explicit bias term b is included. One can compare the numbers of SVs for the case without the bias b ($k = \infty$) and the ones when an explicit bias b is used (cases with $k = 1$ and $k = 10$). Because identifying less support vectors speeds the overall training definitely up, the SVMs implementations with an explicit bias b are faster than the version without bias.

In terms of a generalization, or a performance on a test data set, all the algorithms had very similar results and this demonstrates that the ISDAs produce models that are as good as the standard QP, i.e., SMO based, algorithms. The percentages of the errors on the test data are shown in Fig. 19. Notice the extremely low error percentages on the test data sets for all numerals.

Conclusions

The seminar presents the basics of the standard SVMs models for solving the classification and regression problems first. Then, it also shows why and how, when positive definite kernels are used, the kernel AdaTron, sequential minimal optimization and Gauss-Seidel (i.e., successive over relaxation) algorithms are identical in their analytic form and numerical implementation. Till now, these facts were blurred mainly due to different pace in posing the learning problems and due to the "heavy" heuristics involved in an SMO implementation that shadowed an insight into the possible identity of the methods. It is shown that in the so-called no-bias SVMs, both the KA and the SMO procedure are the coordinate ascent based methods. Based on these equalities the novel ISDA for training SVMs is devised. Finally, due to the many ways how all the three algorithms (KA, SMO and GS i.e., SOR) can be implemented there may be some differences in their overall behavior. The introduction of the relaxation parameter $0 < \omega < 2$ will speed up the algorithm. The exact optimal value ω_{opt} is problem dependent.

Next, we demonstrate the use, the calculation and the effect of incorporating an explicit bias term b in the SVMs trained with the ISDA. The simulation results show that models generated by ISDAs (either with or without the bias term b) are as good as the standard QP (i.e., SMO) based algorithms in terms of a generalization performance. Moreover, ISDAs with an appropriate k value are faster than the standard SMO algorithms on large scale classification problems ($k = 10$ worked particularly well in all our simulations using Gaussian RBF kernels, however it may be that the "best" k value is problem dependent). This is due to both the simplicity of ISDAs and the decrease in the number of SVs chosen after an inclusion of an explicit bias b in the model. The simplicity of ISDAs is the consequence of the fact that the equality constraints (87) do not need to be fulfilled during the training stage. In this way, the *second order heuristics is avoided* during the iterations. Thus, the ISDA is an extremely good tool for solving large scale SVMs problems containing huge training data sets because it is faster than, and it delivers 'same' generalization results as, the other standard QP (SMO) based algorithms. The fact that an introduction of an explicit bias $b$means solving the problem with different kernel suggests that it may be hard to tell in advance for what kind of previously unknown multivariable decision (regression) function the models with bias b may perform better, or may be more suitable, than the ones without it. As it is often the case, the real experimental results, their comparisons and the new theoretical developments should probably be able to tell one day. As for the single data based learning approach presented here, the future work will focus on the development of even faster training algorithms.

References

1. *Abe, S.*, 2004. Support Vector Machines for Pattern Classification, Springer-Verlag, London.
2. *Aizerman, M.A., E.M. Braverman, and L.I. Rozonoer*, 1964. Theoretical foundations of the potential function method in pattern recognition learning // *Automation and Remote Control*, **25**, pp.821–837.
3. *Anlauf, J.K., Biehl, M.*, 1989. The AdaTron — an adaptive perceptron algorithm // *Europhysics Letters*, **10**(7), pp.687–692.
4. *Bartlett, P.L., A. Tewari*, 2004. Sparseness vs estimating conditional probabilities: Some asymptotic results // (submitted for a publication and taken from the P.L. Bartlett's site).

5. *Chang, C., Lin, C.*, 2003. LIBSVM: a library for support vector machines, Available at:
URL: http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

6. *Cherkassky, V., F. Mulier*, 1998. Learning From Data: Concepts, Theory and Methods, John Wiley & Sons, New York, NY.

7. *Cortes, C.*, 1995. Prediction of Generalization Ability in Learning Machines. PhD Thesis, Department of Computer Science, University of Rochester, NY.

