Математические методы распознавания образов

Курс лекций

МГУ, ВМиК, кафедра «Математические методы прогнозирования»
© Местецкий Леонид Моисеевич, 2002–2004

1 Задача распознавания образов

1.1 Предмет распознавания образов

Распознавание образов – это научная дисциплина, целью которой является классификация объектов по нескольким категориям или классам. Объекты называются образами.

Классификация основывается на прецедентах.

Прецедент – это образ, правильная классификация которого известна.

Прецедент – ранее классифицированный объект, принимаемый как образец при решении задач классификации. Идея принятия решений на основе прецедентности – основополагающая в естественно-научном мировоззрении.

Будем считать, что все объекты или явления разбиты на конечное число классов. Для каждого класса известно и изучено конечное число объектов – прецедентов. Задача распознавания образов состоит в том, чтобы отнести новый распознаваемый объект к какому-либо классу.

Задача распознавания образов является основной в большинстве интеллектуальных систем. Рассмотрим примеры интеллектуальных компьютерных систем.

1) Машинное зрение. Это системы, назначение которых состоит в получении изображения через камеру и составление его описания в символьном виде (какие объекты присутствуют, в каком взаимном отношении находятся и т.д.).
2) Символьное распознавание – это распознавание букв или цифр.
    a. Optical Character Recognition (OCR);
    b. Ввод и хранение документов;
    c. Pen Computer;
    d. Обработка чеков в банках;
    e. Обработка почты.
3) Диагностика в медицине.
    a. Маммография, рентгенография;
    b. Постановка диагноза по истории болезни;
    c. Электрокардиограмма.
4) Геология.
5) Распознавание речи.
6) Распознавание в дактилоскопии (отпечатки пальцев), распознавание лица, подписи, жестов.

1.2 Признаки и классификаторы

Измерения, используемые для классификации образов, называются признаками. *Признак – это некоторое количественное измерение объекта произвольной природы.* Совокупность признаков, относящихся к одному образу, называется *вектором признаков*. Вектора признаков принимают значения в *пространстве признаков*. В рамках задачи распознавания считается, что каждому образу ставится в соответствие единственное значение вектора признаков и наоборот: каждому значению вектора признаков соответствует единственный образ.

Классификатором или решающим правилом называется правило отнесения образа к одному из классов на основании его вектора признаков.

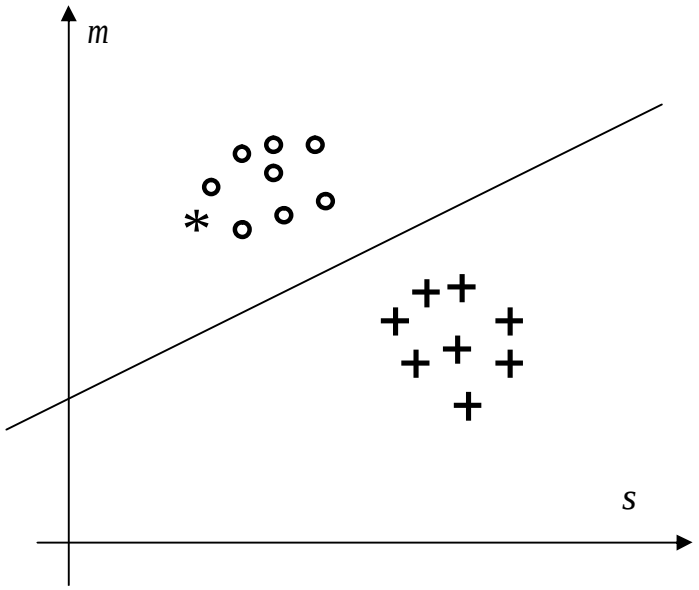


Рис.1. Распределение векторов признаков прецедентах класса A (кружки) и класса B (крестики). Признаки - средние значения и средние отклонения яркости в образах. Прямая линия разделяет вектора из разных классов.

Пример 1. Иллюстрация понятий признаков и классификатора и идеи распознавания (классификации). Рассмотрим задачу диагностики печени по результатам инструментального исследования. Доброкачественные (левый рисунок – класс A) и злокачественные (правый рисунок – класс B) изменения дают разную картину. Предположим, что имеется несколько препаратов в базе данных, про которые известна их принадлежность к классам A и B (правильная классификация). Очевидно, что образцы отличаются интенсивностью точек изображения. В качестве вектора признаков выберем пару: среднее значение (m) и среднеквадратичное отклонение (s) интенсивности в изображении. На рис.1 представлены изображения этих образов в пространстве признаков. Точки, соответствующие прецедентам разных классов, разделяются прямой линией. Классификация неизвестного образа (соответствующая точка изображена звездочкой) состоит в проверке положения точки относительно этой разделяющей прямой.

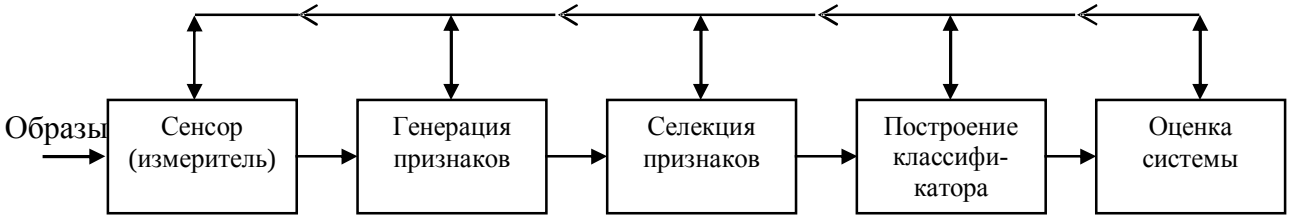


Рис.2. Основные элементы построения системы распознавания образов (классификации)

Практическая разработка системы классификации осуществляется по следующей схеме (рис.2). В процессе разработки необходимо решить следующие вопросы.

1. Как выбрать вектора признаков?
   Задача генерации признаков – это выбор тех признаков, которые с достаточной полнотой (в разумных пределах) описывают образ.

2. Какие признаки наиболее существенны для разделения объектов разных классов?
   Задача селекции признаков – отбор наиболее информативных признаков для классификации.
3. Как построить классификатор?
   Задача построения классификатора – выбор решающего правила, по которому на основании вектора признаков осуществляется отнесение объекта к тому или иному классу.
4. Как оценить качество построенной системы классификации?
   Задача количественной оценки системы (выбранные признаки + классификатор) с точки зрения правильности или ошибочности классификации.

1.3 Классификация с обучением и без обучения

В зависимости от наличия или отсутствия прецедентной информации различают задачи распознавания с обучением и без обучения. Задача распознавания на основе имеющегося множества прецедентов называется классификацией с обучением (или с учителем).

В том случае, если имеется множество векторов признаков, полученных для некоторого набора образов, но правильная классификация этих образов неизвестна, возникает задача разделения этих образов на классы по сходству соответствующих векторов признаков. Эта задача называется *кластеризацией* или распознаванием без обучения.

Пример 2. Рассмотрим съемку со спутника и классификацию поверхности по отраженной энергии (рис.3). На рисунке изображены снимок из космоса (слева) и результат кластеризации векторов признаков, рассчитанных для различных элементов изображения (справа). Распределение образов, изображенных точками (x_1,x_2) по классам осуществляется на основе анализа «скоплений» этих точек в пространстве признаков.

Пример 3. Рассмотрим другой пример распознавания образов – в общественных (социальных) науках. Целью задачи является построение системы классификации государств для определения необходимости гуманитарной поддержки со стороны международных организаций. Необходимо выявить закономерности связей между различными, объективно измеряемыми параметрами, например, связь между ВНП, уровнем грамотности и уровнем детской смертности. В данном случае страны можно представить трехмерными векторами, а задача заключается в построении меры сходства этих векторов и дальнейшем построении схемы кластеризации (выбора групп) по этой мере.

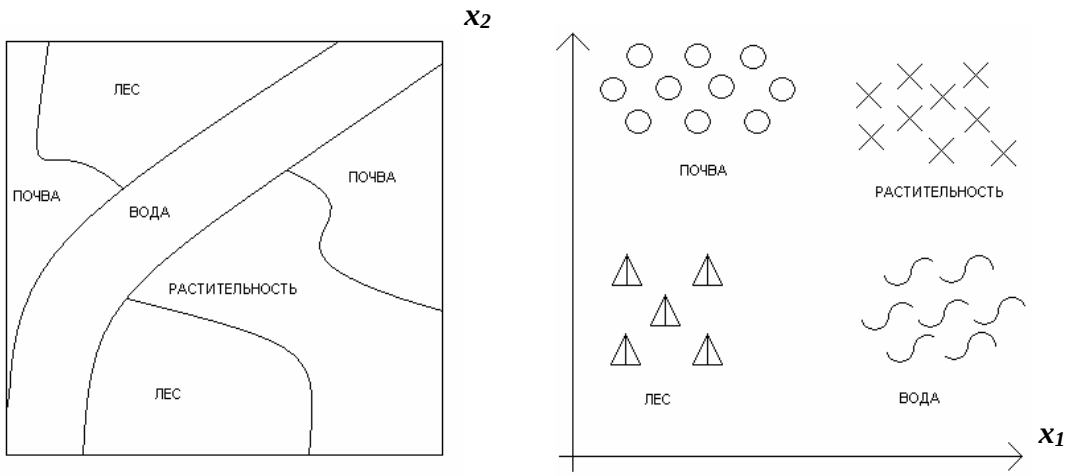


Рис.3. Изображение различных типов поверхности и кластеризация соответствующих векторов признаков.

1.4 Формальная постановка задачи классификации

Будем использовать следующую модель задачи классификации.

Ω – множество объектов распознавания (пространство образов).

$w: v \in \Omega$ – объект распознавания (образ).

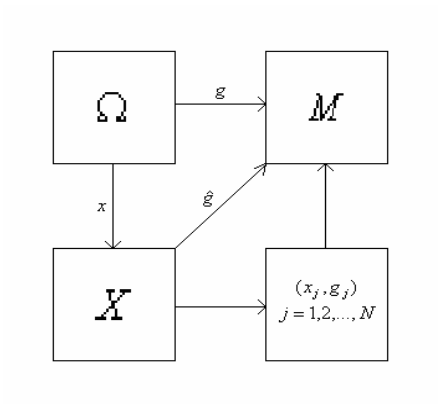
$g(w): \Omega \to \mathrm{M}$, $\mathrm{M} = \{1,2,...,m\}$ – индикаторная функция, разбивающая пространство образов на Ω на m непересекающихся классов $\Omega^1, \Omega^2, ..., \Omega^m$. Индикаторная функция неизвестна наблюдателю.

X – пространство наблюдений, воспринимаемых наблюдателем (пространство признаков).

$x(w): \Omega \to \mathrm{X}$ – функция, ставящая в соответствие каждому объекту w точку $x(w)$ в пространстве признаков. Вектор $x(w)$ - это образ объекта, воспринимаемый наблюдателем. В пространстве признаков определены непересекающиеся множества точек $Ki \subset \mathrm{X}$, $i = 1,2...,m$, соответствующих образам одного класса.

$\hat{g}(x): \mathrm{X} \to \mathrm{M}$ – решающее правило – оценка для $g(w)$ на основании $x(w)$, т.е. $\hat{g}(x) = \hat{g}(x(w))$.

Пусть $x_j = x(w_j)$, $j = 1,2...,N$ – доступная наблюдателю информация о функциях $g(w)$ и $x(w)$, но сами эти функции наблюдателю неизвестны. Тогда (g_j, x_j), $j = 1,2...,N$ – есть множество прецедентов.



Задача заключается в построении такого решающего правила $\hat{g}(x)$, чтобы распознавание проводилось с минимальным числом ошибок.

Обычный случай – считать пространство признаков евклидовым, т.е. $\mathrm{X} = R^l$. Качество решающего правила измеряют частотой появления правильных решений. Обычно его оценивают, наделяя множество объектов Ω некоторой вероятностной мерой. Тогда задача записывается в виде $\min P\{\hat{g}(x(w)) \neq g(w)\}$.

2 Классификация на основе байесовской теории решений

2.1 Байесовский подход

Байесовский подход исходит из статистической природы наблюдений. За основу берется предположение о существовании вероятностной меры на пространстве образов, которая либо известна, либо может быть оценена. Цель состоит в разработке такого классификатора, который будет правильно определять наиболее вероятный класс для пробного образа. Тогда задача состоит в определении “наиболее вероятного” класса.

Задано M классов $\Omega_1, \Omega_2, ..., \Omega_M$, а также $P(\Omega_i|x)$, $i = 1,2,...,M$ – вероятность того, что неизвестный образ, представляемый вектором признаков x, принадлежит классу Ω_i.

$P(\Omega_i|x)$ *называется апостериорной вероятностью,* поскольку задает распределение индекса класса после эксперимента (*a posteriori* – т.е. после того, как значение вектора признаков *x* было получено).

Рассмотрим случай двух классов Ω_1 и Ω_2. Естественно выбрать решающее правило таким образом: объект относим к тому классу, для которого апостериорная вероятность выше. Такое правило классификации по максимуму апостериорной вероятности называется Байесовским: если $P(\Omega_1|x) > P(\Omega_2|x)$, то x классифицируется в Ω_1, иначе в Ω_2. Таким образом, для Байесовского решающего правила необходимо получить апостериорные вероятности $P(\Omega_i|x)$, $i = 1,2$. Это можно сделать с помощью формулы Байеса.

Формула Байеса, полученная Т. Байесом в 1763 году, позволяет вычислить апостериорные вероятности событий через априорные вероятности и функции правдоподобия.

Пусть $A_1, A_2, ..., A_n$ – полная группа несовместных событий. $\bigcup_{i=1}^{n} A_i = \Omega$. $A_i \cap A_j = \varnothing$, при $i \neq j$. Тогда апостериорная вероятность имеет вид:

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^{n} P(A_i)P(B|A_i)},$$

где $P(A_i)$ – априорная вероятность события A_i, $P(B|A_i)$ – условная вероятность события B при условии, что произошло событие A_i.

Рассмотрим получение апостериорной вероятности $P(\Omega|x)$, зная $P(\Omega)$ и $P(x|\Omega)$.

$P(AB) = P(A|B)P(B)$, $P(AB) = P(B|A)P(A)$

$P(A|B)P(B) = P(B|A)P(A)$

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Если $P(A)$ и $P(A|B)$ описываются плотностями $p(x)$ и $p(x|B)$, то

$$P(B|x) = \frac{p(x|B)P(B)}{p(x)} \Rightarrow P(\Omega_i|x) = \frac{p(x|\Omega_i)P(\Omega_i)}{p(x)}.$$

При проверке классификации сравнение $P(\Omega_1|x)$ и $P(\Omega_2|x)$ эквивалентно сравнению $p(x|\Omega_1)P(\Omega_1)$ и $p(x|\Omega_2)P(\Omega_2)$. В случае, когда $P(\Omega_1|x) = P(\Omega_2|x)$, считается, что мера множества x равна нулю.

Таким образом, задача сравнения по апостериорной вероятности сводится к вычислению величин $P(\Omega_1)$, $P(\Omega_2)$, $p(x|\Omega_1)$, $p(x|\Omega_2)$. Будем считать, что у нас достаточно данных для определения вероятности принадлежности объекта каждому из классов $P(\Omega_i)$, $i=1,2$. Такие вероятности называются *априорными* вероятностями классов. А также будем считать, что известны функции распределения вектора признаков для каждого класса $P(x|\Omega_i)$, $i=1,2$. Они называются функциями правдоподобия x по отношению к Ω_i. Если априорные вероятности и функции правдоподобия неизвестны, то их можно оценить методами математической статистики на множестве прецедентов. Например, $P(\Omega_i)\approx\frac{N_i}{N}$, где N_i – число прецедентов из Ω_i, $i=1,2$. N – общее число прецедентов. $P(x|\Omega_i)$ может быть приближено гистограммой распределения вектора признаков для прецедентов из класса Ω_i.

Итак, Байесовский подход к статистическим задачам основывается на предположении о существовании некоторого распределения вероятностей для каждого параметра. Недостатком этого метода является необходимость постулирования как существования априорного распределения для неизвестного параметра, так и знание его формы.

2.2 Ошибка классификации

Определение. *Вероятность* $P_e = P(x\in R_2,\Omega_1)+P(x\in R_1,\Omega_2)$ *называется ошибкой классификации,* $R_1=\{x: P(\Omega_1)p(x|\Omega_1)>P(\Omega_2)p(x|\Omega_2)\}$, $R_2=\{x: P(\Omega_1)p(x|\Omega_1)<P(\Omega_2)p(x|\Omega_2)\}$ *– области решения (*$\Omega_1\cap\Omega_2=\varnothing$*).*

Теорема. *Байесовский классификатор является оптимальным по отношению к минимизации вероятности ошибки классификации.*

Доказательство. Рассмотрим ошибку классификации:

$$P_e = P(x\in R_2,\Omega_1)+P(x\in R_1,\Omega_2)=$$
$$=P(\Omega_1)\int_{R_2}p(x|\Omega_1)dx+P(\Omega_2)\int_{R_1}p(x|\Omega_2)dx=$$
$$=P(\Omega_1)\left(1-\int_{R_1}p(x|\Omega_1)dx\right)+P(\Omega_2)\int_{R_1}p(x|\Omega_2)dx=$$
$$=P(\Omega_1)-P(\Omega_1)\int_{R_1}p(x|\Omega_1)dx+P(\Omega_2)\int_{R_1}p(x|\Omega_2)dx=$$

Учитывая формулу Байеса: $p(x|\Omega_i)=\frac{P(\Omega_i|x)p(x)}{P(\Omega_i)}$, $i=1,2$ получим:

$$=P(\Omega_1)-P(\Omega_1)\int_{R_1}\frac{P(\Omega_1|x)p(x)}{P(\Omega_1)}dx+P(\Omega_2)\int_{R_1}\frac{P(\Omega_2|x)p(x)}{P(\Omega_2)}dx=$$
$$=P(\Omega_1)-\int_{R_1}P(\Omega_1|x)p(x)dx+\int_{R_1}P(\Omega_2|x)p(x)dx=$$
$$=P(\Omega_1)-\int_{R_1}p(x)\big(P(\Omega_1|x)-P(\Omega_2|x)\big)dx$$

Таким образом, минимум достигается, когда $R_1=\{x: P(\Omega_1|x)>P(\Omega_2|x)\}$. R_2 выбирается из остальных точек.

ч.т.д.

Данная теорема была доказана для двух классов Ω_1 и Ω_2. Обобщим ее на M классов.

Пусть вектор признаков x относится к классу Ω_i, если $P(\Omega_i|x) > P(\Omega_j|x)$, при $i \neq j$, $i = 1,2,...,M$, $j = 1,2,...,M$. Соответственно необходимо доказать, что данное правило минимизирует вероятность ошибки классификации. Для доказательства следует воспользоваться формулой правильной классификации $P_r = 1 - P_e$.

Доказательство. Воспользуемся формулой правильной классификации $P_r = 1 - P_e$.

$$P_r = P(x \in R_1, \Omega_1) + P(x \in R_2, \Omega_2) + ... + P(x \in R_l, \Omega_l) =$$

$$= \sum_{i=1}^{l} P(x \in R_i|\Omega_i) P(\Omega_i) =$$

$$= \sum_{i=1}^{l} P(\Omega_i) \int_{R_i} p(x|\Omega_i) dx =$$

$$= P(\Omega_1) \left(1 - \sum_{i=2}^{l} \int_{R_i} p(x|\Omega_1) dx \right) + \sum_{i=2}^{l} P(\Omega_i) \int_{R_i} p(x|\Omega_i) dx =$$

$$= P(\Omega_1) - \sum_{i=2}^{l} \left[P(\Omega_1) \int_{R_i} p(x|\Omega_1) dx - P(\Omega_i) \int_{R_i} p(x|\Omega_i) dx \right] =$$

Учитывая формулу Байеса: $p(x|\Omega_i) = \dfrac{P(\Omega_i|x)\, p(x)}{P(\Omega_i)}$, $i = 1,2,...,l$ получим:

$$= P(\Omega_1) - \sum_{i=2}^{l} \left[P(\Omega_1) \int_{R_i} \frac{P(\Omega_1|x)\, p(x)}{P(\Omega_1)} dx - P(\Omega_i) \int_{R_i} \frac{P(\Omega_i|x)\, p(x)}{P(\Omega_i)} dx \right] =$$

$$= P(\Omega_1) - \sum_{i=2}^{l} \left[\int_{R_i} P(\Omega_1|x)\, p(x) dx - \int_{R_i} P(\Omega_i|x)\, p(x) dx \right] =$$

$$= P(\Omega_1) - \sum_{i=2}^{l} \int_{R_i} p(x) \left[P(\Omega_1|x) - P(\Omega_i|x) \right] dx$$

Таким образом, максимум достигается, когда $P(w_1|x) < P(w_i|x)$. Аналогично для всех $j = 1,2,...,l$ максимум достигается, когда $R_i = \{x : P(w_j|x) < P(w_i|x)\}$.

ч.т.д.

2.3 Минимизация среднего риска

Вероятность ошибки классификации – не всегда лучший критерий проверки классификатора. В том случае, когда цена ошибок различного типа существенно различается, лучше использовать другой критерий качества классификации – *минимум среднего риска*.

Рассмотрим задачу классификации по M классам. R_j, $j = 1,2,...,M$ – области предпочтения классов v_j. Предположим, что вектор x из класса Ω_k лежит в R_i, $i \neq k$, т.е. классификация происходит с ошибкой. Свяжем с этой ошибкой штраф l_{ki} называемый потерями в результате того, что объект из класса Ω_k был принят за объект из класса Ω_i. Обозначим через $L = \| l_{ki} \|$ матрицу потерь.

Определение. *Выражение* $r_k = \sum_{i=1}^{M} l_{ki} P\{x \in R_i | \Omega_k\} = \sum_{i=1}^{M} l_{ki} \int_{R_i} p(x|\Omega_k) dx$ *называется риском при классификации объекта класса* Ω_k.

Определение. *Выражение* $r = \sum_{i=1}^{M} r_k P(\Omega_k)$ *называется общим средним риском.*

Теперь мы можем поставить задачу о выборе классификатора, минимизирующего этот риск. Преобразуем выражение общего среднего риска:

$$r = \sum_{i=1}^{M} r_k P(\Omega_k) = \sum_{k=1}^{M} P(\Omega_k) \sum_{i=1}^{M} l_{ki} \int_{R_i} p(x|\Omega_k) dx =$$

$$= \sum_{i=1}^{M} \left(\sum_{k=1}^{M} P(\Omega_k) l_{ki} \int_{R_i} p(x|\Omega_k) dx \right) =$$

$$= \sum_{i=1}^{M} \int_{R_i} \left(\sum_{k=1}^{M} l_{ki} p(x|\Omega_k) P(\Omega_k) \right) dx$$

Из этого выражения видно, что риск минимален, когда каждый из интегралов в данной сумме минимален, т.е. $x \in R_i$, если $l_i < l_j$, при $i \neq j$, где $l_i = \sum_{k=1}^{M} l_{ki} p(x|\Omega_k) P(\Omega_k)$, $l_j = \sum_{k=1}^{M} l_{kj} p(x|\Omega_k) P(\Omega_k)$.

Пример. Рассмотрим ситуацию радиолокационной разведки. На экране радара отражаются не только цели, но и помехи. Такой помехой может служить стая птиц, которую можно принять за небольшой самолет. В данном случае это двухклассовая задача.

Рассмотрим матрицу штрафов: $L = \| l_{ki} \|$, $i = 1,2$, $k = 1,2$. l_{ki} – это штраф за принятие объекта из класса k за объект класса i. Тогда

$$l_1 = l_{11} p(x|\Omega_1) P(\Omega_1) + l_{21} p(x|\Omega_2) P(\Omega_2)$$

$$l_2 = l_{12} p(x|\Omega_1) P(\Omega_1) + l_{22} p(x|\Omega_2) P(\Omega_2)$$

Пусть x относится у классу Ω_1, если $l_1 < l_2$, т.е.

$$l_{11} p(x|\Omega_1) P(\Omega_1) + l_{21} p(x|\Omega_2) P(\Omega_2) < l_{12} p(x|\Omega_1) P(\Omega_1) + l_{22} p(x|\Omega_2) P(\Omega_2)$$

$$(l_{21} - l_{22}) p(x|\Omega_2) P(\Omega_2) < (l_{12} - l_{11}) p(x|\Omega_1) P(\Omega_1)$$

Т.к. $l_{21} > l_{22}$ и $l_{12} > l_{11}$, то

$$\frac{p(x|\Omega_1)}{p(x|\Omega_2)} > \frac{l_{21} - l_{22}}{l_{12} - l_{11}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}$$

Стоящее в левой части неравенства отношение $l_{12} = \frac{p(x|\Omega_1)}{p(x|\Omega_2)}$ называется отношением правдоподобия. Неравенство описывает условие предпочтения класса Ω_1 классу Ω_2.

Пример. Рассмотрим двухклассовую задачу, в которой для единственного признака x известна плотность распределения:

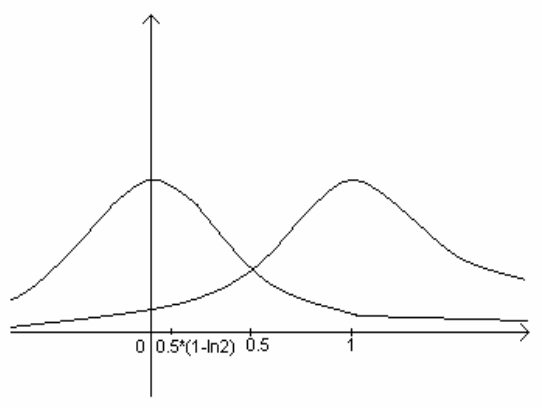
$$p(x|\Omega_1) = \frac{1}{\sqrt{p}} \exp\left(-x^2\right)$$

$$p(x|\Omega_2) = \frac{1}{\sqrt{p}} \exp\left(-(x-1)^2\right)$$

Пусть, также, априорные вероятности $P(\Omega_1) = P(\Omega_2) = \frac{1}{2}$.

Задача – вычислить пороги для

a) минимальной вероятности ошибки
b) минимального риска при матрице риска $L = \begin{pmatrix} 0 & 0.5 \\ 1 & 0 \end{pmatrix}$.



Решение задачи a):

$$p(x|\Omega_1)P(\Omega_1) = p(x|\Omega_2)P(\Omega_2)$$
$$\exp\left(-x^2\right) = \exp\left(-(x-1)^2\right)$$
$$-x^2 = -(x-1)^2$$
$$\hat{x} = \frac{1}{2}$$

Решение задачи b):

$$\frac{p(x|\Omega_1)}{p(x|\Omega_2)} = \frac{l_{21} - l_{22}}{l_{12} - l_{11}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}$$
$$\frac{\exp\left(-x^2\right)}{\exp\left(-(x-1)^2\right)} = \frac{1-0}{0.5-0} = \frac{1/2}{1/2} = 2$$
$$\exp\left(-x^2\right) = 2\exp\left(-(x-1)^2\right)$$
$$-x^2 = \ln 2 - (x-1)^2$$
$$\tilde{x} = \frac{1}{2}(1 - \ln 2)$$

Пример. Рассмотрим двухклассовую задачу с Гауссовскими плотностями распределения $p(x|\Omega_1) \cong N(0, s^2)$ и $p(x|\Omega_2) \cong N(1, s^2)$ и матрицей потерь $L = \begin{pmatrix} 0 & l_{12} \\ l_{21} & 0 \end{pmatrix}$.

Задача – вычислить порог для проверки отношения правдоподобия.

Решение. С учетом матрицы потерь отношение правдоподобия

$$\frac{p(x|\Omega_1)}{p(x|\Omega_2)} = \frac{l_{21} - l_{22}}{l_{12} - l_{11}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}$$

запишется в виде

$$\frac{p(x|\Omega_1)}{p(x|\Omega_2)} = \frac{l_{21}}{l_{12}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}$$

Запишем плотности распределения

$$p(x|\Omega_1) = \frac{1}{\sqrt{2p}\,s}\exp\left(-\frac{x^2}{2s^2}\right);\ p(x|\Omega_2) = \frac{1}{\sqrt{2p}\,s}\exp\left(-\frac{(x-1)^2}{2s^2}\right)$$
$$\frac{p(x|\Omega_1)}{p(x|\Omega_2)} = \frac{l_{21}}{l_{12}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)} = \exp\left(\frac{(x-1)^2}{2s^2} - \frac{x^2}{2s^2}\right)$$
$$x^2 - (x-1)^2 = -2s^2 \ln\left(\frac{l_{21}}{l_{12}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}\right)$$

$$x = \frac{1}{2} - s^2 \ln\left(\frac{l_{21}}{l_{12}} \cdot \frac{P(\Omega_2)}{P(\Omega_1)}\right)$$

Пример. Рассмотрим двух классовую задачу с матрицей потерь $L = \|l_{ki}\|$, $k = 1,2$, $i = 1,2$. Пусть e_1 – вероятность ошибки, соответствующая вектору из класса Ω_1 и e_2 – вероятность ошибки, соответствующая вектору из класса Ω_2. Задача – найти средний риск.

Решение.

$$r = \sum_{i=1}^{M} r_k P(\Omega_k) =$$

$$= \sum_{i=1}^{M}\left(\sum_{k=1}^{M} P(\Omega_k)\, l_{ki} \int_{R_i} p(x|\Omega_k)dx\right) =$$

$$= l_{11}(1-e_1)P(\Omega_1) + l_{12}e_1P(\Omega_1) + l_{21}e_2P(\Omega_2) + l_{22}(1-e_2)P(\Omega_2) =$$

$$= l_{11}P(\Omega_1) + (l_{12} - l_{11})e_1P(\Omega_1) + (l_{21} - l_{22})e_2P(\Omega_2) + l_{22}P(\Omega_2)$$

Пример. Доказать, что в задаче классификации по M классам, вероятность ошибки классификации ограничена: $P_e = \frac{M-1}{M}$.

Указание: показать, что $\max\limits_{i=1,\ldots,M} P(v_i|x) \geq \frac{1}{M}$.

2.4 Дискриминантные функции и поверхности решения

Минимизация риска и вероятности ошибки эквивалентны разделению пространства признаков на M областей. Если области R_i и R_j смежные, то они разделены поверхностью решения в многомерном пространстве. Для случая минимизации вероятности ошибки поверхность решения задается уравнением:

$$P(\Omega_i|x) - P(\Omega_j|x) = 0$$

В данном уравнении приходится оперировать с вероятностями. Иногда вместо вероятностей удобнее работать с функцией от вероятности:

$$g_i(x) = f\left(P(\Omega_i|x)\right),$$

где функция f монотонно возрастает.

Определение. *Функция* $g_i(x) = f\left(P(\Omega_i|x)\right)$ *называется дискриминантной функцией.*

Таким образом, поверхность решения будет задаваться уравнением:

$$g_i(x) - g_j(x) = 0,\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,M,\ i \neq j.$$

Для задачи классификации по вероятности ошибки или риску не всегда удается вычислить вероятности. В этом случае бывает более предпочтительно вычислить разделяющую поверхность на основе другой функции стоимости. Такие подходы дают решения, субоптимальные по отношению к Байесовской классификации.

