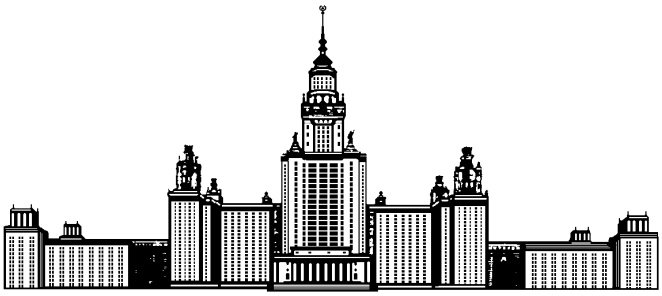
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

КУРСОВАЯ РАБОТА СТУДЕНТА 317 ГРУППЫ

«Обзор алгоритмов классификации документов»

Выполнил:
студент 3 курса 317 группы
Нижибицкий Евгений Алексеевич

Научный руководитель:
д.ф-м.н., доцент
Дьяконов Александр Геннадьевич

Москва, 2012

Содержание

Введение

Ввиду роста количества текстовой информации повсеместно, а особенно в интернете, все большую роль играет возможность классифицировать её, отбирать лишь актуальную её часть – отсюда и возникают сопутствующие задачи машинного обучения. Примерами таких задач могут служить:

1. Автоматизированное разделение текстов или сайтов по тематическим каталогам – например, у компании Яндекс таковой содержит 113 380 сайтов по состоянию на май 2012г. [1], схожий с ним каталог DMOZ (проект Open Directory Project) содержит и того больше — 5 022 597 сайтов [2].

2. Борьба со спамом — по данным Лаборатории Касперского, доля спама в электронной почте составляла от 80 до 85 процентов в разное время в период с 2007-го по 2011-й год [3].

3. Персонификация рекламы — определение тематики сайта или письма, читаемого пользователем, для выдачи так называемой контекстной рекламы. По данным аналитического центра "Видео интернешнл"[4], объем рынка контекстной рекламы в рунете составил 11,3 млрд руб.

После унифицированного представления документов в векторном виде и некоторой специализированной предобработки данных, мы можем применять стандартные методы машинного обучения. Выбор способа перевода в векторный вид сам по себе является довольно сложной задачей — ведь именно от него во многом будет зависеть успешность применения стандартных алгоритмов. Сложность состоит и в размере данных — документы содержат десятки тысяч различных слов, количество классов так же может достигать тысячи — и это все при достаточно скудном описании классов (по несколько документов на класс) и небольшом количестве рубрик у каждого документа (обычно не более 5-8).

Отсюда мы и имеем сравнительно невысокие результаты, достигаемые, например, на открытых конкурсах среди специалистов по машинному обучению — так, на одном из последних тематических конкурсов, проводимом Университетом Варшавы при поддержке проекта SYNAT и сайта TunedIT [5], максимальный полученный результат в виде F-макро меры составил 0.535.

Как уже было сказано выше, алгоритмы классификации текстов состоят из предобработки данных и непосредственной классификации нового координатного пространства. Дополнительно может совершаться уменьшение размерности перед непосредственно классификацией — так, в выше упомянутом конкурсе размерность координатного пространства была больше 25 000 — достаточно, чтобы привести современную вычислительную машину многими стандартными алгоритмами в недееспособное состояние на много часов. Ниже мы сделаем более полную постановку задачи, рассмотрим наиболее используемые способы предобработки данных и собственно сами различные классы алгоритмов, используемые в этом типе задач.

1 Постановка задачи

Пусть существует множество документов $\mathfrak{D}$, множество классов (рубрик) $\mathfrak{C} = \{c_1, \ldots, c_{|\mathfrak{C}|}\}$, к каждому из которых могут относиться документы, и существует некоторая целевая зависимость

$$a^* : \mathfrak{D} \times \mathfrak{C} \to \{0, 1\},$$

значения которой известны лишь на конечном наборе документов обучающей выборки $D^l = \{(\mathbf{d}_1, c_1), \ldots, (\mathbf{d}_l, c_l)\} \subset \mathfrak{D} \times \mathfrak{C}$.

Задачей классификации (рубрикации) текстов является поиск наилучшего приближения a (его также называют алгоритмом ввиду необходимости реализации на компьютере) целевой зависимости a^* на основе обучающей выборки.

Качество алгоритма определяется на контрольной выборке $\bar{D}^k \subset \mathfrak{D}$ — сравниваются ответы, выданные алгоритмом на контрольной выборке, с истинными заранее известными для них ответами с помощью выбранного функционала качества, или метрики.

