# Автоматическая рубрикация текстов

Text categorization

Manning и др. Введение в информационный поиск
гл. 13, 14

# Рубрикация текстов

- Классификация/рубрикация информации – отнесение порции информации к одной или нескольким категориям из конечного множества рубрик

- Применение:
  - Навигация по коллекции документов
  - Поиск информации
  - Замена сложного запроса
  - Иерархическое упорядочение знаний предметной области
    - Анализ распределения документов по тематике
  - Фильтрация потока текстов:
    - Тематический сбор новостей
    - Персонализированная фильтрация потока текстов
    - Фильтрация спама
    - Тематический сбор информации из интернет

# Примеры рубрикаторов

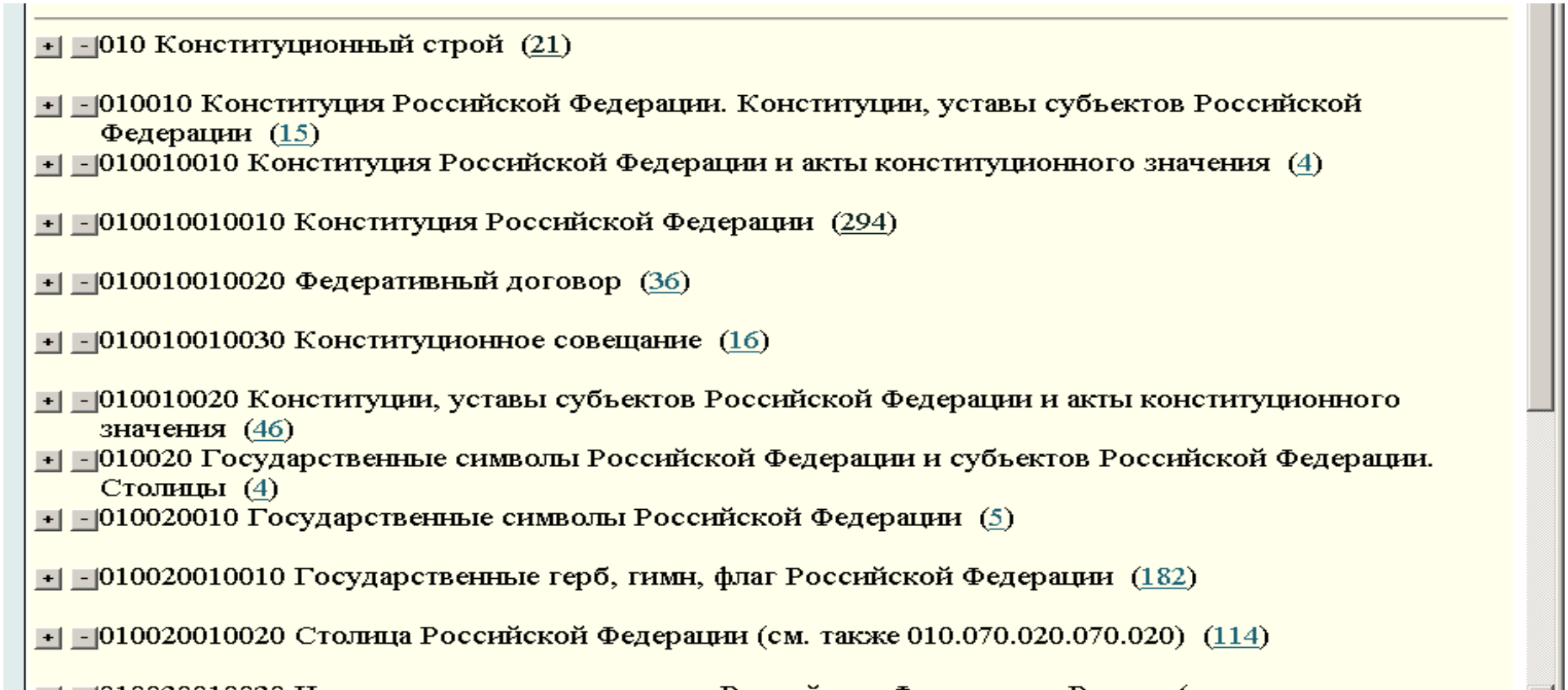
- Каталог Интернет-сайтов:
  Open Directory Project – dmoz.org
  - 4,830,584 sites, 75,151 editors, over 590,000 categories
  - C