2.5 Байесовский классификатор для нормального распределения

Распределение Гаусса очень широко используется по причине вычислительного удобства и адекватности во многих случаях. Рассмотрим многомерную плотность нормального распределения $N(m_i, \Sigma_i)$:

$$p(x|\Omega_i) = \frac{1}{(2p)^{l/2}|\Sigma_i|^{1/2}} \cdot \exp\left(-\frac{1}{2}\frac{(x-m_i)^{\mathrm{T}}}{\Sigma_i \cdot (x-m_i)}\right),\ i = 1,2,...,M$$

где $m_i = \mathrm{E}[X]$ – математическое ожидание случайной величины x в классе Ω_i,

Σ_i – матрица ковариации размерности $l \times l$ для класса Ω_i, $\Sigma_i = \mathrm{E}\left[(x-m_i)(x-m_i)^{\mathrm{T}}\right]$,

$|\Sigma_i|$ – определитель матрицы ковариации.

Здесь x, m_i – это вектора столбцы, а x^T, m_i^T - вектора-строки.

2.5.1 Квадратичная поверхность решения

На основе этих данных необходимо построить байесовский классификатор. Рассмотрим логарифмическую дискриминантную функцию:

$$g_i(x) = \ln(P(\Omega_i|x)) =$$

$$= \ln(p(x|\Omega_i)P(\Omega_i)) =$$

$$= \ln p(x|\Omega_i) + \ln P(\Omega_i) =$$

$$= -\frac{1}{2}\frac{(x-m_i)}{\Sigma_i}(x-m_i)^{\mathrm{T}} + \ln P(\Omega_i) + \ln\frac{1}{(2p)^{l/2}|\Sigma_i|^{1/2}} =$$

$$= -\frac{1}{2}\frac{(x-m_i)}{\Sigma_i}(x-m_i)^{\mathrm{T}} + \ln P(\Omega_i) - \frac{l}{2}\ln(2p) - \frac{1}{2}\ln|\Sigma_i| =$$

$$= -\frac{1}{2}\frac{(x-m_i)}{\Sigma_i}(x-m_i)^{\mathrm{T}} + \ln P(\Omega_i) + C_i, \text{ где } C_i = -\frac{l}{2}\ln(2p) - \frac{1}{2}\ln|\Sigma_i|$$

Эта функция представляет собой квадратичную форму. Следовательно, разделяющая поверхность $g_i(x) - g_j(x) = 0$ является гиперповерхностью второго порядка. Поэтому Байесовский классификатор является квадратичным классификатором.

Пример. Пусть $l = 2$, $\Sigma_i = \begin{pmatrix} s_i^2 & 0 \\ 0 & s_i^2 \end{pmatrix}$. Тогда $\frac{1}{\Sigma_i} = \begin{pmatrix} \frac{1}{s_i^2} & 0 \\ 0 & \frac{1}{s_i^2} \end{pmatrix}$.

$$g_i(x) = -\frac{1}{2s_i^2}(x_1^2 + x_2^2) + \frac{1}{s_i^2}(m_{i1}x_1 + m_{i2}x_2) - \frac{1}{2s_i^2}(m_{i1}^2 + m_{i2}^2) + \ln(P(\Omega_i)) + C_i$$

Разделяющей поверхностью является коническое сечение.

Пример. Пусть $P(\Omega_1) = P(\Omega_2)$, $m_1 = (0,0)$, $m_2 = (1,0)$, $\Sigma_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.15 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.25 \end{pmatrix}$.

Тогда $\frac{1}{\Sigma_1} = \begin{pmatrix} 10 & 0 \\ 0 & 20/3 \end{pmatrix}$, $\frac{1}{\Sigma_2} = \begin{pmatrix} 5 & 0 \\ 0 & 4 \end{pmatrix}$. Найдем поверхность решения.

$$g_1(x) = -\frac{1}{2}(x_1, x_2)\begin{pmatrix} 10 & 0 \\ 0 & 20/3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \ln P(\Omega_1) - \ln(2p) + \frac{1}{2}\ln\frac{200}{3} =$$

$$= -\frac{1}{2}\left(10x_1^2 + \frac{20}{3}x_2^2\right) + \ln P(\Omega_1) - \ln(2p) + \frac{1}{2}\ln\frac{200}{3}$$

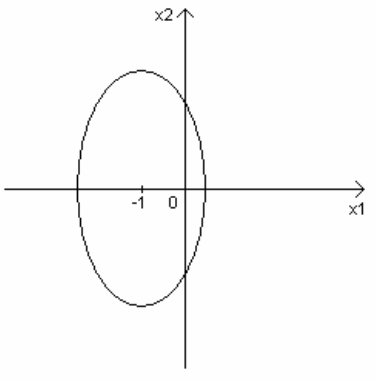
$$g_2(x) = -\frac{1}{2}(x_1 - 1, x_2)\begin{pmatrix} 5 & 0 \\ 0 & 4 \end{pmatrix}\begin{pmatrix} x_1 - 1 \\ x_2 \end{pmatrix} + \ln P(\Omega_2) - \ln(2p) + \frac{1}{2}\ln 20 =$$

$$= -\frac{1}{2}\left(5(x_1-1)^2 + 4x_2^2\right) + \ln P(\Omega_2) - \ln(2p) + \frac{1}{2}\ln 20$$

$$g_1(x) - g_2(x) = -\frac{1}{2}\left(10x_1^2 + \frac{20}{3}x_2^2 - 5(x_1-1)^2 - 4x_2^2\right) + \frac{1}{2}\left(\ln\frac{200}{3} - \ln 20\right) =$$

$$= -\frac{1}{2}\left(5(x_1+1)^2 + \frac{8}{3}x_2^2\right) + 5 + \frac{1}{2}\ln\frac{10}{3}$$

Т.к. $g_1(x) - g_2(x) = 0$, то $-\frac{1}{2}\left(5(x_1+1)^2 + \frac{8}{3}x_2^2\right) + 5 + \frac{1}{2}\ln\frac{10}{3} = 0$



$$5(x_1+1)^2 + \frac{8}{3}x_2^2 = 10 + \ln\frac{10}{3}$$

$$\frac{(x_1+1)^2}{8/3} + \frac{x_2^2}{5} = \frac{3}{40}\left(10 + \ln\frac{10}{3}\right)$$

$$\frac{(x_1+1)^2}{\left(2\sqrt{2/3}\right)^2} + \frac{x_2^2}{\left(\sqrt{5}\right)^2} = \frac{3}{40}\left(10 + \ln\frac{10}{3}\right)$$

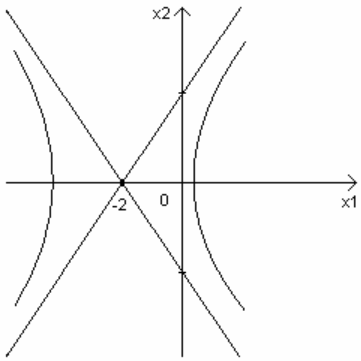
– эллипс центром в точке $(-1,0)$.

Пример. Пусть $P(\Omega_1) = P(\Omega_2)$, $m_1 = (0,0)$, $m_2 = (1,0)$, $\Sigma_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.15 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.1 \end{pmatrix}$.

Тогда $\frac{1}{\Sigma_1} = \begin{pmatrix} 10 & 0 \\ 0 & 20/3 \end{pmatrix}$, $\frac{1}{\Sigma_2} = \begin{pmatrix} 20/3 & 0 \\ 0 & 10 \end{pmatrix}$. Найдем поверхность решения.

Из предыдущего примера:

$$g_1(x) = -\frac{1}{2}\left(5(x_1-1)^2 + 4x_2^2\right) + \ln P(\Omega_2) - \ln(2p) + \frac{1}{2}\ln 20$$



$$g_2(x) = -\frac{1}{2}(x_1-1, x_2)\begin{pmatrix} 20/3 & 0 \\ 0 & 10 \end{pmatrix}\begin{pmatrix} x_1-1 \\ x_2 \end{pmatrix} + \ln P(\Omega_2) + \frac{1}{2}\ln\frac{200}{3}$$

$$g_1(x) - g_2(x) = -\frac{1}{2}\left(10x_1^2 + \frac{20}{3}x_2^2 - \frac{20}{3}(x_1-1)^2 - 10x_2^2\right) =$$

$$= -\frac{1}{2}\left(\frac{10}{3}x_1^2 - \frac{10}{3}x_2^2 + \frac{40}{3}x_1 - \frac{20}{3}\right) =$$

$$= -\frac{1}{2}\cdot\frac{10}{3}\left(x_1^2 - x_2^2 + 4x_1 - 2\right) = -\frac{5}{3}\left((x_1+2)^2 - x_2^2 - 6\right)$$

Т.к. $g_1(x) - g_2(x) = 0$, то $-\frac{5}{3}\left((x_1+2)^2 - x_2^2 - 6\right) = 0$.

$(x_1+2)^2 - x_2^2 = 6$ – гипербола с центром в точке $(-2,0)$.

2.5.2 Линейная поверхность решения

Условие остается тем же: $p(x|\Omega_i) = \frac{1}{(2p)^{1/2}\cdot|\Sigma_i|}\cdot\exp\left(-\frac{1}{2}\frac{x-m_i}{\Sigma_i}(x-m_i)^{\mathrm{T}}\right)$, $i = 1,2,...,M$.

В предыдущем пункте мы получили квадратичную форму:

$$h_i(x) = \ln\left(p(x|\Omega_i)P(\Omega_i)\right) =$$

$$= \ln p(x|\Omega_i) + \ln P(\Omega_i) =$$

$$= -\frac{1}{2}\frac{x-m_i}{\Sigma_i}(x-m_i)^{\mathrm{T}} + \ln P(\Omega_i) + C_i, \text{ где } C_i = \ln\frac{1}{(2p)^{1/2}\left|\Sigma_i\right|^{1/2}}.$$

Пусть $\Sigma_i = \Sigma_j$, тогда

$$h_i(x) = -\frac{1}{2}\left[\frac{x}{\Sigma_i}x^{\mathrm{T}} - \frac{m_i}{\Sigma_i}x^{\mathrm{T}} - \frac{x}{\Sigma_i}m_i^{\mathrm{T}} + \frac{m_i}{\Sigma_i}m_i^{\mathrm{T}}\right] + \ln P(\Omega_i) + C_i =$$

$$= -\frac{1}{2}\left[\frac{x}{\Sigma_i}x^{\mathrm{T}} - 2\frac{m_i}{\Sigma_i}x^{\mathrm{T}} + \frac{m_i}{\Sigma_i}m_i^{\mathrm{T}}\right] + \ln P(\Omega_i) + C_i =$$

$$= -\frac{1}{2}\left[K_i(x) - 2W_i x^{\mathrm{T}} + W_i m_i^{\mathrm{T}}\right] + \ln P(\Omega_i) + C_i =$$

$$= -\frac{1}{2}K_i(x) + L_i(x) + C_i, \text{ где } L_i(x) = W_i x^{\mathrm{T}} + W_{i0};\ W_i = \frac{m_i}{\Sigma_i};$$

$$W_{i0} = \ln P(\Omega_i) - \frac{1}{2}W_i m_i^{\mathrm{T}}$$

При $\Sigma_i = \Sigma_j$ можно сравнивать только $L_i(x)$ и $L_j(x)$. Таким образом, при $\Sigma_i = \Sigma_j$ мы получили линейную поверхность решения.

2.5.3 Линейная поверхность решения с диагональной матрицей ковариации

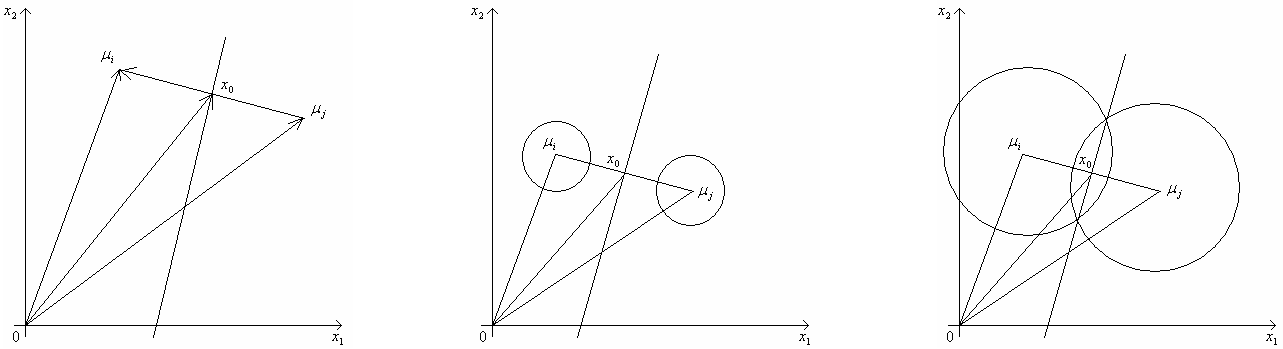
Рассмотрим случай, когда матрица Σ диагональная с одинаковыми элементами: $\Sigma = \begin{pmatrix} s^2 & 0 \\ 0 & s^2 \end{pmatrix}$. Тогда $L_i(x)$ имеет вид: $L_i(x) = \frac{1}{s^2}m_i^{\mathrm{T}}x + W_{i0}$;

$$L_{ij}(x) = L_i(x) - L_j(x) = W^{\mathrm{T}}(x - x_0) = 0,$$

где $W = m_i - m_j$, $x_0 = \frac{1}{2}(m_i + m_j) - s^2\frac{m_i - m_j}{\left\|m_i - m_j\right\|^2}\ln\frac{P(\Omega_i)}{P(\Omega_j)}$ В данном случае под нормой понимается евклидова норма. Поверхностью решения является гиперплоскость, проходящая через точку x_0.

Если $P(\Omega_i) = P(\Omega_j)$, то x_0 – это середина вектора $\overline{m_i m_j}$.

Т.к. $L_{ij}(x) = 0$, то $W^{\mathrm{T}}(x - x_0) = (m_i - m_j)^{\mathrm{T}}(x - x_0) = 0$. Следовательно, поверхность решения ортогональна $\overline{m_i m_j}$.



Пример. Рассмотрим пример разделяющей поверхности решения для двухклассовой задачи с нормальным распределением. Поверхность решения лежит ближе к m_i, если

$P(\Omega_i) < P(\Omega_j)$. Соответственно, поверхность решения лежит ближе к m_j, если $P(\Omega_i) > P(\Omega_j)$. Также, если s^2 мало по отношению к $\|m_i - m_j\|$, то положение поверхности решения не очень чувствительно к изменению $P(\Omega_i)$ и $P(\Omega_j)$. Последнее справедливо, т.к. вектора лежат в малых окрестностях m_i и m_j, поэтому изменение гиперплоскости их затрагивает не сильно. В центре изображен случай малого, а справа случай большого s^2.

2.5.4 Линейная поверхность решения с недиагональной матрицей ковариации

В этом случае уравнение:

$$L_{ij}(x) = L_i(x) - L_j(x) = W^{\mathrm{T}}(x - x_0) = 0$$

будет иметь несколько иные параметры:

$$W = \frac{m_i - m_j}{\Sigma} \text{ и } x_0 = \frac{1}{2}(m_i + m_j) - \frac{m_i - m_j}{\|m_i - m_j\|^2_{\Sigma^{-1}}}$$

В данном случае под нормой понимается так называемая Σ^{-1} норма x, которая имеет вид: $\|x\|_{\Sigma^{-1}} = \left(x^{\mathrm{T}}\Sigma^{-1}x\right)^{1/2}$. Для такой нормы поверхность решения не ортогональна вектору $\overline{m_i m_j}$, Но она ортогональна его образу при преобразовании $\Sigma^{-1}\left(m_i - m_j\right)$.

2.5.5 Классификаторы по минимуму расстояния

Будем рассматривать равновероятные классы с одинаковой матрицей ковариации. Тогда $\Sigma_1 = \Sigma_2 = ... = \Sigma_n = \Sigma$ и выражение

$$L_i(x) = -\frac{1}{2}(x - m_i)\Sigma_i^{-1}(x - m_i)^{\mathrm{T}} + \ln P(\Omega_i) + C_i$$

примет вид

$$L_i(x) = -\frac{1}{2}(x - m_i)\Sigma^{-1}(x - m_i)^{\mathrm{T}}$$

(т.к. логарифм и константа сократятся).

Классификатор по минимуму расстояния с диагональной матрицей ковариации. Рассмотрим случай, когда матрица Σ диагональная с одинаковыми элементами: $\Sigma = \begin{pmatrix} s^2 & 0 \\ 0 & s^2 \end{pmatrix}$. Тогда максимизация $L_i(x)$ влечет минимизацию евклидового расстояния, определяемое выражением $d_E = \|x - m_i\|$. В данном случае будет считаться, что объект относится к данному классу, если он близок в смысле евклидового расстояния.

Классификатор по минимуму расстояния с недиагональной матрицей ковариации. В этом случае максимизация $L_i(x)$ влечет минимизацию *расстояния Махалонобиса*, определяемого выражением $d_M = \left((x - m_i)^{\mathrm{T}}\Sigma^{-1}(x - m_i)\right)^{1/2}$.

Т.к. матрица ковариации является симметрической, ее можно представить в виде:

$$\Sigma = \Phi \cdot \Lambda \cdot \Phi^{\mathrm{T}},$$

где $\Phi^{\mathrm{T}} = \Phi^{-1}$, а Λ – диагональная матрица с собственными значениями матрицы Σ на диагонали. Матрица Φ имеет столбцы, соответствующие собственным векторам матрицы Σ:

$$\Phi = (n_1, n_2, ..., n_l)$$

Таким образом, получаем линию равноудаленных точек x:

$$(x-m_i)^{\mathrm{T}}\cdot\Phi\cdot\Lambda^{-1}\cdot\Phi^{\mathrm{T}}(x-m_i)=C^2$$

Пусть $x'=\Phi^{\mathrm{T}}x$. Тогда координатами x' являются $n_k^{\mathrm{T}}x$, $k=1,2,...,l$, т.е. проекции x на собственные вектора. Другими словами, мы получили координаты в новой системе, у которой оси определяются собственными векторами $n_k x$, $k=1,2,...,l$. Тогда последнее уравнение преобразуется в уравнение эллипсоида в новой системе координат:

$$\frac{\left(x_1'-m_{i1}'\right)^2}{l_1}+\frac{\left(x_2'-m_{i2}'\right)^2}{l_2}+...+\frac{\left(x_l'-m_{il}'\right)^2}{l_l}=C^2$$

При $l=2$ центр эллипса находится в точке $m_i=(m_{i1},m_{i2})$, а главные оси лежат по собственным векторам и имеют длины $2\sqrt{l_1}C$ и $2\sqrt{l_2}C$ соответственно.

Пример. Рассмотрим двумерный двухклассовый случай классификации двух нормально распределенных векторов с ковариационной матрицей $\Sigma=\begin{pmatrix}1.1 & 0.3\\ 0.3 & 1.9\end{pmatrix}$ и средними значениями $m_1=(0,0)^{\mathrm{T}}$ и $m_2=(3,3)^{\mathrm{T}}$.

Найдем Σ^{-1}:

$$|\Sigma|=1.1\cdot1.9-0.3^2=2.09-0.09=2$$

$$\Sigma^{-1}=\frac{1}{2}\begin{pmatrix}1.9 & -0.3\\ -0.3 & 1.1\end{pmatrix}=\begin{pmatrix}0.95 & -0.15\\ -0.15 & 0.55\end{pmatrix}$$

Классифицируем вектор $(1.0,2.2)$. Для этого посчитаем расстояние Махалонобиса:

$$d_m^2(m_1,x)=(x-m_1)^{\mathrm{T}}\Sigma^{-1}(x-m_1)=$$

$$=(1,2.2)\begin{pmatrix}0.95 & -0.15\\ -0.15 & 0.55\end{pmatrix}\begin{pmatrix}1\\ 2.2\end{pmatrix}=$$

$$=(0.95-0.33)+(-0.15+1.21)\cdot2.2=$$

$$=0.57+1\cdot0.6\cdot2.2=0.57+2.332=2.952$$

$$d_m^2(m_2,x)=(x-m_2)^{\mathrm{T}}\Sigma^{-1}(x-m_2)=$$

$$=(-1,-0.8)\begin{pmatrix}0.95 & -0.15\\ -0.15 & 0.55\end{pmatrix}\begin{pmatrix}-2\\ -0.8\end{pmatrix}=$$

$$=-(-1.9+0.12)-(0.3-0.44)\cdot0.8=$$

$$=3.56+0.112=3.672$$

Таким образом, хотя сама точка $(1.0,2.2)$ по евклидову расстоянию ближе к точке $(0,0)$, чем к точке $(3,3)$, но по расстоянию Махалонобиса она ближе к $(3,3)$.

Теперь вычислим главные оси эллипса с центром в точке $(0,0)$. Для этого найдем собственные значения:

$$\begin{vmatrix}1.1-l & 0.3\\ 0.3 & 1.9-l\end{vmatrix}=2.09-3l+l^2-0.09=l^2-3l+2=0$$

$$l_1=1,\ l_2=2.$$

Тогда собственные вектора (и направление главных осей эллипса) будут иметь вид:

$$V_1=\left(\frac{3}{\sqrt{10}},\frac{-1}{\sqrt{10}}\right)^{\mathrm{T}},\ V_2=\left(\frac{1}{\sqrt{10}},\frac{3}{\sqrt{10}}\right).$$

3 Линейный классификатор. Алгоритм персептрона

3.1 Линейная дискриминантная функция

Рассмотрим задачу построения линейной разделяющей гиперповерхности. Главным достоинством линейного классификатора является его простота и вычислительная эффективность.

Рассмотрим линейную дискриминантную функцию: $g(x) = W^{\mathrm{T}}x + W_0$, где $W^{\mathrm{T}} = (W_1, W_2, ..., W_l)^{\mathrm{T}}$ – весовой вектор, W_0 – порог. Поведение решения задается уравнением $g(x) = 0$. Пусть X_1 и X_2 – два конечных множества векторов признаков в евклидовом пространстве, относящихся к классу Ω_1 и Ω_2 соответственно, т.е X_1 принадлежит классу Ω_1 при $g(x) > 0$, а X_2 принадлежит классу Ω_2 при $g(x) < 0$.

Задача состоит в том, чтобы:

установить разделимость этих множеств;

найти разделяющую гиперплоскость.

Рассмотрим сначала в качестве примера двумерную задачу, когда образы представляются точками на плоскости.

Определение. *Множество, содержащее отрезок, соединяющий две произвольные внутренние точки, называется выпуклым.*

Определение. *Выпуклая оболочка – это минимальное выпуклое множество, содержащее данное.*

Утверждение 3.1. *Два множества на плоскости линейно разделимы тогда и только тогда, когда их выпуклые оболочки не пересекаются.*

Из этого утверждения получаем следующее правило проверки разделимости множеств на плоскости:

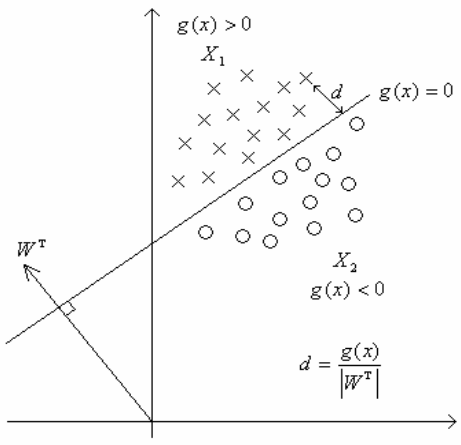
1) Построить выпуклые оболочки.

2) Проверить пересечение выпуклых оболочек. Если они не пересекаются, то множества разделимы.

Очевидно и правило, по которому можно найти разделяющую прямую:

1) Найти ближайшую пару точек в выпуклых оболочках обоих множеств.

2) Построить срединный перпендикуляр к отрезку, соединяющему эти точки. Этот перпендикуляр и будет разделяющей прямой.



Пусть размерность вектора признаков X и вектора коэффициентов W равна l. Рассмотрим «пополненные» вектора X', W' следующего вида: $(W')^{\mathrm{T}} = (W^{\mathrm{T}}, W_0)$ – пополненный весовой вектор, $(X')^{\mathrm{T}} = (X^{\mathrm{T}}, 1)$ – пополненный вектор признаков. Рассмотрим также в $(l+1)$-мерном пространстве однородную линейную функцию $g'(x) = ((W')^{\mathrm{T}}, (X')^{\mathrm{T}}) = \sum_{i=0}^{l} W_i \cdot x_i$.

Очевидно следующее

Утверждение 3.2. Множества X_1 и X_2 линейно разделимы в пространстве R^l дискриминантной функцией $g(x) = W^{\mathrm{T}}x + W_0$ тогда и только тогда, когда они разделимы в

пополненном пространстве R^{l+1} *однородной дискриминантной функцией* $g'(x) = \left((W')^{\mathrm{T}}, (X')^{\mathrm{T}}\right) = \sum_{i=0}^{l} W_i \cdot x_i$.

Далее будем рассматривать дискриминантные функции и вектора в пополненном пространстве.

Определение. *Множество* $\overline{X} = -X$ *называется симметричным множеством к множеству* X .

Утверждение 3.3. *Два замкнутых множества* X_1 *и* X_2 *разделимы тогда и только тогда, когда выпуклая оболочка множества* $X_1 \bigcup \overline{X}_2$ *не содержит начала координат.*

Доказательство. Пусть множества X_1 и X_2 разделимы. Тогда существует линейная функция $g(x)$ такая, что $g(x) > 0$ при $x \in X_1$ и $g(x) < 0$ при $x \in X_2$. Рассмотрим множество $X = X_1 \bigcup \overline{X}_2$, тогда $g(x) > 0$ при $x \in X$. Следовательно, $g(x) > 0$ для выпуклой линейной комбинации из X, а это означает, что $O \notin convX$, т.к. X – замкнутое. Здесь O обозначает начало координат.

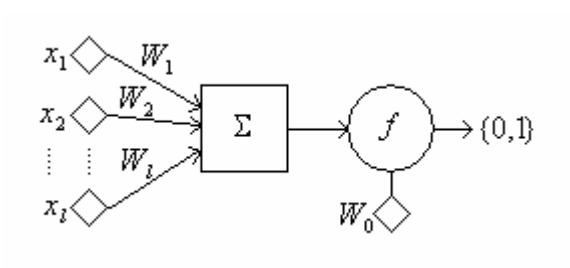
Пусть $O \notin convX$, и пусть $\tilde{x}$ – ближайшая к началу координат O точка из $convX$. Плоскость $(W, x) = 0$ с направляющим вектором $W = \tilde{x}$ не пересекает $convX$, а, значит, $(W, x) > 0$ на $x \in X$. Следовательно, $(W, x) < 0$ на $x \in X_2$.

ч. т. д.

3.2 Алгоритм персептрона

3.2.1 Математическая модель нейрона

В алгоритме персептрона в основу положен принцип действия нейрона. Обобщенная схема нейрона представлена на рисунке. Здесь $x_1, x_2, ..., x_l$ – компоненты вектора признаков $x = (x_1, x_2, ..., x_l)$; Σ – сумматор; $W_1, W_2, ..., W_l$ – синоптические веса; f – функция активации; W_0 – порог. Выходом сумматора является величина $\sum_{i=1}^{l} W_i x_i$, которая является входом (аргументом) функции активации. Значение функции активации вычисляется на основе определения знака суммы $\sum_{i=1}^{l} W_i x_i + W_0$:



$$f(v) = \begin{cases} 0 & при \quad v < 0 \\ 1 & при \quad v > 0 \end{cases}.$$

Таким образом, нейрон представляет собой линейный классификатор с дискриминантная функцией $g(x) = \sum_{i=1}^{l} W_i x_i + W_0$.

Тогда задача построения линейного классификатора для заданного множества прецедентов сводится к задаче обучения нейрона, т.е. подбора соответствующих весов $W_1, W_2, ..., W_l$ и порога W_0. Обучение состоит в коррекции синоптических весов и порога.

3.2.2 Алгоритм персептрона

Алгоритм персептрона представляет собой последовательную итерационную процедуру. Каждый шаг состоит в предъявлении нейрону очередного вектора-прецедента и коррекции весов W_i по результатам классификации. При этом прецеденты предъявляются циклически, т.е. после предъявления последнего снова предъявляется первый. Процесс обучения заканчивается, когда нейрон правильно классифицирует все прецеденты.

Обозначим W_t весовой вектор после t-й итерации, а x_t – прецедент, предъявляемый на t-й итерации.

Основной шаг алгоритма состоит в предъявлении прецедента очередного прецедента x_{t+1}:

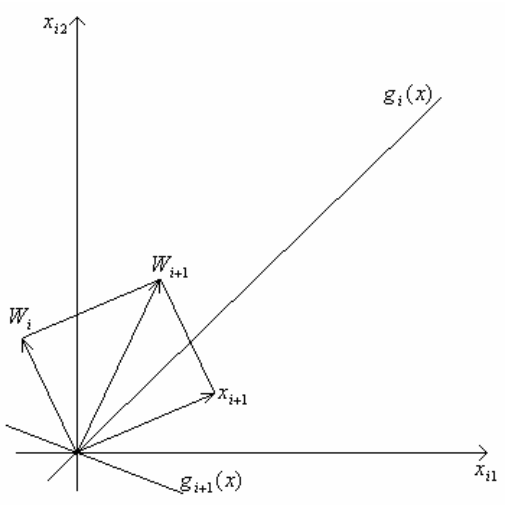
Если $x_{t+1} \in \Omega_1$ и $W_t x_{t+1} > 0$, то $W_{t+1} = W_t$;

Если $x_{t+1} \in \Omega_1$ и $W_t x_{t+1} \le 0$, то $W_{t+1} = W_t + x_{t+1}$;

Если $x_{t+1} \in \Omega_2$ и $W_t x_{t+1} < 0$, то $W_{t+1} = W_t$;

Если $x_{t+1} \in \Omega_2$ и $W_t x_{t+1} \ge 0$, то $W_{t+1} = W_t + x_{t+1}$.

На данном рисунке $g_t(x)$ – дискриминантная функция после t-го шага алгоритма; W_t – весовой вектор после t-го шага алгоритма.



3.2.3 Сходимость алгоритма персептрона

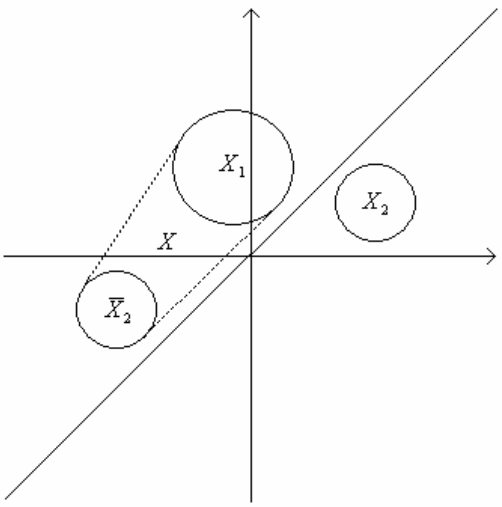
Основной вопрос, связанный с алгоритмом персептрона связан с его сходимостью. Конечен ли построенный итерационный процесс обучения?

