Прикладные задачи анализа данных

Анализ текста: классификация и регрессия

Дьяконов А.Г.

Московский государственный университет
имени М.В. Ломоносова (Москва, Россия)

Этапы работы с текстом

1. Извлечение (из pdf, html и т.п. + кодировка)
2. Разбиение на слова и предложения (tokenization)
3. Удаление стоп-слов
4. Лемматизация (stemming)
5. Машинное обучение

задачи с большими матрицами!

специфичность «признаков»!

специальные функционалы качества!

Токенизация – разбиение на структурные единицы

(кстати, ещё технология шифрования данных электронных платежей)

Границы предложений:

«Тов. Иванов А.А. приехал в г. Иванов.»

Границы структурных единиц - слов:

10/09/2013

123-17

в конце концов

o’neill

oneill

o neill

o’ neill

Вопрос: слово с тремя дефисами

Токенизация

Грубо: разбиение по пробелам

НО: U.S.A. → USA

C++, C# (си и эс)

Lebensversicherungsgesellschaftsangestellter

Стоп-слова

Служебные части речи, предлоги, союзы, артикли

25 стоп-слов

a	from	on
an	has	that
and	he	the
as	in	to
at	is	was
be	it	were
by	its	will
for	of	with

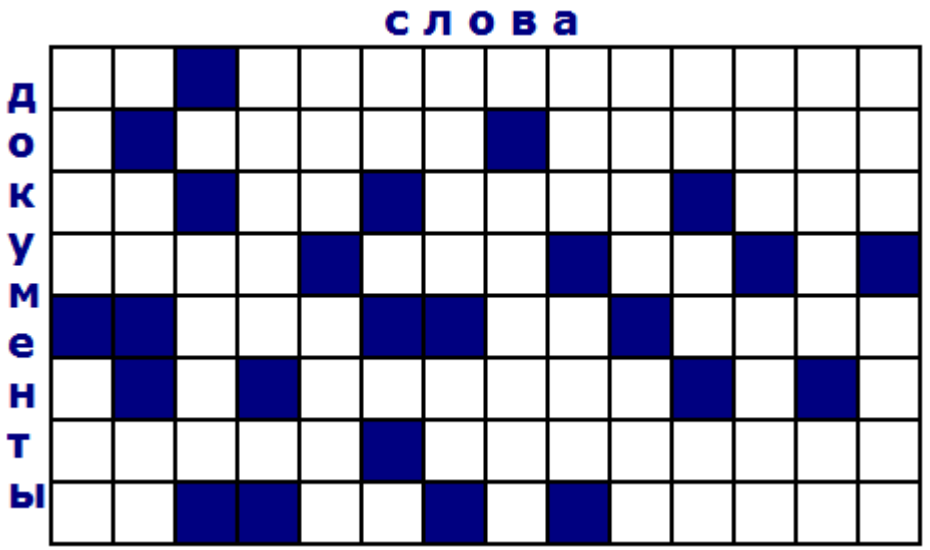
Проблемы: «To be or not to be», «Let it be»

Векторная модель представления документа

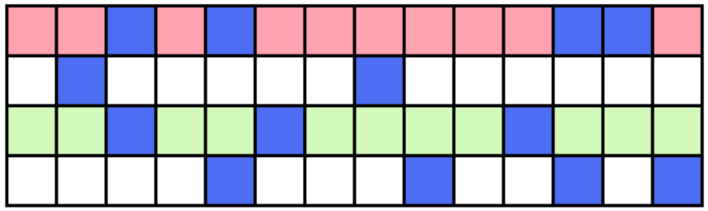
Тов Иванов А А приехал г Иванов

А		Иванов			Тов		г			приехал	
2			2			1		1			1	

Сразу думаем – матричное представление



Матрица «документ-термин» – другой термин!



Как посчитать «похожесть» строк?

Меры близости

Бинарные

$$\frac{2|X \cap Y|}{|X|+|Y|}$$ **Dice**

$$\frac{|X \cap Y|}{|X \cup Y|}$$ **Jaccard**

$$\frac{|X \cap Y|}{\min(|X|,|Y|)}$$ **Overlap**

$$\frac{\sum_i x_i y_i}{\sum_i x_i + \sum_i y_i}$$

$$\frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$$

$$\frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}}$$

Недостатки

1. Что есть слова?

«Нейронная сеть» «Нейронные сети»

Для этого – лемматизация

2. Нет учёта порядка слов

«Петя любит Машу» «Маша любит Петю»

Выход: пары, тройки слов
k-граммы (и на буквах приём!)

«Пет люб» «люб Маш» – «Маш люб» «люб Пет»

3. Стоп слова специфичны

Выход: TF*IDF

Терминология

Лемматизация — приведение словоформы к лемме — её нормальной (словарной) форме.

кошками → кошка
забежав → бежать
краснея → краснеть

нужно уметь определять часть речи (см. дальше)!

Стемминг — нахождение основы слова.
Основа слова необязательно совпадает с морфологическим корнем слова.

Стемминг

Выделение основы

«сети», «сетью», «сетевой» – **СЕТ**

Лемматизация

Приведение к словарной форме

«сети», «сетью» – **СЕТЬ**

«сетевой» – **СЕТЕВОЙ**

Проблемы: грамматическая омония

«Черепах»

«Падали»

«Печь»

«Простой»

«Дорога стала уже»

Стемминг

Первый подход – поиск флективной формы в таблице.

– не работает для новых слов (Ipads → Ipad)
– надо хранить таблицу (для флективных языков большая таблица)
+ таблицу можно генерировать автоматически (run → running, runs, runned, runly)

Для справки: Языки (точнее, словообразование)

флективные	агглютинативные
латинский, немецкий, русский словообразование с использованием аффиксов, сочетающих сразу несколько грамматических значений ДОБРЫЙ (ЫЙ – единственное число, мужской род и именительный падеж)	тюркские, монгольские, тунгусо-маньчжурские, корейский, японский, грузинский, баскский, абхазо-адыгские, дравидийские, шумерский, эсперанто татарский: «в его письмах» хатларында (хат «письмо», -лар- формант множественного числа, -ын- притяжательный формант 3-го лица, -да формант местного падежа)

Стемминг (Алгоритм Портера)

Второй подход – усечения

5 циклов усечения

1 этап	
sses → ss	caresses → caress
ies → i	ponies → poni
ss → ss	caress → caress
s →	cats → cat

ational → ate tional → tion	
(m>1)ement →	replacement → replace cement → cement

Snowball Портера – фреймворк для создания стеммеров

Стемминг

Усечения снабжаются дополнительными правилами

friendlies → friendl (?)

friendlies → friendly → friend (+)

Стемминг

Оригинал текста

Such an analysis can reveal features that are not easily visible from the variations in the individual genes.

Алгоритм Портера (1980, премия Стрикса)

Such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene.

Алгоритм Ловинса (однопроходный, 1968, **первый стеммер**)

Such an analysis can reve featur that ar not eas vis from th vari in th individu gen.

Алгоритм Пейса

Such an analysis can rev feat that are not easy vis from the var y in the individ gen.

Но есть другие языки: русский, словенский…

Стемминг

Недостатки

operate operating operates operation operative ... → oper

news new → new

Вариант решения:

- **стемминг на основе корпуса текстов (затачиваемся под лексику)**
- **учёт контекста (+ вероятностные стеммеры)**

Глобальный недостаток

«Применение рыболовных сетей»,
«Информационная сеть: практика внедрения и применения»,
«Применение байесовских сетей для обработки изображений»,
«Применение правила delta-bar-delta при настройке нейронных сетей»

TF*IDF

Term frequency

$$\frac{f_{ij}}{\sum_k f_{jk}}$$

Inverse document frequency

$$\log \frac{m}{\sum_k I[f_{kj} > 0]}$$

Смысл: инвариантность!

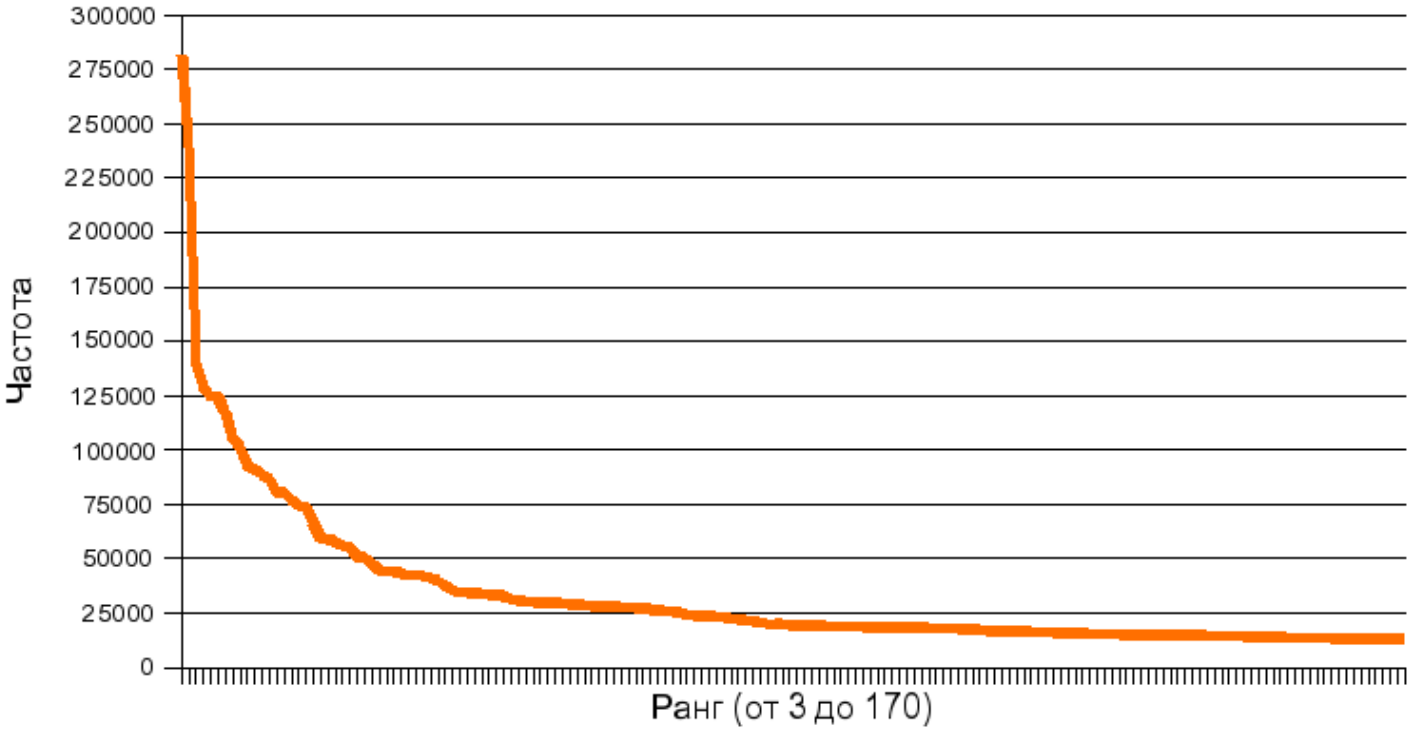
Например, дублирование текста.