2 Отбор признаков

Перед тем, как приступать к любой задаче классификации, одна из наиболее фундаментальных процедур, которую необходимо выполнить, это выбрать представление документа и отобрать признаки. Хоть отбор признаков и применяется довольно часто в других задачах классификации, он особенно важен в задаче классификации текстов ввиду высокой размерности (большого количества признаков) и наличия нерелевантных (шумовых) признаков. В большинстве случаев, представление текста происходит одним из двух способов. Первый – документ как набор слов, в котором документу сопоставляются слова и частота их встречаемости в нем. Такое представление, как видим, независимо от порядка слов, в котором они встречаются в тексте. Второй метод состоит в представлении текста собственно как набор строк, в котором документ является последовательностью слов. Большинство алгоритмов классификации текстов используют первое представление ввиду его простоты и удобства для задач классификации.

Наиболее популярными способами отбора признаков являются удаление стоп-слов и стэмминг. При удалении стоп-слов, мы определяем общие слова документов, которые не являются специфичными или разделяющими для разных классов. При стэмминге, разные формы одного слова объединяются в одно слово (терм). Например, так объединяются слова разного рода/формы/времени/падежа и пр. Множество методов отбора признаков были рассмотрены и экспериментально проверены на эффективность в [6].

2.1 Индекс Джини

Один из наиболее используемых методов для определения разделяющей способности признака – это использование меры, известной как индекс Джини. Пусть $p_1(w), \ldots, p_k(w)$ – частоты наличия метки классов для слова w. Другими словами, $p_i(w)$ – условная вероятность, что документ принадлежит классу i, если известно,

что он содержит слово w. Значит, можно утверждать, что

$$\sum_{i=1}^{k} p_i(w) = 1.$$

Тогда индекс Джини для слова w, обозначаемый $G(w)$, определяется как

$$G(w) = \sum_{i=1}^{k} p_i(w)^2.$$

Значение индекса Джини $G(w)$ всегда лежит на интервале $(1/k, 1)$. Чем выше значение $G(w)$, тем выше разделяющая способность слова w. Так, к примеру, когда все документы, содержащие слово w, принадлежат к одному классу, значение $G(w)$ равно 1. Наоборот же, когда все документы, содержащие слово w равномерно распределены по всем классам, значение $G(w)$ равно $1/k$.

Проблема данного подхода в том, что с самого начала глобальное распределение по классам может быть искажено, что значит, что и введенная выше мера не всегда точно отражает разделительную способность признаков. Тем не менее, возможно построить нормализованный индекс Джини, что отражать разделяющую способность точнее. Пусть $P_1, \ldots, P_k$ являются априорными вероятностями принадлежности документов к соответствующим классам. Тогда мы определим нормализованную вероятность $p'_i(w)$ как

$$p'_i(w) = \frac{p_i(w)/P_i}{\sum_{j=1}^{k} p_j(w)/P_j}.$$

Вычислим нормализованный индекс Джини в новых обозначениях:

$$G(w) = \sum_{i=1}^{k} p'_i(w)^2.$$

Использование вероятностей P_i обеспечивает более точное отражение разделительной способности в случае смещенных распределений по классам во всей выборке.

2.2 Прирост информации

Другая схожая мера, часто используемая в отборе признаков для документов, это прирост информации. Пусть P_i будет априорной вероятностью принадлежности к классу i, а $p_i(w)$ будет условной вероятностью принадлежности к классу i при условии того, что документ содержит слово w. Пусть $F(w)$ – часть документов, содержащая слово w. Тогда мера прироста информации $I(w)$ для данного слова w определяется как

$$I(w) = -\sum_{i=1}^{k} P_i \cdot log(P_i) + F(w) \cdot \sum_{j=1}^{k} p_i(w) \cdot log(p_i(w)) +$$
$$+(1 - F(w)) \cdot \sum_{j=1}^{k} (1 - p_i(w)) \cdot log(1 - p_i(w)).$$

Чем больше значение прироста информации $I(w)$, тем сильнее разделяющая способность слова w.