Теорема Новикова. *Пусть* $\{x_i\}$ *– бесконечная последовательность векторов из двух непересекающихся замкнутых множеств* X_1 *и* X_2*; и пусть существует гиперплоскость, проходящая через начало координат и разделяющая* X_1 *и* X_2 *(не имеет с ними общих точек). Тогда при использовании алгоритма персептрона число коррекций весового вектора конечно.*

Доказательство. Пусть W^*- направляющий вектор разделяющей гиперплоскости (которая существует по условию). Не нарушая общности, будем считать, что он является единичным.

Пусть $X = conv(X_1 \cup \overline{X}_2)$, $\overline{X}_2$ в – симметричное к X_2 множество; $r_0 = r(0, X)$, где r – евклидово расстояние. Согласно утверждению 3.3 $(W^*, X) \ge r_0 > 0 \ \ \forall x \in X$.

Оценим (W_t, W^*).



Пусть W^* – еденичный вектор нормали, разделяющий X_1 и X_2.

$$(W^*, X) \ge r_0 \text{ при } x \in X_1$$

$$(W^*, X) \le -r_0 \text{ при } x \in X_2$$

Пусть W_t – весовой вектор после предъявления вектора x_t; $W_0 = 0$ – начальная итерация весового вектора ($|W^*| = 1$). Тогда, если $(W_t, x_{t+1}) > 0$, то коррекции не происходит. Иначе, если $(W_t, x_{t+1}) \le 0$, то коррекция: $W_{t+1} = W_t + x_{t+1}$

$$|W_{t+1}|^2 = |W_t|^2 + 2(x_{t+1}, W_t) + |x_{t+1}|^2 \le |W_t|^2 + D^2, \text{ т.к. } (x_{t+1}, W_t) \le 0 \text{ и } |x_{t+1}| \le \sup_{x \in X} |x| = D$$

Таким образом, к моменту t происходит k коррекций, то

$$|W_t|^2 \le k \cdot D^2, \text{ т.к. } |W_0| = 0 \qquad (*)$$

В начальный момент времени $(W_0, W^*) = 0$. Если в момент $i+1$ произошла коррекция, то

$$(W_{t+1}, W^*) = (W_0, W^*) + (x_{t+1}, W^*) \ge (W_t, W^*) + r_0$$

Если коррекция не происходит, то

$$(W_{t+1}, W^*) = (W_t, W^*)$$

Если к моменту t произошло k коррекций, то

$$(W_t, W^*) \ge k r_0$$

С другой стороны

$$(W_t, W^*) \le |W_t| \cdot |W^*| = |W_t|$$

Поэтому

$$|W_t| \ge k r_0 \qquad (**)$$

Из неравенств (*) и (**) следует:

$$k^2 r_0 \le |W_t|^2 \le k D^2 \;\Rightarrow\; k r_0 \le D^2 \;\Rightarrow\; k \le \frac{D^2}{r_0}$$

Таким образом, число коррекций k не превосходит $\left\lfloor \frac{D^2}{r_0} \right\rfloor$.

ч. т. д.

3.2.4 Оптимизационная интерпретация

Рассмотрим непрерывную кусочно-линейную функцию $J(W)$:

$$J(W) = \sum_{x \in Y} d_x (W, x), \text{ где } d_x = \begin{cases} -1, x \in X_1 \\ 1, x \in X_2 \end{cases};$$

Y – множество векторов неправильно классифицированных гиперплоскостью W. Тогда $J(W) \ge 0$ и $J(W) = 0 \;\Leftrightarrow\; Y = \varnothing$. Задача состоит в минимизации этой функции:

$$J(W) = \sum_{x \in Y} d_x (W, x) \to \min$$

Построим минимизацию по схеме градиентного спуска:

$$W_{t+1} = W_t - r_t \frac{dJ(W)}{dW}$$

Т.к. $\frac{dJ(W)}{dW} = \sum_{x \in Y} d_x x$, то $W_{t+1} = W_t - r_t \sum_{x \in Y} d_x x$

Таким образом, алгоритм персептрона представляет собой вариант алгоритма градиентного спуска. Выбор последовательности величин r_t для обычно осуществляется так, чтобы:

$$\sum_{t=0}^{\infty} |r_t| > \infty \text{ и } \sum_{t=0}^{\infty} r_t^2 < \infty$$

3.2.5 Схема Кеслера

Идея построения линейного классификатора естественно обобщается на случай классификации с числом классов больше двух. Рассмотрим задачу классификации по M классам. Для каждого класса необходимо определить линейную дискримринантную функцию W_i, $i = 1,2,...,M$. Пусть x – $(l+1)$-мерный вектор в расширенном пространстве. Вектор x относится к классу Ω_i, если

$W_i x > W_j x, \ \forall i \neq j$

Схема Кеслера позволяет применить алгоритм персептрона для решения этой задачи.

Для каждого вектора-прецедента из Ω_i строим $(M-1)$ векторов x_{ij} размерности $(l+1)M$:

$$x_{ij} = (\underbrace{0,...,0}_{1}, \underbrace{0,...,0}_{2},..., \underbrace{x,...,x}_{i}, \underbrace{0,...,0}_{i+1},..., \underbrace{-x,...,-x}_{j}, \underbrace{0,...,0}_{M})^{\mathrm{T}}$$

и вектор $W = (W_1, W_2,..., W_M)^{\mathrm{T}}$, где W_i – весовой вектор i-ой дискриминантной функции.

Пусть $x = (x_1, x_2,..., x_M)$, тогда вектор x_{ij} можно записать в виде:

$$x_{ij} = (\underbrace{0,0,...,0}_{1}, \underbrace{x,}_{i} \underbrace{0,...,0}_{i+1}, \underbrace{-x,}_{j} \underbrace{0,...,0}_{M})$$

Если x относится к классу Ω_i, то $Wx_{ij} > 0 \quad \forall j = 1,2,...,M$, $i \neq j$, т.к. $W_i x > W_j x$ и $Wx_{ij} = W_i x - W_j x > 0$.

Таким образом, задача заключается в построении линейного классификатора в $(l+1)M$-мерном пространстве так, чтобы каждый из $(M-1)N$ векторов-прецедентов лежал в положительном полупространстве. Если вектора в исходной задаче разделимы, то это можно сделать с помощью алгоритма персептрона.

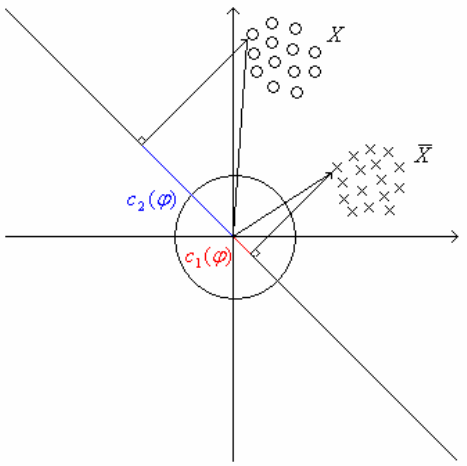
4 Оптимальная разделяющая гиперплоскость

4.1 Существование и единственность

Пусть X и $\bar{X}$ - конечные множества точек в евклидовом пространстве R^l.

Определение. *X и $\bar{X}$ разделимы гиперплоскостью, если существует единичный вектор j и число c, что $(x, j) > c$ при $x \in X$, $(\bar{x}, j) < c$ при $\bar{x} \in \bar{X}$.*

Обозначим $c_1(j) = \min\limits_{x \in X} (x, j)$, $c_2(j) = \max\limits_{\bar{x} \in \bar{X}} (\bar{x}, j)$.



Тогда $(x, j) > c_1(j)$ при $x \in X$, $(\bar{x}, j) < c_2(j)$ при $\bar{x} \in \bar{X}$. Если $c_1(j) \geq c_2(j)$, то гиперплоскость

$$(x, j) = \frac{c_1(j) + c_2(j)}{2} \qquad (4.1)$$

разделяет X и $\bar{X}$.

В силу непрерывности $c_1(j)$ и $c_2(j)$ существует множество разделяющих гиперплоскостей, если существует (4.1).

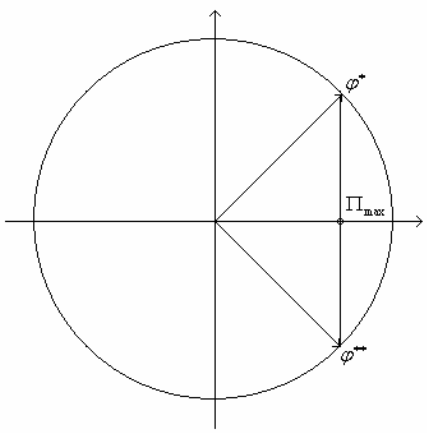
Определение. *Оптимальной называется разделяющая гиперплоскость* (4.1), *соответствующая вектору j^*, при котором достигается максимум* $\Pi(j)$.

Теорема. *Если два множества X и $\bar{X}$ разделимы гиперплоскостью, то оптимальная разделяющая гиперплоскость существует и единственна.*

Доказательство. Функция $\Pi(j)$ непрерывна на сфере $|j| \leq 1$. Значит, $\max\limits_{|j| \leq 1} \Pi(j)$ существует и достигается при некотором значении j^*. Предположим, что он достигается внутри сферы, т.е. $|j^*| < 1$. Тогда для $j^{**} = \dfrac{j^*}{|j^*|}$ получаем

$$\Pi(j^{**}) = c_1(j^{**}) - c_2(j^{**}) =$$
$$= \min_{x \in X} (x, j^{**}) - \max_{\bar{x} \in \bar{X}} (\bar{x}, j^{**}) =$$
$$= \frac{1}{|j^*|} \Pi(j^*) > \Pi(j^*),$$

что противоречит предположению о том, что j^* - точка максимума $\Pi(j)$. Следовательно, максимум достигается на границе сферы, т.е. $|j^*| = 1$.



Докажем единственность максимума. Предположим, что это не так и существуют различные j^* и j^{**} такие, что $\Pi(j^*) = \Pi(j^{**}) = \Pi_{max}$. Рассмотрим значение $j = a j^* + b j^{**}$, $a + b = 1, a > 0, b > 0$, не совпадающее ни с j^*, ни с j^{**}.

$$c_1(j) = \min_{x \in X} (x, a j^* + b j^{**}) =$$
$$= \min_{x \in X} [a (x, j^*) + b (x, j^{**})] \geq$$
$$\geq a \min_{x \in X} (x, j^*) + b \min_{x \in X} (x, j^{**}) =$$

$$= a \cdot c_1(j^*) + b \cdot c_1(j^{**}).$$

Аналогично $c_2(j) \le a \cdot c_2(j^*) + b \cdot c_2(j^{**})$.

Тогда

$$\Pi(j) = c_1(j) - c_2(j) \ge$$
$$\ge a \cdot c_1(j^*) + b \cdot c_1(j^{**}) - a \cdot c_2(j^*) + b \cdot c_2(j^{**}) =$$
$$= a \cdot \Pi(j^*) + b \cdot \Pi(j^{**}) =$$
$$= a \cdot \Pi_{\max} + b \cdot \Pi_{\max} = \Pi_{\max}$$

и j - тоже значение, на котором достигается максимум.

$$|j|^2 = |a j^* + b j^{**}|^2 = a^2 |j^*|^2 + 2ab(j^*, j^{**}) + b^2 |j^{**}|^2 < 1,$$

т.к. $|j^*|^2 = 1$, $|j^{**}|^2 = 1$ и $(j^*, j^{**}) < 1$ при $a + b = 1, a > 0, b > 0$.

Но j лежит внутри сферы $|j| \le 1$ и поэтому не может быть точкой максимума. Следовательно, предположение о существовании двух максимумов неверно и максимум единственный.

ч.т.д.

Таким образом, если максимум функции $\Pi(j)$ достигается при значении $j = j_{опт}$, то гиперплоскость $(x, j_{опт}) = \frac{c_1(j_{опт}) + c_2(j_{опт})}{2}$ максимально удалена от X и $\overline{X}$ и разделяет их.

4.2 Построение оптимальной разделяющей гиперплоскости

Теорема. *Если два множества X и $\overline{X}$ разделимы гиперплоскостью, $Conv(X)$ и $Conv(\overline{X})$ – выпуклые оболочки этих множеств, а $x^* \in Conv(X)$ и $\overline{x}^* \in Conv(\overline{X})$ – пара ближайших точек в выпуклых оболочках, то*

$$\max_{|j|=1} \Pi(j) = |x^* - \overline{x}^*|,$$

где $|x^* - \overline{x}^*|$ – обозначает евклидово расстояние между точками x^* и $\overline{x}^*$.

Доказательство. Положим $j^* = \frac{(x^* - \overline{x}^*)}{|x^* - \overline{x}^*|}$. Из условий $c_1(j) = \min_{x \in X} (x, j)$, $c_2(j) = \max_{\overline{x} \in \overline{X}} (\overline{x}, j)$ следует, что $c_1(j^*) \le (x^*, j^*)$, $c_2(j^*) = (\overline{x}^*, j^*)$ и, следовательно,

$$\Pi(j) = c_1(j) - c_2(j) \le (x^*, j^*) - (\overline{x}^* j^*) = (x^* - \overline{x}^*, j^*) = |x^* - \overline{x}^*| \qquad (4.2)$$

Следовательно $\max_{|j|=1} \Pi(j) \le |x^* - \overline{x}^*|$ и для доказательства теоремы нужно показать, что справедливо неравенство

$$\Pi(j^*) \ge |x^* - \overline{x}^*| \qquad (4.3)$$

Пусть точки $y \in X$ и $\overline{y} \in \overline{X}$ такие, что $c_1(j^*) = (y, j^*)$ и $c_2(j^*) = (\overline{y}, j^*)$.

Тогда

$$\Pi(j^*) = c_1(j^*) - c_2(j^*) = (y - \overline{y}, j^*) =$$
$$= (x^* + (y - x^*) - \overline{x}^* - (\overline{y} - \overline{x}^*), j^*) =$$
$$= (x^* - \overline{x}^*, j^*) + (y - x^*, j^*) - (\overline{y} - \overline{x}^*, j^*) =$$
$$= |x^* - \overline{x}^*| + (y - x^*, j^*) - (\overline{y} - \overline{x}^*, j^*).$$

Теперь покажем, что $(y - x^*, j^*) \ge 0$, а $(\overline{y} - \overline{x}^*, j^*) \le 0$, или, что то же самое:

$$\left(y - x^*, x^* - \bar{x}^*\right) \geq 0,\ \left(\bar{y} - \bar{x}^*, x^* - \bar{x}^*\right) \leq 0 \qquad (4.4)$$

Пусть $z = ly + (1-l)x^*$, $0 < l < 1$ – точка в R^l. Очевидно, что она лежит в выпуклой оболочке X, т.е. $z \in Conv(X)$. Тогда имеем

$$\left|z - \bar{x}^*\right|^2 = \left|l\left(y - \bar{x}^*\right) + (1-l)\left(x^* - \bar{x}^*\right)\right|^2 =$$

$$= \left|l\left(y - x^*\right) + (x^* - \bar{x}^*)\right|^2 =$$

$$= \left|x^* - \bar{x}^*\right|^2 + 2l\left(x^* - \bar{x}^*, y - \bar{x}^*\right) + l^2\left|y - x^*\right|^2 \qquad (4.5)$$

Поскольку точки x^* и $\bar{x}^*$ – ближайшие в выпуклых оболочках $Conv(X)$ и $Conv(\bar{X})$, получаем, что $\left|z - \bar{x}^*\right|^2 \geq \left|x^* - \bar{x}^*\right|^2$.

Тогда из (4.5) следует, что

$$2l(x^* - \bar{x}^*, y - x^*) + l^2\left|y - x^*\right|^2 \geq 0,$$

или $2(x^* - \bar{x}^*, y - x^*) + l\left|y - x^*\right|^2 \geq 0 \quad \forall l > 0$, что возможно лишь при $(x^* - \bar{x}^*,\ y - x^*) \geq 0$.

Таким образом, первое из неравенств (4.4) доказано. Второе неравенство (4.4) доказывается аналогично.

Тем самым доказано неравенство (4.3), а из него (4.2) и утверждение теоремы.

ч.т.д.

Оптимальная разделяющая гиперплоскость ортогональна отрезку, соединяющему ближайшие точки выпуклых оболочек множеств X и $\bar{X}$, и проходит через середину этого отрезка. Задача поиска пары ближайших точек сводится к задаче квадратичного программирования следующим образом.

Каждая точка y, лежащая в выпуклой оболочке $Conv(X)$, представима в виде $y = \sum_{x \in X} a_x x,\ \sum_{x \in X} a_x = 1,\ a_x \geq 0$. Аналогично, точка $\bar{y} \in Conv(\bar{X})$ представима в виде $\bar{y} = \sum_{\bar{x} \in \bar{X}} b_{\bar{x}} \bar{x},\ \sum_{\bar{x} \in \bar{X}} b_{\bar{x}} = 1,\ b_{\bar{x}} \geq 0$. Нужно найти пару точек y и $\bar{y}$, обеспечивающих минимум выражения:

$$\left|y - \bar{y}\right|^2 = \left(\sum_{x \in X} a_x x - \sum_{\bar{x} \in \bar{X}} b_{\bar{x}} \bar{x}, \sum_{x \in X} a_x x - \sum_{\bar{x} \in \bar{X}} b_{\bar{x}} \bar{x}\right) \qquad (4.6)$$

при условиях:

$$\sum_{x \in X} a_x = 1,\ a_x \geq 0, \qquad (4.7)$$

$$\sum_{\bar{x} \in \bar{X}} b_{\bar{x}} = 1,\ b_{\bar{x}} \geq 0. \qquad (4.8)$$

Задача математического программирования (4.6-4.8) имеет два ограничения и квадратичную целевую функцию.

4.3 Алгоритм Гаусса-Зейделя

Задача состоит в нахождении наименьшего расстояния между множествами X и $\overline{X}$.

1. В качестве начальных значений берем произвольную пару x_0 и $\bar{x}_0$. Другими словами в начальный момент $t=0$ $z_t = x_0 \in X$ и $\bar{z}_t = \bar{x}_0 \in \overline{X}$.

2. Необходимо найти точку z_{t+1} ближайшую к $\bar{z}_t$ на отрезке $[z_t, x_t]$. Обозначаем $z_{t+1} = \bar{z}_t$. Напишем условие ортогональности векторов $(z_{t+1} - \bar{z}_t)$ и $(z_t - x_k)$:

$$(z_{t+1} - \bar{z}_t, z_t - x_k) = 0.$$

Т.к. $z_{t+1} = l z_t + (1-l)x_k = x_k + l(z_t - x_k)$, то

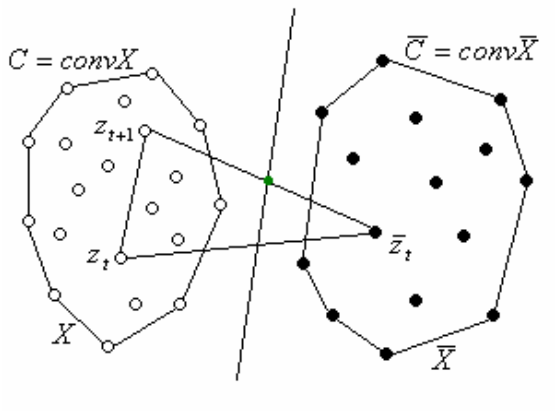
$$(z_{t+1} - \bar{z}_t, z_t - x_k) = (x_k + l(z_t - x_k) - \bar{z}_t, z_t - x_k) =$$
$$= l(z_t - x_k, z_t - x_k) + (x_k - \bar{z}_t, z_t - x_k) = 0$$

Следовательно, $l = \dfrac{(\bar{z}_t - x_k, z_t - x_k)}{|z_t - x_k|^2}$.

Если $l \le 0$, то $z_{t+1} = x_k$. Если $l \ge 1$, то $z_{t+1} = z_t$. Если $0 < l < 1$, то $z_{t+1} = l z_t + (1-l)x_k$.

3. Далее необходимо найти точку $\bar{z}_{t+1}$ ближайшую к z_t на отрезке $[\bar{z}_t, x_r]$. Обозначаем $z_{t+1} = \bar{z}_t$.

Данную процедуру необходимо повторять, пока не найдутся две ближайшие точки множеств X и $\overline{X}$.

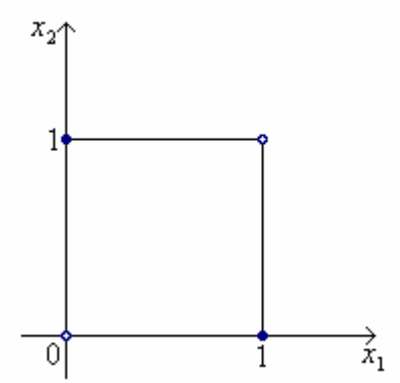


5 Нелинейный классификатор. Многослойный персептрон

5.1 Задача исключающего ИЛИ

Рассмотрим булеву функцию $xor(x_1, x_2)$ как некий классификатор. Вектор признаков имеет вид $x = (x_1, x_2)$. В данном случае имеется четыре прецедента и два класса. Напомним таблицу значений функции $xor(x_1, x_2)$.

№ прецедента	x_1	x_2	$xor(x_1, x_2)$	Класс
1	0	0	0	Ω_1
2	0	1	1	Ω_0
3	1	0	1	Ω_0
4	1	1	0	Ω_1

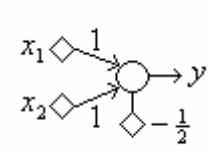


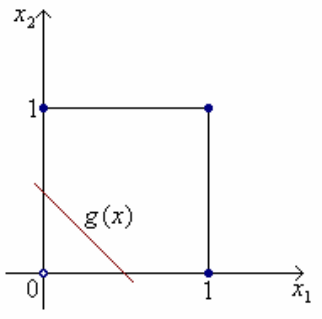
Как видно из рисунка тут нельзя построить разделяющую прямую, поскольку выпуклые оболочки точек, относящихся к первому классу и ко второму классу, пересекаются. Следовательно, и линейный классификатор построить нельзя. Попытаемся построить необходимый нелинейный классификатор как суперпозицию несколько линейных.

Рассмотрим две вспомогательные булевы функции $or(x_1, x_2)$ и $and(x_1, x_2)$. Напомним таблицы значений этих функций:

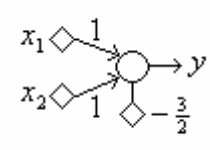
№ прецедента	x_1	x_2	$and(x_1, x_2)$	$or(x_1, x_2)$
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	1

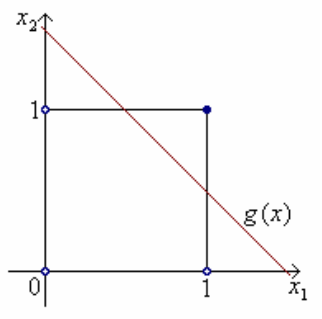
Построение линейного классификатора функции $or(x_1, x_2)$**.** Очевидно, что разделяющей прямой является линия: $x_1 + x_2 = \frac{1}{2}$. Соответствующий персептрон имеет вид:





Построение линейного классификатора функции $and(x_1, x_2)$**.** Здесь также можно построить разделяющую прямую: $x_1 + x_2 = \frac{3}{2}$. Соответствующий персептрон имеет вид:

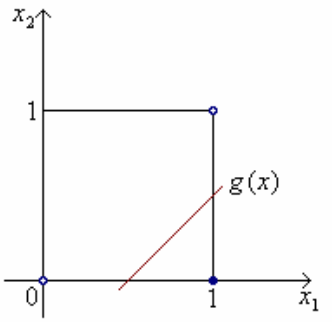




Построение нелинейного классификатора функции $xor(x_1, x_2)$. Пусть на выходе персептрона для функции $or(x_1, x_2)$ – y_1, а на выходе персептрона для функции $and(x_1, x_2)$ – y_2. Посмотрим, какие значения принимает вектор (y_1, y_2).

Исходные вектора		OR	AND	XOR	
x_1	x_2	y_1	y_2		Класс
0	0	0	0	1	Ω_1
0	1	1	0	0	Ω_0
1	0	1	0	0	Ω_0
1	1	1	1	1	Ω_1

Обозначив классы как показано в таблице, получаем разделяющую прямую, изображенную на рисунке и соответствующий линейный классификатор:

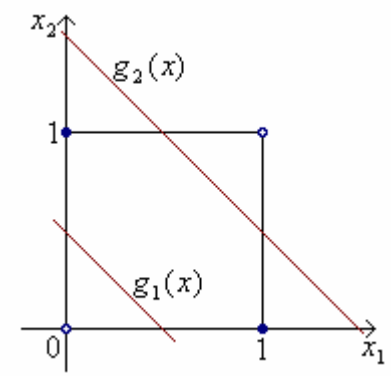


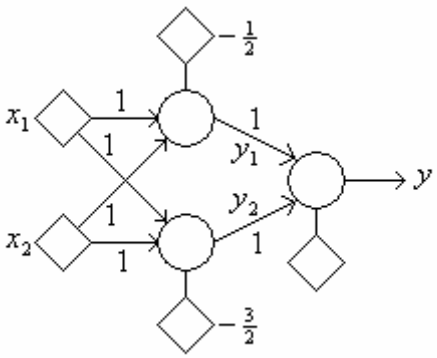
$$y_1 - y_2 = \frac{1}{2}$$

Учитывая вышеизложенное, получаем нелинейный классификатор, который задается через два линейных классификатора, как показано на рисунке слева:

$$x_1 + x_2 = \frac{1}{2} \text{ и } x_1 + x_2 = \frac{3}{2}$$

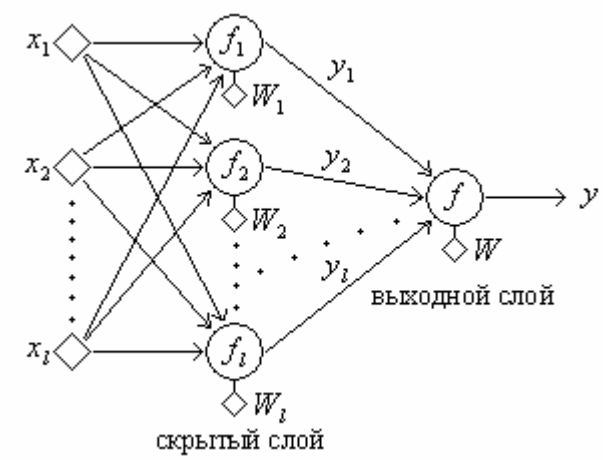
Соответствующий двухслойный персептрон изображен на рисунке справа.





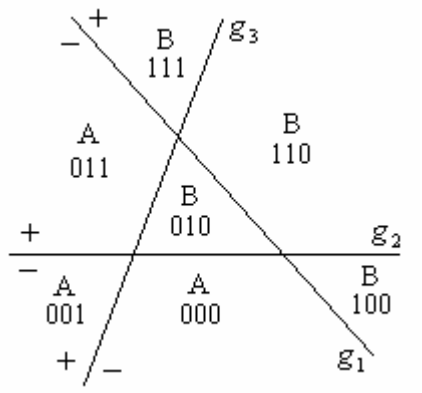
5.2 Классификационные способности двухслойного персептрона

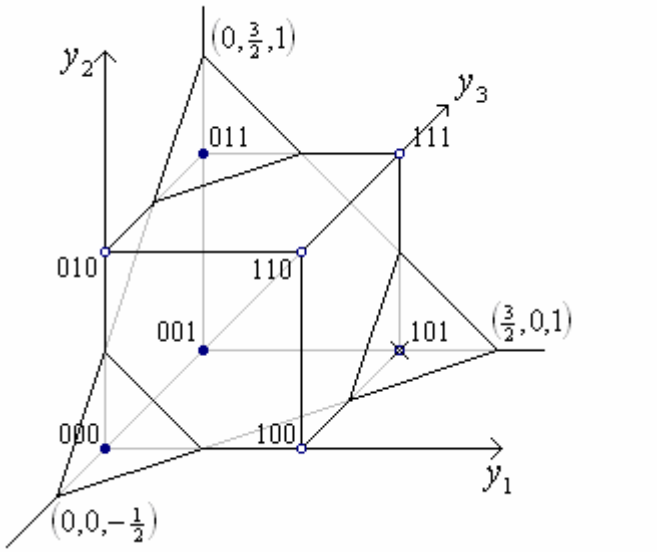
Рассмотрим общий случай двухслойного персептрона. Пусть $x \in R^l$ и в скрытом слое p нейронов. Скрытый слой нейронов отображает R^l в $H_p \in R^p$, где $H_p = \{(y_1, y_2, ..., y_p) \in R^p, y_i \in [0,1], 1 \le i \le p\}$ – гиперкуб. Другими словами, каждый нейрон задает гиперплоскость, которая разделяет пространство пополам, т.е. скрытый слой нейронов делит пространство R^l на полиэдры. Все вектора из каждого полиэдра отображаются в вершину p-мерного



единичного куба. Выходной нейрон разделяет вектора в классах, описанных полиэдрами, т.е. производит сечение гиперкуба, полученного в скрытом слое.

Пример. Рассмотрим нейронную сеть с двумя входами ($l=2$) и тремя нейронами ($k=3$). Тогда пространство $R^l = R^2$. Пусть первый слой нейронов задает разбиение

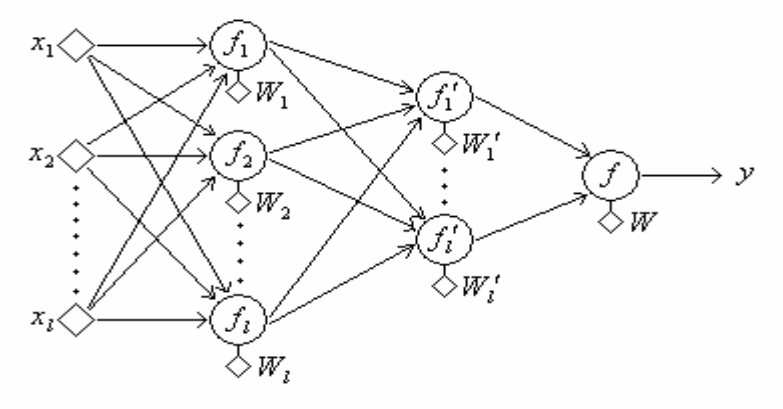




признакового пространства (плоскости) как показано на рисунке. В каждом многоугольнике (возможно, бесконечном) все точки соответствуют одному классу (A или B). При этом в каждом многоугольнике знаки линейных функционалов g_1, g_2, g_3 остаются постоянными. Следовательно, с каждым многоугольником связано определенное значение вектора выходов нейронов первого слоя, причем для разных многоугольников эти значения различны. Поскольку значениями компонент этого вектора являются 0 либо 1, получаем, что каждому многоугольнику соответствует некоторая вершина единичного куба H^3 в пространстве R^3. При этом каждой вершине куба сопоставлен один класс A или B. На рисунке изображен единичный куб H^3, у которого закрашенные вершины относятся к классу A, а не закрашенные – к классу B. Задача нейрона второго слоя состоит в разделении вершин этого куба. Нетрудно видеть, что в нашем примере плоскость $y_1 + y_2 - y_3 = \frac{1}{2}$ является разделяющей для куба H^3. Она и задает параметры нейрона второго слоя. Заметим, что вершина $(1,0,1)$ в кубе не загружена, т.е. в нее не отображается ни один многоугольник.