# Каталог Яндекс – Фасетная классификация

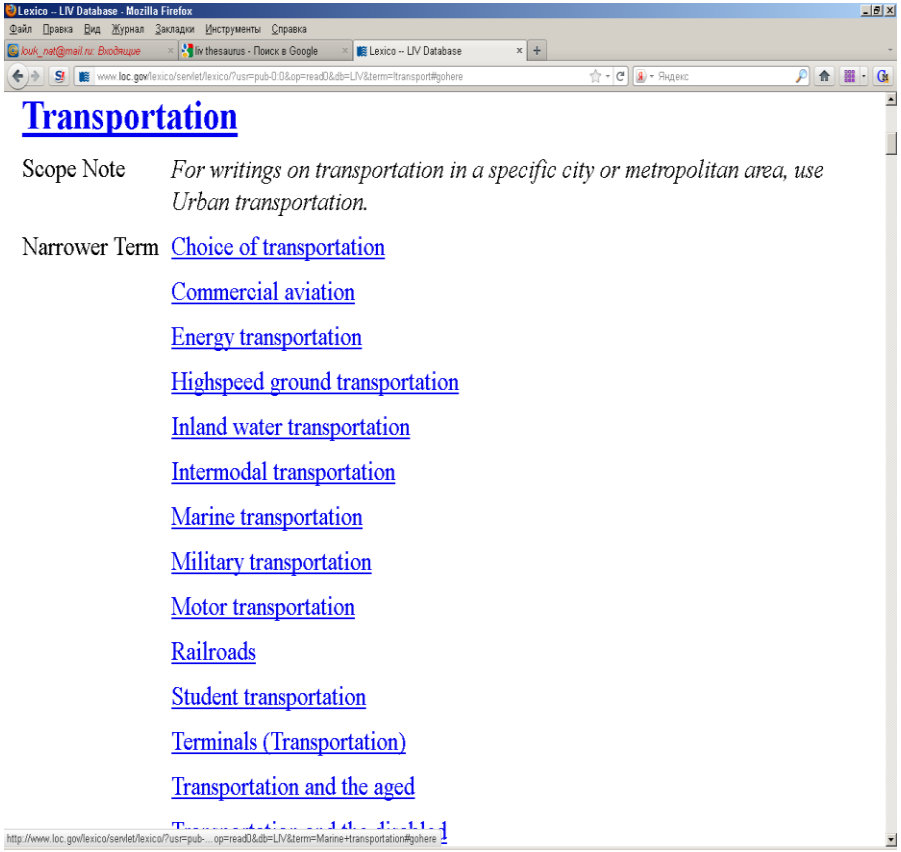
- **Тематическая**
  - Иерархический классификатор, имеет порядка 600 значений и описывает предметную область интернет-ресурса
- **Регион**
  - 230 географических областей. Определяется географическим расположением представляемого объекта, сферой управления и влияния, потенциальной аудиторией информации или информационным содержанием ресурса
- **Жанр**
  - художественная литература; научно-техническая литература; научно-популярная литература; нормативные документы; советы; публицистика
- **Источник информации**
  - Официальный, СМИ, Неформальный, Персональный Анонимный
- **Адресат информации**
  - Партнеры, Инвесторы, Потребители, Коллеги
- **Сектор экономики**
  - Государственный, Коммерческий, Некоммерческий

# Рубрикатор нормативно-правовых актов

- Президентский классификатор (Указ №511 15.03.2000)
- Иерархия рубрик - 1168 рубрик
- Все НПА рубрицируются экспертами в обязательном



# В прошлый раз: тезаурусы




# Отличие рубрикаторов от тезаурусов

- Термины тезауруса являются фундаментально языковыми, в то время как рубрики соответствуют концептуальным категориям **(Bates 1988).**

- Тезаурус
  - Подробное описание предметной области
  - Термины предметной области: синонимы, отношения
  - Отбор декрипторов для описания документов

- Рубрикатор
  - Разделение предметной области сверху
  - Цель: разработать совершенно отдельные концептуальные категории, которые взаимно не пересекаются.
  - Идеально не должно быть пересечений между рубриками и не должно быть промежутков, то есть ни одна подобласть не должна остаться вне рубрик рубрикатора.