«На подумать»

Чем ближе слово к началу документа, тем оно ценнее…

Закон Zipfà

Произведения ранга слова на частоту = const



Пример на частотах слов Википедии

Оценки качества

Точность PRECISION

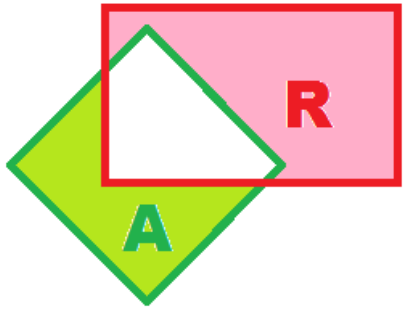
$$\frac{|R \cap A|}{|A|}$$

Полнота RECALL

$$\frac{|R \cap A|}{|R|}$$

R - правильный ответ (релевантные),

A - ответ алгоритма.



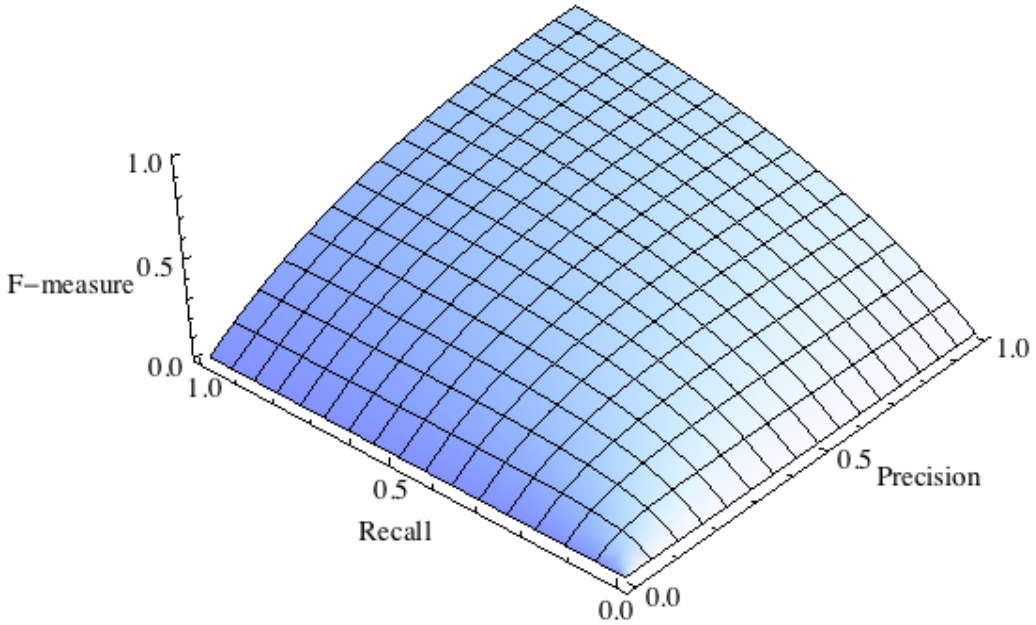
11-точеченый график полноты/точности

F-мера

$$\frac{2}{\frac{1}{\left(\frac{|R\cap A|}{|R|}\right)}+\frac{1}{\left(\frac{|R\cap A|}{|A|}\right)}}$$

$$\frac{2}{\frac{|R|}{|R\cap A|}+\frac{|A|}{|R\cap A|}}$$

$$2\frac{|R\cap A|}{|R|+|A|}$$



Примеры задач

1. Иерархическая классификация
2. Обнаружение оскорблений

Методы

1. kNN

косинусная мера сходства

Если AUC – не надо думать о нормировкам по классам

2. SVM

без ядер или с полиномиальными ядрами

3. Метод Роше/Роккио

Классификация спама

Сами мы легко распознаём спам

«Контекстность» спама

«диваны», «погрузка»

Письмо:

Заголовок

Текст

Вложение

```
Return-path: <vasya@bk.ru>
Received: from mail by f64.mail.ru with local
id 1INyJO-0008s3-00
for djakonov@mail.ru; Thu, 23 Aug 2007 02:04:26 +0400
Received: from [84.163.74.42] by win.mail.ru with HTTP;
Thu, 23 Aug 2007 02:04:26 +0400
From: =?windows-1251?Q?=CA=EE=ED=F1=F2=E0=ED=F2=E8=ED_=C2=EE=F0=EE=ED=F6=EE=E2?= < vasya@bk.ru>
To: igor@mail.ru
Subject: =?windows-1251?Q?RE=3A_=E0=F2=E5=F0=EE=F1=EA=EB=E5=F0=EE=E7?=
Mime-Version: 1.0
X-Mailer: mPOP Web-Mail 2.19
X-Originating-IP: [84.163.74.42]
Date: Thu, 23 Aug 2007 02:04:26 +0400
Reply-To: =?windows-1251?Q?=CA=EE=ED=F1=F2=E0=ED=F2=E8=ED_=C2=EE=F0=EE=ED=F6=EE=E2?= <
vasya@bk.ru>
Content-Type: text/plain; charset=windows-1251
Content-Transfer-Encoding: 8bit
Message-Id: <E1INyJO-0008s3-00. vasya-bk-ru@f64.mail.ru>
```

Рассматриваем только текстовую классификацию

Р.А.С.С.Ы.Л.К.А.

jpg с рекламой

$ПАМ

ff0000

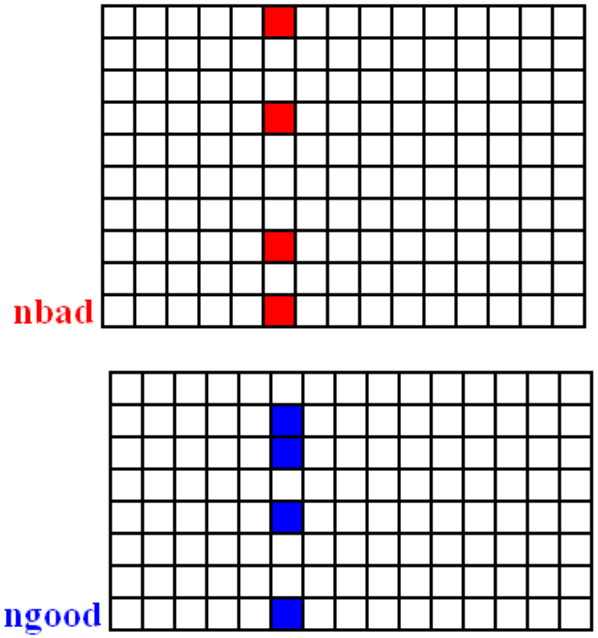
url

Классификация спама
(один из первых методов)

Анализ только текста, даже без доп. инф. типа ff0000 (что это???)

Paul Graham

http://www.paulgraham.com/antispam.html



если

$$g + b \geq 5$$

коэф. спамовости слова

$$\gamma = \frac{b/nbad}{(b/nbad)+(g/ngood)}$$

обрезка коэф.

$$\gamma = \max[\min[\gamma, 0.99], 0.01]$$

незнакомые слова = 0.4

Оценка спамовости слов

perl 0.01
pop3 0.03
difficult 0.07

madam 0.99
promotion 0.99
investment 0.86

approach offers 0.1

special offers 0.95

из письма выбираются 15 «наиболее интересных слов»

Какие это слова?

на их основе – коэф. спамовости письма

Как например?

пороговое решающее правило

Плюсы

Интерпретация всех коэффициентов

Возможность параметризации и настройки на конкретный ящик

Реальная задача

Обучающая выборка (4000/100):

Новости – Жертвы спам-ловушек

Контрольная выборка (2500/400):

ящики пользователей (3/15)

Задача:

адаптация под новый ящик

(с другим распределением)

Хорошие чужие идеи:

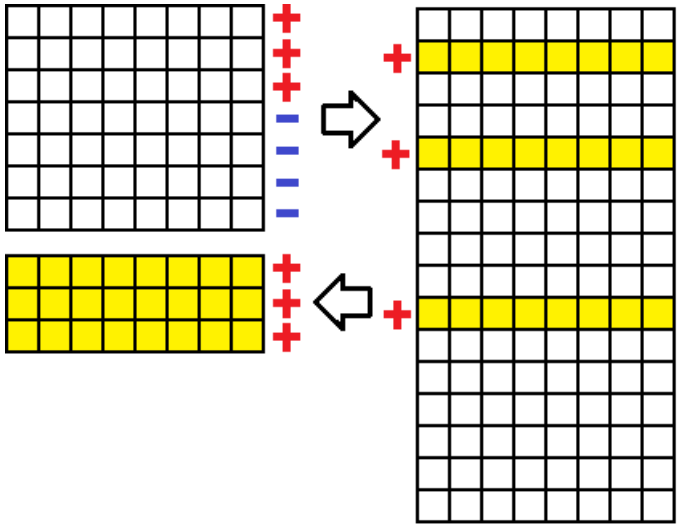
применить алгоритм к новому ящику

выделить те письма, что точно не спам

пополнить ими выборку

повторить

Обучение лексике пользователя!