5.3 Трехслойный персептрон

Внешний (выходной) нейрон реализует лишь одну гиперплоскость. Очевидно, что одна разделяющая гиперплоскость не всегда может обеспечить желаемое разделение вершин гиперкуба. Например, если два конца одной его главной диагонали относятся к классу A, а два конца другой диагонали – к классу B. С аналогичной ситуацией мы уже сталкивались в задаче *исключающего или*. Попробуем ввести еще один слой нейронов.



Утверждение. *Трехслойная нейронная сеть позволяет описать любые разделения объединений полиэдров.*

Доказательство. Рассмотрим первый слой из p нейронов. На первом формируются гиперплоскости, т.к. строится полиэдральное разбиение пространства гиперплоскостями. Очевидно, что для заданного конечного множества прецедентов всегда можно построить разбиение пространства признаков на полиэдры такое, что ни в каком полиэдре не окажется пары точек из разных классов. Как было показано выше, первый слой отображает полиэдры в вершины p-мерного единичного гиперкуба. Поскольку с каждым полиэдром связаны образы одного класса, то и с каждой вершиной гиперкуба связан лишь один класс.

Каждый нейрон второго слоя описывает сечение полученного гиперкуба. Выберем в качестве таких сечений гиперплоскости, отсекающие ровно одну вершину гиперкуба. Поскольку число вершин в гиперкубе равно 2^p, число нейронов второго слоя также равно 2^p. Таким образом, выход нейронов второго слоя имеет следующий вид. Это вектор размерности 2^p, у которого всегда лишь одно значение равно 1, а остальные равны нулю. Назовем нейроны второго слоя нейронами класса A или B в соответствии с классом вершины гиперкуба, которую отсекает этот нейрон. Теперь становится понятно, каким образом строить третий слой нейронной сети. Нужно в выходном нейроне третьего слоя реализовать оператор логического сложения выходов нейронов второго слоя, относящихся к классу A. Таким образом разделяющая гиперплоскость выходного нейрона задается уравнением:

$$c_1 z_1 + c_2 z_2 + \ldots + c_л z_л = \frac{1}{2}, \text{ где } k=2^p, \text{ а } c_i = \begin{cases} 1 & \textit{если нейрон } i \textit{ относится к классу } A \\ 0 & \textit{в противном случае} \end{cases}$$

Таким образом, можно построить трехслойный персептрон следующим образом. Нейроны первого слоя разделяют пространство признаков на полиэдры одного класса и отображают их в вершины гиперкуба. Нейроны второго слоя отсекают вершины гиперкуба. Нейрон третьего слоя собственно осуществляет классификацию через оператор логического сложения. Тем самым утверждение доказано.

ч.т.д.

Рассмотрим, как строится уравнение гиперплоскости, отсекающей вершину p-мерного единичного гиперкуба. Диагональ куба имеет длину $\sqrt{p}$. Длины диагоналей $(p-1)$-мерных единичных гиперкубов, являющихся боковыми гранями p-мерного куба, равны $\sqrt{p-1}$. Центр куба находится в точке $\left(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2}\right)$. Расстояние от центра куба до любой вершины равно $\sqrt{p}/2$. Плоскость проводим перпендикулярно главной диагонали куба, инцидентной вершине, которую надо отсечь, так, чтобы расстояние от этой вершины до секущей

плоскости было равно $\frac{\sqrt{p}-\sqrt{p-1}}{2}$, причем данная точка должна находиться на диагонали куба, проведенной к отсекаемой вершине.

Пусть V – отделяемая вершина, $\overline{V}$ – диагонально противоположная вершина ($\overline{V}=E-V$, где E обозначает p-мерный вектор, состоящий из единиц). Следовательно, $W=V-\overline{V}$ – направляющий вектор разделяющей гиперплоскости. Тогда гиперплоскость проходит через точку:

$$U=\overline{V}+\left(V-\overline{V}\right)\cdot\frac{\sqrt{p}+\sqrt{p-1}}{2\sqrt{p}}$$

Обозначим:

$$g=\frac{\sqrt{p}+\sqrt{p-1}}{2\sqrt{p}}=\frac{1}{2}\cdot\left(1+\sqrt{1-\frac{1}{p}}\right)$$

Тогда

$$U=\overline{V}+\left(V-\overline{V}\right)\cdot g$$

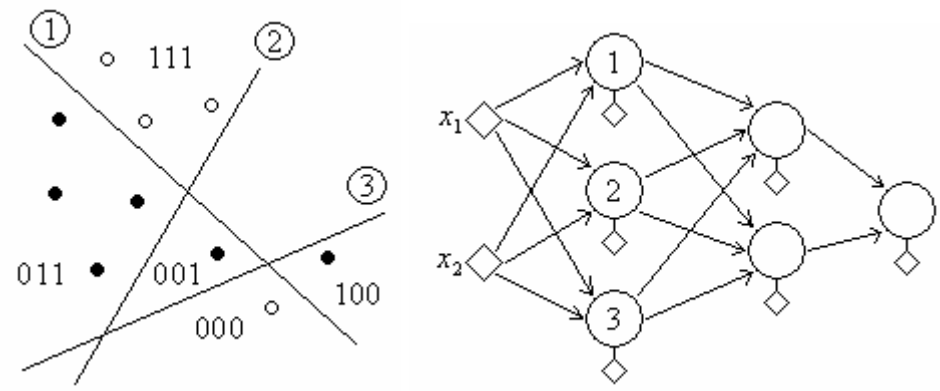
и уравнение гиперплоскости запишется в виде: $((z-U),W)>0$.

5.4 Построение нейронной сети

Существует два подхода к задаче построения нейронной сети-классификатора. Первый подход заключается в построении сети, варьируя архитектуру. Данный метод основан на точной классификации прецедентов. Второй подход состоит в подборке параметров (весов и порогов) для сети с заданной архитектурой.

5.4.1 Алгоритм, основанные на точной классификации множества прецедентов.

Опишем общую идею метода. За основу берется один нейрон. Далее наращиваем нейрон, пока не получим правильную классификацию всех прецедентов.



Рассмотрим более подробно алгоритм. Начинаем с одного нейрона $n(X)$, называемого мастером. После его тренировки получаем разделение множества X на X^+ и X^-. Если X^+ содержит вектора из двух классов, то вводим новый узел $n(X^+)$, называемый последователем.

Таким образом, на первом слое нейронов находится один мастер и несколько последователей. Никакие вектора из разных классов не имеют одинакового выхода из первого слоя.

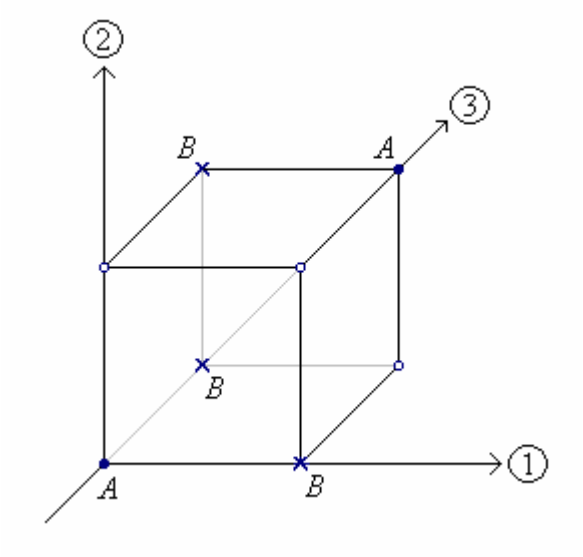
$X_1=\{y: y=f_1(x), x\in X\}$,

где f_1 – отображение, задаваемое первым слоем.

Аналогичным образом строим второй слой, третий слой и т.д.

Утверждение. *При правильном выборе весов каждый очередной слой правильно классифицирует все вектора, которые правильно классифицировал мастер и еще хотя бы один вектор.*

Таким образом, получаем архитектуру, имеющую конечное число слоев, правильно классифицирующие все прецеденты.



5.4.2 Алгоритм ближайших соседей

Нейроны первого слоя – это биссекторы, разделяющие пары. Второй слой – нейроны *and*, определяющие полиэдры. Третий слой – нейроны *or*, определяющие классы.

Основным недостатком данного метода является слишком большое количество нейронов. Уменьшить количество нейронов можно путем удаления внутренних ячеек:

$R_i = \{x : d(x, x_i) < d(x, x_j), i \neq j\}$.

5.4.3 Алгоритм, основанный на подборе весов для сети с заданной архитектурой

Идея данного метода состоит в том, чтобы ввести критерий в виде функции стоимости, которую необходимо минимизировать.

Пусть

L – число слоев в сети;

k_r – число нейронов в слое r, где $r = 1,2,...,L$;

k_L – число выходных нейронов;

$k_0 = l$ – размер входа;

$x(i) = \left(x_1(i), x_2(i),..., x_{k_0}(i)\right)$ – входной вектор признаков;

$y(i) = \left(y_1(i), y_2(i),..., y_{k_L}(i)\right)$ а – выходной вектор, который должен быть правильно классифицирован.

Текущем состоянии сеть при обучении дает результат $\hat{y}(i)$ не совпадающий с $y(i)$. Обозначим:

$$J = \sum_{i=1}^{N} e(i),$$

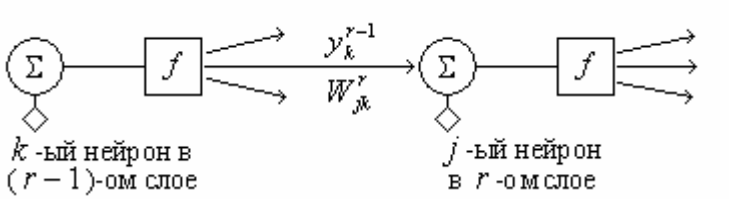
где N – число прецедентов; $e(i)$ – ошибка на i-ом прецеденте;

$$e(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) = \frac{1}{2}\sum_{m=1}^{k_L}\left(y_m(i) - \hat{y}_m(i)\right)^2,$$

где $i = 1,2,...,N$. J – функция всех синоптических весов и порогов. Таким образом, целью обучения является решение оптимизационной задачи:

$$J(W) \to \min,$$

где W – множество синоптических весов.



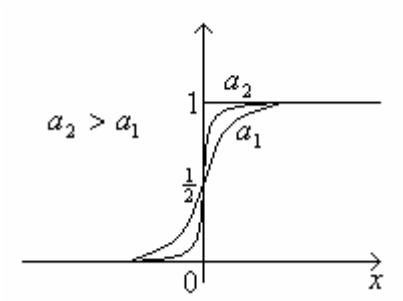
Пусть y_k^{r-1} – выход k-ого нейрона $(r-1)$-ого слоя; W_j^r – весовой вектор (включая порог) j-ого нейрона в r-ом слое, т.е. $W_j^r = \left(W_{j0}^r, W_{j1}^r, ..., W_{jk_{r-1}}^r\right)$, где k_{r-1} а – число нейронов в $(r-1)$-ом слое. Таким образом, J – разрывная функция M переменных, где

$$M = \sum_{r=1}^{L} k_{r-1} k_r$$

J разрывна, т.к. разрывна функция активации f:

$$f(x) = \begin{cases} 1, x > 0 \\ 0, x < 0 \end{cases}$$

5.4.4 Алгоритм обратной волны



Суть – аппроксимация J непрерывной дифференцируемой функцией за счет замены функции активации “сигмовидной” функцией:

$$f(x) = \frac{1}{1+e^{-ax}}$$

Вычислим производную функции:

$$f'(x) = \frac{1}{\left(1+e^{-ax}\right)^2} \cdot ae^{-ax} = a\left(\frac{1}{1+e^{-ax}} - \frac{1}{1+e^{-ax}}\right) = af(x)\left(1 - f(x)\right)$$

При данном чисто формальном приеме вектора признаков уже могут отображаться не только в вершины, но и внутрь гиперкуба. Необходимо решить задачу минимизации:

$$J(W) \to \min$$

Метод градиентного спуска решения задачи минимизации.

Пусть $W = \left\{W_j^r;\ j = 1,2,...,k_r;\ r = 1,2,...,L\right\}$. Тогда метод градиентного спуска выглядит так:

$$\Delta W = -m\frac{dJ}{dW},$$

где m а – шаг градиентного спуска. Очевидно, для его реализации необходимо уметь градиент $\frac{dJ}{dW_j^r}$.

Вычисление градиента. Аргумент функции активации j-ого нейрона r-ого слоя

$$V_j^r = \sum_{k=1}^{k_{r-1}} W_{jk}^r y_k^{r-1}(i) + W_{j0}^r = \sum_{k=0}^{k_{r-1}} W_{jk}^r y_k^{r-1}(i)$$

принимает различные значения в зависимости от индекса прецедента. В данном случае $y_0^{r-1}(i) = 1$.

Во входном слое, при $r = 1$ $y_k^{r-1}(i) = x_k(i)$, $k = 1,2,...,k_0$. В выходном слое, при $r = L$ $y_k^r(i) = \hat{y}_k(i)$, $k = 1,2,...,k_L$.

Рассмотрим выходной слой $r = L$.

$$e(i)=\frac{1}{2}\sum_{m=1}^{k_L}(e_m(i))^2=\frac{1}{2}\sum_{m=1}^{k_L}\left(f\left(V_m^L(i)\right)-y_m(i)\right)^2=e\left(V_m^L(i)\right)=e\left(V_m^L\left(W_m^L\right),i\right)$$

$$\frac{\partial e(i)}{\partial W_j^L}=\frac{\partial e(i)}{\partial V_j^L}\cdot\frac{\partial V_j^L}{\partial W_j^L}$$

$\frac{\partial V_j^L}{\partial W_j^L}=y^{r-1}(i)$ – не зависит от j-ого номера нейрона в слое, т.е. имеем одинаковый вектор производных для всех нейронов $(r-1)$-ого слоя.

$$\frac{\partial e(i)}{\partial V_j^L}=\left(f\left(V_j^L(i)\right)-y_j(i)\right)\cdot f'\left(V_j^L(i)\right)=e_j(i)\cdot f'\left(V_j^L(i)\right)$$

Следовательно, для последнего слоя $\frac{\partial e(i)}{\partial W_j^L}=y^{r-1}(i)\cdot e_j(i)\cdot f'\left(V_j^L(i)\right)$

Рассмотрим скрытый слой $r<L$. Имеется зависимость:

$$V_k^r=V_k^r\left(V_j^{r-1}\right)$$

$$\frac{\partial e(i)}{\partial V_j^{r-1}(i)}=\sum_{k=1}^{k_r}\frac{\partial e(i)}{\partial V_k^r(i)}\cdot\frac{\partial V_k^r(i)}{\partial V_j^{r-1}(i)}$$

$$\frac{\partial V_k^r(i)}{\partial V_j^{r-1}(i)}=\frac{\partial}{\partial V_j^{r-1}(i)}\left[\sum_{m=0}^{k_{r-1}}W_{km}^r y_m^{r-1}(i)\right],$$

но $y_m^{r-1}(i)=f\left(V_m^{r-1}(i)\right)$, следовательно:

$$\frac{\partial V_k^r(i)}{\partial V_j^{r-1}(i)}=W_{kj}^r\frac{\partial y_j^{r-1}(i)}{\partial V_j^{r-1}(i)}=W_{kj}^r f'\left(V_j^{r-1}(i)\right)$$

$$\frac{\partial e(i)}{\partial V_j^{r-1}(i)}=\left[\sum_{k=1}^{k_r}\frac{\partial e(i)}{\partial V_k^r(i)}W_{kj}^r\right]\cdot f'\left(V_j^{r-1}(i)\right)$$

Сумма, заключенная в квадратных скобках, известна из предыдущего шага.

Описание алгоритма.

0. Начальное приближение. Случайно выбираются веса небольших значений: W_{jk}^r, $r=1,2,...,L$, $j=1,2,...,k_r$, $k=01,2,...,k_{r-1}$.

1. Прямой проход. Для каждого вектора прецедента $x(i)$, $i=1,2,...,N$ вычисляются все $V_j^r(i)$, $y_j^r(i)=f\left(V_j^r(i)\right)$, $j=1,2,...,k_r$, $r=1,2,...,L$. Вычисляется текущее значение ценовой функции $J(W)$:

Цикл по $i=1,2,...,N$ (по прецедентам):

Вычислить:

$y_k^0(i)=x_k(i)$, $k=1,2,...,k_0$.

$y_0^0(i)=1$.

Цикл по $r=1,2,...,L$ (по слоям):

Цикл по $j=1,2,...,k_r$ (по нейронам в слое):

$$V_j^r(i)=\sum_{k=0}^{k_{r-1}}W_{jk}^r y_k^{r-1}(i)$$

$$y_j^r(i)=f\left(V_j^r(i)\right)$$

Конец цикла по j.

Конец цикла по r.

Конец цикла по i.

$$J(W) = \sum_{i=1}^{N} \frac{1}{2}\left(y_j^L(i) - y_j(i)\right)^2$$

2. Обратный проход. Для каждого значения $i = 1,2,...,N$ и $j = 1,2,...,k_L$ вычисляется $\frac{\partial e(i)}{\partial V_j^L(i)}$.

Затем последовательно необходимо вычислить $\frac{\partial e(i)}{\partial V_j^r(i)}$ для всех $r = L-1,...,1$ и $j = 1,2,...,k_r$:

Цикл по $i = 1,2,...,k_r$ (по нейронам в слое):

Вычислить:

$$e_j(i) = y_j^L(i) - y_j(i)$$

$$d_j^L(i) = e_j(i) \cdot f'\left(V_j^{r-1}(i)\right)$$

Цикл по $r = L, L-1,...,2$ (по слоям):

Цикл по $j = 1,2,...,k_r$ (по нейронам в слое):

$$e_j^{r-1}(i) = \sum_{k=1}^{k_r} d_k^r(i) \cdot W_{kj}^r$$

$$d_j^{r-1}(i) = e_j^{r-1}(i) \cdot f'\left(V_j^{r-1}(i)\right)$$

Конец цикла по j.

Конец цикла по r.

Конец цикла по i.

3. Пересчет весов. Для всех $r = 1,2,...,L$ и $j = 1,2,...,k_r$ $W_j^r(new) = W_j^r(old) + \Delta W_j^r$, где $\Delta W_j^r = -m\sum_{i=1}^{N} \frac{\partial e(i)}{\partial V_j^r(i)} y^{r-1}(i)$.

- Останов алгоритма может происходить по двум критериям: либо $J(W)$ стала меньше порога, либо градиент стал очень мал.
- От выбора m зависит скорость сходимости. Если m мало, то скорость сходимости также мала. Если m велико, то и скорость сходимости высока, но при такой скорости можно пропустить min.
- В силу много экстремальности существует возможность спустить в локальный минимум. Если данный минимум по каким-то причинам не подходит, надо начинать алгоритм с другой случайной точки.
- Данный алгоритм быстрее, чем алгоритм с обучением.

6 Метод потенциальных функций

Рассмотрим множество прецедентов. Пусть каждый из них имеет поле притяжения. Берем новый объект и смотрим, каким классом притягивается.

Пусть $L = 2$ – число классов. Обозначим эти классы через K_1 и K_2 соответственно. Рассмотрим обучающую последовательность $x_1,..., x_{r_1}, x_{r_1+1},..., x_{r_2}$. Без ограничения общности будем считать, что $x_1,...., x_{r_1} \in K_1$ и $x_{r_1+1},..., x_{r_2} \in K_2$.

Каждая точка образует в пространстве признаков X некоторое поле притяжения. Например, можно рассматривать каждую точку как единичный заряд. Поле описывается потенциалом, создаваемым системой зарядов во всем пространстве.

В пространстве задана потенциальная метрика: $K(x, y)$ – потенциальная функция, $x, y \in X$ такая, что

$K(x, y) > 0$, при $x \neq y$,

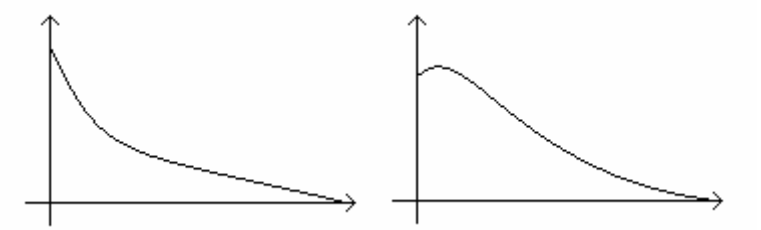
$K(x, y) = K(x, x + m(y - x)) = \tilde{K}(m)$,

где $\tilde{K}(m)$ – монотонно убывающая функция и $\tilde{K}(0)$ – ее максимальное значение.

Пример. Пусть $d(x, y)$ – расстояние в R^2. Рассмотрим функцию $K(x, y) = K(d(x, y))$. Пусть a – параметр функции. Рассмотрим два примера функций $K(x, y)$:

$K(x, y) = e^{-a^2 d^2(x,y)}$ (рис. слева),

$K(x, y) = \dfrac{1}{1 + a^2 d^2(x, y)}$ (рис. справа).



Пусть X и $\overline{X}$ – прецеденты первого и второго класса соответственно, y – пробный образ. Тогда потенциалы, создаваемые в пространстве точками из классов X и $\overline{X}$ будут иметь вид:

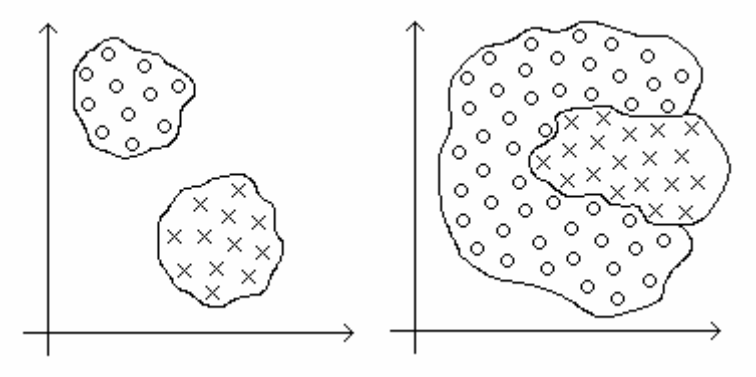
$K_X(y) = \sum_{x \in X} K(x, y)$, $K_{\overline{X}}(y) = \sum_{\overline{x} \in \overline{X}} K(\overline{x}, y)$

соответственно.

Тогда *правило классификации* можно записать следующим образом: Если $K_X(y) > K_{\overline{X}}(y)$, то пробный образ y относится к классу X, иначе к классу $\overline{X}$, при равенстве выдавать ответ «не знаю» (отказ от распознавания).

Если рассмотреть дискриминантную функцию $\Phi(x) = K_X(x) - K_{\overline{X}}(x)$, то задача сводится к поиску этой функции по обучающей последовательности.

Рассмотрим вариант метода потенциальных функций называется «наивным». Он имеет следующие недостатки. Если, например, в первом классе точек много больше, чем в остальных, то голоса первого класса подавят голоса других классов, то есть потенциал o больше потенциала x (рис. слева). Но даже, если множества соизмеримы, могут возникнуть проблемы другого характера, например, погружение одних точек в другие (рис. справа). Поэтому наивный метод подлежит усовершенствованию.



6.1 Общая рекуррентная процедура

Пусть $\{j_i(x)\}$ – конечная или бесконечная система функций на X. Будем искать дискриминантную функцию в виде

$$\Phi(x) = \sum_i c_i j_i(x).$$

Требования к рассматриваемому ряду:

Для бесконечного ряда требуем поточечную сходимость.

Также желательно, чтобы c_i убывали быстро с ростом i. Это необходимо для обеспечения хорошего совпадения “обрезанного” бесконечного ряда с $\Phi(x)$.

Итак, пусть $\{j_i(x)\}$ – базовая система функций. В качестве потенциальной функции будем рассматривать функцию вида

$$K(x,y) = \sum_i l_i^2 j_i(x) j_i(y),$$

где l_i удовлетворяет условиям: $\sum_i l_i^2 < \infty$ и $l_i \neq 0$. Обозначим $y_i(x) = l_i j_i(x)$. Тогда

$$K(x,y) = \sum_i y_i(x) j_i(y).$$

Предположим, что

$$K(x,x) = \sum_i y_i^2(x) \le M = const,$$

Тогда

$$K(x,y) = \sum_i y_i(x) j_i(y) \le M.$$

Для приближения $\Phi(x)$ предлагается рекуррентная процедура, называемая общей рекуррентной процедурой:

$$\Phi_{n+1}(x) = q_n * \Phi_n(x) + r_n * K(x_{n+1}, x),\ n = 1,2,\ldots.$$

Пусть

$\{x_{n+1}\}$ – обучающая последовательность прецедентов;

q_n, r_n – некоторые числовые последовательности, которые должны задаваться так, чтобы обеспечить сходимость $\Phi_n(x)$ к $\Phi(x)$ при $n \to \infty$ в том или ином смысле.

Зададим начальное приближение $\Phi_0(x) = 0$. Как уже отмечалось, мы ищем функцию $\Phi(x)$ в виде:

$$\Phi(x) = \sum_{i=1}^{\infty} c_i j_i(x).$$

Мы сделали достаточно сильное допущение, сказав, что наше решение будем выражать через базовую систему функций. Т.е. мы априорно предполагаем, что $\Phi_n(x)$ разложимо по системе функций $\{j_i(x)\}$:

$K(x,y)=\sum_{i=1}^{\infty} l_i^2 j_i(x) j_j(x)$ и $\Phi_k(x)=\sum_{i=1}^{\infty} c_i^k j_i(x)$.

Тогда, учитывая, что $K(x_{n+1},x)=\sum_i y_i(x_{n+1}) y_i(x)$, получаем:

$$\sum_i c_i^{n+1} j_i(x) = q_n \sum_i c_i^n j_i(x) + r_n \sum_i y_i(x_{n+1}) y_i(x).$$

Обозначим через

$\bar{c}_i^k = \frac{c_i^k}{l_i}$, $i,k=1,2,...$

Тогда

$\bar{c}_i^{n+1} y_i(x) = q_n c_i^n y_i(x) + r_n y_i(x_{n+1}) y_i(x)$.

Откуда получаем вторую форму общей рекуррентной процедуры:

$\bar{c}_i^{n+1} = q_n c_i^n + r_n y_i(x_{n+1})$.

Для нахождения связи коэффициентов $\{c_i^k\}$ и $\{c_i^{k+1}\}$ воспользуемся второй формой для формулы общей рекуррентной процедуры и соотношением

$\bar{c}_i^k = \frac{c_i^k}{l_i}$, $i,k=1,2,...$

Получим соотношение, связывающее коэффициенты $\{c_i^k\}$ и $\{c_i^{k+1}\}$:

$\frac{c_i^{n+1}}{l_i} = q_n c_i^n + r_n y_i(x_{n+1})$, где $y_i = l_i j_i(x)$.

Для возможности итерационных вычислений необходимо понять, как вычислять параметры q_n и r_n, а также начальное приближение c_i^0.

Зададим функцию

$$\Phi_i(x)=\begin{cases} K(x,x_i), \text{при } x_i \in X \\ -K(x,x_i), \text{при } x_i \in \bar{X} \end{cases},$$

где

$K(x,x_m)=\sum_{i=1}^{\infty} l_i^2 j_i(x) j_i(x_m)$ и $c_i^0 = \pm(l_i^2 j_i(x_1))$.

Тогда процесс перехода от Φ_n к Φ_{n+1} суть процесс подсчета коэффициентов. Обычно $q_n=1$, r_n вычисляется по следующему правилу:

$$r_n=\begin{cases} 0, \text{если } \Phi_n(x_{n+1})>0 \text{ и } x_{n+1}\in X \\ 0, \text{если } \Phi_n(x_{n+1})<0 \text{ и } x_{n+1}\in \bar{X} \\ 1, \text{если } \Phi_n(x_{n+1})<0 \text{ и } x_{n+1}\in X \\ -1, \text{если } \Phi_n(x_{n+1})>0 \text{ и } x_{\tau+1}\in \bar{X} \end{cases}.$$

Возьмем следующее начальное приближение:

$$\Phi_1(x)=\begin{cases} K(x,x), \text{при } x_1 \in X \\ -K(x,x), \text{при } x_1 \in \bar{X} \end{cases}$$

Таким образом, при правильном определении Φ_{n+1} получаем, что $\Phi_{n+1}(x) = \Phi_n(x)$; а в случае ошибки

$$\Phi_{n+1}(x) = \begin{cases} \Phi_n(x) + K(x_{n+1}, x), \text{при } x_{n+1} \in X \\ \Phi_n(x) - K(x_{n+1}, x), \text{при } x_{n+1} \in \overline{X} \end{cases}$$

Данный процесс напоминает обучение в алгоритме персептрона.

Возникают следующие естественные вопросы:

Есть ли поточечная сходимость функции $\Phi_n(x)$ к $\Phi(x)$?

Где взять базисные функции $j_i(x)$ в многомерном пространстве?

Попробуем ответить на эти вопросы.

Рассмотрим аналогию данного алгоритма с алгоритмом персептрона. Для функции

$$\Phi_n(x) = \sum_{i=1}^{\infty} c_i^k j_i(x)$$

произведем замену $j_i(x) = z_i$ и $z = (z_1, z_2, ...)$, $z \in Z$ – это вектор в бесконечномерном пространстве, тогда

$$\Phi(x) = \sum_{i=1}^{\infty} c_i j_i(x) = \sum_{i=1}^{\infty} c_i z_i ,$$

где z – спрямляющее пространство. Таким образом, если $\Phi(x) > 0$ или $\Phi(x) < 0$, то $\sum_{i=1}^{\infty} c_i z_i > 0$ или $\sum_{i=1}^{\infty} c_i z_i < 0$ соответственно. Пусть $x_k \in X \cup \overline{X}$, тогда $x_k \to (z_1^k, z_2^k, ...)$ и $z_i^k = j_i(x_k)$.

6.2 Выбор системы функций

Система функций $\{j_i(x)\}$ задается априорно. Обычно используют некую полную систему функций, например, на конечном отрезке можно взять систему тригонометрических функций. Эта система к тому же ортогональна.

Утверждение. *Если задана полная ортогональная система функций одной переменной, то можно построить полную ортогональную систему функций любого числа переменных.*

Доказательство. Пусть $\{j_i(x)\}$ – полная ортогональная система функций на конечном интервале I. Рассмотрим систему

$$\{j_{i_1,...,i_m}(x_1,..., x_m) = j_{i_1}(x_1),..., j_{i_m}(x_m)\},\ i_1,...,i_m = 0,1,2,...$$

Эта система полна и ортогональна на декартовом произведении m экземпляров I, то есть на $\underbrace{I \times I \times ... \times I}_{m \text{ раз}}$.

Проверим ортогональность. В скалярном произведении двух различных функций $j_{i_1,....,i_m}$ и $j_{j_1,...,j_m}$:

$$\int_I ... \int_I (j_{i_1,...,i_m}, j_{j_1,...,j_m}) dx_1 ... dx_m$$

всегда найдется такое k, что $j_{i_k}(x_k) \neq j_{j_k}(x_k)$ и, в силу ортогональности системы $\{j_i(x)\}$, имеем:

$$\int_I (j_{i_k}(x_k), j_{j_k}(x_k)) dx_k = 0 .$$

Далее, пусть $F(x_1,.., x_m)$ – произвольная функция m переменных. Фиксируем все переменные, кроме x_1, и получаем разложение функции F:

$$F = \sum_{i_1} c_{i_1}(x_2,..,x_m)j_{i_1}$$

Повторяем это рассуждение для c_{i_1} последовательности $m-1$ раз:

$$F(x_1,...,x_m) = \sum_{i_1,...,i_m} c_{i_1...i_m} j_{i_1}(x_1)...j_{i_m}(x_m),$$

что и доказывает полноту системы

$\{j_{i_1,...,i_m}(x_1,..., x_m)\}$, $i_1,...,i_m = 0,1,2,...$

ч.т.д.