2.3 Взаимная информация

Понятие меры взаимной информации произошло из теории информации [7], оно предоставляет формальный путь смоделировать корреляцию между признаками и классами. Точечная взаимная информация $M_i(w)$ между словом w и классом i определяется как смещение уровня совместной встречаемости класса i и слова w. Заметим, что выборочная совместная встречаемость класса i и слова w в смысле взаимной информации определена как $P_i \cdot F(w)$. Истинная совместная встречаемость, очевидна, равна $F(w) \cdot p_i(w)$. На практике, значение $F(w) \cdot p_i(w)$ может сильно отличаться от $P_i \cdot F(w)$ в зависимости от корреляции класса i и слова w. Взаимная информация определена в терминах отношения двух выше рассмотренных величин. А именно,

$$M_i(w) = \log\left(\frac{F(w) \cdot p_i(w)}{F(w) \cdot P_i}\right) = \log\left(\frac{p_i(w)}{P_i}\right).$$

Как видно, при наличии слова w вероятность принадлежности к классу i возрастает, когда $M_i(w) > 0$, и наоборот – наличие слова w отрицательно сказывается на вероятности принадлежности к классу i, когда $M_i(w) < 0$. Заметим, что $M_i(w)$ определено для класса i. Нам нужно как-то вычислить общую взаимную информацию для слова w для всех классов. Для этого можно рассмотреть максимум или среднее по классам:

$$M_{avg}(w) = \sum_{i=1}^{k} P_i \cdot M_i(w)$$

$$M_{max}(w) = \max_i \{M_i(w)\}$$

Любая из этих мер может быть использована для определения релевантности слова w. Вторая из мер особенно полезна, когда нам более важно выделить высокие корреляции слов хотя бы с одним из классов.

2.4 Статистика хи-квадрат

Статистика χ^2 – это еще один способ выявить зависимость (а точнее, отсутствие независимости) между словом w и определенным классом i. Как и ранее, m будет общим количеством документов в выборке, $p_i(w)$ – условная вероятность принадлежности документа к классу i при наличии слова w, P_i – априорная вероятность принадлежности к классу i, $F(w)$ – часть документов, содержащих слово w. Статистика χ^w слова между словом w и классом i определяется как

$$\chi_i^2(w) = \frac{n \cdot F(w)^2 \cdot (p_i(w) - P_i)^2}{F(w) \cdot (1 - F(w)) \cdot P_i \cdot (1 - P_i)}.$$

Как и в случае с взаимной информацией, мы можем вычислить глобальную статистику χ^2 через значения, определенные для классов. Мы можем использовать как среднее значение, так и максимум, чтобы получить комбинированную величину:

$$\chi_{avg}^2(w) = \sum_{i=1}^{k} P_i \cdot \chi_i^2(w),$$

$$\chi^2_{max}(w) = \max_i\{\chi^2_i(w)\}.$$

Заметим, что статистика χ^2 и взаимная информация являются различными способами измерения корреляции между термами и категориями. Главное преимущества статистики χ^2 перед взаимной информацией состоит в том, что это нормализованная величина, а, значит, эти значения могут подвергаться сравнению между различными термами одной категории.

2.5 Линейный дискриминантный анализ

Другим методом для преобразования признаков является использование линейных дискриминантов, которые явно пытаются построить направления в признаковом пространстве, вдоль которых имеет место быть наилучшее разделение различных классов. Самым распространённым таким методом является линейный дискриминант Фишера [8]. Главная идея этого метода – определить направления в данных, вдоль которых точки как можно лучше разделяются. Подпространство меньшей размерности строится с помощью итерационного поиска таких векторов α_i среди данных, где α_i определяется на i-ой итерации. Мы также хотим, чтобы различные α_i были между собой ортонормированы. На каждом шаге мы определяем этот α_i дискриминантным анализом и проектируем данные на оставшееся ортонормированное подпространство. Качество вектора α_i определяется функцией, которая измеряет степень разделенности различных классов. Значения этой функции уменьшаются с каждым разом, т.к. на данной итерации значение α_i является оптимальным дискриминантом в том подпространстве. Процесс поиска линейных дискриминантов продолжается, пока значения, выдаваемые функцией, не станут меньше некоторого порога. Возможности данного подхода хорошо были показаны в [9], где было получено, что простое дерево решений работает куда лучше на таким образом преобразованных данных, чем более сложные алгоритмы.