8. *Cortes, C., Vapnik, V.* 1995. Support Vector Networks // *Machine Learning*, **20**: 273–297.

9. *Cristianini, N., Shawe-Taylor, J.*, 2000, An introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, Cambridge, UK.

10. *Dong, X., Krzyzak, A., Suen, C.Y.*, 2003. A fast SVM training algorithm // *International Journal of Pattern Recognition and Artificial Intelligence*, vol. **17**, No.3, pp.367–384.

11. *Drucker, H., C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik*. 1997. Support vector regression machines // *Advances in Neural Information Processing Systems*, **9**, pp.155–161, MIT Press, Cambridge, MA.

12. *Eisenhart, C.*, 1962. Roger Joseph Boscovich and the Combination of Observationes // *Actes International Symposium on R. J. Boskovic*, pp.19–25, Belgrade–Zagreb–Ljubljana, YU.

13. *Evgeniou, T., Pontil, M., Poggio, T.*, 2000. Regularization networks and support vector machines // *Advances in Computational Mathematics*, **13**, pp.1–50.

14. *Friess, T., R.F. Harrison*, 1998. Linear programming support vectors machines for pattern classification and regression estimation and the set reduction algorithm, TR RR-706, University of Sheffield, Sheffield, UK.

15. *Friess, T.-T., Cristianini, N., Campbell, I.C.G.*, 1998. The Kernel-Adatron: a Fast and Simple Learning Procedure for Support Vector Machines // *In Shavlik, J.*, editor, *Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, pp.188–196, San Francisco, CA.

16. *Girosi, F.*, 1997. An Equivalence Between Sparse Approximation and Support Vector Machines // *AI Memo* 1606, MIT.

17. *Graepel, T., R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K.-R. Müller, K. Obermayer, R. Williamson*, 1999. Classification on proximity data with LP–machines // *Proc. of the* 9th *Intl. Conf. on Artificial NN*, ICANN 99, Edinburgh, 7–10 Sept.

18. *Hadzic, I., V. Kecman*, 1999. Learning from Data by Linear Programming // *NZ Postgraduate Conference Proceedings*, Auckland, Dec. 15–16.

19. *Huang T.-M., Kecman V.*, 2004. Bias Term b in SVMs Again // *Proc. of the* 12th *European Symposium on Artificial Neural Networks*, ESANN 2004, pp.441–448, Bruges, Belgium.

20. *Huang T.-M., V. Kecman, I. Kopriva*, 2006. Kernel Based Algorithms for Mining Huge Data Sets, Supervised, Semi-supervised, and Unsupervised Learning, Series "Computational Intelligence", Springer-Verlag, in print.

21. *Huang, T.-M.*, 2006. Large-Scale Support Vector Machines and Semi-Supervised Learning Algorithms, PhD thesis, The University of Auckland, School of Engineering, Auckland, NZ.

22. *Kecman, V., Arthanari T., Hadzic I.*, 2001 LP and QP Based Learning From Empirical Data // *IEEE Proceedings of IJCNN 2001*, Vol.4, pp.2451–2455, Washington, DC.

23. *Kecman, V.*, 2001. Learning and Soft Computing, Support Verctor machines, Neural Networks and Fuzzy Logic Models, The MIT Press, Cambridge, MA, the book's web site is:
URL: http://www.support-vector.ws

24. *Kecman, V., Hadzic I.*, 2000. Support Vectors Selection by Linear Programming // *Proceedings of the International Joint Conference on Neural Networks* (IJCNN 2000), Vol.5, pp.193–198, Como, Italy.

25. *Kecman, V., Vogt, M., Huang, T.-M.*, 2003. On the Equality of Kernel AdaTron and Sequential Minimal Optimization in Classification and Regression Tasks and Alike Algorithms for Kernel Machines // Proc. of ESANN 2003, 11th *European Symposium on Artificial Neural Networks*, Bruges, Belgium, downloadable from
URL: http://www.support-vector.ws

26. *Kecman, V., T.M. Huang, M. Vogt, 2005*. Iterative Single Data Algorithm for Training Kernel Machines from Huge Data Sets: Theory and Performance, Chapter in a Springer-Verlag book, "Support Vector Machines: Theory and Applications", ed. *L. Wang*.