6.3 Сходимость общей рекуррентной процедуры

Предположим, что обучающая последовательность есть выборка конечного объема из пространства $\tilde{X}$ ($\tilde{X}$ – пространство признаков). Тогда последовательность $\{\Phi_n(x)\}$ есть последовательность случайных функций, и последовательность c_i – последовательность случайных чисел. Поэтому будем говорить о сходимости $\{\Phi_n(x)\}$ в вероятностном смысле, то есть либо по вероятности, либо с вероятностью равной 1, либо в среднем.

Пусть x – случайные величины из $\tilde{X}$, а $X, \overline{X}$ – выборка для конечного объекта.

Теорема. *Пусть заданы два множества X и $\overline{X}$ и выполнены следующие условия.*

1) Существует функция $\Phi(x)$ такая, что

$\Phi(x) \geq e$, *при* $x \in X$,

$\Phi(x) \leq -e$, *при* $x \in \overline{X}$,

где константа $e > 0$.

2) Задана система функций $\{j_i(x)\}$, $i = 1,2,...$ *такая, что*

$$\Phi(x) = \sum_i c_i j_i(x),$$

$$K(x,y) = \sum_i l_i^2 j_i(x) j_i(y),$$

$$\sum_i \left(\frac{c_i}{l_i}\right)^2 < \infty.$$

3) Точки из обучающей последовательности независимые случайные величины, с одной и той же плотностью $p(x)$.

Тогда общая рекуррентная процедура, определяемая формулой

$$\Phi_{n+1} = q_n \Phi_n + r_n K(x, x_{n+1}),$$

где $\Phi_1(x) = 0$, $q_n = 1$ *и*

$$r_n = \begin{cases} 0, \text{если } \Phi_n(x_{n+1}) > 0 \text{ и } x_{n+1} \in X \\ 0, \text{если } \Phi_n(x_{n+1}) < 0 \text{ и } x_{n+1} \in \overline{X} \\ 1, \text{если } \Phi_n(x_{n+1}) < 0 \text{ и } x_{n+1} \in X \\ -1, \text{если } \Phi_n(x_{n+1}) > 0 \text{ и } x_{m+1} \in \overline{X} \end{cases}$$

сходится в следующем смысле: $\mathrm{E}\left[\left|sign\Phi(x) - sign\Phi_n(x)\right|\right] \to 0$, *при* $n \to \infty$.

Теорема. *Пусть выполнены все условия предыдущей теоремы. Пусть также на каждом n-ом шаге работы общей рекуррентной процедуры существует строго положительная вероятность исправления ошибки, если функция $\Phi_n(x)$ к n-ому шагу еще не разделила*

классы K_1 *и* K_2. *Пусть с вероятностью единица для каждой реализации процедуры существует конечное число* l *такое, что*

$\Phi_l(x)$ – *правильно разделяет* X *и* $\overline{X}$,

Z – *конечный интервал на прямой* ($Z \in R^1$),

$\{j_i(x)\}$ – *полная ортогональная система функций,* $j_i(x): Z \to R^1$,

$\int_Z j_i(x) j_j(x) dx = 0$ *при* $i \neq j$.

Тогда система функций

$\{j_{i_1,\ldots,i_m}(x_1,\ldots,x_m) = j_{i_1}(x_1) * \ldots\ldots * j_{i_m}(x_m)\}$

полна и ортогональна на пространстве $\underbrace{Z \times \ldots \times Z}_{m \text{ раз}}$.

Доказательство. Ортогональность очевидна. Докажем полноту. Пусть $F(x_1,..,x_m)$ – произвольная функция в Z^m такая, что $Z^m \to R^1$. Фиксируем переменные начиная с x_2, тогда

$$F = F(x_1) = \sum_{i=1}^{\infty} c_i(x_1,\ldots,x_m) \cdot j_i(x_1), \text{ где } c_i(x_2,\ldots,x_m) = \sum_{i=1}^{\infty} c_i(x_3,\ldots,x_m) \cdot j_i(x_l).$$

ч.т.д.

6.4 Функции Эрмита

Если в качестве системы функций $\{j_i(x)\}$ взять функции вида

$$j_i(x) = \frac{1}{\left(2^i i! \sqrt{p}\right)^{1/2}} e^{-\frac{x^2}{2}} H_i(x),$$

где $H_i(x) = (-1)^i e^{x^2} \left(\frac{d}{dx}\right)^i e^{-x^2}$ – полином Эрмита.

Тогда

$$K(x,y) = \sum_{i=0}^{\infty} l_i^2 j_i(x) j_j(x).$$

Обозначим $a^i = l_i^2$, где $|a| < 1$, тогда

$$K(x,y) = \frac{1}{\sqrt{p(1-a^2)}} \exp\left[\frac{2xya - (x^2+y^2)a^2}{1-a^2}\right].$$

7 КОМИТЕТНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ РАСПОЗНАВАНИЯ

7.1 Теоретико-множественная постановка задачи выбора алгоритма.

Байесовский подход исходит из статистической природы наблюдений. За основу берется предположение о существовании вероятностной меры на пространстве образов, которая либо известна, либо может быть оценена. Цель состоит в разработке такого классификатора, который будет правильно определять наиболее вероятный класс для пробного образа. Тогда задача состоит в определении "наиболее вероятного" класса.

Пусть J – индексное множество; D_j, $j \in J$ – подмножество некоторого множества (например, множества алгоритмов); $D = \{D_j | j \in J\}$ – система подмножеств. Пусть Y – множество, в котором необходимо найти решение. Задача заключается в нахождении такого элемента $y \in Y$ такое, что $y \in D_j \ \forall j \in J$.

Пример. Пусть $X_1 = \{x_1, x_2, ..., x_{m_1}\}$, $X_2 = \{x_{m_1+1}, x_{m_1+2}, ..., x_m\}$, $x_j \in \Omega$, $J = \{1,2,...,m\}$.

$F : \Omega \to \{0,1\}$ так, что $F(x) = \begin{cases} 0, x \in X_1 \\ 1, x \in X_2 \end{cases}$

Тогда D_j – множество алгоритмов, дающих правильную классификацию x_j:

$$D_j = \left\{ F \middle| F : \Omega \to \{0,1\}, F(x_j) = \begin{cases} 0,1 \le j \le m_1 \\ 1, иначе \end{cases} \right\}, \ j = 1,2,...,m$$

Определение. *Пусть* $J' \in J$, $D' = \{D_j | j \in J'\}$. *Тогда система подмножеств* D' *называется совместной, если* $\bigcap_{J'} D_j \neq \varnothing$.

В примере условием совместности является не пересекаемость множеств X_1 и X_2. Тогда, очевидно, что в пересечении $\bigcap_J D_j$ лежит $\Phi : \Omega \to \{0,1\}$, где

$$\Phi(x_j) = \begin{cases} 0,1 \le j \le m_1 \\ 1, иначе \end{cases}$$

Тогда возникает вопрос: что делать, если $D^* = \bigcap_{j \in J} D_j = \varnothing$? Существует два способа решения данной проблемы:

Смягчить условия, описывающие D_j, т.е. построить $\tilde{D} = \{\tilde{D}_j | j \in J, D_j \subseteq \tilde{D}_j\}$.

Решить задачу поиска максимальных совместных подсистем системы $D' = \{D_j | j \in J\}$, $J' \subset J$

Определение. Теоретико-множественная задача называется разрешимой в классе Y, если $Y \cap D^* \neq \varnothing$, где $D^* = \bigcap_{j \in J} D_j$.

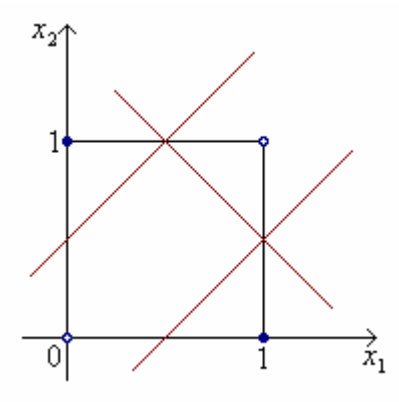
7.2 Комитеты

Нас интересует случай, когда теоретико-множественная задача не разрешима. Идея комитетного метода распознавания состоит в использовании нескольких классификаторов,

каждый из которых дает свой результат. Далее по какому-либо общему правилу голосования на основе полученных результатов от каждого классификатора выдается итоговый результат.

Определение. *Для исходной системы* D *и числа* p*:* $0 \le p < 1$ *конечное подмножество* $K \subseteq Y$ *называется* p*-комитетом в классе* Y*, если для всех* $j \in J$ *выполнено неравенство* $\left|K \cap D_j\right| > p|K|$ *(относительная доля* K*, лежащая в* D_j*, превосходит* p*). Если* $p = ½$*, то* p*-комитет называется просто комитетом.*

Пример комитета для несовместной системы. Рассмотрим задачу исключающего или. $x_0 = (0,0)$, $x_1 = (1,1)$, $x_2 = (0,1)$, $x_3 = (1,0)$. Пусть D а – множество линейных классификаторов. Опишем множество D^*: $D_0 = \{F : F(x_0) = 0\}$, $D_1 = \{F : F(x_1) = 0\}$, $D_2 = \{F : F(x_2) = 1\}$, $D_3 = \{F : F(x_3) = 1\}$, $D^* = D_0 \cap D_1 \cap D_2 \cap D_3 \neq \varnothing$. Пусть $Y = D$. Построим комитет $K = \{f_1, f_2, f_3\} \subset D$:



$$f_1 = \left(-x_1 + x_2 - \frac{1}{2} > 0\right) \quad f_1 \in D_0 \cap D_1 \cap D_2$$

$$f_2 = \left(x_1 - x_2 - \frac{1}{2} > 0\right) \quad f_1 \in D_0 \cap D_1 \cap D_3$$

$$f_3 = \left(-x_1 - x_2 + \frac{3}{2} > 0\right) \quad f_1 \in D_1 \cap D_2 \cap D_3$$

	z_1	z_2	Класс	f_1	f_2	f_3
x_0	0	0	B(0)	0	0	1
x_1	1	1	B(0)	0	0	0
x_2	0	1	A(1)	1	0	1
x_3	1	0	A(1)	0	1	1

$K \cap D_0 = \{f_1, f_2\}$,
$K \cap D_1 = K$,
$K \cap D_2 = \{f_1, f_3\}$,
$K \cap D_3 = \{f_2, f_3\}$.
$\left|K \cap D_j\right| \ge 2 > \frac{1}{2}|K| = \frac{3}{2}$.

Следовательно, K есть комитет в классе линейных классификаторов.

Определение. Пусть $A, B \subseteq \Omega$ (подмножества, возможно, бесконечные) и $\tilde{F} = \{F | F : \Omega \to R\}$ – класс функционалов. Набор функционалов $\{F_1, F_2, ..., F_q\}$ называется разделяющим комитетом для множеств A и B, если

$$\left|\{k | F_k(a) > 0\}\right| > \frac{1}{2}q, \ \forall a \in A$$

$$\left|\{k | F_k(b) > 0\}\right| > \frac{1}{2}q, \ \forall b \in B$$

Утверждение. *Чтобы набор* $\{F_1, F_2, ..., F_q\}$ *был разделяющим комитетом для* A *и* B *необходимо, чтобы для каждой пары* $a \in A$ *и* $b \in B$ *нашелся такой* F_k*, что* $F_k(a) > 0$ *и* $F_k(b) < 0$.

Доказательство. Если n_a – число функционалов $F_k(a) > 0$, n_b – число функционалов $F_k(b) > 0$, то

$$n_a + n_b > \frac{1}{2}q + \frac{1}{2}q = q$$

И, т.к. найдется функционал, обладающий обоими свойствами, утверждение доказано.

ч.т.д.

Теорема. *Пусть* $\Omega = R^l$, $l \geq 2$; $A = \{x_1, x_2, ..., x_{m_1}\}$, $B = \{x_{m_1+1}, x_{m_2+2}, ..., x_m\}$, $0 < m_1 < m$. *И пусть* $x_k = 0$, $\forall k = 1,2,...,m$ *(нет нулевой точки);* $x_i \neq x_j a$, $a \neq 0$, $\forall i, j, a$ *(не коллинеарны). Тогда для таких* A *и* B *существует разделяющий комитет в классе аффинных функционалов:* $\tilde{F} = \{F | F(x) = (W, x) + W^0, W \in R^l, W^0 \in R\}$.

Доказательство. Построим комитет из $2m-1$ элементов (функционалов):

$K = \{F_1, F_1', F_2, F_2', ..., F_{m-1}, F_{m-1}', F_m\}$

Для каждого функционала необходимо найти W_k и W_k^0 в – пару, которая определяет функционал $F_k = (W_k, x) + W_k^0$, причем $(x_k, W_k) = 0$, т.е. $W_k \perp S_k$ и $\forall r \neq k$, $r = 1,2,...,m$ $(W_k, x_r) \neq 0$, т.е. W_k не ортогонален остальным x_r. Другими словами каждая гиперплоскость должна иметь направляющий вектор, ортогональный своему прецеденту и не ортогональный всем остальным.

Пусть $d_k = \frac{1}{2} \min_{r \neq k} |(W_k, x_r)| > 0$. Выберем W_k^0 следующим образом:

$$W_k^0 = \begin{cases} d_k, \text{ при } k = 1,2,...,m_1 \\ -d_k, \text{ при } k = m_1 + 1,...,m \end{cases}$$

$F_k'(x) = -(W_k, x) + W_k^0$

$F_k(x) = (W_k, x) + W_k^0$

Покажем, что построенное множество функционалов является комитетом для A и B. Рассмотрим

$$F_k(x_k) = (W_k, x_k) + W_k^0 = W_k^0 = \begin{cases} > 0, k \leq m_1 \\ < 0, k > m_1 \end{cases}$$

$$F_k'(x_k) = -(W_k, x_k) + W_k^0 = W_k^0 = \begin{cases} > 0, k \leq m_1 \\ < 0, k > m_1 \end{cases}$$

$F_k'(x)$ и $F_k(x)$ правильно классифицируют x_k. Посмотрим, как будет работать каждй такой функционал на остальных x_r:

$F_k(x_r) = (W_k, x_k) + W_k^0$

Т.к. $W_k^0 < (W_k, x_k)$, то знак $F_k(x_r)$ определяется знаком (W_k, x_r).

Рассмотрим $1 \leq k \leq m-1$. $F_k'(x_k)$ и $F_k(x_k)$ голосуют правильно, т.е. x_k соответствует правильное положение гиперплоскостей. $F_k'(x_r)$ и $F_k(x_r)$ имеют разные знаки. Следовательно, каждая пара F_k' и F_k правильно классифицирует на всех x_k и дает одну правильную классификацию на остальных x_r. Таким образом, количество правильно голосующих за x_k равно $2 + (m-2) = m$.

ч.т.д.

7.3 Комитеты линейных функционалов

Пусть $A = \{x_1, x_2, ..., x_{m_1}\}$, $B = \{x_{m_1+1}, x_{m_2+2}, ..., x_m\}$, $A, B \subseteq R^l$ в – конечные множества в пространстве признаков; $x_1, x_2, ..., x_m$ – точки общего положения.

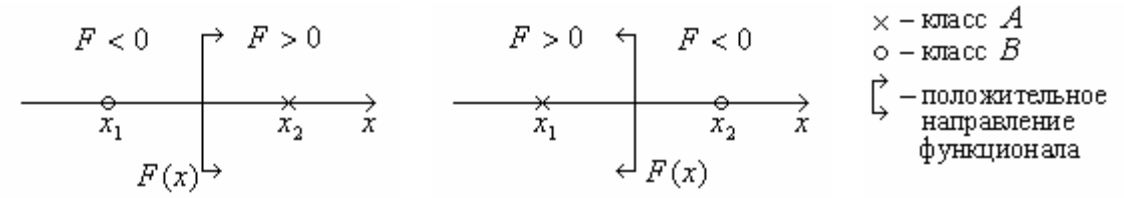
Определение. *Точки* $x_1, x_2, ..., x_m$ *пространства* R^l *называются точками общего положения, если никакая* $l+1$ *точка не лежит в гиперплоскости размерности* $l-1$.

Пример. Пусть $l=2$, т.е. рассматривается пространство R^2 (плоскость). Тогда точки $x_1, x_2, ..., x_m$ – точки общего положения, если никакие три из них не лежат на одной прямой.

Теорема. *Существует разделяющий комитет аффинных функционалов, состоящий из не более, чем* m *членов при нечетном* m *и не более, чем* $m-1$ *при четном* m.

Доказательство. Рассмотрим случай $l=1$, т.е. пространство R^1.

Пусть $m=2$, $m_1=1$. Тогда возможны два случая.



Для первого случая (рис. слева) функционал имеет вид:

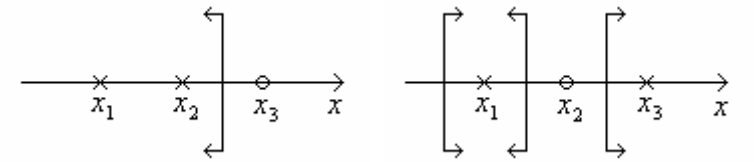
$$F(x)=x-\frac{x_1+x_2}{2}$$

Для второго случая (рис. справа) функционал имеет вид:

$$F(x)=-\left(x-\frac{x_1+x_2}{2}\right)$$

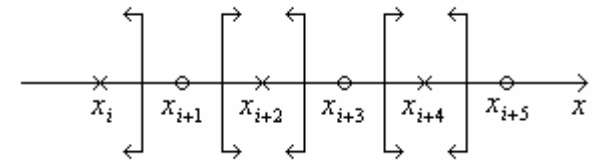
$|k|=1$ – количество функционалов для худшего случая.

Пусть $m=3$, $m_1=2$. Тогда возможны следующие варианты.



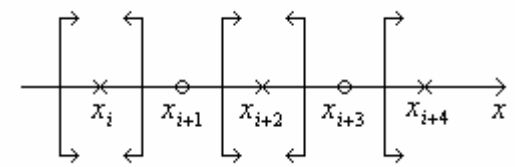
Все случаи вида показанного на рис. слева сводятся к предыдущему ($m=2$, $m_1=1$). Во всех остальных случаях функционалы надо располагать аналогично рис. справа. Для худшего случая $|k|=3$.

Пусть $m=2n$ (четное количество точек). Рассмотрим худший из возможных вариантов.



В данном случае функционалы надо располагать как показано на рис. $|k|=m-1$.

Пусть $m=2n-1$ (нечетное количество точек). Рассмотрим худший из возможных вариантов.



В данном случае функционалы надо располагать как показано на рис. $|k|=m$. Все остальные случаи можно свести либо к этим двум, либо к предыдущим.

Таким образом, по методу математической индукции существует разделяющий комитет аффинных функционалов из не более, чем m членов при нечетном m и не более, чем $m-1$ при четном m в пространстве R^1.

Многомерный случай сводится к одномерному следующим образом. Ищем подпространство $W \in R^l$ такое, что $(W, x_i) \neq (W, x_j)$, при $i \neq j$. Проектируем все x_i на соответствующие подпространства, пока не получим одномерную задачу. В многомерном случае для разделения x_i и x_j служит гиперплоскость:

$$(W, x) = \frac{1}{2}\left[(W, x_i) + (W, x_j)\right]$$

ч.т.д.

7.4 Функция Шеннона

Пусть $L_n(m_1, m - m_1)$ – это число гиперплоскостей, достаточное для разделения любых точечных множеств m_1 и $m - m_1$ точек общего положения в пространстве R^n.

Лемма 1. *Если* $m_1 \leq m - m_1$, *то*

$$L_n(m_1, m - m_1) \leq 2\left\lceil \frac{m_1}{n} \right\rceil$$

Доказательство. Если $m_1 \leq n$, то добавим точки общего положения до n. Через n точек из m_1 проводим гиперплоскость:

$F(x_1) = F(x_2) = ... = F(x_n) = 0$

Для x_k такого, что $k > n$ $F(x_k) \neq 0$.

Выберем $e = \frac{1}{2} \min\limits_{n<i\leq m_1} |F(x_i)|$ и возьмем гиперплоскости $G_1 = F + e$ и $G_2 = F - e$. G_1 и G_2 отделяют точки $x_1, x_2, ..., x_n$ от всех остальных.

Аналогичным образом из оставшихся $(m_1 - n)$ точек выделяем еще n и строим еще пару гиперплоскостей. Далее из оставшихся $(m_1 - 2n)$ точек выделяем еще n и строим еще пару гиперплоскостей и т.д. В конце получим $(m_1 - nm)$ точек. Следовательно:

$$L_n(m_1, m - m_1) \leq 2\left\lceil \frac{m_1}{n} \right\rceil$$

ч.т.д.

Утверждение 1. *Если* $W_1, W_2, ..., W_q$ *разделяют множества* A *и* B, *и* $r(t)$ *– непрерывная кривая в* R^l *такая, что* $r(0) \in A$, *а* $r(1) \in B$, *то существует* $k \in \{1,2,...,q\}$ *и* $t_0 \in (0,1)$ *такие, что* $(W_{k_0}, r(t_0)) = 0$.

Утверждение 2. *Любая гиперплоскость пересекает кривую* $r(t)$ *не более, чем в* n *точках.*

Доказательство. Рассмотрим линейный функционал W. Запишем условие пересечения гиперплоскости и кривой $r(t)$:

$(W, r(t)) = 0$.

Кривая $r(t)$ задана многочленом степени n. Следовательно, $(W, r(t))$ – то же многочлен степени n. Значит, уравнение $(W, r(t)) = 0$ является уравнением степени n. Следовательно, т.к. корни могут быть кратными, данное уравнение имеет не более n корней.

ч.т.д.

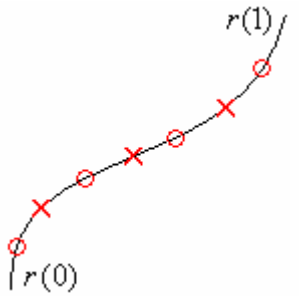
Лемма 2. $L_n(m_1, m-m_1) \geq \left\lceil \frac{2m_1-1}{n} \right\rceil$.

Доказательство. Построим $L_n(m_1, m-m_1)$. Рассмотрим последовательность точек:

$0 < t_1 < t_2 < \ldots < t_m = 1$

Пусть $r(t) = (r_1, r_2, \ldots, r_n)$, где $r_i = r_i(t) = t^i$, $i = 1,2,\ldots,n$. Тогда $x_j = r(t_j) = \left(t_j, t_j^2, t_j^3, \ldots, t_j^n\right)$ – точки в R^n.

Без ограничения общности положим $x_j \in A$, при j нечетном, и $x_j \in B$, при j четном. Тогда получим непрерывную кривую (см. рис).



Каждая гиперплоскость дает не более, чем n пересечений. Кривая должна иметь $(m-1)$ разделение, т.е. должно быть $(m-1)$ гиперплоскостей. Следовательно, всего гиперплоскостей должно быть не менее, чем $\left\lceil \frac{m-1}{n} \right\rceil$, т.е. $L_n(m_1, m-m_1) \geq \left\lceil \frac{m-1}{n} \right\rceil$

Т.к. $m_1 = \begin{cases} m/2, \text{ при четном } m \\ (m-1)/2, \text{ при нечетном } m \end{cases}$, то $\begin{matrix} m = 2m_1, \text{ при четном } m \\ m = 2m_1 + 1, \text{ при нечетном } m \end{matrix}$.

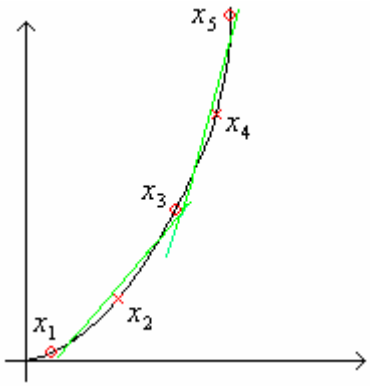
Следовательно,

$L_n(m_1, m-m_1) \geq \left\lceil \frac{2m_1}{n} \right\rceil$, при нечетном m,

$L_n(m_1, m-m_1) \geq \left\lceil \frac{2m_1-1}{n} \right\rceil$, при четном m.

Окончательно получаем: $L_n(m_1, m-m_1) \geq \left\lceil \frac{2m_1-1}{n} \right\rceil$, $\forall m$.

ч.т.д.



Пример. Пусть $m = 5$, $m_1 = 2$, $n = 2$. Обозначим $A = \{x_1, x_3, x_5\}$ и $B = \{x_2, x_4\}$. Тогда

$L_n(m_1, m-m_1) = L_2(2,3) \geq \left\lceil \frac{2 \cdot 2 - 1}{2} \right\rceil = 2$ и

$L_n(m_1, m-m_1) = L_2(2,3) \leq 2 \cdot \left\lceil \frac{2}{2} \right\rceil = 2$

§5. Метод построения комитета.

Пусть X – множество прецедентов; l в – размерность пространства признаков; m_1 и $m-m_1$ – количество прецедентов в каждом классе.

Построим $W(x)$ – линейный функционал такой, что, если $W(x_k) > 0$, то объект из класса A $(k = 1,2,\ldots,m_1)$, и, если $W(x_k) < 0$, то объект из класса B $(k = m_1+1, m_2+2, \ldots, m)$. Если данный функционал правильно классифицирует меньше половины объектов, то возьмем его со знаком минус.

Итак, пусть линейный функционал $W(x)$ правильно классифицирует больше половины объектов. Разобьем множество прецедентов X на множество правильно классифицированных объектов X_1 и множество неправильно классифицированных объектов $\overline{X}_1$, т.е. $X = X_1 \cup \overline{X}_1$.

Далее строим последовательно пары функционалов W_s и W_s':

$W_1, W_2, W_2', W_3, W_3', ..., W_s, W_s'$

Делаем очередной шаг. $X = X_s \bigcup X_s'$. Пусть на X_s – (s) правильно классифицированных объектов, а на $\overline{X}_s$ – $(s-1)$ правильно классифицированных объектов. Строим пару W_{s+1}, W_{s+1}'. В $\overline{X}_s$ выделяем l точек одного класса. Эти точки можно перевести в X_{s+1}, т.е. $X_{s+1} = X_s + \{l \text{ точек}\}$, а $\overline{X}_{s+1} = \overline{X}_s$.

На каждом шаге множество неправильно классифицированных объектов уменьшается на l, следовательно, процесс сходится.

Общее число функционалов: $1 + 2 \cdot \left\lceil \frac{m}{2} \cdot \frac{1}{l} \right\rceil = 1 + \left\lceil \frac{m}{l} \right\rceil$.

Теорема. *Существует комитет линейных функционалов, в котором число членов не превосходит* $\left\lceil \frac{m}{l} + 1 \right\rceil$.

8 Классификация на основе сравнения с эталоном.

Пусть задано множество образов (эталонов). Задача состоит в том, чтобы для тестируемого объекта выяснить, какой эталон ближе на основе меры сходства (расстояния между объектами). Данная задача и получила название “сравнение с эталонами”.

В качестве эталонов могут рассматриваться следующие объекты:

1) Буквы в словах рукописного текста (применительно к распознаванию рукописного текста);

2) Силуэты объектов в сцене (применительно к машинному зрению);

3) Слова (команды), произносимые человеком (применительно к распознаванию речи).

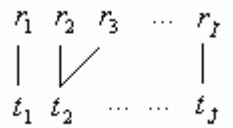
В этих примерах признаки не выделены, но можно измерить сходство. Например, сравнение слов: *кошка ~ мошка ~ кора ~ норка* и.т.д. Или силуэт объекта в сцене, чье положение и ориентация заранее не известны (применительно к машинному зрению, робототехнике).

8.1 Мера близости, основанная на поиске оптимального пути на графе

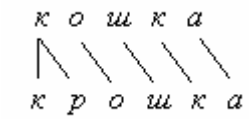
Рассмотрим строчный образ (слово). В данном случае можно выделить два критерия, на основе которых можно строить меру близости:

совпадение букв,

монотонность (совпадение порядка букв).

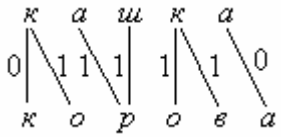


Пусть $r_1 r_2 ... r_i$ – эталон, $t_1 t_2 ... t_j$ – пробный образ, причем $i \neq j$. Построим соответствие между эталоном и пробным образом по следующему правилу: каждому символу в первом слове должен соответствовать хотя бы один символ во втором слове и каждому символу во втором слове должен соответствовать хотя бы один символ в первом слове, (но соответствие между символами не взаимнооднозначное, в частности, поскольку $i \neq j$).



Введем меру следующим образом:

$$r(r_i, t_i) = \begin{cases} 1, r_i \neq t_i \\ 0, r_i = t_i \end{cases}$$



В качестве меры сходства двух слов принимаем соответствие, при котором суммарный вес всех дуг (изображенных на рисунках) минимален:

$n(\bar{r}, \bar{t}) = \min_S m(S)$, где $m(S) = \sum_{(i,j) \in S} r(r_i, t_j)$.

Через $n(\bar{r}, \bar{t})$ далее будем обозначать меру близости двух слов $\bar{r}$ и $\bar{t}$.

Соответствие S должно быть двудольным графом без изолированных вершин с непересекающимися ребрами. Рассмотрим задачу сравнения цепочек упорядоченных символов. В данной задаче могут возникать следующие ошибки:

неправильно определенный символ (*ко<u>ш</u>ка – ко<u>р</u>ка*),

ошибка вставки (*кошка – кошр<u></u>ка*),

ошибка потери (*к<u>о</u>шка – кшка*).

Определение. *Редакторским расстоянием называется минимальное общее число изменений, вставок и потерь, требуемое для изменения образа A в образ B:*

$$D(A,B)=\min_j\left[C(j)+P(j)+R(j)\right],$$

где минимизация происходит по всем возможным комбинациям символьных преобразований таких, чтобы получить B *из* A.

Пусть

$$d(i,j\,|\,i-1,j-1)=\begin{cases}1, & \text{при } t(i)=r(i)\\ 0, & \text{при } t(i)\neq r(i)\end{cases}.$$

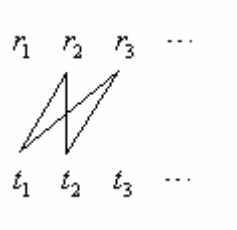
Тогда

$$d(i,j\,|\,i-1,j)=d(i,j\,|\,i,j-1)=1.$$

Построим таблицу, в которой столбцы – это символы образа, строки – символы эталона. Количество точек в матрице есть $P\cdot J$.