Далее, рассмотрим, как же строится дискриминант Фишера. Во-первых, мы рассмотрим функцию $J(\bar{\alpha})$, которая определяет степень разделенности различных классов вдоль данного направления $\bar{\alpha}$. Так мы задаем задачу оптимизации – поиска значения $\bar{\alpha}$, которое максимизирует $J(\bar{\alpha})$. Для простоты, рассмотрим случай двух классов. Пусть D_1 и D_2 будут двумя наборами документов, принадлежащих двум классам. Тогда проекция документа $\mathbf{d} \in D_1 \cup D_2$ на вектор $\bar{\alpha}$ будет выражаться, как $\mathbf{d} \cdot \bar{\alpha}$. Далее, квадратичное разделение классов $S(D_1, D_2, \bar{\alpha})$ по направлению $\bar{\alpha}$ определяется как

$$S(D_1, D_2, \bar{\alpha}) = \left(\frac{\sum_{\mathbf{d} \in D_1} \mathbf{d} \cdot \bar{\alpha}}{|D_1|} - \frac{\sum_{\mathbf{d} \in D_2} \mathbf{d} \cdot \bar{\alpha}}{|D_2|} \right)^2.$$

Вдобавок, нам нужно как-то нормализовать эти абсолютные значения скрытыми внутриклассовыми отклонениями. Пусть $Var(D_1, \bar{\alpha})$ и $Var(D_2, \bar{\alpha})$ будут классовыми разбросами по направлению $\bar{\alpha}$. Другими словами,

$$Var(D_i, \bar{\alpha}) = \frac{\sum_{\mathbf{d} \in D_i} (\mathbf{d} \cdot \bar{\alpha})^2}{|D_i|} - \left(\frac{\sum_{\mathbf{d} \in D_i} \mathbf{d} \cdot \bar{\alpha}}{|D_i|} \right)^2.$$

Тогда нормализованную меру разделения классов $J(\bar{\alpha})$ можно определить так:

$$J(\bar{\alpha}) = \frac{S(D_1, D_2, \bar{\alpha})}{Var(D_1, \bar{\alpha}) + Var(D_2, \bar{\alpha})}.$$

Оптимальное значение α нужно определить исходя из ограничения, что $\bar{\alpha}$ – единичный вектор. Пусть μ_1 и μ_2 будут средними (центроидами) наборов данных D_1 и D_2, а C_1 и C_2 – соответствующие им ковариационные матрицы. Можно показать, что оптимальное (ненормированное) направление $\bar{\alpha}$=$\bar{\alpha*}$ может быть выражено формулой

$$\bar{\alpha*} = \left(\frac{C_1 + C_2}{2}\right)^{-1} (\mu_1 - \mu_2).$$

Главной сложностью в вычислении формулы выше является необходимость обращения разреженной матрице большой размерности (как мы помним, это одна из особенностей задачи классификации текстов). тем не менее, можно применить метод градиентного спуска для определения значения $\bar{\alpha}$ более эффективно с точки зрения количества вычислений. Более подробно этот подход рассматривается в [9].

3 Алгоритмы классификации

3.1 Метрические алгоритмы классификации

Одним из простейших способов классифицировать объект является поиск схожего по некоторым параметрам уже классифицированного объекта. Если мера сходства объектов введена достаточно удачно, то, как правило, оказывается, что схожим объектам очень часто соответствуют схожие ответы. В задачах классификации это означает, что классы образуют компактно локализованные подмножества. Это предположение принято называть *гипотезой компактности*. Для формализации понятия «сходства» вводится функция расстояния в пространстве объектов. Методы обучения, основанные на анализе сходства объектов, называются *метрическими*, даже если функция расстояния не удовлетворяет всем аксиомам метрики (в частности, аксиоме треугольника).

3.1.1 Классификатор Роше

Классификатор Роше проводит рубрикацию документа исходя из его близости к эталонам рубрик. Эталоном для рубрики c является вектор $(w_1, w_2, \dots)$ в признаковом пространстве, вычисленный по формуле:

$$w_i = \frac{\alpha}{|POS(c)|} \sum_{d \in POS(c)} w_{di} - \frac{\beta}{|NEG(c)|} \sum_{d \in NEG(c)} w_{di},$$

где $POS(c)$ и $NEG(c)$ — множества документов из обучающей выборки, которые принадлежат и не принадлежат рубрике c соответственно, а w_{di} — веса i-го признака документа d. Обычно, положительные примеры гораздо важнее отрицательных, поэтому $\alpha \gg \beta$. Если $\beta = 0$, то эталоном рубрики будет просто центроид всех её

документов. Отсюда название этого частного случая классификатора Роше — центроидный алгоритм.

Классификатор Роше очень легко реализуем, а также не является ресурсоёмкий. Его качество, тем не менее, тоже посредственно — особенно, если есть рубрики, представляющие собой объединения несвязанных кластеров.