27. *Lawson, C. I., Hanson, R. J.*, 1974. Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, N.J.

28. *Mangasarian, O.L., 1965*. Linear and Nonlinear Separation of Patterns by Linear Programming // *Operations Research*, **13**, pp.444–452.

29. *Mangasarian, O.L., Musicant, D.R.*, 1999. Successive Overrelaxation for Support Vector Machines // *IEEE Trans. Neural Networks*, **11**(4), pp.1003–1008.

30. *Mercer, J.*, 1909. Functions of positive and negative type and their connection with the theory of integral equations // *Philos. Trans. Roy. Soc. London*, A 209:415{446}.

31. *Ostrowski, A.M.*, 1966. Solutions of Equations and Systems of Equations, 2nd ed., Academic Press, New York.

32. *Osuna, E., R. Freund, F. Girosi*. 1997. Support vector machines: Training and applications // *AI Memo* 1602, Massachusetts Institute of Technology, Cambridge, MA.

33. *Platt, J.C.* 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

34. *Platt, J.C.*, 1999. Fast Training of Support Vector Machines using Sequential Minimal Optimization. Chapter 12 in Advances in Kernel Methods — Support Vector Learning, edited by *B.Schölkopf, C. Burges, A. Smola*, The MIT Press, Cambridge, MA.

35. *Poggio, T., Mukherjee, S., Rifkin, R., Rakhlin, A., Verri, A.*, 2001. *b*, CBCL Paper # 198/AI Memo #2001–011, Massachusetts Institute of Technology, Cambridge, MA.

36. *Schölkopf, B., Smola, A.*, 2002. Learning with Kernels — Support Vector Machines, Optimization, and Beyond, The MIT Press, Cambridge, MA.

37. *Smola, A., Schölkopf, B.* 1997. On a Kernel-based Method for Pattern Recognition, Regression, Approximation and Operator Inversion. GMD Technical Report No.1064, Berlin.

38. *Smola, A., T.T. Friess, B. Schölkopf*, 1998, Semiparametric Support Vector and Linear Programming Machines, NeuroCOLT2 Technical Report Series, NC2-TR-1998-024, also In: *Advances in Neural Information Processing Systems 11*

39. *Steinwart, I.*, 2003. Sparseness of support vector machines // *Journal of Machine Learning Research*, **4** (2003), pp.1071–1105.

40. Support Vector Machines Web Site:
URL: http://www.kernel-machines.org/

41. *Suykens, J.A.K., T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle*, 2002. Least Squares Support Vector Machines, World Scientific Pub. Co., Singapore.

42. *V. Vapnik and A. Chervonenkis*, 1964. A note on one class of perceptrons // *Automation and Remote Control*, **25**.

43. *Vapnik, V.N., A.Y. Chervonenkis*, 1968. On the uniform convergence of relative frequencies of events to their probabilities // *Doklady Akademii Nauk USSR*, **181**, (4) (In Russian).

44. *Vapnik, V.* 1979. Estimation of Dependences Based on Empirical Data. Nauka, Moscow. (In Russian, English translation: 1982, Springer Verlag, New York).

45. *Vapnik, V.N., A.Y. Chervonenkis*, 1989. The necessary and sufficient conditions for the consistency of the method of empirical minimization [in Russian] // *Yearbook*

of the Academy of Sciences of the USSR on Recognition, Classification, and Forecasting, **2**, 217–249, Moscow, Nauka, (English transl.: The necessary and sufficient condititons for the consistency of the method of empirical minimization. Pattern Recognitio and Image Analysis, **1**, 284–305, 1991)