По данной таблице построим граф по следующему правилу. Если отображается точка (r_2,t_1), то далее выбираем (r_2,t_2), (r_3,t_2) или (r_3,t_1) (т.е. возможны три варианта). Соответствие слов реализуется в виде маршрута в графе. Этот маршрут обязательно начинается с точки (r_1,t_1) (иначе появится изолированная точка) и заканчивается в (r_P,t_J).

	t_1	t_2	t_3	…	t_J
r_1	0	0	0	…	0
r_2	0	0	0	…	0
r_3	0	0	0	…	0
…	…	…	…	…	…
r_P	0	0	0	…	0



Таким образом, получили задачу выбора кратчайшего пути на графе от точки (r_1,t_1) до точки (r_P,t_J), где каждая вершина имеет свою стоимость: 0 или 1.

8.2 Задача сравнения контуров

В качестве примера рассмотрим задачу сравнения контуров. Контура изображаются ломаными линиями, вершины которых будем называть узлами. Пусть заданы две линии – эталон и тестируемый объект. Используем следующую модель для сравнения объектов. Будем считать, что они изготовлены из проволоки и будем сравнивать близость этих ломаных путем оценки величины механической работы, которую нужно выполнить, чтобы преобразовать один объект в другой. Определим элементарную работу, которую надо совершить для перевода отдельных прямолинейных элементов ломаных. Достаточно рассмотреть два основных вида деформаций: растяжение (сжатие) и изгиб в узлах.

Каждой такой деформации припишем элементарную работу:

$f(|l_1-l_2|)$ – работа по изменению длины при растяжении и сжатии,

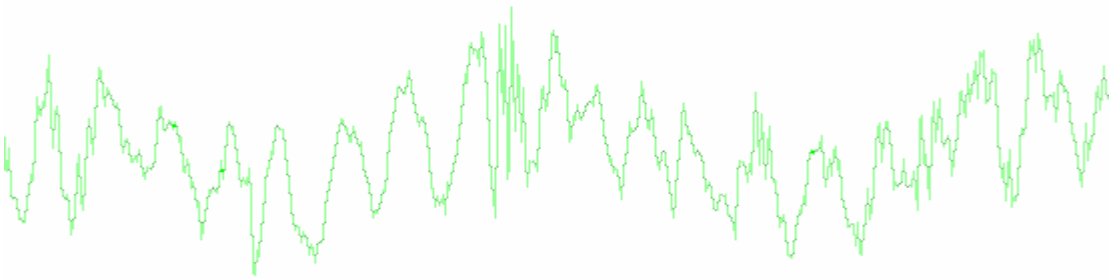
$f(|j_1-j_2|)$ – работа по изменению угла при изгибе.

Задача состоит в поиске такого преобразования, чтобы затраченная работа была минимальной, т.е. надо найти

$$f_\Sigma\to\min_S.$$

Эта задача сводится к установлению соответствия узлов одной ломаной узлам другой. При этом не требуется взаимно-однозначное соответствие, но требуется сохранение монотонности. Задача установления такого соответствия, которое минимизирует общую работу по деформации ломаных, также сводится к поиску минимального пути на графе такого же типа, как и рассмотренный в предыдущем пункте. В графе каждая дуга получает вес $f(|l_1 - l_2|)$ – работу по сжатию или растяжению, а для каждой вершина – вес $f(|j_1 - j_2|)$ – работу по изменению угла.

8.3 Задача сравнения речевых команд



В обработке речи можно выделить следующие основные направления:

Распознавание отдельных слов (IWR – Isolated Word Recognition),

Распознавание слитной речи (CSR – Continuous Speech Recognition).

CDR – Speaker Dependent Recognition,

SIR – Speaker Independent Recognition.

Ядром IWR-систем является совокупность эталонов и мера. Отрезок сигнала (см. рис.) $[0, T]$ разбивается на сегменты, т.е. сигнал квантуется (с перекрытием). С каждым сегментом связывается вектор коэффициентов Фурье.

Обработка звука происходит в два этапа.

Первый этап. Строим цепочку $r(i)$, $i = 1, \ldots, I$ – разговорные сегменты. Далее строим преобразование Фурье с разбиением на $t_f = 512$ отрезков. Обозначим через $x_i(n)$, $n = 0, \ldots, 511$ – отчеты для i-ого сегмента, $i = 1, .., I$. Тогда

$$X_i(m) = \frac{1}{\sqrt{512}} \sum_{n=0}^{511} x_i(n) \cdot \exp\left(-j \frac{2p}{512}\right), \; m = 0, \ldots, 511.$$

Рассмотрим первые l, $l << t_f$ (пусть $l \approx 50$), коэффициентов Фурье в качестве вектора признаков:

$$r(i) = \begin{bmatrix} X_i(0) \\ X_i(1) \\ \vdots \\ X_i(l-1) \end{bmatrix}, \; i = 1, \ldots, I.$$

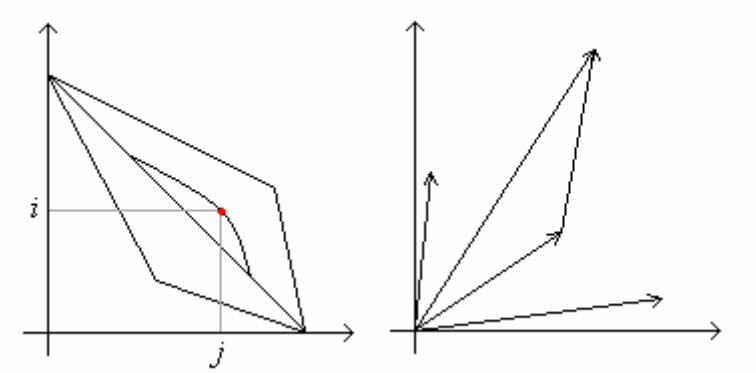
Второй этап. Определяем ограничения в графе соответствия сегментов эталонной и тестируемой команд.

Глобальные ограничения – ограничения поля для оптимального маршрута, например, $|i - j| \le k$ (рис. слева).

Локальные ограничения – монотонность на сети маршрутов (рис. справа),

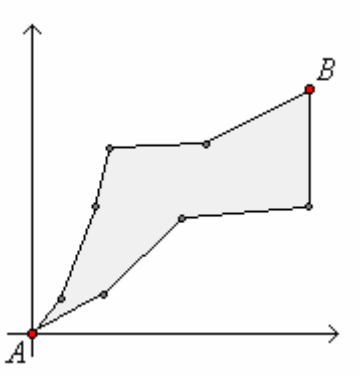
Ограничения конечной точки,

Стоимость d – Евклидово расстояние между $r(i_k), t(j_k)$: $d(i_k, j_k \mid i_{k-1}, j_{k-1}) = \|r(i_k) - t(j_k)\| = d(i_k, j_k)$.



Таким образом, и эта задача также сводится к поиску кратчайшего пути на графе.

8.4 Динамическое программирование



Задача поиска кратчайшего пути на графе может быть решены методом динамического программирования. Пусть (i_0, j_0) – начальный узел (отправной город), (i_f, j_f) – конечный узел (город – пункт назначения). Тогда задача состоит в поиске оптимального маршрута через промежуточные узлы (города):

$$(i_0, j_0) \xrightarrow{opt} (i_f, j_f).$$

Пусть (i, j) – промежуточный узел. Тогда по принципу оптимальности Беллмана имеем

$$(i_0, j_0) \xrightarrow{opt} (i_f, j_f) = \left((i_0, j_0) \xrightarrow{opt} (i, j)\right) \oplus \left((i, j) \xrightarrow{opt} (i_f, j_f)\right),$$

причем $D_{\min}(i_k, j_k) = \min\limits_{(i_{k-1}, j_{k-1})} \left[D_{\min}(i_{k-1}, j_{k-1}) + d(i_k, j_k \mid i_{k-1}, j_{k-1})\right].$

9 КОНТЕКСТНО-ЗАВИСИМАЯ КЛАССИФИКАКЦИЯ

9.1 Постановка задачи

Рассмотренные ранее задачи предполагали, что нет зависимости между различными классами, т.е. имея вектор x из класса Ω_i, мы могли получить следующий вектор из любого класса. Далее мы будем предполагать зависимость классов, т.е. классификация каждого нового вектора осуществляется в зависимости от классификации предыдущих векторов. Выбор класса, к которому следует отнести вектор, зависит от его собственного значения, значений других векторов, существующих отношений между различными классами.

Такие задачи возникают во многих приложения: распознавание речи, обработка изображений и др.

Эта классификация называется контекстно-зависимой.

Отправной точкой является Байесовский классификатор. Но зависимость между различными классами требует более общей формулировки проблемы. Общая информация, которая присутствует в векторах, требует, чтобы классификация была выполнена с использованием всех векторов одновременно и также была организованна в той же последовательности, в которой получена в экспериментах. Поэтому мы будем называть вектор признаков *наблюдением*, выстроенным в последовательность $x_1, x_2, ..., x_N$ из N наблюдений.

9.2 Байесовский классификатор

Пусть $X = (x_1, x_2, ..., x_N)$ – последовательность N наблюдений и w_i, $i = 1, 2, ..., M$ – классы, в которые эти вектора можно классифицировать. Пусть также $\Omega_i = (w_{i_1}, w_{i_2}, ..., w_{i_N})$ – одна из возможных последовательностей соответствия классов последовательности наблюдений, где $i_k \in \{1, 2, ..., M\}$, $k = 1, 2, ..., N$. Общее число таких последовательностей классов Ω_i есть M^N. Задача заключается в том, чтобы решить, к какой последовательности классов отнести последовательность наблюдений. Это эквивалентно отнесению x_1 к w_{i_1}, x_2 к w_{i_2} и т.д.

Подходом к решению проблемы является рассмотрение каждой конкретной последовательности X как расширенного вектора признаков на Ω_i, $i = 1, 2, ..., M^N$ как на возможных классах. В данном случае Байесовское правило $P(\Omega_i \mid X) > P(\Omega_j \mid X)$, при $i \neq j$ эквивалентно $P(\Omega_i)\, p(X \mid \Omega_i) > P(\Omega_j)\, p(X \mid \Omega_j)$, при $i \neq j$.

Рассмотрим, как это правило выглядит для некоторого типичного класса контекстно-зависимых моделей.

9.3 Модель Марковской цепи

Одна из наиболее используемых моделей, описывающих зависимость классов, является правило Марковской цепи. Если $w_1, w_2, ..., w_N$ есть последовательность классов, то Марковская модель предполагает, что

$$P(w_{i_k} \mid w_{i_{k-1}}, w_{i_{k-2}}, ..., w_{i_1}) = P(w_{i_k} \mid w_{i_{k-1}})$$

Тогда зависимость классов ограничивается только внутри двух последовательных классов. Такой класс моделей называется *Марковской моделью первого порядка*. Возможны обобщения на второй, третий и т.д. порядок.

Другими словами, даны наблюдения $x_{k-1}, x_{k-2}, ..., x_1$, принадлежащие классам $w_{i_{k-1}}, w_{i_{k-2}}, ..., w_{i_1}$ соответственно. Вероятность того, что наблюдение x_k на шаге k принадлежит классу w_{i_k}, зависит только от того класса, к которому принадлежит наблюдение $x_{i_{k-1}}$ на шаге $k-1$.

$$P(\Omega_i) = P(w_{i_1}, w_{i_2}, ..., w_{i_N}) = P(w_{i_N} \mid w_{i_{N-1}}, w_{i_{N-2}}, ..., w_{i_1}) \cdot P(w_{i_{N-1}} \mid w_{i_{N-2}}, w_{i_{N-3}}, ..., w_{i_1}) \cdot ... \cdot P(w_{i_1})$$

или

$$P(\Omega_i) = P(w_{i_1}, w_{i_2}, ..., w_{i_N}) = P(w_{i_1}) \cdot \prod_{k=2}^{N} P(w_{i_k} \mid w_{i_{k-1}}) \quad (*)$$

Сделаем два общих предположения:

1) в последовательности классов наблюдения статистически независимы;
2) функция плотности вероятностей в одном классе не зависит от других классов.

Это означает, что зависимость существует только на последовательности, в которой классы встречаются, но внутри классов наблюдений “подчиняются” собственным правилам. Таким образом, получаем, что

$$P(X \mid \Omega_i) = \prod_{k=1}^{N} p(x_k \mid w_{i_k}) \quad (**)$$

Комбинируя $(*)$ и $(**)$, получаем Байесовское правило в виде следующего утверждения.

Байесовское правило: для последовательности наблюдений векторов $X = (x_1, x_2, ..., x_N)$ проводим их классификацию в соответствующие последовательности классов $\Omega_i = (w_{i_1}, w_{i_1}, ..., w_{i_N})$ так, чтобы величина

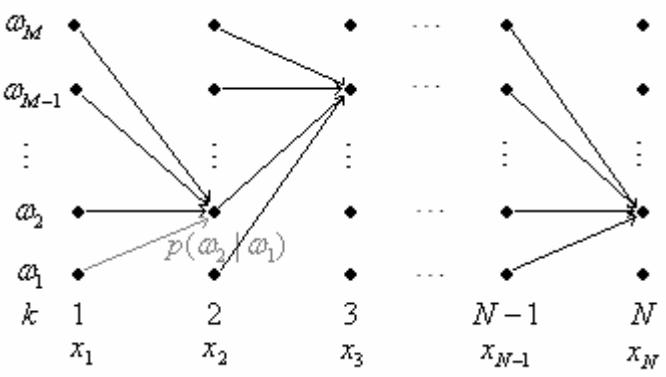
$$p(X \mid \Omega_i) \cdot P(\Omega_i) = P(w_{i_1}) \cdot p(x_1 \mid w_{i_1}) \cdot \prod_{k=2}^{N} P(w_{i_k} \mid w_{i_{k-1}}) \cdot p(x_k \mid w_{i_k}) \quad (***)$$

была максимальной.

Поиск требует вычисления последнего выражения для каждого Ω_i, $i = 1, 2, ..., M^N$, что, в свою очередь, требует $O(NM^N)$ умножений, а это очень много. Но существуют пути экономии вычислений. Если в Ω_i и Ω_j отличаются только последние классы, т.е. $w_{i_k} = w_{j_k}$, при $k = 1, 2, ..., N-1$ и $w_{i_N} \neq w_{j_N}$, то большая часть вычислений дублируется.

9.4 Алгоритм Витерби (Viterbi)

Пусть задано N столбцов; каждая точка в столбце соответствует одному из M возможных классов $w_1, w_2, ..., w_M$; столбцы соответствуют наблюдениям x_k, $k = 1, 2, ..., N$. Стрелками обозначены переходы от одного класса к другому в последовательности получения наблюдений. Каждая последовательность классов Ω_i соответствует конкретному маршруту последовательных переходов. Каждый переход от i-го класса к j-му характеризуется вероятностью $P(w_j \mid w_i)$, которая предполагается известной. Предположим, что эти вероятности одинаковы для всех k. Далее предположим, что условные вероятности – плотности $p(x_k \mid w_i)$, $k = 1, 2, ..., N$,



$i = 1, 2, ..., M$ – также известны. Тогда задача максимизации (***) ставится как поиск последовательности переходов.

Пусть $\hat{d}(w_{i_k}, w_{i_{k-1}}) = P(w_{i_k} \mid w_{i_{k-1}}) \cdot p(x_k \mid w_{i_k})$ – цена, связанная с переходом $(w_{i_{k-1}}, w_{i_k})$. Начальное условие при $k = 1$ есть $\hat{d}(w_{i_1}, w_{i_0}) = P(w_{i_1}) \cdot p(x_1 \mid w_{i_1})$. Учитывая данные предположения, получаем общую формулу, которую нужно оптимизировать:

$$\hat{D} = \prod_{k=1}^{N} \hat{d}(w_{i_k}, w_{i_{k-1}})$$

или, логарифмируя, имеем

$$\ln(\hat{D}) = \sum_{k=1}^{N} \ln \hat{d}(w_{i_k}, w_{i_{k-1}}) = \sum_{k=1}^{N} d(w_{i_k}, w_{i_{k-1}}) = D$$

Используем принцип Беллмана:

$$D_{\max}(w_{i_k}) = \max_{i_{k-1}=1,2,...,M} \left[D_{\max}(w_{i_{k-1}}) + d(w_{i_k}, w_{i_{k-1}}) \right], \text{ при } D_{\max}(w_{i_0}) = 0.$$

Обозначим через

$$w_{i_N}^* = \arg\max_{w_{i_N}} D_{\max}(w_{i_n}).$$

Получаем обратный ход для вычисления $w_{i_k}^*$. Получаем число операций $O(NM^2)$, что существенней меньше $O(NM^N)$. Данная процедура динамического программирования известна как алгоритм Витерби.

9.5 Скрытые Марковские модели

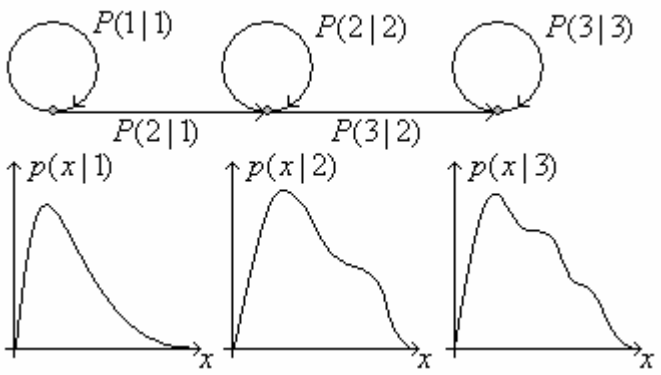
Теперь рассмотрим системы, в которых состояния напрямую не наблюдаются и могут быть лишь оценены из последовательности наблюдений с помощью некоторой оптимизационной техники. Этот тип Марковских моделей известен как скрытые Марковские модели (HMM). HMM – это тип стохастической аппроксимации нестационарных стохастических последовательностей со статистическими свойствами, которые подвергаются различным случайным переходам среди множества различных стационарных процессов. Иными словами, HMM моделирует последовательность наблюдений как кусочно-стационарный процесс.

Такие модели широко используются в распознавании речи. Рассматриваются так называемые *высказывания* – это может быть слово, часть слова, даже предложение или параграф. Статистические свойства речевого сигнала внутри высказывания подвергаются серии переходов. Например, слово содержит порцию гласных и согласных звуков. Они характеризуются различными статистическими свойствами, которые в свою очередь отражены в переходах в речевых сигналах от одной к другой. Такие примеры дает распознавание рукописного текста, распознавание текстур, где успешно применяется HMM.

HMM есть в основе своей конечный автомат, который генерирует строку наблюдений – последовательность векторов наблюдений $x_1, x_2, ..., x_N$. Таким образом, HMM содержит k состояний, и строка наблюдений получается как результат последовательных переходов из одного состояния i в другое состояние j. Нам подходит так называемая модель “машины Моора”, в соответствии с которой наблюдения получаются как результаты (выходы) из состояний на прибытие (по переходу) в каждом состоянии.

Пример. HMM с тремя состояниями. Стрелки обозначают переходы. Такая модель может соответствовать короткому слову с тремя различными стационарными частями, например, для слова “оса”.

Модель предоставляет информацию о последовательных переходах между состояниями $P(i \mid j)$, $i, j = 1,2,3$. Такой тип HMM известен как “слева-направо”, поскольку индекс состояний определяется выделенным числом фонем в одном слове. В действительности, несколько состояний (обычно 3 или 4) используется для каждой фонемы.



10 СЕЛЕКЦИЯ ПРИЗНАКОВ

10.1 Задача селекции признаков

Рассмотрим этапы решения задачи распознавания образов:

- Генерация признаков – выявление признаков, которые наиболее полно описывают объект.
- Селекция признаков – выявление признаков, которые имеют наилучшие классификационные свойства для конкретной задачи.
- Построение классификатора.
- Оценка классификатора.

Пусть

$X \in R^m$ – множество признаков,

$Y \in R^l$ – множество признаков, которые нужно отобрать в процессе селекции, причем $l < m$.

Тогда задача селекции задается следующим образом: $X \to Y$.

10.1.1 Постановка задачи селекции признаков.

Пусть задан вектор признаков $X \in R^m$. Среди них необходимо выбрать наиболее информативные, т.е. получить новый вектор признаков $Y \in R^l$, причем $l < m$.

Определение. *Процедура выделения из множества признаков меньшего подмножества с наилучшим сохранением информативности для классификации называется селекцией признаков.*

Суть выбора признаков – это выделение признаков, которые приводят к большим расстояниям между классами и к малым внутри классов.

Зачем нужна селекция признаков?

Основной мотивацией для сокращения числа признаков является уменьшение вычислительной сложности. Наряду с признаками, имеющими низкие классификационные способности весьма вероятна ситуация двух хороших признаков (с почти равными классифицирующими способностями), сильно коррелированных между собой.

Вторая причина для уменьшения числа признаков – повышение общности классификатора.

10.1.2 Общность классификатора.

Пусть

N – число прецедентов,

k – число степеней свободы классификатора (для нейронной сети – это количество синоптических весов).

Ясно, что чем больше степеней свободы, тем легче настроить классификатор. Обозначим через $\frac{N}{k}$ характеристику общности. Тогда получаем, что, чем больше $\frac{N}{k}$, чем выше общность классификатора.

Чем больше признаков, тем больше k. Поэтому при ограниченном N уменьшение числа признаков согласуется с уменьшением k, т.е. с усложнением настройки классификатора.

Различают скалярную и векторную селекцию признаков. При скалярной селекции рассматривается отдельно один признак из данного множества. Таким образом, получили

одномерную задачу. При векторной селекции одновременно исследуются свойства группы признаков.

10.2 Предобработка векторов признаков

Пусть задано множество признаков.

Селекции признаков предшествует предобработка, позволяющая привести их в единый масштаб измерений и произвести некоторые дополнительные улучшения..

Основные операции предобработки описываются следующими тремя пунктами.

2.1. Удаление выбросов – точек, лежащих “очень далеко” от среднего значения. Обычно измеряется расстояние в средних отклонениях, например, $2s \sim 95\%$, $3s \sim 99\%$ для нормального распределения.

2.2. Нормализация. Признаки, имеющие большие значения, могут влиять на классификатор сильнее остальных, что искажает правильность классификатора. Поэтому необходимо уменьшить их влияние путем, который носит название нормализации. Пусть

x_i – прецедент,

$x_i = (x_{i_1},...., x_{i_l})$ – признаки.

Тогда

$$\bar{x}^{(k)} = \frac{1}{N}\sum_{i=1}^{N} x_{i_k},\ k = 1,2,...,l$$

есть усреднение признака (фактически его математическое ожидание).

Обозначим через

$$(s^{(k)})^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_{i_k} - \bar{x}^{(k)})^2$$

оценку разброса. Тогда нормализованные признаки задаются следующим образом.

$$\tilde{x}_{i_k} = \frac{x_{i_k} - \bar{x}^{(k)}}{s^{(k)}}.$$

2.3. Пропуск данных (потери). По многим прецедентам могут быть известны не все признаки. В таком случае, если данных много, то можно отобрать те у которых набор признаков одинаковый. Если же отбрасывать признаки нельзя, то их можно дополнить, например, с помощью эвристик.

10.3 Селекция на основе проверки статистических гипотез

Этот метод относится к скалярной селекции признаков.

Рассмотрим значение признаков как реализацию случайных величин. Методами математической статистики можно выяснить их распределение. Если распределение совпадает для разных классов, то признак не различает эти классы; если распределения различны, то признак их различает. Такова суть метода селекции на основе проверки статистических гипотез.

Таким образом, задача скалярно селекции на основе проверки статистических гипотез решается путем оценивания дискриминантной способности каждого отдельного признака.

10.3.1 Постановка задачи

Пусть x признак. Пусть также известны его значения для разных классов Ω_1 и Ω_2. Тогда задача состоит в оценке, существенно ли различаются распределения признака для разных классов.

Примем следующие соглашения. Обозначим через H_0 и H_1 две гипотезы:

H_0 – значения признаков отличаются существенно – нуль-гипотеза.

H_1 - значения признаков отличаются несущественно – альтернативная гипотеза.

10.3.2 Общая теория проверки гипотез

Пусть

x – случайная величина с известной плотностью и неизвестным параметром q,

$x_1,...,x_N$ – экспериментальные значения x,

$q = f(x_1,...,x_N)$ – статистика, где плотность есть $P_q(q,q)$.

Тогда гипотезы примут вид: $H_0: q \neq q_0$ и $H_1: q = q_0$. Задача состоит в построении интервала D такого, что в D высокая вероятность выполнения гипотезы H_0.

Пусть $\overline{D} = R \setminus D$ – дополнение к D. Тогда, если q попадает в D, то принимается H_0, иначе отвергается.

Назовем вероятностью ошибки решения следующую величину:

$P(q \in \overline{D} \mid H_0) = r$,

причем r выбирается заранее и называется *уровнем значимости*:

$$r = \int_{\overline{D}} P_q(q \mid H_0)dq.$$

Случай известной дисперсии.

Пусть

$Ex = m$ – неизвестное среднее,

$E((x-m)^2) = d^2$ – известная дисперсия.

У нормализованных признаков дисперсия равна единице, следовательно, дисперсия известна.

Оценка m задается следующим образом:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

причем $E\bar{x} = m$, $m = q$, $\tilde{m} = q_0$. Тогда гипотезы примут вид: $H_0: m = \tilde{m}$ и $H_1: m \neq \tilde{m}$.

В данном случае статистика имеет вид:

$$q = \frac{\bar{x} - \tilde{m}}{\left(\frac{d}{\sqrt{N}}\right)},$$

где $\frac{d}{\sqrt{N}}$ – среднеквадратичное отклонение для $\bar{x}$.

По центральной предельной теореме имеем:

$$P_{\bar{x}}(x) = \frac{\sqrt{N}}{\sqrt{2p}\,d}\exp\left(-\frac{N(x-\tilde{m})^2}{d^2}\right).$$

Далее, $q \sim N(0,1)$. Следовательно, находим доверительный интервал D по r: т.к. $\Phi(x_r) = r$, то $D = \lfloor -x_r; x_r \rfloor$. Для уровня значимости r интервал принятия гипотезы $D = \lfloor -x_r; x_r \rfloor$ выбирается как интервал, в котором q лежит с вероятностью 1- r.

Случай неизвестной дисперсии. Если дисперсия неизвестна, то оценка $\tilde{d}^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2$ есть несмещенная оценка дисперсии и $q = \frac{\bar{x} - m}{\frac{\tilde{d}}{\sqrt{N}}}$ есть статистика (не гауссова величина).

Если $\bar{x}$ гауссова величина, то q имеет *t*-распределение Стьюдента с $N-1$ степенями свободы. Тогда доверительный интервал $D = \left[-x_r ; x_r\right]$ вычисляется по таблицам.

10.3.3 Приложение к селекции признаков

Наша основная забота теперь – проверить отличие m_1 и m_2 между средними значениями признака в двух классах.

Пусть

$x_1,.....,x_N$ – значение признака в первом классе со средним m_1. Соответственно,

$y_1,.....,y_N$ - значение признака во втором классе со средним m_2.

Предположим, что дисперсии одинаковы в обоих классах. Пусть m_1 и m_2 – средние для значений признаков в первом и втором классе соответственно. Тогда соответствующие гипотезы имеют вид:

$H_0: \Delta m = m_1 - m_2 = 0$,

$H_1: \Delta m \neq 0$.

Для решения о близости двух классов мы проверим эти гипотезы.

Пусть $x_i = x_i - y_i$.

Гипотеза о равенстве параметров распределения говорит о попадании в этот интервал величины $x = x - y$, где x и y – случайные величины, причем $E(x) = m_1 - m_2$ и $d_x^2 = 2d^2$

Для случая неизвестной дисперсии статистика имеет вид:

$$q = \frac{(\bar{x} - \bar{y}) - (m_1 - m_2)}{s_x\sqrt{\frac{2}{N}}}$$

и несмещенная оценка дисперсии записывается следующим образом:

$$s_x^2 = \frac{1}{2N-2}\left(\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2 + \sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2\right)$$

s_x^2 имеет c^2 распределение с $2N-2$ степенями свободы.

Если x, y – нормально распределенные с одинаковыми дисперсиями, тогда случайная величина q имеет *t*-распределение Стьюдента с $2N-2$ степенями свободы.

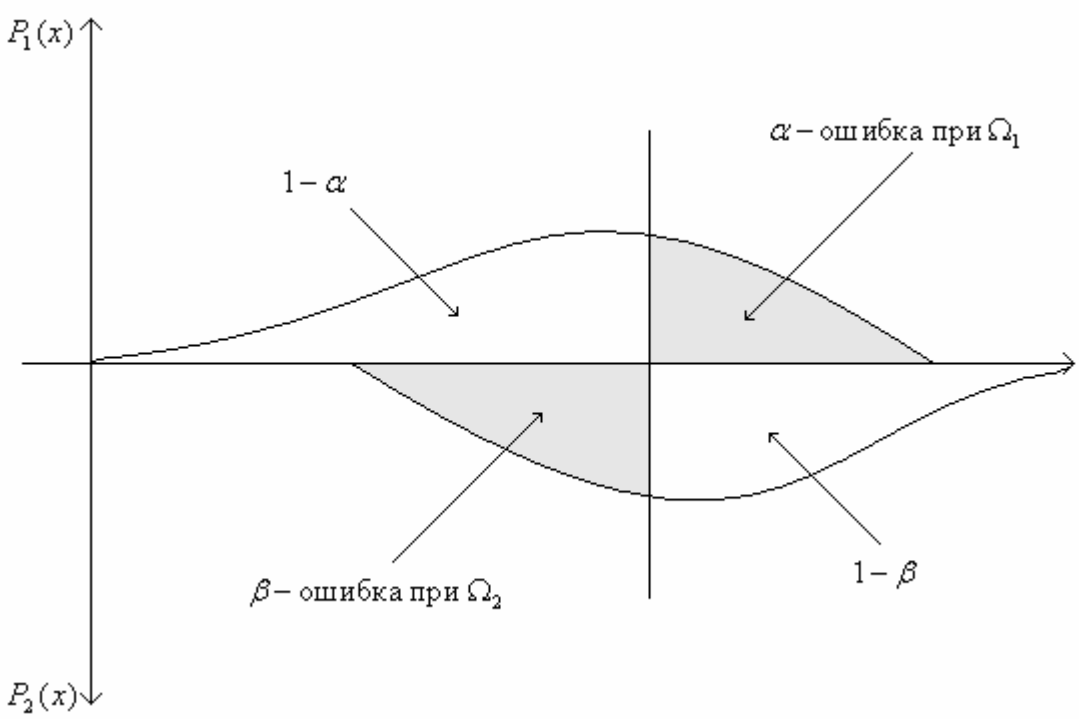
Если числа прецедентов в обоих классах не совпадают, то формулы модифицируются.