3.2 Классификаторы на основе решающих правил

В классификаторах на основе решающих правил, пространство данных моделируется набором правил, на левой стороне которого находится условие на набор признаков, а на правой – метка класса. Набор правил генерируется из обучающей выборки. Для данного тестового объекта мы получаем набор правил, для которых он удовлетворяет их условию на левой стороне. Затем определяем метку класса как функцию от полученных меток класса правил.

В более общем виде, левая сторона правила является логическим условием, выраженным в виде дизъюнктивной нормальной формы (ДНФ). Тем не менее, в большинстве случаев, условие на левой стороне гораздо проще и представляет собой набор термов, все из которых должны наличествовать в документе для удовлетворения условию. *Отсутствие* термов как признак редко используется, потому что такие правила не являются сколь либо информативными для разреженных текстовых данных, в которых большинство слов лексикона обычно и не будут присутствовать по умолчанию (собственно, свойство разреженности в чистом виде). Также, несмотря на то, что пересечение условий на наличие терма используется часто, объединение таких условий редко используется в одном правиле. Это потому, что такие правила могут быть разделены на два. Например, правило $Honda \cup Toyota \Rightarrow$ *Машины* может быть заменено двумя правилами $Honda \Rightarrow$ *Машины* и $Toyota \Rightarrow$ *Машины* без потери информативности. С другой стороны, правило $Honda \cap Toyota \Rightarrow$ *Машины* точно более информативно, чем два различных правила. Таким образом, на практике, для таких данных как текст, правила лучше всего выражать в виде простого объединения условий на наличие терма.

Одним из основных принципов при построение набора правил является то, что пространство ответов должно быть покрыто хотя бы одним правилом. В большинстве случаев, это достигается построением различных наборов правил для каждого класса, и одного правила «по умолчанию», которое покрывает все остальные экземпляры.

Несколько критериев могут быть использованы для получения правил из тренировочных данных. Два наиболее используемых условия, которые используются для создания правил, это поддержка (support) и уверенность (confidence). Эти условия являются общими для всех классификаторов на основе решающих правил [10] и могут быть определены следующим образом:

- **Поддержка:** Показывает абсолютное число экземпляров в тренировочных данных, для которых данное правило релевантно. Например, при наборе из 100 000 документов, правило, обе стороны которого верны для 50 000 документов, является более важным, чем правило, которому удовлетворяют всего 20 документов. Проще говоря, поддержка показывает статистический *объем*, связанный с правилом. Тем не менее, она не показывает *силу* правила.

- **Уверенность:** Показывает условную вероятность, что правая часть правила удовлетворена, если удовлетворена его левая часть. Она является более прямой мерой силы оцениваемого правила.

Заметим, что это не все возможные меры, но очень широко используемые в литературе по анализу данных и машинному обучению [10] как для текстовых, так и для иного рода данных ввиду своей интуитивной природы и простоты интерпретации.

Во время обучения мы строим правила, основываясь на выше определенных мерах качества. Для данного тестового объекта мы определяем все соответствующие ему правила. Так как возможны наложения, возможно, что больше чем одно правило окажется релевантным. Если все метки классов на правой их стороне окажутся одинаковыми, метка для объекта выбирается очевидным образом. С другой стороны, задача становится интереснее, когда имеются конфликты меток между этими правилами. Имеется широкий набор различных методов для ранжирования правил [10] и выбора наиболее релевантного правила как функции от них. Например, популярным подходом является ранжирование правил по величине их уверенности и выбор k самых «уверенных» как наиболее релевантных. Тогда ответом для тестируемого объекта будет служить множество меток отобранных правил.

3.3 Вероятностные классификаторы

Вероятностные классификаторы рассматривают решение об отнесении документа $\mathbf{d}$ к классу c как вероятность $\mathsf{P}(c|\mathbf{d})$ принадлежности этого документа к этому классу, и, соответственно, вычисляют её по теореме Байеса:

$$\mathsf{P}(c|d) = \frac{\mathsf{P}(\mathbf{d}|c)\,\mathsf{P}(c)}{\mathsf{P}(\mathbf{d})}$$

3.3.1 Наивный байесовский классификатор

Вероятность $\mathsf{P}(\mathbf{d})$ не требует вычислений ввиду того, что это константа для всех рубрик. Чтобы вычислить $\mathsf{P}(\mathbf{d}|c)$, тем не менее, нужно сделать некоторые предположения насчет документа $\mathbf{d}$. При представлении документа в виде вектора признаков ($\mathbf{d} = (t_1, \ldots, t_n)$) наиболее общим является предположение, что все координаты независимы, т.е.

$$\mathsf{P}(\mathbf{d}|c) = \prod_i \mathsf{P}(w_i|c).$$

Классификаторы, основанные на этом предположении, называют наивными Байесовскими классификаторами. «Наивными», т.к. предположение никогда не проверяется и, скорее всего, вообще не верно. Тем не менее, попытки опустить это предположение и воспользоваться вероятностными моделями с зависимостями координат пока не дали никаких заметных улучшений.