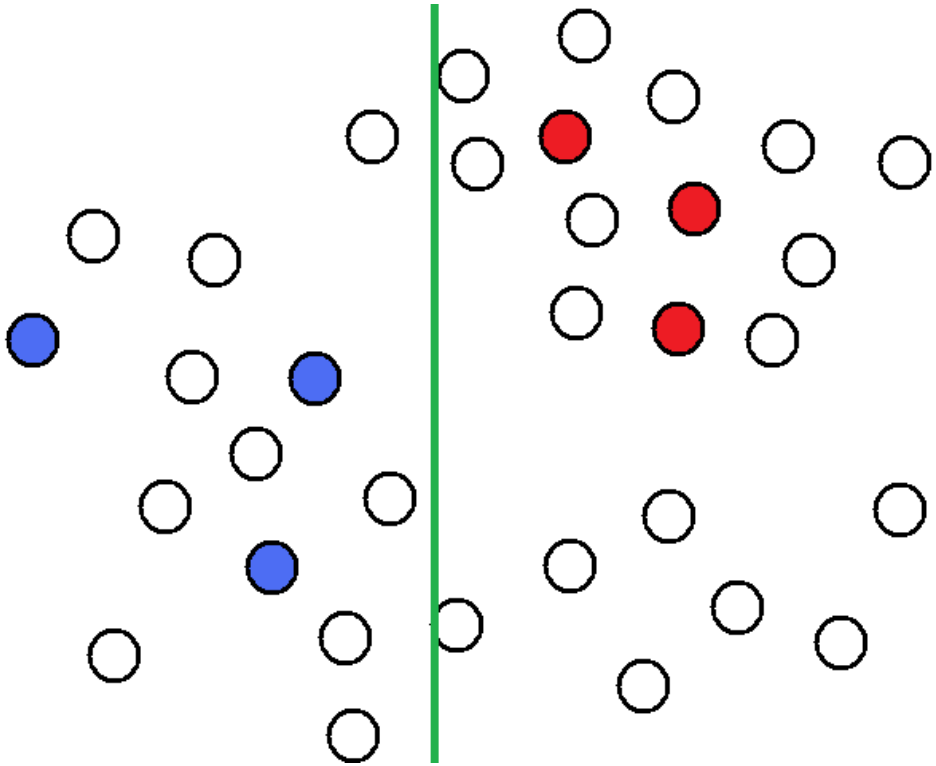


Хорошие чужие идеи:

оценка попарных сходств писем

хранятся только k похожих

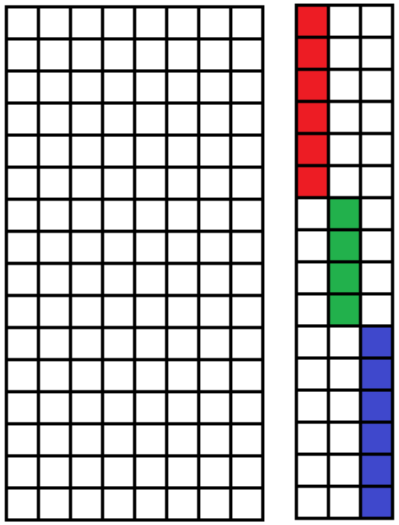
- основа для Global And Local Consistency

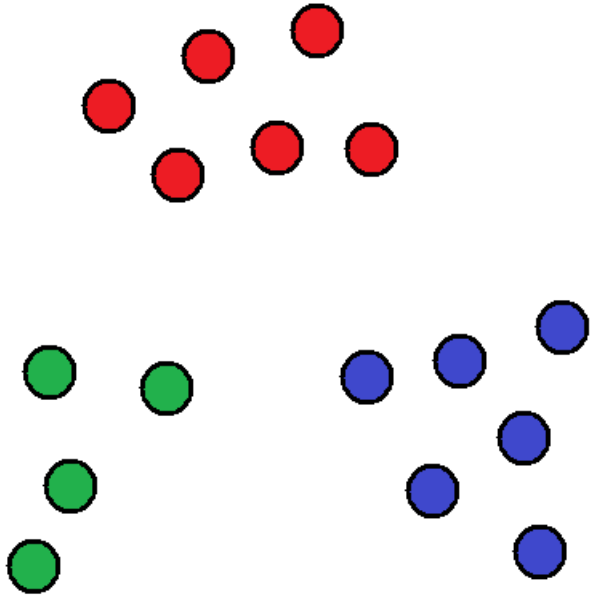


Semi-supervised learning

Хорошие чужие идеи:

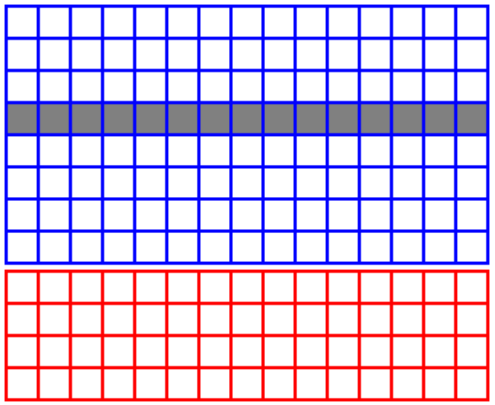
кластеризация
пополнение новыми признаками
(номера кластеров)





Наш алгоритм:

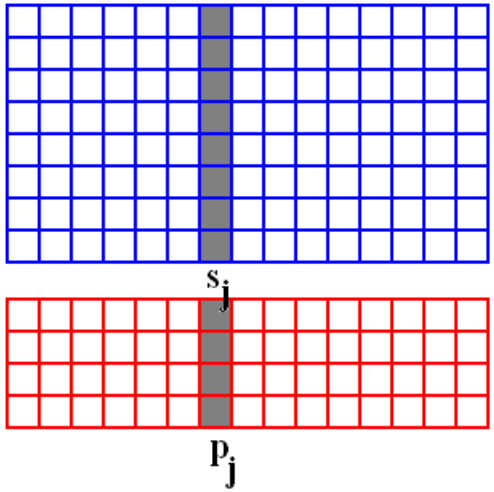
1. **Построковая обработка**



Например:

нормировка/обезличивание

2. **Постолбцовое «суммирование»**



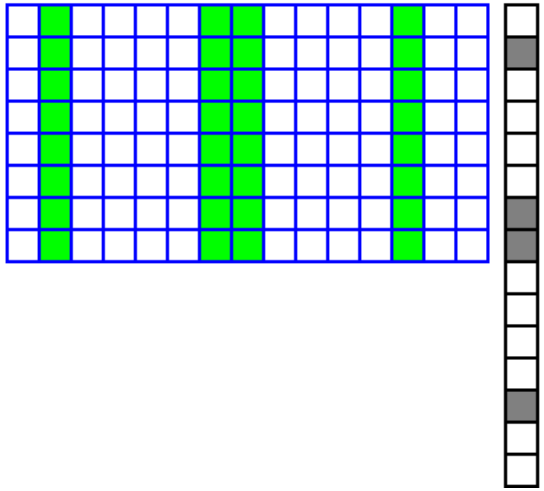
Например: среднее арифметическое

Получили оценки спамовости/неспамовости

3. Применение оценок

Например: линейная комбинация,
сравнить с порогом

Кстати:
хранение sparse-матриц
и перемножение! (можно выделять ненулевые области)



Применение алгоритма:

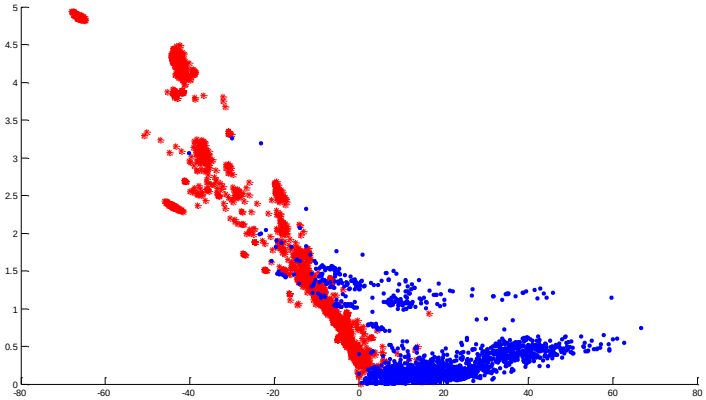
перебор и оптимизация этапов

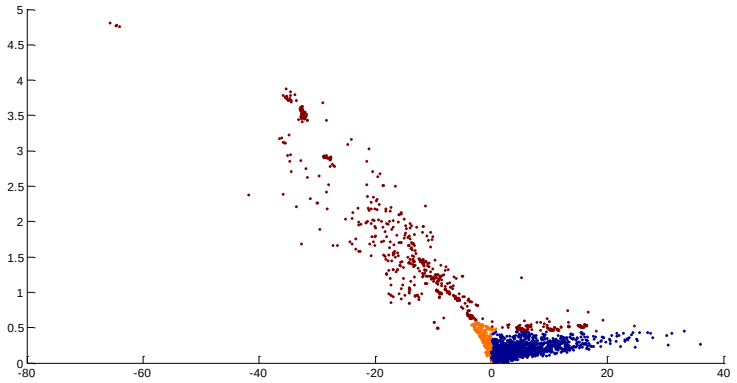
Признак:

F(H(G(S)))

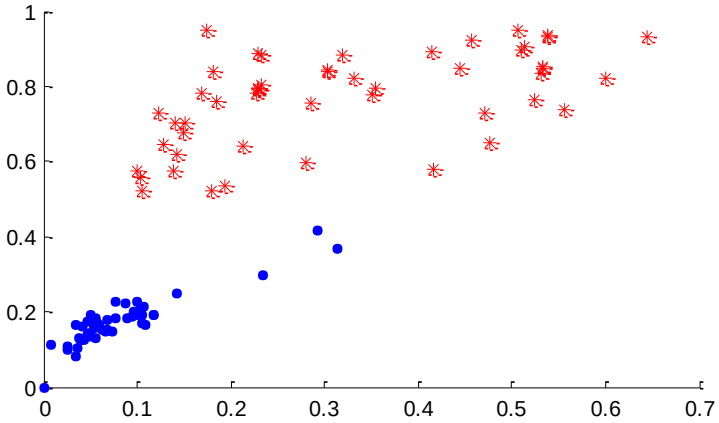
F(H(G(M)))

Именно признаки и искал!