10.3.4 Мера различия плотностей признаков

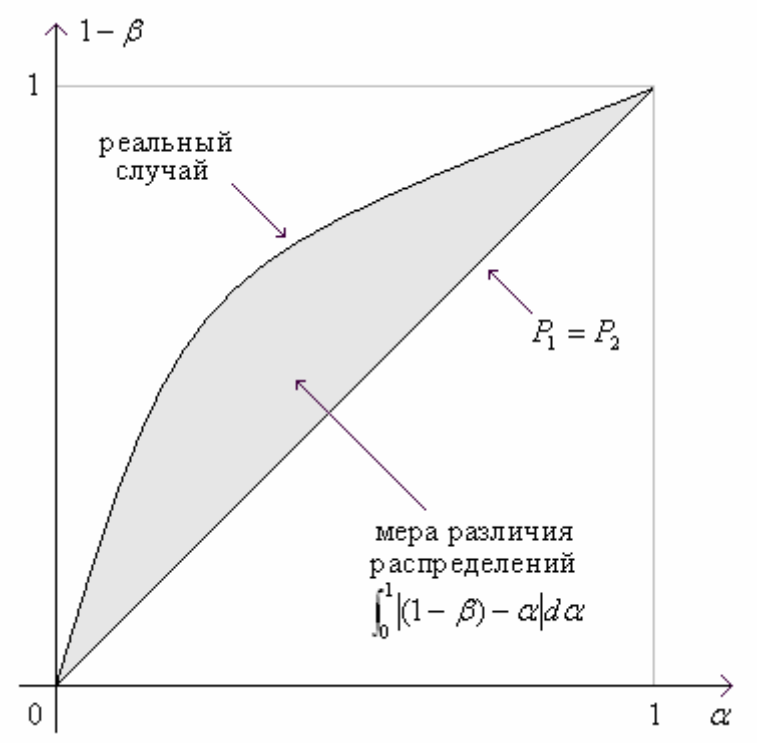
Если средние значения различаются, и дисперсии очень большие, то признак может не обладать хорошими разделительными свойствами. Средние значения могут не совпадать и хорошо разделяться, но при больших дисперсиях признак становиться неудовлетворительным. Поэтому нужна информация о перекрытиях между классами. Рассмотрим способ анализа информации о перекрытии плотностей распределения признаков.

Принимать решение, к какому классу отнести объект будем по значению t. Пусть $a(t)$ и $b(t)$ – ошибки при пороге классификации t.

Идеальным случаем является случай, когда $P_1(x) = P_2(x)$, т.е. у признака нет селективных способностей: $a + b = 1$. Рассмотрим параметрическую кривую: $(a(t), 1 - b(t))$.



Тогда в качестве меры различия распределений можно использовать площадь разности между кривой реального случая и идеального случая, которая выражается следующим интегралом $\int_0^1 |(1-b)-a|da$.



10.4 Векторная селекция признаков. Мера отделимости классов

Ранее обсуждались дискриминантные свойства отдельных признаков. Теперь рассмотрим дискриминантные способности векторов признаков.

Пусть

X – множество признаков,

X_r – подмножество из r признаков,

$C(X_r)$ – мера отделимости классов на множестве признаков X_r.

Тогда задача выглядит следующим образом:

$C(X_r) \to \max\limits_{X_r \le X}$.

Существует различные подходы к описанию меры отделимости. Мы рассмотрим два таких подхода: дивергенцию и матрицу рассеивания.

10.4.1 Дивергенция

Будем рассматривать Байесовское правило:

$$P(\Omega_1 \mid x) >< P(\Omega_0 \mid x),$$

по которому выбирается Ω_1. Ошибка классификации задается следующим интегралом:

$$P_l = P(\Omega_0) - \int_{R_1} \left[P(\Omega_1 \mid x) - P(\Omega_0 \mid x)\right] p(x) dx.$$

Пусть

R_1 – область решения по классу Ω_1; если $x \in R_1$, то класс Ω_1;

$R_1 \cup R_0 = R^l$ – все пространство признаков,

$P(\Omega_1 \mid x) - P(\Omega_0 \mid x)$ – очень важный показатель разделимости классов. От этой разности зависит ошибка классификации;

$\dfrac{P(\Omega_1 \mid x)}{P(\Omega_0 \mid x)}$ – информация о разделяющих свойствах вектора признаков (другая форма этого показателя).

Информацию о разделяющих свойствах вектора признаков можно записать следующим образом:

$$\ln\left(\frac{P(\Omega_1)}{P(\Omega_0)} \cdot \frac{p(x \mid \Omega_1)}{p(x \mid \Omega_0)}\right).$$

Если

$$\frac{P(\Omega_1)}{P(\Omega_0)} = const,$$

то

$$\int_{-\infty}^{+\infty} \left[\ln \frac{p(x \mid \Omega_1)}{p(x \mid \Omega_0)}\right] p(x \mid \Omega_1) dx = D_{01},$$

где D_{01} – характеристика отделимости. Аналогично:

$$D_{10} = \int_{-\infty}^{+\infty} \left[\ln \frac{p(x \mid \Omega_0)}{p(x \mid \Omega_1)}\right] p(x \mid \Omega_0) dx.$$

Обозначим через $d_{01} = D_{01} + D_{10}$ дивергенцию разделения классов по вектору признаков x. Аналогично для случая многоклассовой задачи d_{ij} – дивергенция классов Ω_i и Ω_j. Тогда

$$d = \sum_{i=0}^{k}\sum_{j=0}^{k} P(\Omega_i)P(\Omega_j)d_{ij} \ .$$

Дивергенция есть мера расстояния между плотностями. Она имеет следующие свойства: $d_{ij} \geq 0$; $d_{ij}=0$ при i=j; $d_{ij}=d_{ji}$. Если компоненты вектора признаков независимы, можно показать, что $d_{ij}(x_1,x_2,\ldots,x_l)=Sd_{ij}(x_r)$.

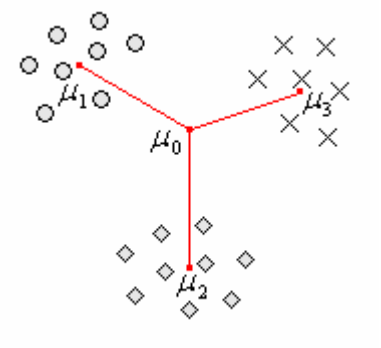
Дивергенция учитывает различия и в средних, и в дисперсии. Однако, она очень чувствительна к разности средних, что затрудняет использование.

10.4.2 Мера на основе матриц рассеивания

Главный недостаток многих критериев отделимости классов – сложность вычисления, если не проходит предположение о гауссовых плотностях. Мы рассмотрим простой критерий, не требующий нормальности распределения, построенный на информации, относящейся к тому, как вектора признаков разбросаны в пространстве.

Пусть

S_i – матрица ковариации, $S_i = E\left[(x-m_i)(x-m_i)^T\right]$, где x – вектор признаков и m_i – среднее значение по x, принадлежащим данному классу, $m_i = E(x \mid \Omega_i)$.



$S_W = \sum_{i=1}^{M} P_i S_i$ – матрица внутриклассового рассеивания есть мера дисперсии признаков, где P_i – априорная вероятность данного класса, $P_i = P(\Omega_i)$.

$S_b = \sum_{i=1}^{M} P_i (m_i - m_0)(m_i - m_0)^T$ – матрица внеклассового рассеивания, где $m_0 = \sum_{i=1}^{M} P_i m_i$ – общее среднее – разброс относительно общего среднего всех классов (центр тяжести).

S_m – смешанная матрица рассеивания (ковариация относительно общего среднего), $S_m = E\left[(x-m_0)(x-m_0)^T\right] = S_w + S_b$.

Определение. *Следом (обозначается* $trace$*) называется сумма диагональных элементов матрицы.*

Пример. Пусть задана матрица $A = \left\| a_{ij} \right\|_{l\times l}$. Тогда $trace(A) = \sum_{i=1}^{l} a_{ii}$.

Пусть

$J_1 = \dfrac{trace(S_m)}{trace(S_w)}$ – критерий, принимающий большие значения, когда образы хорошо кластеризуются вокруг своих средних в границах каждого класса и кластеры разных классов хорошо разделены. Иногда вместо S_m используют S_b. Тогда получаем задачу $J_1 \to \max$.

Вместо критерия J_1 можно использовать другие критерии:

$$J_2 = \frac{|S_m|}{|S_W|} = \left|S_m S_W^{-1}\right|,$$

$$J_3 = \left\{S_m S_W^{-1}\right\}.$$

Последний критерий очень удобен на практике для аналитических выкладок.

10.4.3 Стратегия наращивания вектора признаков

Стратегия наращивания вектора признаков заключается в использовании признаков, дающих наибольший прирост меры отделимости.

Рассмотрим алгоритм наращивания вектора признаков. Пусть из l признаков нужно отобрать m.

Рассмотрим множество признаков $\{x_1\},\{x_2\},...,\{x_l\}$. Необходимо найти признак, имеющий наибольшую селективную способность. Это аналогично нахождению $\max\limits_{\{x_i\}} C(\{x_i\})$. Пусть $X_1 = \{x_j\}$, где $\{x_j\} = \arg\max\limits_{\{x_i\}} C(\{x_i\})$.

Пусть X_r – построенное множество признаков. Далее, $X \setminus X_r = \max\limits_{X \in X \setminus X_r} C(X_r \cup \{x_i\})$.

Условие остановки: $C(X_{r+1}) - C(X_r) < e$, либо $r=m$.

10.4.4 Стратегия сокращения вектора признаков

Пусть X – множество признаков.

Шаг алгоритма: набор признаков X_r, чтобы выполнялось $\max\limits_{X_i \le X_r} C(X_r \setminus \{x_i\})$ и $X_{r-1} = X_r \setminus \{x_i\}$.

Условие остановки: $|C_{t+1} - C_t| < e$, либо $r=m$.

10.4.5 Выбор стратегии

Пусть l: $1<l<m$. Если $l > \frac{1}{2}$, то используем стратегию сокращения вектора признаков. Если $l < \frac{1}{2}$, то используем стратегию наращивания вектора признаков.

В качестве альтернативы можно использовать сравнение $l-m$ с l. Если $l-m<l$, то используем стратегию сокращения вектора признаков. Если $l-m>l$, то используем стратегию наращивания вектора признаков.

Обе стратегии являются жадными.

Определение. *Стратегия называется "жадной", если она не допускает шагов возврата.*

10.4.6 Алгоритм плавающего поиска

Примером нежадной стратегии является метод плавающего поиска. Плавающий поиск базируется на стратегии вставки и исключения.

Пусть все признаки упорядочены по убыванию меры $C(\{x_i\})$. Пусть также $X_k = \{x_1,...,x_k\}$ – первые k признаков, имеющие наибольшее $C(\{x_i\})$. Тогда остальные признаки строятся следующим образом:

$Y_{l-k} = X \setminus X_k$,

$X_{k+1} = \{X_k, x_{k+1}\}$.

Предположим, что построены множества $X_2, X_3, ..., X_{k-1}$. Рассмотрим алгоритм плавающего поиска.

Шаг 1. Вставка.

Добавление признаков: $x_{k+1} = \arg\max C(\{X_k, y\})$ и $X_{k+1} = \{X_k, x_{k+1}\}$.

Шаг 2. Проверка

$x_r = \arg\max C(X_{k+1} \setminus \{x_s\}), x_s \in X_{k+1}$, где x_r – признак дающий наименьший вклад (минимальные потери при выбросе).

Если $r = k+1$, то увеличиваем k и переходим на шаг 1.

Если $r \neq k+1$ и $C(X_{k+1} \setminus \{x_r\}) < C(x_k)$, то переходим на шаг 1.

Если $k = 2$, то $X_k = X_{k+1} \setminus \{x_r\}$ и переходим на шаг 1.

Шаг 3. Исключение x_r.

$X'_k = X_{k+1} \setminus \{x_r\}$.

Поиск наименее значительного элемента в новом множестве $x_s = \arg\max C(X'_k \setminus \{y\}), y \in X'_k$.

Если $C(X'_K \setminus \{x_s\}) < C(X_{k-1})$, то $X_k = X'_k$ и переходим на шаг 1.

Если $X'_{k-1} = X'_k - \{x_s\}$, то уменьшаем k на 1.

Если $k = 2$, то $X_k = X'_k$, $C(X_k) = C(X'_k)$ и переходим на шаг 1.

Переходим на шаг 3.

10.5 Оптимальная селекция признаков

Существуют две формы использования критериев (мер отделимости классов): “пассивная” и “активная”. Пассивная селекция – это работа с уже полученными признаками. Активная селекция аналогична процессу генерации признаков: она позволяет построить из исходного набора признаков новый набор меньшего размера, в котором состав признаков, вообще говоря, не является подмножеством исходного набора признаков. Все типы селекции, рассмотренные в предыдущих разделах – пассивные.

Пусть $x \in R^l$ и $y \in R^m \subset R^l$. Рассмотрим конструирование критериев с использованием активной селекции: $y = Ax$ или $y = F(x)$.

Пусть

x и y – вектора столбцы, тогда x^T, y^T – строки,

$x \in R^m$ – исходное пространство признаков,

$y \in R^l$ – результирующее пространство признаков,

A – матрица преобразования исходного пространства в результирующее,

m – число классов.

Тогда

$y = A^T x$,

или

$y_{l\times 1} = A^T x_{m\times 1}$,

следовательно, матрица $A^T_{l\times m}$ имеет размер $l \times m$.

Рассмотрим критерий $J_3 = \left\{S_m S_W^{-1}\right\}$. Будем максимизировать критерий J_3 путем выбора матрицы A. Для вектора признаков x имеем матрицы S_{xW} и S_{xb}. Для вектора признаков y имеем матрицы S_{yW} и S_{yb}.

$S_{yW} = \sum_i P_i S_{yi}$.

Проведем несколько преобразований.

$$S_{yi} = E\left[(y-m_i)(y-m_i)^T\right] = E\left[A^T(x-m_{xi})(x-m_{xi})^T A\right] = A^T E\left[(x-m_{xi})(x-m_{xi})^T\right]A = A^T S_{si} A$$

$$S_{yW} = \sum_i P_i A^T S_{xi} A = A^T \left(\sum_i P_i S_{xi}\right) A = A^T S_{sW} A$$

Аналогично: $S_{yb} = A^T S_{xb} A$. Тогда $J_3(A) = trace\left(\left(A^T S_{xW} A\right)^{-1}\left(A^T S_{xb} A\right)\right)$ – критерий разделимости вектора признаков.

Теперь необходимо преобразовать A из соображений $J_3 \to \max$. Будем искать решение из условия максисума

$$\frac{dJ_3(A)}{dA} = 0.$$

Утверждение о вычислении производной. *Пусть S_1 и S_2 - некоторые квадратные матрицы размера $m \times m$. Тогда*

$$\frac{d}{dA} trace\left\{(A^T S_1 A)^{-1}(A^T S_2 A)\right\} = -2S_1 A(A^T S_1 A)^{-1}(A^T S_2 A)(A^A S_1 A)^{-1} + 2S_2 A(A^T S_1 A)^{-1}.$$

Для получения максимума по критерию, необходимо, чтобы

$$-2S_{xW} A(A^T S_{xW} A)^{-1}(A^T S_{xb} A)(A^T S_{xW} A)^{-1} + 2S_{xb} A(A^T S_{xW} A)^{-1} = 0$$

или

$$-S_{xW} A S_{yW}^{-1} S_{yb} S_{yW}^{-1} + S_{xb} A S_{yW} = 0$$

или

$$A(S_{yW}^{-1} S_{yb}) = (S_{xW}^{-1} S_{xb}) A$$

есть условие того, что

$$\frac{dJ_3(A)}{dA} = 0.$$

Утверждение. *Пусть S_{yW} и S_{yb} – симметрические, положительно определенные матрицы. Тогда существует преобразование, приводящее одну из них к единичной, а другую к диагональной.*

Доказательство. Приведем эти преобразования

$$B^T S_{yW} B = I,$$

$$B^T S_{yb} B = D,$$

где B, I, D – матрицы размера $l \times l$.

ч.т.д.

Утверждение. *J_3 инвариантно относительно преобразований вектора y в R^l.*

Доказательство. Рассмотрим

$$\tilde{y} = B^T y = B^T A x.$$

Тогда

$$J_3(\tilde{y}) = trace\left\{S_{\tilde{y}W}^{-1} S_{\tilde{y}b}\right\} = trace\left\{(B^T S_{yW} B)^{-1}(B^T S_{yb} B)\right\} =$$

$$= trace\left\{B^{-1} S_{yW}^{-1} (B^T)^{-1} B^{-1} S_{yb} B\right\} = trace\left\{B^{-1}(S_{yW}^{-1} S_{yb} B)\right\} =$$

$$= trace\left\{(S_{uW}^{-1} S_{yb}) B B^{-1}\right\} = trace\left\{S_{yW}^{-1} S_{yb}\right\} = J_3(y).$$

Т.к. $(S_{xW}^{-1} S_{xb}) A = A(S_{yW}^{-1} S_{yb})$ – условие того, что производная равна нулю, то

$$(S_{xW}^{-1} S_{xb}) AB = A(S_{yW}^{-1} S_{yb}(B^T)^{-1} B^T) B =$$

$$= A(B(B^{-1} S_{yW}^{-1} (B^T)^{-1})(B^T S_{yb} B) = AB(B^T S_{yW} B)^{-1}(B^T S_{yb} B) = ABD$$

Используя предыдущее утверждение, подбираем матрицу B и получаем:

$$(S_{xW}^{-1} S_{xb}) AB = ABD.$$

Обозначим $AB = C$ – матрица размера $m \times l$.

ч.т.д.

Утверждение. *Если матрица* F *положительно определенная (положительно полуопределенная), то*

все собственные значения F *положительны,*

если F *симметричная, то все собственные вектора, соответствующие разным собственным значениям, ортогональны,*

для симметричной матрицы F *существует преобразование* $\Phi^T F\Phi = \Delta$, *где* Φ *состоит из собственных векторов этой матрицы или столбцы* $\Phi = [v_1, v_2, ...v_m]$ *– собственные вектора, причем* Δ *– диагональная матрица, на диагоналях которой стоят собственные значения.*

Т.к. случайные величины ортогональны, то $\Phi^T = \Phi^{-1}$.

Теперь рассмотрим алгоритм оптимальной селекции признаков:

Поиск $S_{xW}^{-1} S_{xb}$.

Поиск собственных значений и выбор l наилучших (наибольших).

Формирование матрицы C из собственных векторов, соответствующих этим собственным значениям

Вычисление $y = C^T x$.

10.6 Оптимальная селекция признаков с помощь нейронной сети

Пусть задано m признаков, x – вектор признаков. Для применения теории нейронных сетей к задаче селекции признаков немного изменим обычное представление о нейронной сети. Теперь будем рассматривать нейронную сеть с линейными функциями активации. Таким образом, теперь вектор признаков, попавший на вход нейронной сети, просто суммируется и подается на выход, т.е. выход нейрона превращается в обычную сумму.

Рассмотрим так называемую автоассоциативную сеть. Сеть имеет l входных и l выходных узлов и единственный скрытый слой с m узлами и линейными функциями активации. В процессе обучения выходы сети те же, что и входы. Такая сеть имеет единственный максимум и выходы скрытого слоя определяют проекцию l -мерного пространства на m-мерное подпространство.

Интерес представляет выходной слой из l нейронов. Если восстанавливать исходный вектор с целью максимального правдоподобия, то получим задачу квадратичного программирования с одним экстремумом.

11 МЕТОДЫ ГЕНЕРАЦИИ ПРИЗНАКОВ

11.1 Генерация признаков на основе линейных преобразований

В данном разделе рассматриваются способы генерации признаков через линейные преобразования исходных измерений образов. Целью такой генерации признаков является сокращение информации до “значимой”, т.е. надо просто преобразовать исходное множество измерений в новое множество признаков. Обычно задача состоит в выделении низкочастотных компонент, содержащих основную информацию.

11.1.1 Базисные вектора

Пусть

$x(0), x(1), ..., x(N-1)$ – множество исходных измерений,

$X^T = [x(0), ..., x(N-1)]$ – соответствующий вектор столбец.

Рассмотрим унитарную матрицу $A_{N\times N}$. Для действительной матрицы $A_{N\times N}$ условие унитарности обозначает, что матрица $A_{N\times N}$ ортогональная, т.е. $A_{N\times N}{}^{-1} = A_{N\times N}^T$. Для комплексной матрицы $A_{N\times N}$ условие унитарности обозначает, что $A_{N\times N}{}^{-1} = A_{N\times N}^H$, где матрица $A_{N\times N}^H$ - транспонированная (сопряженная).

Пусть

$$y = A^H X = \begin{pmatrix} a_0^H \\ a_1^H \\ \mathbb{M} \\ a_{N-1}^H \end{pmatrix} \cdot X ,$$

где $a_0^H, a_1^H, ..., a_{N-1}^H$ – строки из транспонированных столбцов a_i и $A = (a_0, a_1, ..., a_{N-1})$. Тогда

$$x = (AA^{-1})x = (AA^H)x = AA^H x = Ay = \sum_{i=0}^{N-1} y(i) a_i .$$

Вектора a_i называются *базисными векторами.* Таким образом, в силу ортогональности a_i между собой, $y(i)$ – это проекция вектора x на базисные вектора.

11.1.2 Случай двумерных образов

Пусть $x(i, j)$, $i, j = 0,1,..., N-1$ – двумерные измерения. Очевидно, что представление его в виде вектора размерности N^2 неэффективно. Альтернативой является преобразование x через базисные матрицы.

Пусть $U_{N\times N}$ и $V_{N\times N}$ – унитарные матрицы. Определим матрицу преобразования X в Y:

$Y = U^H X V$.

Учитывая, что $UU^H = I$ и $VV^H = I$, имеем

$X = UYV^H$.

Следовательно

$$X = \sum_{i=0}^{N-1}\sum_{J=0}^{N-1} Y(i, j) \cdot u_i \cdot v_j{}^H \quad (*)$$

Пусть

$U = (u_0, u_1, ..., u_{N-1})$, где u_i – вектор-столбец,

$V = (v_0, v_1, ..., v_{N-1})$, где v_j^H – вектор-строка.

Тогда

$$A_{ij} = u_i v_j^H = \begin{pmatrix} u_{i,0}v_{j,0}^* & u_{i,0}v_{j,1}^* & \mathbf{L} & u_{i,0}v_{j,N-1}^* \\ u_{i,1}v_{j,0}^* & u_{i,1}v_{j,1}^* & \mathbf{L} & u_{i,1}v_{j,N-1}^* \\ \mathbb{M} & \mathbb{M} & \mathbf{O} & \mathbb{M} \\ u_{i,N-1}v_{j,0}^* & u_{i,N-1}v_{j,1}^* & \mathbf{L} & u_{i,N-1}v_{j,N-1}^* \end{pmatrix}.$$

Таким образом (∗) есть выражение x в терминах N^2 базисных матриц. Если Y – диагональная, то $X = \sum_{i=0}^{N-1} Y(i,i) \cdot u_i \cdot v_i^H$ – это разложение по базисным матрицам или образам.

Также возможна следующая запись:

$Y(i, j) = \langle X, A_{ij} \rangle$.

Тогда

$\langle A, B \rangle = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A(m,n) \cdot B^*(m,n)$.

11.2 Преобразование Карунена-Лоева

Пусть x – вектор измерений образа. Целью преобразования является построение такого вектора признаков, что

$E[y(i)y(j)] = 0$ при $i \neq j$.

т.е. чтобы признаки были взаимно некоррелированны.

Пусть

A – матрица базисных векторов,

y и x – вектора-столбцы.

Будем считать, что

$y = A^T x$.

Обозначим $R_y = E[yy^T]$, тогда

$R_y = E[yy^T] = E[A^T xx^T A] = A^T R_x A$,

где R_x – симметричная матрица и ее собственные вектора ортогональны.

Выберем в качестве a_i собственные вектора матрицы R_x. Тогда R_y – диагональная матрица, у которой на диагонали стоят собственные значения R_x: l_i, $i = 0,1,...,N-1$. Таким образом

$R_y = A^T \cdot R_x \cdot A = \Lambda$.

Если R_x положительно определенная матрица, то собственные значения $l_i > 0$, $i = 0,1,...,N-1$.

Описанное преобразование называется *преобразованием Карунена-Лоева*. Оно имеет фундаментальное значение, т.к. оно приводит к построению некоррелированных признаков.

11.2.1 Свойства преобразования Карунена-Лоева

Пусть

$x = Ay$ или $x = \sum_{i=0}^{N-1} y(i) a_i$ – разложение по базисным векторам.

Определим новый m-мерный вектор ($m < N$):

$$\hat{x} = \sum_{i=0}^{m-1} y(i) a_i$$

где $\hat{x}$ – проекция x на подпространство. Если мы аппроксимируем x с помощью $\hat{x}$, то ошибка есть (выбираем те векторов, m для которых ошибка минимальна):

$$E\left\|x-\hat{x}\right\|^2 = E\left[\left\|\sum_{i=0}^{N-1} y(i) a_i\right\|^2\right] = E\left[\sum_i \sum_j \left(y(i) a_i^T\right)\left(y(i) a_i\right)\right] =$$

$$= \sum_{i=m}^{N-1} E\left[y^2(i)\right] = \sum_{i=m}^{N-1} a_i^T E\left[xx^T\right] a_i = \sum_{i=m}^{N-1} a_i^T l_i a_i = \sum_{i=m}^{N-1} l_i .$$

Тогда очевидно, что выбирать нужно m базисных векторов с максимальными собственными значениями.

Отметим еще раз соотношение преобразования Карунена-Лоева с методом селекции признаков. В методе селекции признаков в качестве критерия выступали дискриминантные свойства полученного вектора признаков. В преобразовании Карунера-Лоева в качестве критерия выступает наилучшее приближение исходных измерений.

11.2.2 Применение преобразования Карунена-Лоева к задаче классификации

В данном случае основная концепция заключается в том, что подпространство главных собственных значений может быть использовано для классификации.

Алгоритм:

для каждого класса Ω_i строим корреляционную матрицу R_i,

выбираем m главных собственных значений и собственных векторов,

строим соответствующие матрицы A_i, у которых столбцы – значения собственных векторов.

неизвестный (пробный) вектор x классифицируем по правилу $\left\|A_j^T x\right\| > \left\|A_i^T x\right\|$ при $i \neq j$, т.е. в ближайшее подпространство.

Если все подпространства одинаковой размерности, то разделяющие поверхности – это гиперплоскости, иначе гиперповерхности второго порядка. Такой классификатор интегрирует все: генерация, селекция, классификация.

11.2.3 Декомпозиция сингулярных значений

Пусть задана матрица A ранга r. Покажем, что существуют такие унитарные матрицы $U_{N\times N}$ и $V_{N\times N}$, что

$$X = U \cdot \begin{bmatrix} \Lambda^{\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} \cdot V^H,\ Y = \begin{bmatrix} \Lambda^{\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} = U^H \cdot X \cdot V,$$

где $\Lambda^{\frac{1}{2}}_{r\times r}$ – диагональная матрица с элементами $\sqrt{l_i}$ и l_i – r ненулевых собственных значений матрицы $X^H X$. Иначе существуют такие унитарные матрицы $U_{N\times N}$ и $V_{N\times N}$, что преобразованная X путем $U^H XV$ есть диагональная матрица. Следовательно

$$X = \sum_{i=0}^{r-1} \sqrt{l_i} \cdot u_i \cdot v_i^H \quad (*)$$

где u_i и v_i – первые r столбцов матриц $U_{N\times N}$ и $V_{N\times N}$ соответственно, т.е. u_i и v_i – собственные вектора матриц XX^H и $X^H X$ соответственно.

Собственные значения l_i называются *сингулярными значениями* матрицы X. Преобразование $(*)$ – преобразование сингулярных значений или спектральное представление X.

Если X аппроксимировать следующим образом

$$\hat{X} = \sum_{i=0}^{k-1} \sqrt{l_i} \cdot u_i \cdot v_i^H, k \le r-1 ,$$

то $\hat{X}$ есть сумма k одноранговых матриц и имеет ранг равный k. Можно показать, что квадратичная ошибка

$$e^2 = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \left| X(m,n) - \hat{X}(m,n) \right|^2$$

является минимальной для всех k-ранговых матриц. Ошибка аппроксимации есть

$$e^2 = \sum_{i=k}^{r-1} l_i ,$$

следовательно, и в данном случае нужно выбирать максимальное l_i.

Таким образом, $\hat{X}$ есть наилучшая аппроксимация в смысле нормы Фробениуса. Данная аппроксимация напоминает преобразование Карунена-Лоева.

11.3 Дискретное преобразование Фурье (ДПФ)

Преобразования типа Карунена-Лоева есть результат специальной обработки (оптимизации) применительно к конкретной выборке требует больших вычислительных затрат. Если разложить по некоторому заданному базису, то можно снизить затраты, правда снизив требования к разложению.

11.3.1 Одномерное дискретное преобразование Фурье

Пусть $x(0), x(1), ..., x(N-1)$ – N исходных измерений. Тогда ДПФ определяется следующим образом:

$$y(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \exp\left(-j \frac{2p}{N} kn \right),$$

где $k = 0,1,...,N-1$ и $\exp\{a_j\} = \cos(a) + j\sin(a)$.

Обратное преобразование есть:

$$x(k) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y(n) \exp\left(j \frac{2p}{N} kn \right),$$

где $n = 0,1,...,N-1$.

Определим

$$W_N = \exp\left\{ -j \frac{2p}{N} \right\}.$$

Тогда

$$\exp\left\{ -j \frac{2p}{N} kn \right\} = W_N^{kn}.$$

Пусть

$$y = W^H x ,$$

тогда

$$x = Wy ,$$

$$W^H = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \mathbf{L} & 1 \\ 1 & W_N & \mathbf{L} & W_N^{N-1} \\ 1 & W_N^2 & \mathbf{L} & W_N^{2(N-1)} \\ \mathbb{M} & \mathbb{M} & \mathbf{O} & \mathbb{M} \\ 1 & W_N^{N-1} & \mathbf{L} & W_N^{(N-1)(N-1)} \end{pmatrix}.$$

Утверждается, что W – унитарная симметрическая матрица. Пусть W^* – сопряженная матрица: $W^* = W^H = W^{-1}$. Тогда базисные вектора – это столбцы матрицы W. Таким образом, имеет место разложение по заданному базису (по определению $x = \sum_{i=0}^{N-1} y(i)a_i$ – разложение по базисным векторам).

Прямое вычисление

$y = W^H x$ или $x = Wy$

имеет сложность $O(N^2)$, однако, специфика структуры матрицы W позволяет строить алгоритмы сложности $O(N \ln N)$.

ДПФ можно рассматривать как разложение последовательности $X(n)$ в множество N базисных последовательностей $h_k(n)$:

$$X(n) = \sum_{k=0}^{N-1} y(k) h_k(n),$$

где $h_k(n) = \begin{cases} \frac{1}{N} \exp\left\{ j \frac{2p}{N} kn \right\}, & \text{при } n = 0,1,\ldots,N\text{-}1, \\ 0, & \text{иначе.} \end{cases}$

$y(k)$ – коэффициенты разложения,

а последовательности $h_k(n)$ ортогональные:

$$(h_k, h_l) = d_{kl} = \begin{cases} 1, & \text{при } k = l, \\ 0, & \text{иначе.} \end{cases}$$

11.3.2 Двумерные ДПФ

Пусть $X(i, j)$, $i, j = 0,1,\ldots,N-1$ – двумерные измерения. Тогда двумерное ДПФ есть:

$$Y(k,l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X(m,n) W_N^{k \times m} W_N^{l \times n}.$$

Обратное преобразование:

$$X(m,n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} Y(k,l) W_N^{-k \times m} W_N^{-l \times n}.$$

Данную запись компактно можно переписать в следующем виде:

$Y = W^H X W^H$, $X = WYW$.