# Методы рубрицирования текстов

- Ручное рубрицирование
- Автоматическое
  - Инженерный подход (=методы, основанные на знаниях, экспертные методы)
  - Методы машинного обучения
- Полуавтоматическое

# Точность *P* и полнота *R*

| | in the class | not in the class |
|---|---|---|
| predicted to be in the class | true positives (TP) | false positives (FP) |
| predicted to not be in the class | false negatives (FN) | true negatives (TN) |

$$P = TP / ( TP + FP)$$

$$R = TP / ( TP + FN)$$



9

# Комбинированная мера: *F*

- $$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P + R}$$

$$\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$



10

# Усреднение: Micro vs. Macro

- Посчитали меру оценки ($F_1$) for one class.
- Как агрегировать оценки F1 для многих классов.
- Macroaveraging - макроусреднение
  - Посчитать $F_1$ для каждого из $C$ классов
  - Среднее арифметическое для этих $C$ чисел
- Microaveraging - микроусреднение
  - Посчитать TP, FP, FN для каждого из $C$ classes
  - Суммировать эти C чисел для каждого показателя
  - Посчитать $F_1$ для суммированных TP, FP, FN



11

# Коллекция и рубрикатор Reuters для автоматического рубрицирования

- Более 21 тысячи информационных сообщений из области биржевой торговли и слияния предприятий
- Массив разделен на две части: документы для обучения, документы для тестирования
- Большинство текстов имеют рубрики, проставленные людьми
- Основные рубрики: 135 без иерархии
- Примеры рубрик: Золото (товар), Свинец (товар), Кофе и др. товары, Торговля
- Средняя длина текста - 133 слова

# Ручное рубрицирование

- Высокая точность рубрицирования
  - Обычно процент документов, в которых проставлена явно неправильная рубрика, чрезвычайно мал (если работают специалисты)
- Низкая скорость обработки документов
- Используется:
  - Парламентские службы,
  - Looksmart, about.com, ODP, PubMed
  - Библиотеки (УДК)

# Автоматическая рубрикация: Инженерный подход

- Основное предположение: рубрикатор создается осмысленно, содержание рубрики можно выразить ограниченным количеством понятий в виде формулы
- Эксперты описывают смысл рубрики в виде булевских выражений, правил продукции
- Construe system (Hayes)
  - Reuter news story
  - 674 рубрики: 135 тематических рубрик + география…
  - 4 человеко-года
  - 94 % полноты и 84 % точности на 723 текстах

# Reuters: пример описания рубрики

```
if      (wheat & farm) or
        (wheat & commodity) or
        (bushels & export) or
        (wheat & tonnes) or
        (wheat & winter and (¬ soft))
then
        WHEAT
else
   (not WHEAT)
```

# Коммерческая система Verity: описание правила

```
comment line            # Beginning of art topic definition
top-level topic         art ACCRUE
                            /author = "fsmith"
topic definition modifiers  /date   = "30-Dec-01"
                            /annotation = "Topic created
                                          by fsmith"
subtopic topic          * 0.70 performing-arts ACCRUE
  evidence topic        ** 0.50 WORD
  topic definition modifier     /wordtext = ballet
  evidence topic        ** 0.50 STEM
  topic definition modifier     /wordtext = dance
  evidence topic        ** 0.50 WORD
  topic definition modifier     /wordtext = opera
  evidence topic        ** 0.30 WORD
  topic definition modifier     /wordtext = symphony
subtopic                * 0.70 visual-arts ACCRUE
                        ** 0.50 WORD
                            /wordtext = painting
                        ** 0.50 WORD
                            /wordtext = sculpture
subtopic                * 0.70 film ACCRUE
                        ** 0.50 STEM
                            /wordtext = film
subtopic                ** 0.50 motion-picture PHRASE
                        *** 1.00 WORD
                            /wordtext = motion
                        *** 1.00 WORD
                            /wordtext = picture
                        ** 0.50 STEM
                            /wordtext = movie
subtopic                * 0.50 video ACCRUE
                        ** 0.50 STEM
                            /wordtext = video
                        ** 0.50 STEM
                            /wordtext = vcr
                        # End of art topic
```

- Note:
  - maintenance issues (author, etc.)
  - Hand-weighting of terms

[Verity was bought by Autonomy.]