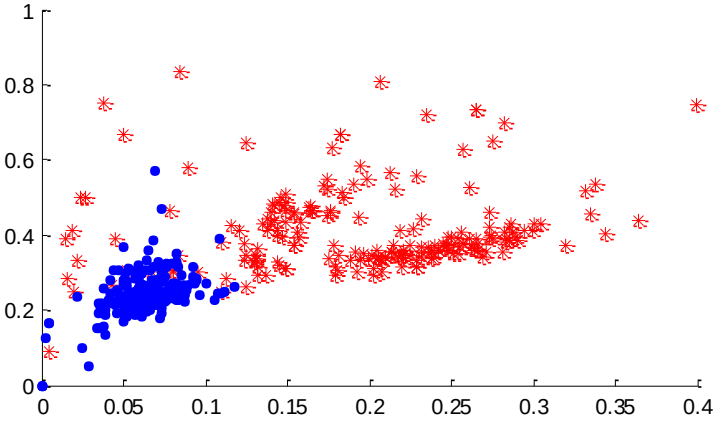




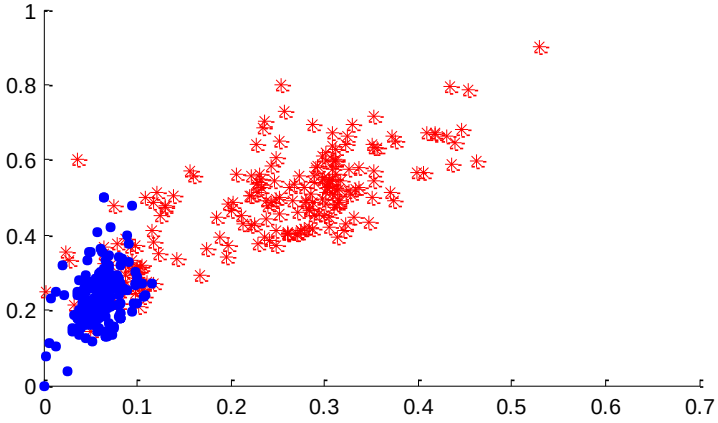
Изменение распределения



база

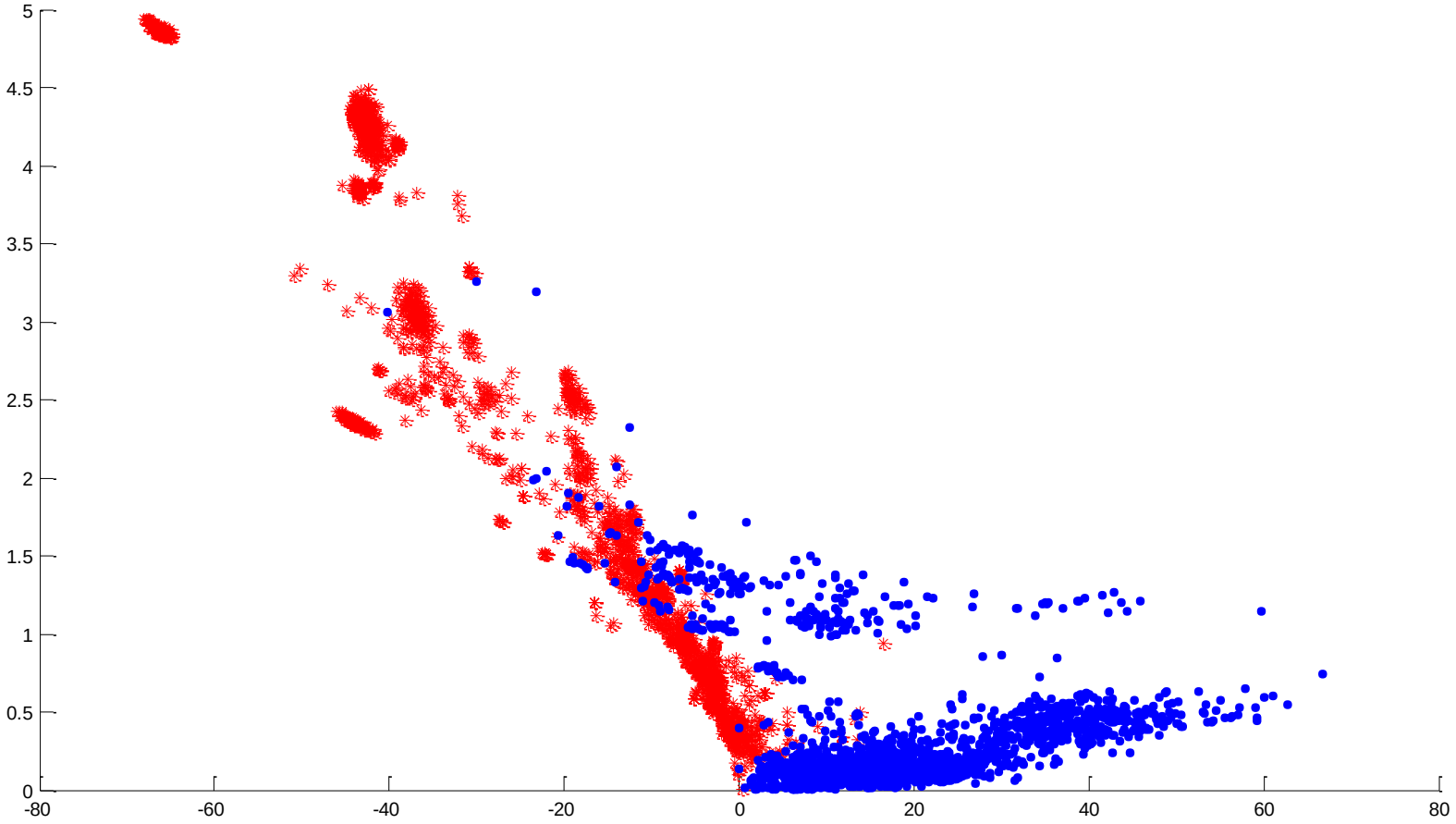


ящик 1



ящик 2

Если искать признаки, а не «решать задачу», то можно «увидеть» данные



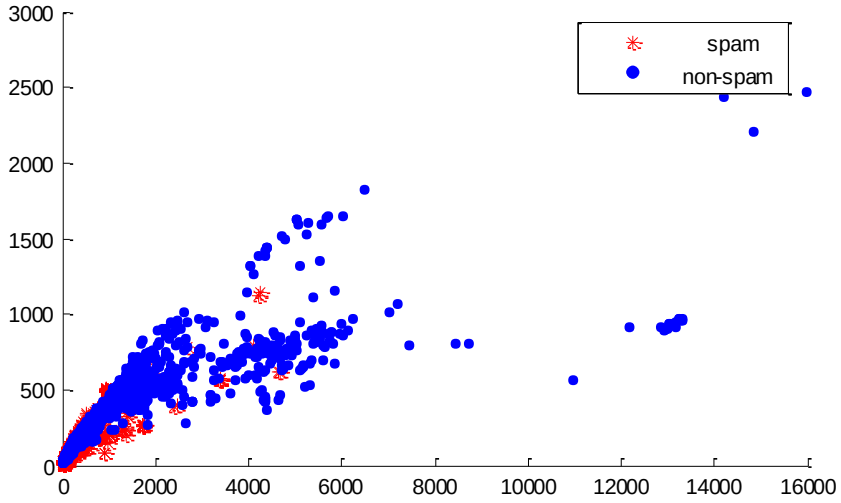
Спамовые кучки

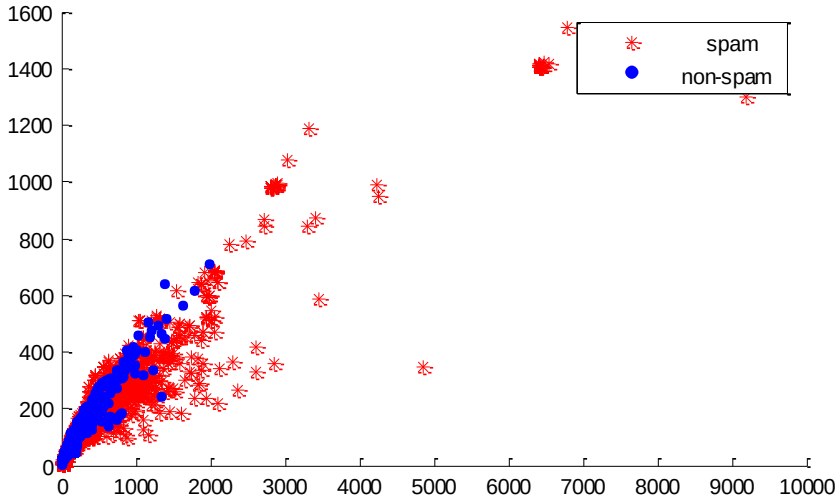
Вот почему рулила идея кластеризации!

Есть «небольшие» группы очень похожих писем – разные виды спама.

Кстати,
а что такое неспамовые кучки?
И почему это не совсем кучки?