3.4 Линейные классификаторы

Линейными называют классификаторы, в которых предсказанное значение вычисляется в виде $c = \mathbf{w} \cdot \mathbf{d} + b$, где $\mathbf{d} = (d_1, \ldots, d_l)$ – нормализованный вектор из частот слов в документе, $\mathbf{w}$ – вектор линейных весов той же размерности, что и признаковое пространство, а b – некоторое скалярное значение. Естественной интерпретацией для c в дискретном случае будет разделяющая гиперплоскость между различными классами. Машина опорных векторов (SVM) [11] – один из классификаторов данного вида, который пытается искать «хорошие» линейные разделители сежду разными классами.

Регрессионные модели (такие, как метод наименьших квадратов) – более явные и традиционные статистические методы для классификации текстов. Тем не менее, они обычно используются в случаях, когда целевые переменные являются числовыми, а не номинальными. Несколько подходов были предложены в литературе для адаптации таких методов для случая классификации текстовых данных [12]. Сравнение различных методов линейной регрессии для классификации, включая SVM, можно найти в [13].

Наконец, простые нейронные сети также являются одной из форм линейных классификаторов. Самая простая из них, известная как персептрон (или однослойная нейронная сеть) сама по себе создана для линейного разделения и хорошо работает для классификации документов. Тем не менее, используя несколько слоев нейронов, возможно обобщить подход для нелинейного разделения. Далее мы рассмотрим различные линейные методы для классификации текстов.

3.4.1 Классификатор SVM

Машина опорных векторов (Support Vectors Machine, SVM) была предложена в [11] для числовых данных. Главным принципом SVM является определение разделителя в искомом пространстве, который разделяет классы наилучшим образом. Рассмотрим пример, показанный на рисунке 1, на котором мы имеем два класса, обозначенных через «x» и «o». На рисунке также обозначены три гиперплоскости – A, B, и C. Очевидно, что гиперплоскость A производит наилучшее разделение классов, потому что расстояние от неё до любых точек максимально. Иными словами, гиперплоскость A максимизирует *зазор*. Отметим, что вектор нормали к этой гиперплоскости является направлением в признаковом пространстве, вдоль которого мы имеем максимальную разделенность. Одно из преимуществ SVM-метода в том, что так как он пытается определить оптимальное направление разделения признакового пространства, рассматривая комбинации признаков, он достаточно устойчив к большим размерностям. В [14] было отмечено, что текстовые данные идеально подходят для классификатора SVM из-за разреженных данных большой размерности, что является отличительной особенностью текстов.

Заметим, что задача поиска лучшего разделителя является задачей оптимизации, которая может быть сведена к задаче квадратичного программирования. Так, большинство методов используют метод Ньютона для итерационной минимизации выпуклой функции. Он бывает достаточно медлителен, особенно для данных высокой размерности, таких, как текстовые документы. В [15] было показано, что раз-

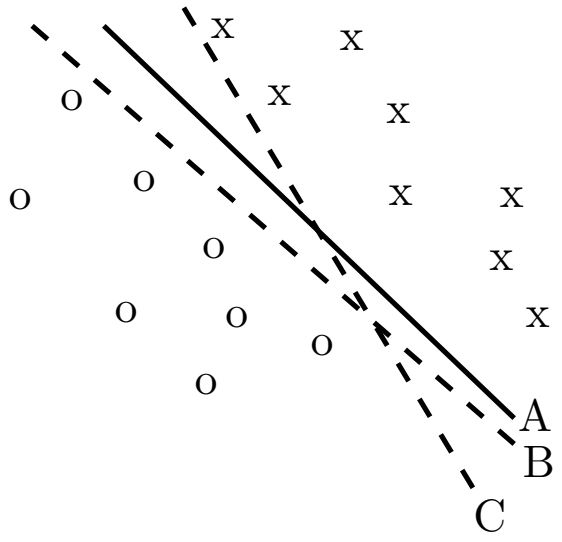


Рис. 1: Какая разделяющая поверхность лучше?

бивая большую задачу квадратичного программирования на набор меньших задач, можно найти эффективное решение.