46. *Vapnik, V.N.*, 1995. The Nature of Statistical Learning Theory, Springer Verlag Inc, New York, NY.
47. *Vapnik, V., S. Golowich, A. Smola*. 1997. Support vector method for function approximation, regression estimation, and signal processing // In: *Advances in Neural Information Processing Systems* **9**, MIT Press, Cambridge, MA.
48. *Vapnik, V.N.*, 1998. Statistical Learning Theory, J.Wiley & Sons, Inc., New York, NY.
49. *Veropoulos, K.*, 2001. Machine Learning Approaches to Medical Decision Making, PhD Thesis, The University of Bristol, Bristol, UK.
50. *Vogt, M.*, 2002. SMO Algorithms for Support Vector Machines without Bias, Institute Report, IAT, TU Darmstadt, Darmstadt, Germany.
URL: http://w3.rt.e-technik.tu-darmstadt.de/∼vogt/
51. *Vogt, M., V. Kecman*, 2005. Active-Set Methods for Support Vector Machines, Chapter in a Springer-Verlag book, “Support Vector Machines: Theory and Applications”, ed. *L. Wang*.

Vojislav KECMAN, PhD, MSc, Dipl.-Ing., Associate Professor and a Head of Dynamics and Control Systems Group, Department of Mechanical Engineering, the University of Auckland, Auckland, New Zealand. He is an Associate Editor in IEEE Transaction on Neural Networks and a member of the Advisory Board of Interdisciplinary Journal of Information, Knowledge, and Management.
His newest research interests are machine learning from huge data sets in high dimensional spaces by support vector machines and/or neural networks; fuzzy logic systems; kernel machines; pattern recognition and/or multivariate function approximation; knowledge modelling and knowledge acquiring; bioinformatics; text categorization. Other research interests: neural networks based adaptive control systems; system dynamics modelling, simulation, identification and control; unified approach to the modelling of physically different systems; singularly perturbed control systems.

He has visited many European, North American and Pacific region universities, research institutions or industries and upon the invitations delivered seminars from the fields of research at Harvard University, MIT, University of California Santa Barbara, Rutgers University, TU Bremen, TU Dresden, TH Darmstadt, FH Heilbronn, Research Institute for Knowledge Systems in Maastricht, Melbourne University, The University of Auckland, Universities of Belgrade, Novi Sad and BanjaLuka.
Dr. Kecman has published more than 100 journal and conference papers, monographs, books or other bound publications.

Воислав КЕЦМАН, руководитель группы динамики и систем управления, доцент кафедры машиностроения Университета в Окленде, Новая Зеландия. Он является членом редакционной коллегии журнала IEEE Transaction on Neural Networks, а также членом консультативного совета журнала Interdisciplinary Journal of Information, Knowledge, and Management.
Его основные научные интересы лежат в настоящее время в таких областях, как обучение машин на сверхбольших наборах данных в многомерных пространствах с использованием машин опорных векторов и/или нейронных сетей; нечеткие системы; kernel machines; распознавание образов и/или аппроксимация функций многих переменных; модели знаний и извлечение знаний; биоинформатика; категоризация текста. К числу других областей научных интересов д-ра Кецмана относятся также нейросетевые адаптивные системы управления; моделирование, идентификация и управление для динамических систем; единый подход к моделированию систем различной физической природы; системы управления с сингулярными возмущениями.
Д-ра Кецмана приглашали для проведения семинаров по тематике его научных интересов многие университеты и исследовательские институты Европы, Северной Америки и Тихоокеанского региона.
Д-р Кецман является автором более 100 журнальных статей, докладов на конференциях и книг.

НАУЧНАЯ СЕССИЯ МИФИ–2007

НЕЙРОИНФОРМАТИКА–2007

**IX ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ

Часть 1

Оригинал-макет подготовлен *Ю. В. Тюменцевым*
с использованием издательского пакета LaTeX 2_ε
и набора PostScript–шрифтов *PSCyr*

Подписано в печать 11.11.2006 г. Формат 60×84 1/16
Печ. л. 11, 25. Тираж 190 экз. Заказ №

*Московский инженерно-физический институт
(государственный университет)
Типография МИФИ
115409, Москва, Каширское шоссе, 31*