# Автоматическая рубрикация: Методы машинного обучения

- Имеется коллекция отрубрицированных людьми текстов.

- Для каждой рубрики имеется множество положительных и отрицательных примеров

# Методы машинного для задачи автоматической рубрикации

- Метод Байеса (Naive Bayes )
- Метод Roccio
- Метод ближайшего соседа (k-Nearest Neighbors – knn)

- Метод опорных векторов (Support-vector machines – SVM)

- !!Должно быть размечено много данных.

# Метод Байеса

# Релевантность документов рубрике

- Relevance feedback
- Пользователь (эксперт) – размечает документы, относящиеся к рубрике
  - Эта разметка производится оффлайн.
  - Может быть результатом ручной рубрикации
- Определение лучшего класса для документа
  - P (C|$d_i$)
  - Метод Байеса

# Правило Байеса для классификации текстов

- Для документа *d* и класса *c*

$$P(c,d) = P(c \mid d)P(d) = P(d \mid c)P(c)$$

$$\boxed{P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}}$$

# Байесовский классификатор (Naive Bayes)



Задача: Классифицировать новый документ *d на основе совокупности признаков в один из классов* $c_j \in C$

$$d = (\langle x_1, x_2, \ldots, x_n \rangle)$$

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\ \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\ P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

MAP is “maximum a posteriori” = наиболее вероятный класс

# Предположение классификатора Naive Bayes

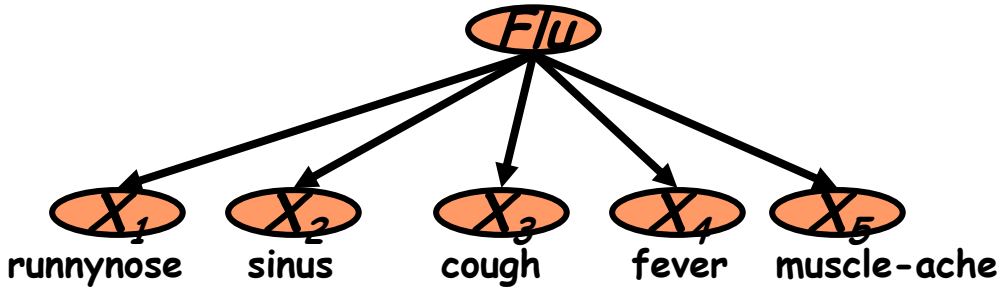
- $P(c_j)$
  - Может быть оценено из частотности классов в обучающих примерах

- $P(x_1,x_2,\ldots,x_n|c_j)$
  - O($|X|^n \bullet |C|$) параметры
  - Может быть оценено, но нужно очень-очень много обучающих примеров

Naive Bayes Предположение о независимости:

- Предположим, что вероятность наблюдения конъюнкции атрибутов равна произведению индивидуальных вероятностей $P(x_i|c_j)$.

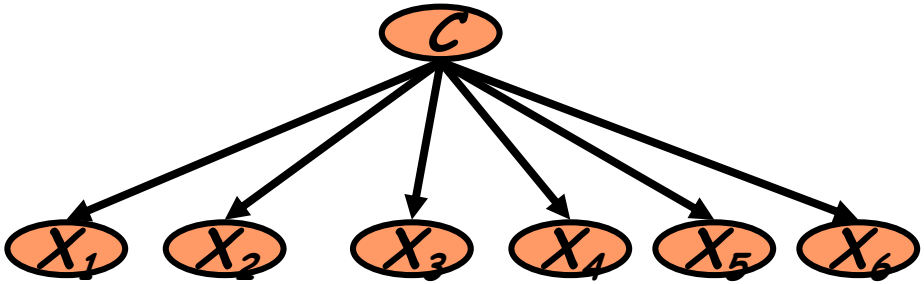
# Классификатор Naive Bayes




- **Предположение о Conditional Independence Assumption:** признаки присутствия термов независимы друг от друга в заданном классе кдлассификатора:

$$P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

# Обучение модели

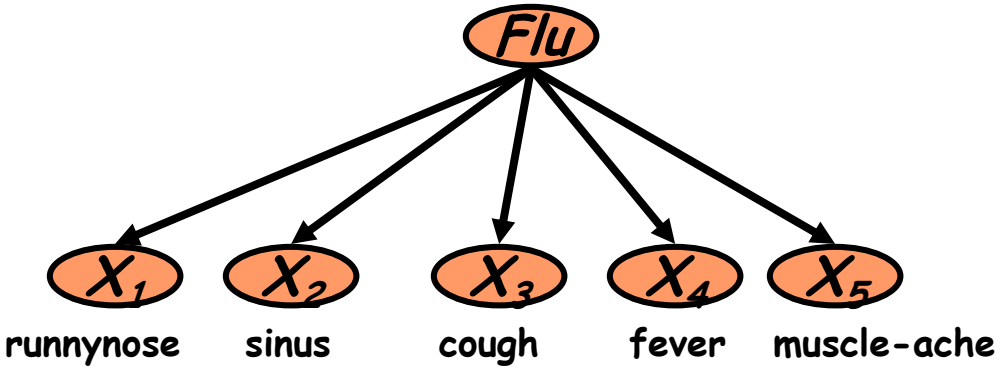



- Первая попытка: MLE maximum likelihood estimates
  - Используем просто частоты в обучающих примерах

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

# Проблемы с Maximum Likelihood




$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

- Предположим, что у нас не было обучающих документов со словами ***muscle-ache*** и классифицированных в тему Грипп?

$$\hat{P}(X_5 = t \mid C = Flu) = \frac{N(X_5 = t, C = Flu)}{N(C = Flu)} = 0$$

# Сглаживание: правило Лапласа

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

# of values of $X_i$

- Более «мягкая» версия

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + m}{N(C = c_j) + mK}$$

# Базовый подход к классификации текстов на основе метода Байеса

- Атрибуты - позиции в тексте, значения - слова

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)\prod_i P(x_i \mid c_j)$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)P(x_1 = \text{"our"} \mid c_j)\cdots P(x_n = \text{"text"} \mid c_j)$$

- Предположим еще, что классификация не зависит от позиции слов
  - Используем те же параметры в любой позиции
  - Результат – модель мешка слов

# Наивный Байес: Обучение

- Из обучающего корпуса извлекаем *Vocabulary*
- Вычисляем $P(c_j)$ and $P(x_k \mid c_j)$
  - Для каждого $c_j$ in $C$ do
    - $docs_j \leftarrow$ *множество документов, в которых проставлен класс* - $c_j$
    - $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
    - $Text_j \leftarrow$ документ, содержащий все I $docs_j$
    - Для каждого слова $x_k$ in *Vocabulary*
      - $n_k \leftarrow$ number of occurrences of $x_k$ in $Text_j$
      - $$P(x_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

# Наивный Байес: Классификация

- Return $c_{NB}$, where

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \; P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

# Вычисление: переход к логарифмам

- Перемножение большого числа вероятностей от 0 до1 может привести к проблемам типа floating-point underflow.
- Так как log(*xy*) = log(*x*) + log(*y*), лучше выполнять все операции, суммируя логарифмы
- Класс с максимальным логарифмом – является наиболее вероятным

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} [\log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)]$$

- Заметим, что полученная модель классификации – это просто сумма весов

frequency of WTO occurring in UK documents.

TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC($V, d$)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] += \log condprob[t][c]$
6 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

**Figure 13.2:** Naive Bayes algorithm (multinomial model): Training and testing

# Naive Bayes vs. Другие методы

(a)

| | NB | Rocchio | kNN | SVM |
|---|---|---|---|---|
| micro-avg-L (90 classes) | 80 | 85 | 86 | 89 |
| macro-avg (90 classes) | 47 | 59 | 60 | 60 |

(b)

| | NB | Rocchio | kNN | trees | SVM |
|---|---|---|---|---|---|
| earn | 96 | 93 | 97 | 98 | 98 |
| acq | 88 | 65 | 92 | 90 | 94 |
| money-fx | 57 | 47 | 78 | 66 | 75 |
| grain | 79 | 68 | 82 | 85 | 95 |
| crude | 80 | 70 | 86 | 85 | 89 |
| trade | 64 | 65 | 77 | 73 | 76 |
| interest | 65 | 63 | 74 | 67 | 78 |
| ship | 85 | 49 | 79 | 74 | 86 |
| wheat | 70 | 69 | 77 | 93 | 92 |
| corn | 65 | 48 | 78 | 92 | 90 |
| micro-avg (top 10) | 82 | 65 | 82 | 88 | 92 |
| micro-avg-D (118 classes) | 75 | 62 | n/a | n/a | 87 |

Evaluation measure: $F_1$

Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

# Преимущества и недостатки Байесовского подхода к классификации

- Преимущества
  - Легко реализовать (нет оптимизации, матриц и т.п.)
  - Эффективный для обучения и использования
  - Легко обновить при появлении новых данных
  - Предположение о независимости позволяет оценивать параметры на разных данных
  - Оценить признаки
- Недостатки
  - Предположение о независимости признаков часто неверно
  - Во многих задачах классификации текстов - не самый лучший метод

# Метод Байеса в распознавании почтового спама

- Применяется во многих системах распознавания спама
  - «убийца спама»
    - Classic Naive Bayes superior when appropriately used

# Почтовый спам

- **Спам** - рассылка коммерческой и иной рекламы или иных видов сообщений лицам, не выражавшим желания их получать
  - Лаборатория Касперского: доля спама в почтовом трафике в июле 2013 года составила 71%
- Методы (http://www.securelist.com/en/threats/spam?chapter=97)
  - Черные списки сайтов
  - Bulk email – массовая рассылка..
  - Фильтрация по контенту
    - Ручные правила
    - Метод Байеса

# Извлечение признаков

- Заголовок: получатель, отправитель, доменные имена
- Текст
  - Слова, фразы, строки символов
  - Могут быть бинарными или числовыми
- URL, HTML tags, картинки

Случайно порожденное имя и адрес

From: Sam Elegy <aj6xfdou7@yahoo.com>
To: ddlewis4@att.net
Subject: you can buy V!@gra

Типограф. варианты


Спам-контент в форме картинки

Irrelevant legit content; doubles as hash buster

# SpamAssassin признаки:

- Basic (Naïve) Bayes spam probability
- Mentions: Generic Viagra
- Regex: millions of (dollar) ((dollar) NN,NNN,NNN.NN)
- Phrase: impress ... girl
- Phrase: ‘Prestigious Non-Accredited Universities’
- From: starts with many numbers
- Subject is all capitals
- HTML has a low ratio of text to image area
- Relay in RBL, http://www.mail-abuse.com/enduserinfo_rbl.html
- RCVD line looks faked
- http://spamassassin.apache.org/tests_3_3_x.html

# Байес для распознавания спама

- Два класса: спам – неспам
- Вычисление вероятностей
  - Вероятности классов: доля каждого класса в обучающей выборке
  - *Вероятности признаков* $P(x_k \mid c_j)$: – *количество вхождений в классе/число признаков в классе*
- Могут быть варианты формул, связанные с тем, что пропустить спам менее опасно, чем отправить нормальное письмо в спам
- Используется во многих системах фильтрации и в настоящее время
  - Возможно, что предположение о независимости признаков, больше соответствует действительности