3.4.2 Классификаторы на основе регрессии

Регрессионное моделирование – метод, который часто используется для поиска связей между вещественными параметрами. Тем не менее, его можно применить и для задач классификации, потому что бинарные значения можно рассматривать как частный случай вещественных, и некоторые регрессионные методы, такие как логистическая регрессия, также естественным образом моделируют и дискретные значения.

Одним из ранних применений регрессии к классификации текстов является линейный метод наименьших квадратов (ЛМНК) [12], который работает следующим образом. Предположим, что предсказанная метка класса будет равна линейной комбинации $p_i = \mathbf{w} \cdot \mathbf{d} + b$, а истинной меткой будет y_i, тогда целью нашего обучения будут значения параметров $\mathbf{w}$ и b такие, что сумма квадратичных отклонений $\sum_{i=1}^{n}(p - y_i)^2$ минимальна. На практике значение b берется равным 0 во время обучения. Пусть P будет вектором размерности $1 \times d$ бинарных значений принадлежности классам. Таким образом, если X будет матрицей термов размерности $n \times d$, то нашей задачей будет найти вектор размерности $1 \times d$ регрессионных коэффициентов $\mathbf{w}$, для которого норма Фробениуса $||\mathbf{w} \cdot X^T - P||$ минимальна. Задачу можно легко обобщить от

бинарной классификации на случай k классов за счет выбора матрицы P размера $k \times n$ из все также бинарных значений. В этой матрице только одно из значений в каждом столбце будет равно 1, и соответствующая строка с единицей будет отвечать за класс, к которому принадлежит объект. Аналогично, A будет матрицей размера $k \times d$ в этом случае. Линейный метод наименьших квадратов сравнивали с набором других методов [12, 13], и он показал себя очень устойчивым на практике.

Более естественным способом решения задачи классификации на основе регрессии является логистическая регрессия [16], которая происходит из предыдущего метода заменой функции p_i на функцию правдоподобия. Точнее, вместо использования $p_i = \mathbf{w} \cdot \mathbf{d} + b$ для прямой подгонки метки y_i, мы полагаем, что вероятность наблюдения этой метки равна

$$p(c = y_i|\mathbf{d}_i) = \frac{\exp(\mathbf{w} \cdot \mathbf{d} + b)}{1 + \exp(\mathbf{w} \cdot \mathbf{d} + b)}.$$

Проведя логарифмические преобразования, можно заметить, что

$$\log \frac{p(c = y_i|\mathbf{d}_i)}{1 - p(c = y_i|\mathbf{d}_i)} = \exp(\mathbf{w} \cdot \mathbf{d} + b).$$

Таким образом, логистическая регрессия все еще остается линейным классификатором, т.к. разделяющая поверхность является линейной функцией параметров. В случае бинарной классификации, вероятность $p(c = y_i|\mathbf{d}_i)$ может быть использована для определения метки класса (например, используя порог, равный 0.5). В случае классификации со многими классами, метка класса с наибольшим значением данной вероятности будет назначена для $\mathbf{d}_i$. Если нам дан тренировочный набор данных $\{(\mathbf{d}_1, y_1, \ldots, \mathbf{d}_l, y_l)\}$, то модель логистической регрессии может быть построена на основе выбора таких параметров $\mathbf{w}$, которые максимизируют функцию правдоподобия $\prod_{i=1}^{n} p(y_i|\mathbf{d}_i)$.

Видно, что регрессионные классификаторы очень схожи с моделью SVM. Более того, т.к. ЛМНК, логистическая регрессия и SVM суть линейные классификаторы, они идентичны на уровне концепции; главное отличие между ними лежит в постановке задачи оптимизации и её решении. Как и в случае случае классификатора SVM, обучение регрессионной модели так же использует затратный оптимизационный процесс, включающий в себя вычисления сингулярных разложений матриц.

3.4.3 Нейронные сети

Базовой единицей нейронной сети является нейрон. Каждый нейрон получает набор входов, которые будем обозначать $\mathbf{d}_i$, которые в нашей задаче будут обозначать вхождение термов в i-й документ. Каждый нейрон также ассоциирован с набором весов $\mathbf{w}$, которые используются для подсчета функции входов $f(\cdot)$. Типичной такой функцией, используемой в нейронной сети, является линейная:

$$p_i = \mathbf{w} \cdot \mathbf{d}_i.$$

Таким образом, для вектора $\mathbf{d}_i$ и лексикона из d слов, вектор весов $\mathbf{w}$ должен также содержать d элементов. Теперь рассмотрим задачу бинарной классификации, в

которой метки будут из множества $\{+1, -1\}$. Предположим также, что истинным классом $\mathbf{d}_i$ является y_i. В этом случае, знак предсказанной функции p_i будет определять как раз метку класса.