Данное преобразование – это преобразование с базисными матрицами или образами $w_i w_j^T$, $i, j = 0,1,\ldots,N-1$. Число требуемых операций “в лоб” равно $O(N^3)$. Учитывая специфическую структуру W, существуют методы сложности $O(N^2 \ln N)$.

11.3.3 Дискретное косинусное преобразование (ДКП)

Данное преобразование имеет вид:

$$y(k) = a(k)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{p(2n+1)k}{2N}\right),\ k = 0,1,..,N-1,$$

где

$$x(n) = \sum_{k=0}^{N-1} a(k)y(k)\cos\left(\frac{p(2n+1)k}{2N}\right),\ n = 0,1,...,N-1,$$

где $a(k) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{при } k = 0, \\ \sqrt{\frac{2}{N}}, & \text{иначе.} \end{cases}$

Его можно переписать в векторной форме:

$y = C^T x$,

где $C(n,k) = \begin{cases} \frac{1}{\sqrt{N}}, & \text{при } k = 0, 0 \le n \le N-1, \\ \sqrt{\frac{2}{N}}\cos\left(\frac{p(2n+1)k}{2N}\right), & \text{при } k = 1,2,...,N\text{-}1,\ 0 \le n \le N\text{-}1. \end{cases}$

и C – действительная матрица, причем $C^{-1} = C^T$.

Двумерное ДПФ определяется так

$Y = C^T XC$ и $X = CYC^T$.

11.3.4 Дискретное синусное преобразования (ДСП)

Данное преобразование вычисляется аналогично косинусному через матрицу:

$$S(k,n) = \sqrt{\frac{2}{N-1}}\sin\left(\frac{p(k+1)(n+1)}{N+1}\right),\ k,n = 0,1,...,N-1.$$

Вычислительная сложность затрат на ДКП и ДСП есть $O(N\ln N)$.

ДКП и ДСП обладают хорошими “упаковочными” свойствами для большинства изображений в том смысле, что концентрируют основную информацию в небольшом числе коэффициентов. Объясняется это тем, что оба они дают хорошее приближение для большого класса реальных образов, моделируемых случайных сигналов, известные как Марковский процесс 1-ого порядка.

11.4 Преобразования Адамара и Хаара

Преобразование Адамара и Хаара имеют такие же вычислительные достоинства, как и ДПФ, ДКП, ДСП. Их матрицы состоят из ± 1, поэтому они вычисляются через сложения и вычитания без умножений.

11.4.1 Преобразование Адамара

Определение. *Унитарная матрица Адамара порядка* n *– это* $N \times N$ *матрица , где* $N = 2^n$, *сгенерированная по следующему итерационному правилу*

$H_n = H_1 \otimes H_{n-1}$,

где

$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

и $\otimes$ *обозначает кронекерово произведение двух матриц:*

$$A \otimes B = \begin{pmatrix} A(1,1)B & A(1,2)B & \mathbf{L} & A(1,N)B \\ A(2,1)B & A(2,2)B & \mathbf{L} & A(2,N)B \\ \mathbb{M} & \mathbb{M} & \mathbf{O} & \mathbb{M} \\ A(N,1)B & A(N,2)B & \mathbf{L} & A(N,N)B \end{pmatrix}.$$

Распишем H_2:

$$H_2 = H_1 \otimes H_1 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

По аналогии можно выписать все H_n, $n = 1, 2, \ldots$. Нетрудно установить ортогональность H_n, $n = 1, 2, \ldots$:

$H_n^{-1} = H_n^{\mathrm{T}} = H_n$.

Для вектора x из N образцов пара преобразований есть:

$y = H_n x$, $x = H_n y$.

Преобразование Адамара имеет очень хорошие “упаковочные” свойства. Алгоритм для вычисления выделений и сложений достаточно быстрый: $O(N - \ln N)$.

11.4.2 Преобразование Хаара

Начальной точкой для определения преобразования Хаара являются функции Хаара, которые являются непрерывными и определенными на замкнутом сегменте $[0,1]$.

Порядок k функций Хаара единственным образом раскладывается через два целых числа p и q:

$k = 2^p + q - 1$, $k = 0,1,\ldots,L-1$, $L = 2^n$,

где $0 \le p \le n-1$, $0 \le q \le 2^p$ для $p \ne 0$ при $q = 0$ или $p = 0$ при $q = 1$.

Определение. *Функции Хаара:*

$$h_0(z) \equiv h_{00}(z) = \frac{1}{\sqrt{L}},\ z \in [0,1];$$

$$h_k(z) \equiv h_{pq}(z) = \frac{1}{\sqrt{L}} \cdot \begin{cases} 2^{\frac{p}{2}}, \text{ при } \dfrac{q-1}{2^p} \le z < \dfrac{q-0,5}{2^p}, \\ -2^{\frac{p}{2}}, \text{ при } \dfrac{q-0,5}{2^p} \le z < \dfrac{q}{2^p}, \\ 0, \text{ для остальных } z \in [0,1]. \end{cases}$$

11.5 Генерация признаков на основе нелинейных преобразований. Выделение текстуры изображений.

Пусть дано изображение или его часть (область). Задача состоит в генерации признаков, которые впоследствии будут использоваться при классификации.

Определение. *Цифровое изображение (монохромное) есть результат процесса дискретизации непрерывной функции $I(x,y)$ в виде двумерного массива $I(m,n)$, где $m = 0,1,\ldots,N_X - 1$, $n = 0,1,\ldots,N_Y - 1$. Значение функции $I(x,y)$ – интенсивность, число градаций N_g – глубина изображения.*

Определение. *Генерацией признаков называется эффективное кодирование необходимой для классификации информации, содержащейся в оригинальных (исходных) данных.*

11.5.1 Региональные признаки. Признаки для описания текстуры

Дадим не точное определение текстуры.

Определение. Текстурой называется распределение оттенков серого цвета среди пикселов в регионе.

Рассмотрим основные типы характеристик:

тонкие – грубые,

гладкие – резкие (нерегулярные),

однородные – неоднородные.

Отметим, что в основе подхода лежит гипотеза о том, что внутри региона значения интенсивностей описываются одинаково, т.е. одним и тем же распределением вероятностей.

Пусть интенсивность внутри региона есть случайная величина. Тогда, при условии, что внутри региона характеристики одинаковы, данная случайная величина внутри региона одинаково распределенная, чем обеспечивается свойство однородности в регионе.

Нашей целью является генерация признаков, которые как-то квантуют свойства фрагментов изображения (регионов).

Данные признаки появляются при анализе пространственных соотношений по распределению серых цветов.

11.5.2 Признаки, основанные на статистиках первого порядка

Пусть I – интенсивность случайной величины, представляющая собой значение (уровень интенсивности) серого цвета в регионе. Пусть также $P(I = I_0)$ – вероятность, того что интенсивность в регионе равна I_0.

Определение. Гистограммой первого порядка называется величина $P(I)$, равная отношению числа пикселов с уровнем интенсивности I_0 к общему числу пикселов в регионе и обозначается $P(\hat{I})$.

Рассмотрим центральный момент:

$$m_i = \sum_{\hat{I}=0}^{N_g-1} (\hat{I} - m_1) P(\hat{I}),$$

где m_1 – среднее значение интенсивности – первый момент, который в общем случае определяется из формулы:

$$m_k = \sum_{\hat{I}=0}^{N_g-1} I^k P(\hat{I})$$

при $k = 1$.

Среди центральных моментов наиболее часто используются

m_2 – дисперсия $\hat{I}$,

m_3 – ассиметрия,

m_4 – эксцесс.

В качестве признаков, основанных на статистиках первого порядка, также может использоваться абсолютный момент:

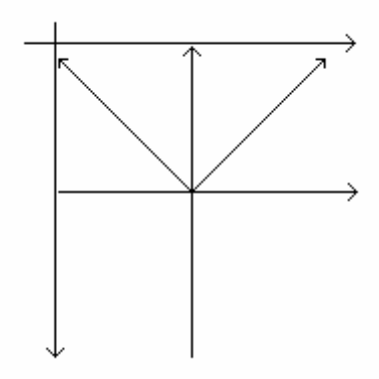
$$\tilde{m}_i = \sum_{\hat{I}=0}^{N_g-1} \left| \hat{I} - m_1 \right| \cdot P(\hat{I})$$

и энтропия:

$$H = -E\left[\log_2 P(\hat{I})\right] = -\sum_{\hat{I}=0}^{N_g-1} P(\hat{I}) \log_2 (P(\hat{I})),$$

которая определяет меру равномерности распределения. Чем энтропия выше, тем распределение равномернее.

11.5.3 Признаки, основанные на статистиках второго порядка. Матрицы сочетаний.



Пусть d – относительное расстояние между пикселами, j – ориентация. Тогда можем ввести метрику следующим образом:

$$r(p_1, p_2) = \max\left\{|p_1 x - p_2 x|, |p_1 y - p_2 y|\right\},$$

причем пикселы рассматриваются в парах.

Рассмотрим соседство для четырех пикселей. Пусть $j = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, т.е у нас имеется горизонтальное, вертикальное, диагональное и антидиагональное соседство.

Обозначим через $P_j(\hat{I}(m,n), \hat{I}(m_1,n_1))$ совместную плоскость. Рассмотрим $j = 0^\circ$. $P_j(\hat{I}(m,n) = I_1, \hat{I}(m \pm d, n) = I_2)$ – вероятность того, что точки, расположенная на горизонтали $r = d$ имеют интенсивности I_1 и I_2, равные отношению числа пар пикселов с расстоянием d и значением I_1 и I_2 к общему числу пикселов в регионе.

Аналогично считается $P_j(\hat{I}(m,n) = I_1, \hat{I}(m \pm d, n \mp d) = I_2)$ для $j = 45^\circ$; $P_j(\hat{I}(m,n) = I_1, \hat{I}(m, n \mp d) = I_2)$ для $j = 90^\circ$; $P_j(\hat{I}(m,n) = I_1, \hat{I}(m \pm d, n \pm d) = I_2)$ для $j = 135^\circ$. Каждый такой массив называют матрицей сочетаний или матрицей пространственной зависимости.

Рассмотрим конкретный пример матрицы $\hat{I}$. Пусть $N_g = 4$, т.е. уровни интенсивности изменяются от 0 до 3. Пусть также матрица $\hat{I}$ задана следующим образом:

$$\hat{I} = \begin{pmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{pmatrix}.$$

Т.к. просмотр происходит в обе стороны, то общее количество пар равно 24.

Рассмотрим $j = 0^\circ$ и $d = 1$. $0 \le \hat{I}_1, \hat{I}_2 \le 3$. Очевидно, что матрица A является симметрической.

$$A = \begin{pmatrix} P(0,0) & P(0,1) \\ P(0,1) & P(\hat{I}_1, \hat{I}_2) \end{pmatrix} = \frac{1}{24} \begin{pmatrix} 4 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 6 & 3 \\ 0 & 0 & 3 & 2 \end{pmatrix}$$

Для $j = 45^\circ$ и $d = 1$ матрица A выглядит следующим образом:

$$A=\frac{1}{18}\begin{pmatrix}0&1&2&1\\1&0&1&1\\2&1&0&3\\1&1&3&0\end{pmatrix}$$

Существуют следующие основные виды признаков, основанные на статистиках второго порядка:

Угловой момент второго порядка: $ASM=\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}(P(i,j))^2$ – мера гладкости изображения. При малой вариации $ASM\approx 1$, а при больших вариациях (например при увеличении) контраста $ASM\to 0$.

Контраст (по заданной паре): $CON=\sum_{n=0}^{N_g-1}n^2\left\{\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}P(i,j)\right\}$ – мера локальной дисперсии серого.

Момент обратной разности: $IDF=\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}\frac{P(i,j)}{1+(i-j)^2}$. Момент обратной разности имеет большое значение для слабоконтрастных изображений.

Энтропия: $H=\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}P(i,j)\log_2 P(i,j)$ – мера равномерности. Энтропия связана с фиксированной ориентацией и фиксированным расстоянием.

11.6 Признаки формы и размера

Рассмотрим методы генерации признаков, описывающих структуру. Существует два основных пути описания формы:

- Полное описание формы в регенеративной манере (например, признаки Фурье). По такому описанию полностью можно восстановить образ.
- Не восстановительное описание формы (дескриптивные признаки). По такому описанию можно отличить заданную форму от других, но не полностью восстановить образ.

11.6.1 Признаки Фурье

Отметим, что полное описание позволяет восстанавливать границу образа. Частичное же описание дает признаки для распознавания. Нас интересует вопрос о зависимости изменения признаков от преобразований.

Пусть (x_k,y_k), где $k=0,1,...,N-1$, – координаты последовательных точек границы; $u_k=x_k+j*y_k$ – комплексные числа. Для N точек u_k определим ДФП (DFT):

$$f_l=\sum_{k=0}^{N-1}u_k\exp\left(-j\cdot\frac{2p}{N}\cdot l\cdot k\right),\ l=0,1,...,N-1,$$

где f_l – Фурье-описание границы.

Рассмотрим, как изменяется f_l при сдвиге, повороте, масштабировании и сдвиге начальной точки.

Сдвиг описывается следующим образом: $x'_k=x_k+\Delta x$, $y'_k=y_k+\Delta y$ и $u'_k=u_k+\Delta u'$. Тогда

$$f'_l=f_l+\Delta u d(l),\ \text{где } d=\begin{cases}1,\ \text{при } l=0\\0,\ \text{при } l\neq 0\end{cases}.$$

При $l=0$ $f'_0\neq f_0$, т.к.

$f_0' = f_0 + \Delta u d(0) = f_0 + \Delta u \neq f_0$.

При $l \neq 0$ $f_l' = f_l$, т.к.

$f_l' = f_l + \Delta u d(l) = f_l + \Delta u \cdot 0 = f_l$

Поворот описывается следующим соотношением: $u_k' = u_k \cdot \exp(jq)$. Следовательно, $f_l' = f_l \cdot \exp(jq)$, т.е. поворот не меняет модулей, а именно $|f_l'| = |f_l|$.

Масштабирование описывается следующим соотношением: $u_k' = a \cdot u_k$. Следовательно, $f_l' = a \cdot f_l$. Т.к.

$$\frac{f_i'}{f_i} = a \text{ и } \frac{f_j'}{f_j} = a,$$

то масштабирование не меняет соотношения

$$\frac{f_i'}{f_j'} = \frac{f_i}{f_j}.$$

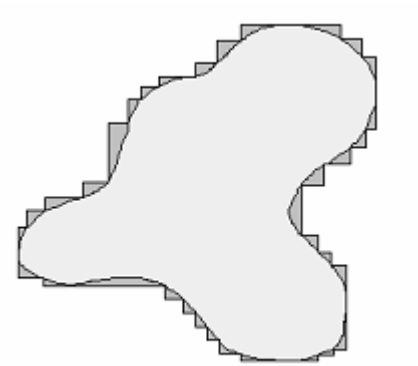
Сдвиг начальной точки определяется следующим образом: $u_k' = u_{k-k_0}$. Следовательно

$$f_l' = f_l \cdot \exp\left(-j \cdot \frac{2p}{N} \cdot k_0 \cdot l\right),$$

т.е. сдвиг начальной точки сохраняет модули: $|f_l'| = |f_l|$.

11.6.2 Цепной код

Определение. *Цепным кодом называется кодирование (запоминание) последовательности поворота вектора по пикселям на границе описываемой области – маршрута обхода.*



Из построенного цепного кода конструируются следующие признаки:

относительная доля каждого направления,

относительная доля разных сочетаний кривизны.

Недостатком представления изображения цепным кодом является появления шума. Способом борьбы с данным недостатком является использование более мелкой (точной) сетки.

11.6.3 Геометрические свойства фигуры

Пусть P – периметр фигуры, A – площадь фигуры. Рассмотрим следующие свойства: некруглость фигуры и энергию изгиба.

Некруглость фигуры определяется по следующей формуле:

$$r = \frac{P^2}{4pA}.$$

Рассмотрим два крайних значения для данного свойства. Наиболее лучшее (наибольшая “круглость”) значение должно быть для круга, оно равно

$$r = \frac{P^2}{4pA} = \frac{(2pR)^2}{4p \cdot pR^2} = \frac{4p^2R^2}{4p^2R^2} = 1.$$

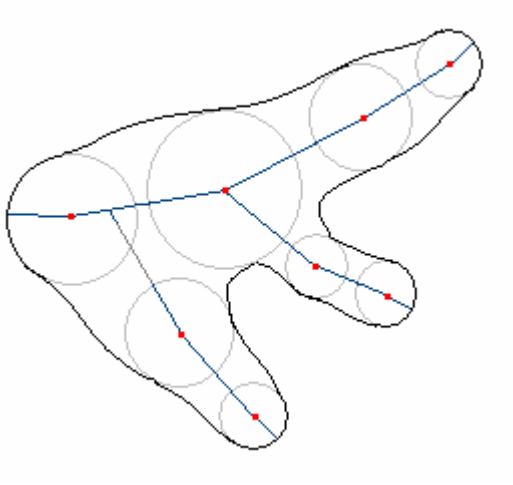
Наиболее худший вариант (наименьшая “круглость”) наблюдается у квадрата. Соответствующее значение равно

$$r = \frac{P^2}{4pA} = \frac{(4a)^2}{4p \cdot a^2} = \frac{16a^2}{4pa^2} = \frac{4}{p}.$$

Энергия изгиба. Пусть задано n точек фигуры. Тогда Энергия изгиба описывается следующей формулой:

$$E(n) = \frac{1}{P}\sum_{i=0}^{n-1} |k_i|^2,$$

где $k_i = q_{i+1} - q_i$ и $q_i = \arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$. k_i характеризует изменение угла в вершине.



11.6.4 Скелетизация

Определение. *Скелетизацией называется построение скелета, описывающего форму фигуры.*

Определение. *Скелетом называется множество всех центров вписанных в фигуру максимальных окружностей.*

MAT (Medial Area Transform) определяется как скелет плюс функция ширины фигуры.

12 Обучение по прецедентам (по Вапнику, Червоненкису)

12.1 Задача построения классификатора

Пусть

Ω – пространство образов,

X – признаковое пространство,

$g(w)$, $w \in \Omega$ – индикаторная функция,

M – множество признаков.

Тогда $g : \Omega \to M$.

Пусть также

$\mathrm{X} = \langle x(w_i), g(w_i) \rangle$, $i = 1,..,N$ – множество прецедентов,

$\hat{g}(x)$ – решающее правило.

Тогда $\hat{g} : \mathrm{X} \to M$.

Выбор решающего правила исходит из минимизации $d(g, \hat{g}) \to \min$, где d – метрика, мера близости функций $g(w)$ и $\hat{g}(x(w))$. Построение $\hat{g}$ называют задачей обучения. $\hat{g}$ – это ученик, процедура формирования – это учитель, прецеденты – это обучающая последовательность.

12.2 Качество обучения классификатора

Относительная доля несовпадений классификации с учителем для решающего правила есть: $K = \frac{m}{N}$, где $m = |\{w_i : g(w_i) \neq \hat{g}(x(w_i)), i = 1, 2, ..., N\}|$. Надежность обучения классификатора – это вероятность получения решающего правила с заданным качеством.

Пусть $f(x, a)$ – класс дискриминантных функций, где $a \in A$ – параметр. Число степеней свободы при выборе конкретной функции в классе определяется количеством параметров в векторе a, т.е. размерностью A.

Например, для классов линейных и квадратичных функций имеем:

Линейная дискриминантная функция: $f(x, a) = \sum_{i=1}^{n} a_i x_i + a_0$. В таком случае имеем $n+1$ степень свободы.

Квадратичная дискриминантная функция: $f(x, a) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n} b_i x_i + b_0$. В таком случае имеем $n^2 + n + 1$ степеней свободы.

С увеличением степеней свободы увеличивается способность классификатора по разделению.

12.3 Вероятностная модель

Пусть прецеденты – это результат реализации случайных величин. Рассмотрим величину риска (т.е. ошибки) связанной с классификацией. Определим понятия риск среднего и риска эмпирического.

Пусть на Ω заданы s -алгебра и мера P. Пусть также

x – вектор признаков,

$\tilde{f}$ – класс функций, из которых выбирается решающее правило,

$f(x,a)$ – решающее правило (результат классификации), которое принимает значение 0 или 1 при фиксированном векторе параметра,

c – характеристическая функция множества,

A – множество параметров, описывающие различные функции в $\tilde{f}$.

Тогда $\hat{g} = f(x,a)$, где $f \in \tilde{f}$ и $f : X \times A \to M$, $y = g(w)$.

В данных обозначениях средний риск выглядит следующим образом:

$$K(a) = \int_X c\{y \neq f(x,a)\} dP .$$

Для случая двух классов, при $M = \{0,1\}$, имеем:

$$K(a) = \int_\Omega (y - f(x,a))^2 dP$$

или

$$K(a) = \int_{(X,M)} (y - f(x,a))^2 dP(x,y),$$

где dP – это вероятностная мера на пространстве X.

12.4 Задача поиска наилучшего классификатора

Рассмотрим минимизацию функционала:

$$K(a) \to \min_{a \in A}$$

Задача же поиска наилучшего классификатора состоит в нахождении a^* такого, что

$$K(a^*) = \min_{a \in A} K(a).$$

Если же минимума не существует, то надо найти a^* такое, что

$$\left| K(a^*) - \inf_{a \in A} K(a) \right| < d .$$

Другими словами, необходимо решить задачу минимизации среднего риска.

Поскольку dP неизвестно, будем решать задачу минимизации эмпирического риска. Пусть l – число прецедентов. Тогда эмпирический риск задается выражением:

$$K_{эмп}(a) = \frac{1}{l} \sum_{i=1}^{l} \left| y - f(x,a) \right| .$$

Таким образом, задача минимизации эмпирического риска выглядит так:

$$K_{эмп}(a) \to \min_{a \in A},$$

где случайные величины мы минимизируем по параметру a – любой возможный параметр.

В идеале надо получить взаимосвязанные оценки эмпирического и среднего риска.

Отметим, что чем меньше l, тем легче построить $f(x,a)$ такую, что $K_{эмп}(a)$ обращается в ноль, либо очень мало. Но при этом истинное значение $K(a)$ может сильно отличаться от $K_{эмп}(a)$. Необходимо выбрать $f(x,a)$ такую, чтобы имела место равномерная сходимость по a выражения:

$$P\left\{ \sup_a \left| K_{эмр}(a) - K(a) \right| > e \right\} \xrightarrow[l \to \infty]{} 0 .$$

Фактически это есть сходимость частот к математическому ожиданию.

В дальнейшем будем считать, что в зависимости от конкретного набора прецедентов можем получить любые a. Но необходимо, чтобы полученные эмпирическое решающее хорошо работало (отражало общие свойства) для всех образов. Поэтому в формуле присутствует равномерная сходимость.

12.5 Сходимость эмпирического риска к среднему. Случай конечного числа решающих правил.

Пусть

$K(a)$ – математическое ожидание ошибки классификатора $f(x,a)$,

A – событие – ошибка классификатора при решающем правиле $f(x,a)$,

$P(A)$ – вероятность,

$u(A)$ – частота в l испытаниях.

Воспользуемся неравенством Бернштейна, тогда

$$P\{|u(A)-P(A)|>e\}\le e^{-2e^2l}$$

есть оценка – соотношение между частотой и вероятностью при заданном количестве испытаний.

Пусть x_j – случайная величина. Тогда $E(x_j)=0$ – математическое ожидание x_j, $Ex_j^2=d^2$ – дисперсия, причем $|x_j|\le L$. Обозначим $S_0=x_1+x_2+...+x_n$. Тогда соответствующая оценка имеет вид:

$$P\left\{|S_n|>td\sqrt{n}\right\}\le 2\cdot\exp\left\{-\frac{t^2}{2\cdot\left(1+\frac{a}{3}\right)}\right\},\ \text{где}\ a=\frac{L\cdot t}{\sqrt{nd}}.$$

$$l=\frac{\ln N-\ln h}{2e^2}\ \text{и}\ e=\sqrt{\frac{\ln N-\ln h}{2l}},$$

где l – необходимое количество прецедентов для обеспечения близости.

Теорема. *Пусть из множества, состоящего из N решающих правил, выбирается правило, частота ошибок которого на прецедентах составляет u. Тогда с вероятностью $1-h$ можно утверждать, что вероятность ошибочной классификации с помощью данного правила $f(x,a)$ составит величину, меньшую $u+e$, если длина обучающей последовательности не меньше $l=\frac{\ln N-\ln h}{2e^2}$, где $e=\sqrt{\frac{\ln N-\ln h}{2l}}$, h и e заданы и последовательность независима.*

Данная теорема справедлива для случая конечного числа решающих правил. Вапник и Червоненкис смогли обобщить эти оценки на случай бесконечного числа решающих правил.

12.6 Случай бесконечного числа решающих правил

Введем понятие “разнообразия класса функций для бесконечного множества”. Пусть $x_1,x_2,...,x_l$ – прецеденты.

Определение. *Дихотомией называется разбиение множества на два подмножества.*

В нашем случае имеем 2^l дихотомий. Итак, пусть $f(x,a)$, $a\in A$ – это класс решающих правил, причем $f(x,a)=\{0,1\}$. Пусть $\Delta(x_1,x_2,...,x_l)$ есть количество дихотомий на классе решающих правил. Тогда зададим энтропию следующим образом:

$$H(l)=E\{\log_2\Delta(x_1,x_2,...,x_l)\},$$

где математическое ожидание берется по всем выборкам $(x_1,x_2,...,x_l)$. Тогда

$$H^S(l)=E\{\log_2\Delta^S(x_1,x_2,...,x_l)\}$$

есть энтропия класса S решающих правил на выборках длины l.

12.6.1 Критерий равномерной сходимости $u(a)$ к вероятностям $P(a)$

Теорема. *Для равномерной сходимости* $u(a) = K_{эмп}(a)$ *к* $P(a) = K(a)$ *по классу* $a \in A$ *необходимо и достаточно, чтобы* $\frac{H(l)}{l} \xrightarrow[l\to\infty]{} 0$.

Суть данного критерия – не пытаться выделить очень точный классификатор, так как это отдаляет от общности.

Сразу же возникает проблема необходимость перехода к бесконечным системам решающих правил. Существенно, что значение имеет лишь конечное подмножество систем решающих правил, необходимое для разделения конечного числа прецедентов.

12.6.2 Достаточное условие равномерной сходимости

Проверка условия критерия равномерной сходимости по вероятности затрудняется неопределенностью распределения выборки. Поэтому достаточные условия формулируются таким образом, чтобы не зависеть от распределения и при этом гарантировать равномерную сходимость. В таком случае вместо энтропии рассматривается величина:

$$m^S(l) = \max_{x_1,\dots,x_l} \Delta^S(x_1, x_2, \dots, x_l),$$

где $m^S(l)$ – это функция роста класса решающих функций $f(x,a)$.

Т.к. логарифм максимума равен максимуму логарифмов, что, в свою очередь, не меньше математического ожидания от логарифма, то $\log_2 m^S(l) \ge H^S(l)$. Если $\lim\limits_{l\to\infty}\left[\log_2 m^S(l)/l\right] \to 0$, то по свойствам пределов $\lim\limits_{l\to\infty} \frac{H^S(l)}{l} \to \infty$.

Данное условие легко проверятся для различных классов решающих правил.

Другими словами $m^S(l)$ можно трактовать как максимальное число способов разделения l точек на два класса с помощью решающих правил $f(x,a)$, $a \in A$.

Теорема. *Функция роста либо тождественно равна* 2^l, *либо, мажорируется функцией* $\sum\limits_{i=0}^{n-1} C_l^i$, *где* n *– минимальное значение* l, *при котором* $m^S(l) \ne 2^l$, *т.е. либо* $m^S(l) \equiv 2^l$, *либо* $m^S(l) \le \sum\limits_{i=0}^{n-1} C_l^i$.

В свою очередь $\sum\limits_{i=0}^{n-1} C_l^i \le 1{,}5 \cdot \frac{l^{n-1}}{(n-1)!}$. Значит $m^S(l) \le 1{,}5 \cdot \frac{l^{n-1}}{(n-1)!}$, где $l = 1, 2, \dots, n$, и $\frac{l^{n-1}}{(n-1)!}$ – степенная функция, мажорирующая $m^S(l)$.

Существует максимум $n-1$ точка, которая еще разбивается всеми возможными способами с помощью правила $f(x,a)$, но никакие n точек этим свойством не обладают.

Определение. $n-1$ *называется емкостью класса решающих функций или мера разнообразия решающих правил в классе* $f(x,a)$ *или VC–размерностью класса – универсальная характеристика класса решающих функций.*

Отметим, что если $m^S(l) = 2^l$ для всех l, то емкость бесконечна.

Теорема. *Если емкость класса решающих функций конечна, то всегда имеет место равномерная сходимость частот к вероятностям такое, что*

$$\lim_{l\to\infty}\left(\frac{\log_2 m^S(l)}{l}\right) \le \lim_{l\to\infty}\left(\frac{(n-1)\log l + \log 1.5}{l}\right) = 0$$

и достаточное условие выполнено.

12.6.3 Скорость сходимости

Запишем оценку для бесконечного числа решающих правил. Ее вид аналогичен случаю конечного числа решающих правил:

$$P\left\{\sup_a \left|P(a)-u(a)\right| > e\right\} < 3m^S(2l)\cdot e^{-\frac{e^2(l-1)}{4}}.$$

Если емкость бесконечна, то оценка тривиальная (не больше единицы). Пусть r – конечная емкость класса решающих функций. Тогда

$$P\left\{\sup_a \left|P(a)-u(a)\right| > e\right\} < 4,5\cdot\frac{(2l)^r}{r}\cdot e^{-\frac{e^2(l-1)}{4}}.$$

Введем обозначение: $h = 4,5\cdot\frac{(2l)^r}{r}\cdot e^{-\frac{e^2(l-1)}{4}}$,

Тогда $P\left\{\sup_a \left|P(a)-u(a)\right| > e\right\} < h$.

Отсюда следует, что $e = \sqrt{\frac{r\left(\ln\frac{2l}{r}+1\right)-\ln\frac{h}{5}}{l-1}}$.

Значит, с вероятностью, превышающей $1-h$ качество эмпирического оптимального решающего правила отличается от истинно оптимально решающего правила не более чем на величину $\Delta = 2e$.

В следующей таблице представлен некоторый итог наших рассуждений.

	Малая емкость класса решающих функций (бедный)	Большая емкость класса решающих функций (богатый)
Близость эмпирического решающего правила к оптимальному решающему правилу	Хорошая	Плохая
Качество разделения (минимизация ошибки)	Низкое	Высокое

Таким образом, необходимо минимизировать степени свободы.

12.6.4 Случай класса линейных решающих функций

Пусть $f(x,a)$ – линейная решающая функция, m – размерность пространства.

Как уже отмечалось выше, имеем 2^l дихотомий, где l – длина выборки. Хотим выяснить, какое количество дихотомий реализуется с помощью гиперплоскостей?

Максимальное число точек в пространстве размерности m, которое с помощью гиперплоскостей можно разбить всеми возможными способами на два класса есть $m+1$.

Если $m^S(l) \le 1,5\cdot\frac{l^{m+1}}{(m+1)!}$,

то линейный риск будет равномерно сходиться к среднему риску. Емкость класса конечна и равна $m+1$.