# Классификаторы на основе пространства векторов

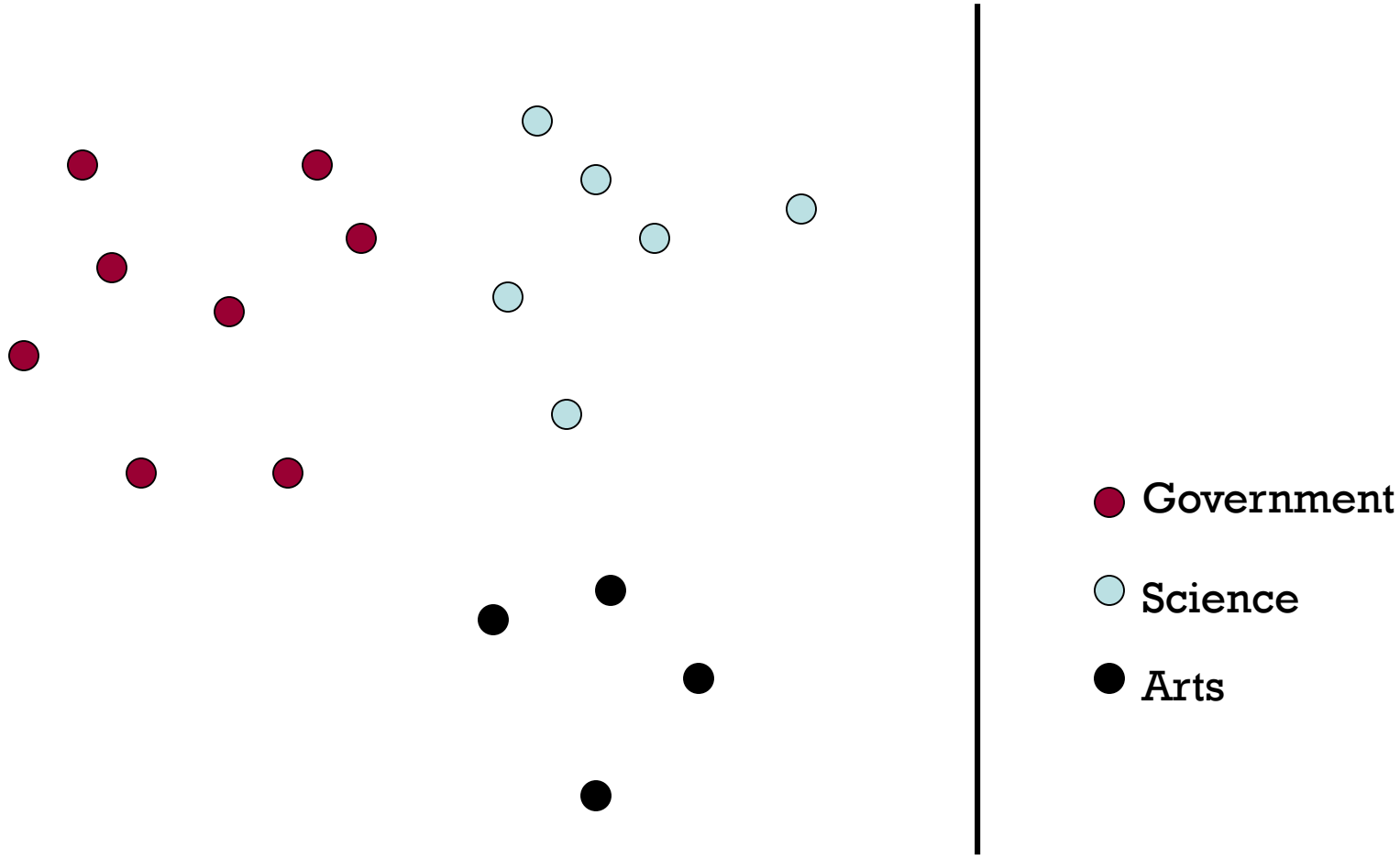
# Векторная модель

- Преобразование множества текстов в векторы пространства $R^n$
  - Пословная модель – bag of words
  - Удаление стоп-слов (предлоги, союзы…), которые заданы списком
  - Приведение к нормальной морфологической форме (stemming, лемматизация – приведение к словарной форме)
  - Определение весов слов: tf.idf
  - Построение вектора слов документа