Пример плохих (переобученных) признаков



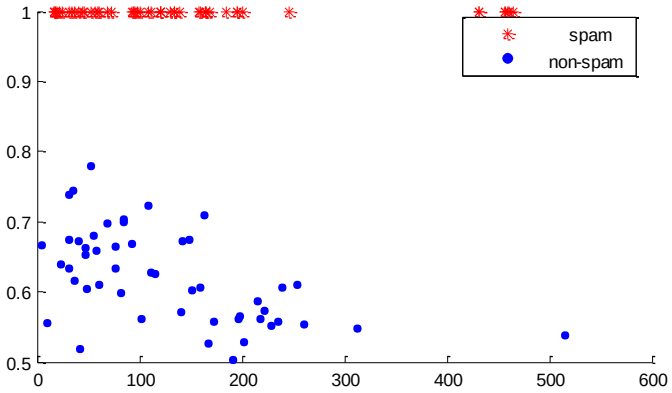


база **ящик**

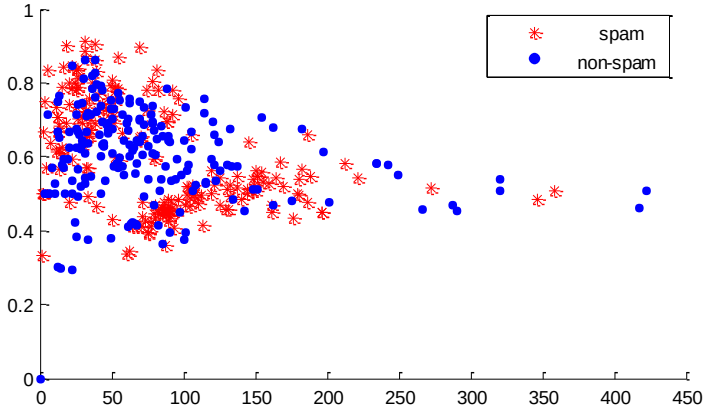
```
sum(B)
sum(B>0)
```

«Длина письма» - плохой признак –
нет устойчивости в ящиках

Пример плохих (переобученных) признаков



База



ящик

Признаки: процент спамовых слов и их число.

Т.е. наши признаки не просто «хорошие», а стабильные!

Вопрос: как измерить стабильность?

Оптимальные этапы:

1. Обезличить и пронормировать

$$\frac{I[f_{ij} > 0]}{\sum_k I[f_{ik} > 0] + \varepsilon}$$

2. Сумма

3. Окончательные признаки для $(x_1, \ldots, x_n)$

$$\sum_{i:\, x_i > 0} \frac{s_i}{s_i + 1} \quad (1)$$

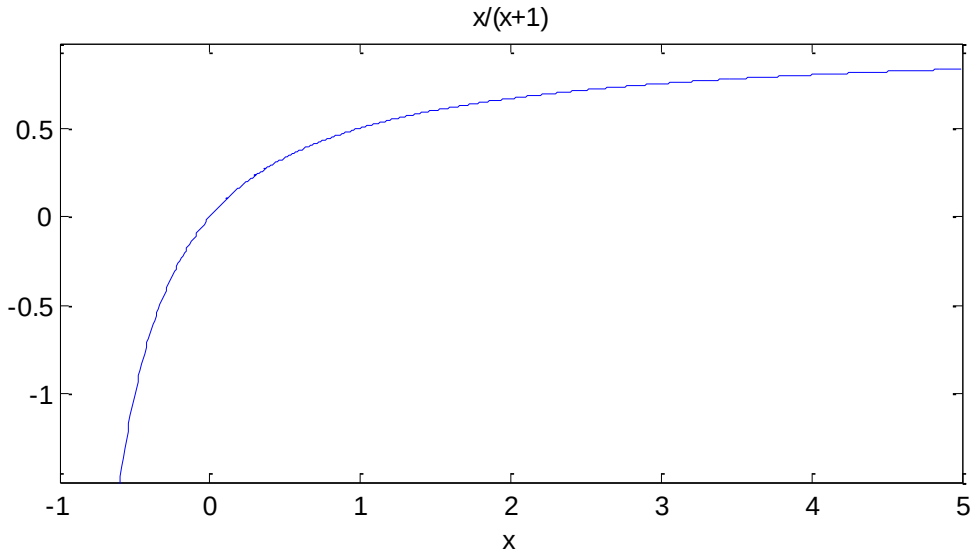
$$\frac{1}{|\{i : x_i > 0\}|} \sum_{i:\, x_i > 0} s_i \quad (2)$$

$$\sum_{i:\, x_i > 0} \frac{n_i}{n_i + 1} \quad (3)$$

$$\sum_{i:\, x_i > 0} \frac{x_i n_i}{n_i + 1} \quad (4)$$

Итоговая картина: в осях (1)-(3), (4).

Почему возникли такие функции



Изначально был логарифм – т.к. сумма логарифмов – произведение (это байесовский подход)

$$\sum_{i:\, x_i>0} f(s_i)$$

Потом – подбор функции.

Итог: что похоже на логарифм, но растёт очень медленно…

Ещё реальная задача: иерархическая классификация текстов

каждый год проходит соревнование

Особенность: классы организованы иерархически.

Интересно: не всегда иерархичность влияет на решение.

Интересно: простейший метод Роккио/Роше

$$\frac{\alpha}{|K_1|}\sum_{\tilde{x}\in K_1}\tilde{x}-\frac{\beta}{|CK_1|}\sum_{\tilde{x}\notin K_1}\tilde{x}$$

В этой задаче:

Переход к центроидам улучшал метод ближайшего соседа

И скорость классификации!

Наверное, это связано с геометрией задачи.

Интерпретация метода: создаём универсальный текст («все новости каталога»).

Метод решения

1. Нормировка

$$f_{ij} \to \ln(1+ f_{ij})$$

часто: разная нормировка у обучения/контроля

2. Вычисление центроидов

$$\frac{1}{|K_j|}\sum_{\tilde{x}\in K_1}\tilde{x}$$

3. Стандартно: tf*idf + cos

Лучше tf*idf не было!

Лучше cos не было!

4. Учёт / не учёт иерархии

kNN	kNN+centroid	ln+kNN+centr.	+ удаление пр.
35%	39%	42%	44%

Large Scale Hierarchical Text Classification

Completed • Swag • 119 teams

Large Scale Hierarchical Text Classification

Wed 22 Jan 2014 – Tue 22 Apr 2014 (2 years ago)

#	Δrank	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	—	anttip *	0.33932	93	Tue, 22 Apr 2014 23:43:14 (-0.4h)
2	—	Alexander D'yakonov (MSU, Moscow, Russia)	0.33568	78	Tue, 22 Apr 2014 20:12:47 (-1.6h)
3	—	nagadomi	0.33025	35	Mon, 21 Apr 2014 15:55:26 (-2.2d)
4	—	Martin Martin	0.29485	10	Mon, 21 Apr 2014 22:37:41 (-0.1h)
5	—	paithan	0.28211	5	Tue, 22 Apr 2014 19:58:23
6	—	student2012	0.28155	9	Tue, 22 Apr 2014 07:24:25
7	—	Dmitriy Anisimov	0.27966	9	Tue, 22 Apr 2014 18:51:57 (-11h)
8	—	Harvard Stat 183	0.27031	38	Tue, 15 Apr 2014 00:52:21 (-24.3h)
9	—	echo	0.26298	5	Tue, 22 Apr 2014 16:53:55 (-3h)
10	—	machine learner	0.25810	21	Tue, 22 Apr 2014 17:52:45

Large Scale Hierarchical Text Classification

325 056 категорий (образуют иерархию)

на вход – матрица текст-слово (число вхождений)

Оценка решения:

$tp_i \;\; fp_i \;\; fn_i$ **– отдельно вычисляются по классам** $i \in \{1,2,\ldots,l\}$

$$P' = \frac{1}{l}\sum_{i=1}^{l}\frac{tp_i}{tp_i + fp_i}$$ **– макроточность**

$$R' = \frac{1}{l}\sum_{i=1}^{l}\frac{tp_i}{tp_i + fn_i}$$ **– макрополнота**

$$F' = \frac{2}{1/P' + 1/R'}$$ **– макро F1-мера**

Large Scale Hierarchical Text Classification

Методы решения

1 место

старый подход (Multinomial Naive Bayes, разные нормировки: tf-idf, BM25,...) + для каждого класса предсказываются объекты, (почти не использовали иерархию)

2 место

tf-idf + kNN (свой взвешенный), не использовал иерархию

3 место

Роше + чуть изменённый tf-idf

7 место

Роше

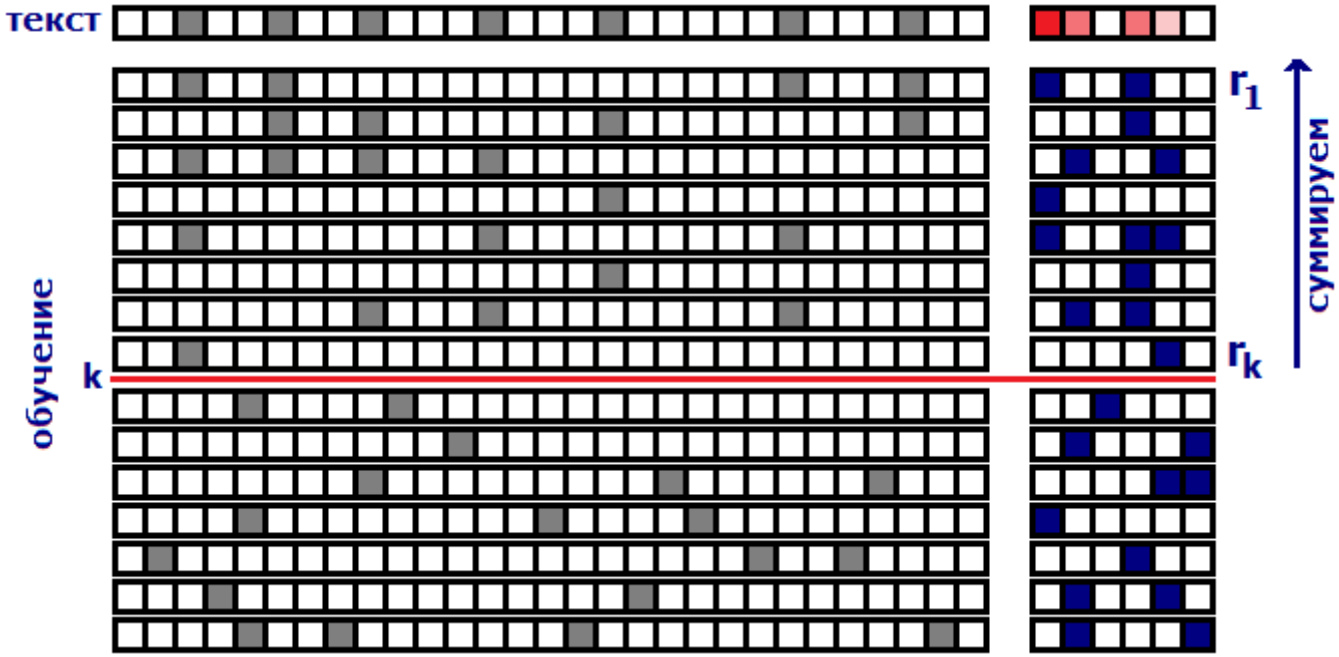
10 место

Ансамбль 4х kNN с разными нормировками (tf-idf, BM25, ...)