Главной идеей обучения является изначально произвольный выбор весов, а затем пошаговое обновление при наличии ошибок функции на тренировочных данных. «Силу» обновления на каждом шаге будет определять параметр μ, называемый также скоростью обучения. Итого, мы получили т.н. *персептронный алгоритм*:

Вход: Скорость обучения μ, обучающая выборка $(\mathbf{d}_i, y_i), i = 1 \dots l$;
Выход: Веса $\mathbf{w}$;

1: Инициализировать $\mathbf{w}$ произвольными маленькими числами;
2: **повторять**
3: **для** $i = 1, \dots, n$
4: **если** знак $\mathbf{w} \cdot \mathbf{d}_i$ не совпадает с y_i **то**
5: Обновить веса $\mathbf{w}$ в соответствии со скоростью обучения μ;
6: **конец условия**
7: **конец цикла**
8: **пока** веса $\mathbf{w}$ не стабилизируются.

Веса $\mathbf{w}$ обычно изменяются на величину, пропорциональную $\mu \cdot \mathbf{d}_i$. Мы также отметим, что много различных способов обновления было предложено в литературе. Например, можно изменять веса каждый раз на μ, а не на $\mu \cdot \mathbf{d}_i$. Это разумно в делать в области классификации текстов, где все признаки имеют небольшие неотрицательные значения. Несколько реализаций методов на основе нейронных сетей были предложены в [17, 18, 19].

Естественным вопросом является то, как же использовать нейронную сеть, когда классы могут быть линейно неразделимы. В этом случае, если использовать *многослойные нейронные сети*, можно получить более полные классы разделяемых поверхностей. В таких сетях выходы одного слоя подаются на входы нейронов следующего слоя. Двух слоев достаточно для приближения сколько угодно сложных структур из многоугольников. Самым распространённым алгоритмом обучения является алгоритм обратного распространение ошибки [20]. Тем не менее, эксперименты показали [19], что для классификации текстов рассмотрение многослойных нейронный сетей не дает значимого выигрыша перед однослойным персептроном.

Список использованных источников

[1] Яндекс. Яндекс.Каталог. http://yaca.yandex.ru/, 2012.

[2] DMOZ. Open Directory Project. http://www.dmoz.org/, 2012.

[3] Лаборатория Касперского. Kaspersky Security Bulletin. Спам в 2011 году. http://www.securelist.com/ru/analysis/208050743/rss/analysis, 2012.

[4] Газета «Коммерсантъ». «Яндекс» добавил выручки. http://www.kommersant.ru/doc/1314584/, 2010.

[5] TunedIT. JRS 2012 Data Mining Competition. http://tunedit.org/challenge/JRS12Contest, 2012.

[6] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. pages 412–420. Morgan Kaufmann Publishers, 1997.

[7] T. Cover and J. Thomas. *Elements of information theory.* Wiley, New York, 1991.

[8] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[9] Soumen Chakrabarti, Shourya Roy, and Mahesh V. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 658–669. VLDB Endowment, 2002.

[10] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. pages 80–86, 1998.

[11] C.Cortes and V.Vapnik. Support vector networks. In *Machine Learning*, volume 20, pages 237–297, 1995.

[12] Yiming Yang and Christopher G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Trans. Inf. Syst.*, 12(3):252–277, July 1994.

[13] Jian Zhang and Yiming Yang. Robustness of regularized linear classification methods in text categorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 190–197, New York, NY, USA, 2003. ACM.

[14] Thorsten Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. pages 143–151, 1997.

[15] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, CIKM '98, pages 148–155, New York, NY, USA, 1998. ACM.

[16] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, 2002.

[17] Ido Dagan, Yael Karov, and Dan Roth. Mistake-driven learning in text categorization. In *The second conference on empirical methods in natural language processing*, pages 55–63, 1997.

[18] Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. Feature selection, perceptron learning, and a usability case study for text categorization. *SIGIR Forum*, 31(SI):67–73, July 1997.

[19] Erik Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting, 1995.

[20] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical neural networks for text categorization. In *In Proceedings of the 22 nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 281–282, 1999.

[21] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[22] Thomas J. Santner and Diane E. Duffy. *The Statistical Analysis of Discrete Data.* Springer-Verlag, 1989.

[23] R. J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[24] M. F. Porter. *Readings in information retrieval.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.