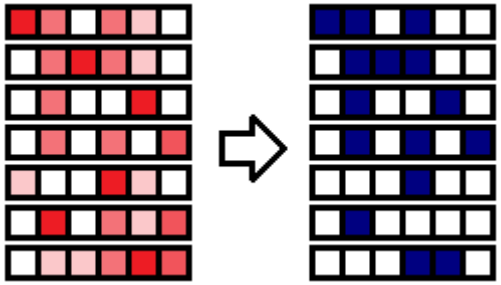
13 место

kNN (свой), удаление ненужных признаков, BM25, ансамбль по объединению ответов

Взвешенный kNN



Итог – матрица оценок



Концепция алгебраического подхода

$$A = B \cdot C$$

распознающий оператор × решающее правило

$$C(\| h_{ij} \|_{q \times l}) = \| \alpha_{ij} \|_{q \times l}$$

простейшее РП – пороговое

$$\alpha_{ij} = \begin{cases} 1, & h_{ij} \geq c, \\ 0, & h_{ij} < c, \end{cases}$$

в специальных задачах –

$$\alpha_{ij} = \begin{cases} 1, & h_{ij} \geq \max[h_{i1}, \ldots, h_{il}], \\ 0, & h_{ij} < \max[h_{i1}, \ldots, h_{il}], \end{cases}$$

Решающие правила

Очень хорошее…

$$\alpha_{ij} = \begin{cases} 1, & h_{ij} / \max[h_{i1}, \ldots, h_{il}] \geq c, \\ 0, & h_{ij} / \max[h_{i1}, \ldots, h_{il}] < c, \end{cases}$$

Почему?

Секрет успеха

$$h_{i1},\ldots,h_{il} : h_{i,\text{rank}_i^1(1)} \ge h_{i,\text{rank}_i^1(2)} \ldots \ge h_{i,\text{rank}_i^1(l)}$$
$$h_{1j},\ldots,h_{qj} : h_{\text{rank}_j^2(1),j} \ge h_{\text{rank}_j^2(2),j} \ldots \ge h_{\text{rank}_j^2(q),j}$$

Дизъюнкция трёх РП

$$\alpha_{ij} = \begin{cases} 1, & h_{ij} \ge (h_{i,\text{rank}_i^1(1)} + \ldots + h_{i,\text{rank}_i^1(k_0)})/k_0, \\ 0, & h_{ij} < (h_{i,\text{rank}_i^1(1)} + \ldots + h_{i,\text{rank}_i^1(k_0)})/k_0, \end{cases}$$

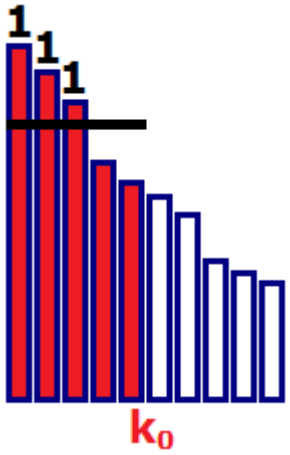
$$\alpha_{ij} = \begin{cases} 1, & h_{ij} \ge h_{i,\text{rank}_i^1(k_1)}, \\ 0, & h_{ij} < h_{i,\text{rank}_i^1(k_1)}, \end{cases}$$

$$\alpha_{ij} = \begin{cases} 1, & h_{ij} \ge h_{\text{rank}_j^2(k_2)}, \\ 0, & h_{ij} < h_{\text{rank}_j^2(k_2)}, \end{cases}$$

Что это значит?

Секрет успеха

Порог определяем по среднему первых k_0 оценок



Каждый объект принадлежит как минимум k_1 классу

В каждом классе как минимум k_2 объектов

Многоклассовая классификация текстов

Completed • $680 • 120 teams

Greek Media Monitoring Multilabel Classification (WISE 2014

Mon 2 Jun 2014 – Tue 15 Jul 2014 (2 years ago)

#	Δrank	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	Alexander D'yakonov (MSU, Moscow, Russia) *	0.79685	26	Tue, 15 Jul 2014 23:53:59 (-1h)
2	↑3	anttip	0.79482	21	Tue, 15 Jul 2014 23:49:20 (-24.9h)
3	↓2	honzas (Univ. of West Bohemia, Pilsen, Czechia)	0.79463	52	Tue, 15 Jul 2014 07:37:11
4	—	Stanislav Semenov (HSE Yandex)	0.79388	72	Tue, 15 Jul 2014 21:04:06 (-4.1d)
5	↓2	KazAnova&Rafa	0.79302	70	Tue, 15 Jul 2014 20:41:29 (-26.3h)
6	—	Yanir Seroussi	0.79176	7	Sat, 12 Jul 2014 10:48:49 (-7.6d)
7	↑1	Eleftherios Spyromitros-Xioufis	0.78271	23	Tue, 15 Jul 2014 08:02:08
8	↓1	dkay	0.78259	15	Tue, 15 Jul 2014 23:38:52 (-4d)
9	—	nagadomi	0.77897	11	Mon, 14 Jul 2014 08:46:39 (-4.9d)
10	—	David Thaler	0.77454	14	Tue, 15 Jul 2014 08:25:39 (-2.1h)

Многоклассовая классификация текстов

Медиа-статьи

Ручная каталогизация

301561 признаков

203 класса

64857 статей в обучении

34923 статей в контроле

на вход – матрица текст-слово (уже как-то нормированные)

качество = F1-мера

Методы победителей

1 место

Алгоритм	Параметр	Качество
kNN	k ∈ {1, 2, 3, 50}	0.68
ridge regression	Alpha ∈ {0.4, 0.8, 1.2}	0.76
logistic regression	L1 regularization ∈ {2, 6, 10}	0.78

Стэкинг с ridge regression

обучение = 50000 + 14857 (для стэкинга)

хитрость: обучение упорядочено, оптимальные размеры разбиения

2 место

Восстановление исходных значений матрицы документ-слово

(до tf-idf)

Пары слов, LDA (всего 3 признаковых пространства)

100 классификаторов в ансамбле

Методы победителей

3 место

Линейная модель + semi-supervised learning

4 место

Linear SVM + L1 + техника выбора порога

Решающие правила

$$\alpha_{ij} = 1 \Leftrightarrow h_{ij} \geq \min[c, \max[h_{i1}, \ldots, h_{il}]] \quad \textbf{(1)}$$

$$\Leftrightarrow h_{ij} \geq c \cdot \max[h_{i1}, \ldots, h_{il}] \quad \textbf{(2)}$$

$$\Leftrightarrow h_{ij} \geq c \cdot (h_{i1} + \ldots + h_{il}) \quad \textbf{(3)}$$

$$\Leftrightarrow h_{ij} - \bar{h} \geq c \cdot \max[h_{i1} - \bar{h}, \ldots, h_{il} - \bar{h}] \quad \textbf{(4)}$$

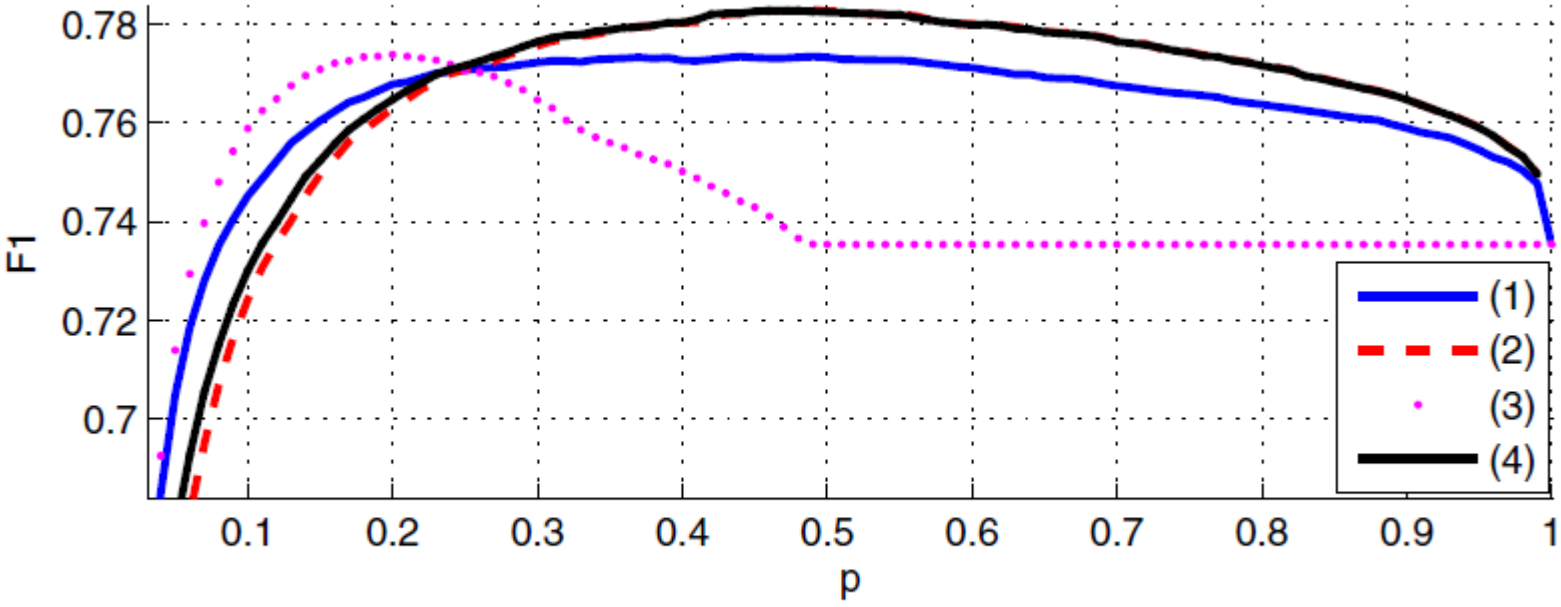


Fig. 1. Performance of decision rules

Решающие правила

1 место $\alpha_{ij} = 1 \Leftrightarrow h_{ij} \geq 0.55 \cdot \max[h_{i1}, \ldots, h_{il}]$

2 место $\alpha_{ij} = 1 \Leftrightarrow h_{ij} \geq 0.5 \cdot \max[h_{i1}, \ldots, h_{il}]$

3 место
$$\alpha_{ij} = 1 \Leftrightarrow \left[\begin{array}{c} h_{ij} \geq 0.5 \cdot \max[h_{i1}, \ldots, h_{il}] \\ j = \text{rank}_i^1(1) \\ j = \text{rank}_i^1(2), h_{ij} \geq c_2 \cdot \max[h_{i1}, \ldots, h_{il}] \\ j = \text{rank}_i^1(3), h_{ij} \geq c_3 \cdot \max[h_{i1}, \ldots, h_{il}] \\ j = \text{rank}_i^1(4), h_{ij} \geq c_4 \cdot \max[h_{i1}, \ldots, h_{il}] \end{array} \right.$$

4 место – определение оптимального порога – отдельная регрессионная задача

Реальная задача банка ТКС

```
199;022003;КИРОВСКИМ РУВД ГОР.УФЫ РЕСП БАШКОРТОСТАН;01М2005
200;782084;УВД ПУШКИНСКОГО Р-НА С-ПЕТЕРБУРГА;09М2003
201;592001;ОТДЕЛОМ ВНУТРЕННИХ ДЕЛ Г. ПЕРМЬ ДЗЕРЖИНСКОГО РАЙОНА;02М2005
202;743004;ОТДЕЛОМ МИЛИЦИИ №1 УВД Г. ЗАЛТОУСТА ЧЕЛЯБИНСКОЙ ОБЛАСТИ;09М2004
203;592026;УВД Кунгура Пермской области;12М2003
204;612071;ОВД ПРОЛЕТАРСКОГО Р-НА Г РОСТОВ-НА-ДОНУ;04М2003
```

Первый шаг –

просто набор эвристик

г.	обл.
город	об.
гор.	область
г-д	области
города	обл-ть
г-да	обл-ти

Вопрос: как их придумать?

Реальная задача: обнаружение оскорблений в форумах

```
1,20120619015242Z,"""You really think shes speaking spanish? You are a fool aren't you?""",PrivateTest
1,20120611202936Z,"""Hey race baiter take a hike you are a typical idiot""",PrivateTest
0,20120618223003Z,"""when Obamabots blame everything else for their own failures, including Bush, then the tsunami in Japan, etc.....................they don't realize that Americanvoters hate BLAME and love real leadership.\xa0 That is a losing strategy . Maybe they should fire AxelFRAUD.""",PrivateTest
0,20120619005130Z,"""Zimmerman and his wife look like brother and sister.""",PrivateTest
0,20120609203436Z,"""Is that you, Yannick??????? Give it a rest.""",PrivateTest
0,20120529172838Z,"""nah...that'd be your daughter""",PrivateTest
1,20120320115718Z,"""Maybe next time you say it, you'll write ""You're a moron"".\xa0 \n\nYou're a cretin.""",PrivateTest
1,20120620132122Z,"""you are a closet, flaming homo""",PrivateTest
1,20120619145438Z,"""go away you trashy\xa0 tramp""",PrivateTest
0,20120529144915Z,"""You and your ilk are the reason why Paul will never be elected to any position except the one he currently has. Sadly this will effect his son''s ratings.""",PrivateTest
0,20120530151416Z,"""LOL Lame old woman, mother of yellow chicken hawks.""",PrivateTest
1,20120619052459Z,"""Wow you attack others and show your ignorance. You get exactly what you deserve. You are one those typle scum bag employees who don't pull their wait and hide behind the union. You make me sick.""",PrivateTest
```

Фрагменты кода

```
% удалить всякие служебные символы + вычислить признаки
function Slist = convertS(Slist)

    % удаление того, что в кавычках
...
    S = [S ' citat'];

    S = lower(S); % сразу в нижний регистр

    % убрать все апострофы
    S = strrep(S, '''', '');
%{  S = strrep(S, 'n''t', ' not');
    S = strrep(S, '''s', ' is');
    S = strrep(S, '''m', ' am');
    S = strrep(S, '''re', ' are');
    S = strrep(S, '''ll', ' will');
    S = strrep(S, '''ve', ' have');
%}  S = strrep(S, '''d', ' should');

    inds = strfind(S,'\x'); % было - '\x80' -> ' 80 '
    if ~isempty(inds)
        S = [S ' xspecsimb'];
...
    end;

    % убрать 2 пробела подряд
```

```
    S(findstr(S, '  '))= [];

    % схлопывание слов, выдел. пробелами 'I D I O T' -> 'IDIOT'

...
    S =  delAllButOneFirst(S,'.'); %S = strrep(S, '.', ' pointsymb ');
    S =  delAllButOneFirst(S,'!'); %S = strrep(S, '!', ' vosklsymb ');
    S =  delAllButOneFirst(S,'?'); %S = strrep(S, '?', ' voprsymb ');

    S =  delAll(S,'\n');
    S =  delAll(S,'\r');

    S = strrep(S, ':))', ' smilik1 ');
    S = strrep(S, ':)',  ' smilik2 ');
    S = strrep(S, ':-)', ' smilik2 ');
    S = strrep(S, ':p',  ' smilik5 ');
    S = strrep(S, ':(',  ' smilik6 ');
    S = strrep(S, ':-(', ' smilik6 ');
    S = strrep(S, '=)',  ' smilik3 ');
    S = strrep(S, ';)',  ' smilik9 ');
    S = strrep(S, ';-)',  ' smilik9 ');
    S = strrep(S, ':d',  ' smilik4 ');

    if ~isempty(strfind(S,'jpg'))
        S = [S ' jpgspecsimb'];
    end;

    if ~isempty(strfind(S,'youtube'))
```

```
        S = [S ' youtubespecsimb'];
    end;

    if ~isempty(strfind(S,'facebook'))
        S = [S ' facebookspecsimb'];
    end;

    if ~isempty(strfind(S,'http'))||(~isempty(strfind(S,'www')))
        S = [S ' inetspecsimb'];
    end;

    % удаление html-тэгов
    S = regexprep(S, '<[^>]*>', '');

    % удалить все интернет ссылки
...

    S = strrep(S, ' ', ' ');
    S = strrep(S, '&amp', ' ');

    % aa bbb cccc -> aa b c
    Irep = find((S(1:end-2)==S(2:end-1))&(S(1:end-2)==S(3:end)));
    S([Irep+1 Irep+2])=[];

    S = regexprep(S, 's[!@#$%^&*.+-][!@#$%^&*.+-]t', ' shit ');
...
```

Сам алгоритм

```
% мой алгоритм - мои нормировки + kNN (тут ищу только соседа)
function [myans R] = MYA9082(DS, Y, ds, N)
% DS - матрица "текст-слово"
% Y - классы
% ds - рассматриваемый текст
% N - число соседей

% нормировки
W = sum(DS(Y==0,:),1);

%DS  = DS*sparse(1:size(DS,2), 1:size(DS,2), 1./(log(W+1.01)+1)); % new
DS  = sparse(1:size(DS,1), 1:size(DS,1), 1./sqrt(sum(DS.^2, 2)+0.01))*DS; % new

% нормировки объекта
ds = ds./sqrt(W+0.01); % new
ds = ds./sqrt(sum(ds.^2)+0.01); % new

% оценки близости
R = DS*ds';

[R,I] = sort(R, 'descend');

% классы N ближайших
myans = Y(I(1:N));
R = R(1:N);
```

Интересные закономерности

1. Интуиция подводит

(нельзя решать не видя данных)

2. Грустный смайлик – нет оскорбления

(обратное неверно)

Аналогично часто со ссылками

3. Ответ на оскорбление – часто оскорбление

4. Проблема цитирования

(в цитате может быть что угодно)

Topical Classification of Biomedical Research Papers

Challenges / JRS 2012 Data Mining Competition: Topical Classification of Biomedical Research Papers

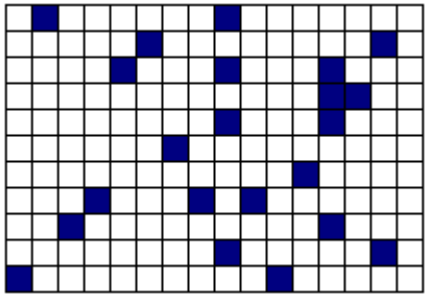
- Summary
- News
- Task
- Leaderboard
- Submit
- Register
- Forum

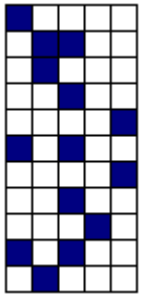
Status	Closed
Type	Scientific
Start	2012-01-02 00:00:00 CET
End	2012-03-30 23:59:59 CET
Prize	1,500$

Rank	Team	Time of Submission	Preliminary Result	Final Result
1	+ ULjubljana	Mar 31, 00:00:03	0.535	0.53579
2	+ Kebi	Mar 30, 16:54:39	0.530	0.53343
3	+ D'yakonov Alexander	Mar 31, 00:00:10	0.530	0.53242
4	+ purexa	Mar 30, 23:24:31	0.528	0.53094
5	+ asrk	Mar 15, 08:47:53	0.523	0.53032
6	+ UMoscow	Mar 30, 23:23:50	0.531	0.52939
7	+ Andrew Ostapets	Mar 30, 16:33:56	0.528	0.52885
8	+ Dmitry Kondrashkin	Mar 29, 16:43:55	0.528	0.52871
9	+ MLKD	Mar 30, 18:25:32	0.526	0.52719
10	potapenko	Mar 29, 21:10:25	0.521	0.52174

Topical Classification of Biomedical Research Papers

Написать алгоритм для автоматической классификации биомедицинских научных статей (10000 статей) на 83 класса. Каждая статья описывается 25000 признаками.

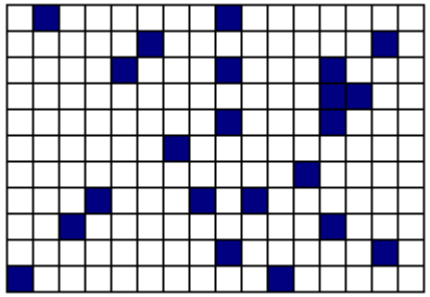


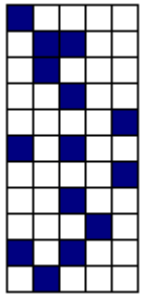


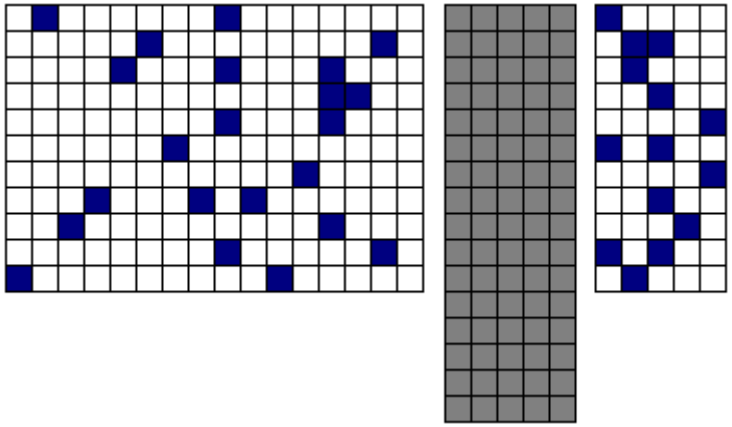
Качество – F-мера

Первый метод – решение матричного уравнения

Данные







$$X_{q\times n}\cdot W_{n\times l}=Y_{q\times l}$$

Почему нельзя решать напрямую?

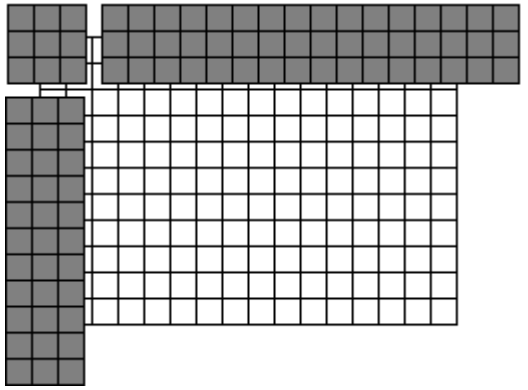
$$X_{q\times n}\cdot W_{n\times l}=Y_{q\times l}$$

$$q=10000,\ n=25000,\ l=83$$

$$X_{q\times n}\approx U_{q\times k}L_{k\times k}V_{k\times n}$$

$$U_{q\times k}\cdot W_{n\times k}=Y_{q\times l}$$

SVD-преобразование



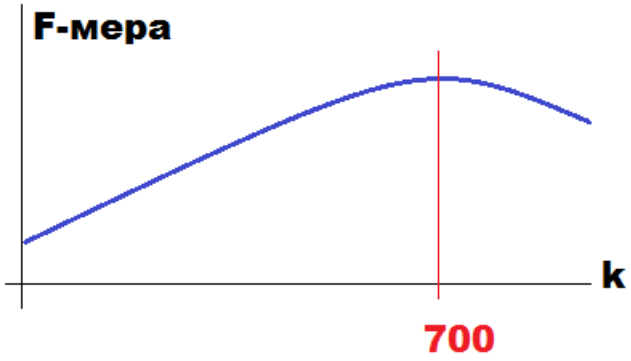
Первый метод – решение матричного уравнения

1. TF*IDF (?)
2. Нормировка

$$f_{ij} \to \frac{f_{ij}}{\sum_k f_{ik}}$$

3. SVD + решение матричного уравнения

700 сингулярных векторов!



Второй метод – весовой kNN

1. Нормировка

$$f_{ij} \to \frac{f_{ij}}{s_j}$$

$$s_i = \frac{1}{l}\sum_{j=1}^{l} \log(1+s_{ij})$$

$$\| s_{ij} \|_{l\times n} = Y^T X$$

аналог tf*idf для многоклассовых задач

tf∗idf	наша нормировка
47%	49.7%

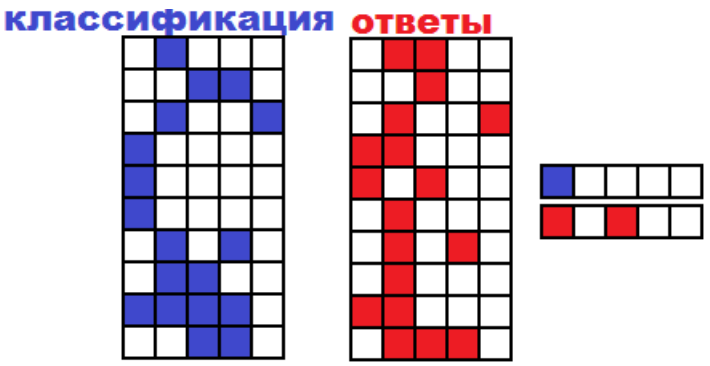
2. kNN

k=200, квадратичная весовая схема

Специфика задачи:

Функционал качества – F-мера

$$\frac{2|R \cap A|}{|R|+|A|}$$

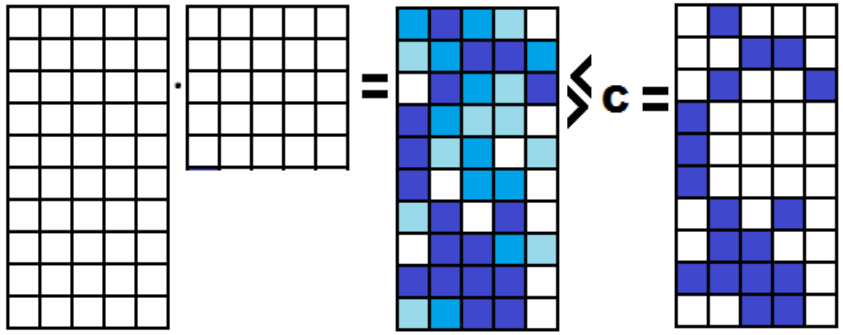



```
2*mean(sum(A.*Y,2)./(sum(A,2)+sum(B,2))
```

Финальный результат ~ 0.53

Вроде бы нельзя решать независимо (для каждого класса отдельно).

Выход: усложнённое решающее правило



т.е. «сверхлинейное» + порог

Ещё методы решения:

Пакет LIBLINEAR

(история ошибки: пакет LIBSVM)

Используется для реализации линейного SVM на больших разреженных матрицах

Чтобы получались «разные алгоритмы» для блендинга – сделать разные нормировки.

По строкам

$$f_{ij} \to \frac{f_{ij}}{\max\limits_k [\, f_{ik} \,]}$$

По столбцам

$$f_{ij} \to \frac{f_{ij}}{\max\limits_k [\, f_{kj} \,]}$$

Блендинг

После синтеза всех методов:

$$\sum_i c_i N(B_i)$$

- построение распознающего оператора

$$N(\| h_{ij} \|_{q\times l}) = \left\| \frac{h_{ij}}{\sqrt{\max[h_{i1},\ldots,h_{il}]}} \right\|_{q\times l}$$

Потом решающее правило – сравнение с порогом

Почему так… чуть позже.

Ещё примеры задач…

Ключевые слова

1. Подсвечены тегами, в заголовках, …
2. Высокая частота (но не стоп-слова)
3. Место в тексте (начало/конец, не в вопросах, …)

Устойчивые словосочетания

«Российская федерация»,
«Московский государственный университет»,
«база данных»

Понижение признакового пространства

1. Использование части слов
2. SVD (LSI – латентно-семантическое индексирование)
3. Кластеризация терминов

Ещё примеры задач...

Интересное направление:

Регрессия на текстах

- определение возраста человека по тексту (пола, образования)

- по объявлению о работе определение зарплаты

Фонетические алгоритмы

1. Пишем первую букву
2. A, E, I, O, U, H, W, Y → 0
3. B, F, P, V → 1
4. C, G, J, K, Q, S, X, Z → 2
5. D, T → 3
6. L → 4
7. M, N → 5
8. R → 6
9. Схлопываем соседние одинаковые цифры
10. Удаляем все нули, дополняем нулями в конец
11. Возвращаем букву и следующие 3 цифры

Hermann → H655

Фонетический код

Soundex

N251	Нагимов, Нагмбетов, Назимов, Насимов, Нассонов, Нежнов, Незнаев, Несмеев, Нижневский, Никонов, Никонович, Нисенблат, Нисенбаум, Ниссенбаум, Ногинов, Ножнов

Улучшенный Soundex

N8030802	Насимов, Нассонов, Никонов
N80308108	Нисенбаум, Ниссенбаум
N8040802	Нагимов, Нагонов, Неганов, Ногинов
N804810602	Нагмбетов
N8050802	Назимов, Нежнов, Ножнов

Что можно почитать

К.Д. Маннинг,
П. Рагхаван,
Х. Шютце
«Введение в информационный поиск»

Очень хорошая книга для ознакомления с областью обработки текста.