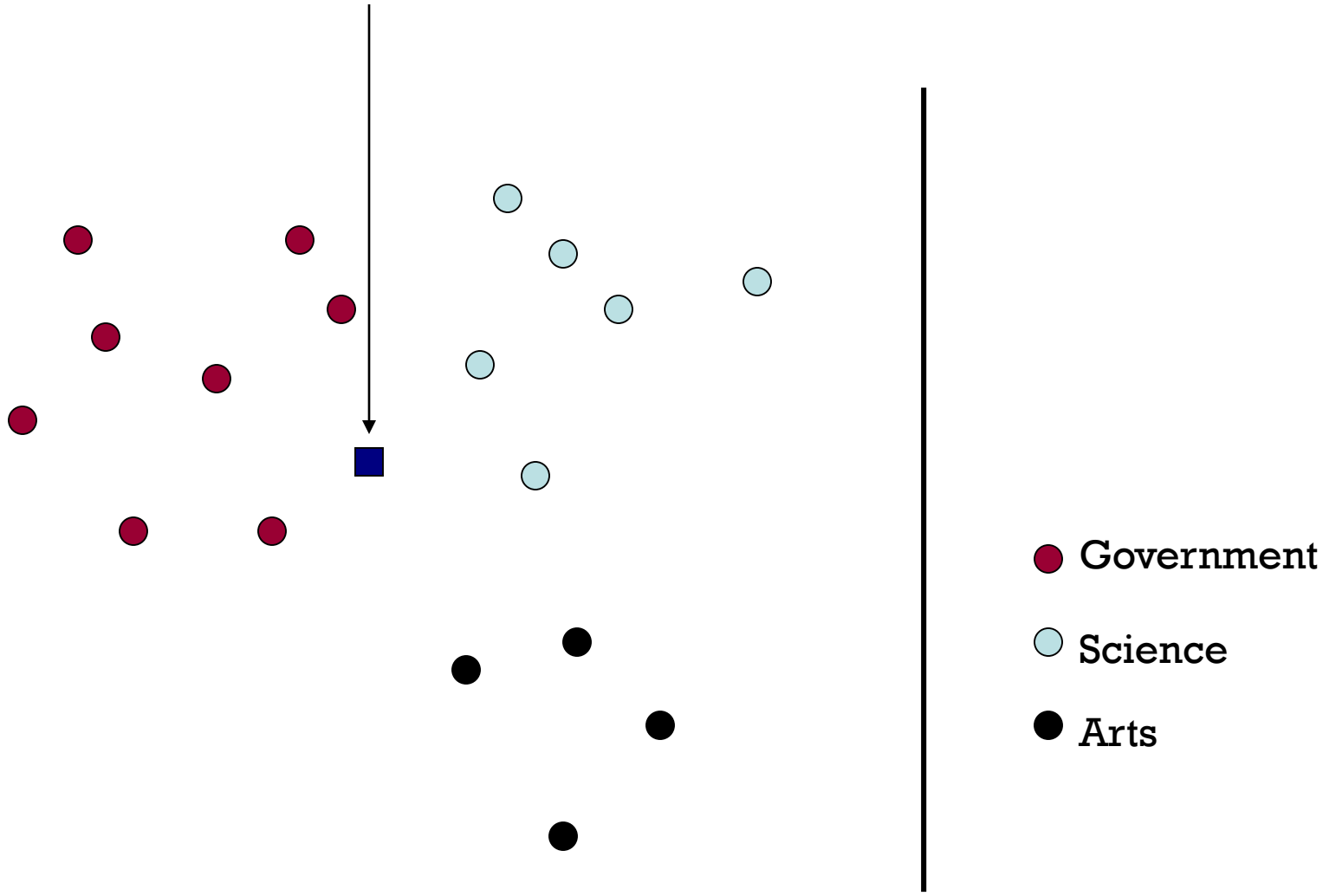
# Классификация на основе пространства векторов

- Документы – вектора, точки в векторном пространстве
- Предположения:
  - Документы одного класса находятся в одной области пространства
  - Документы из разных классов находятся в непересекающихся областях
  - Таким образом: нужно найти разделяющую поверхность

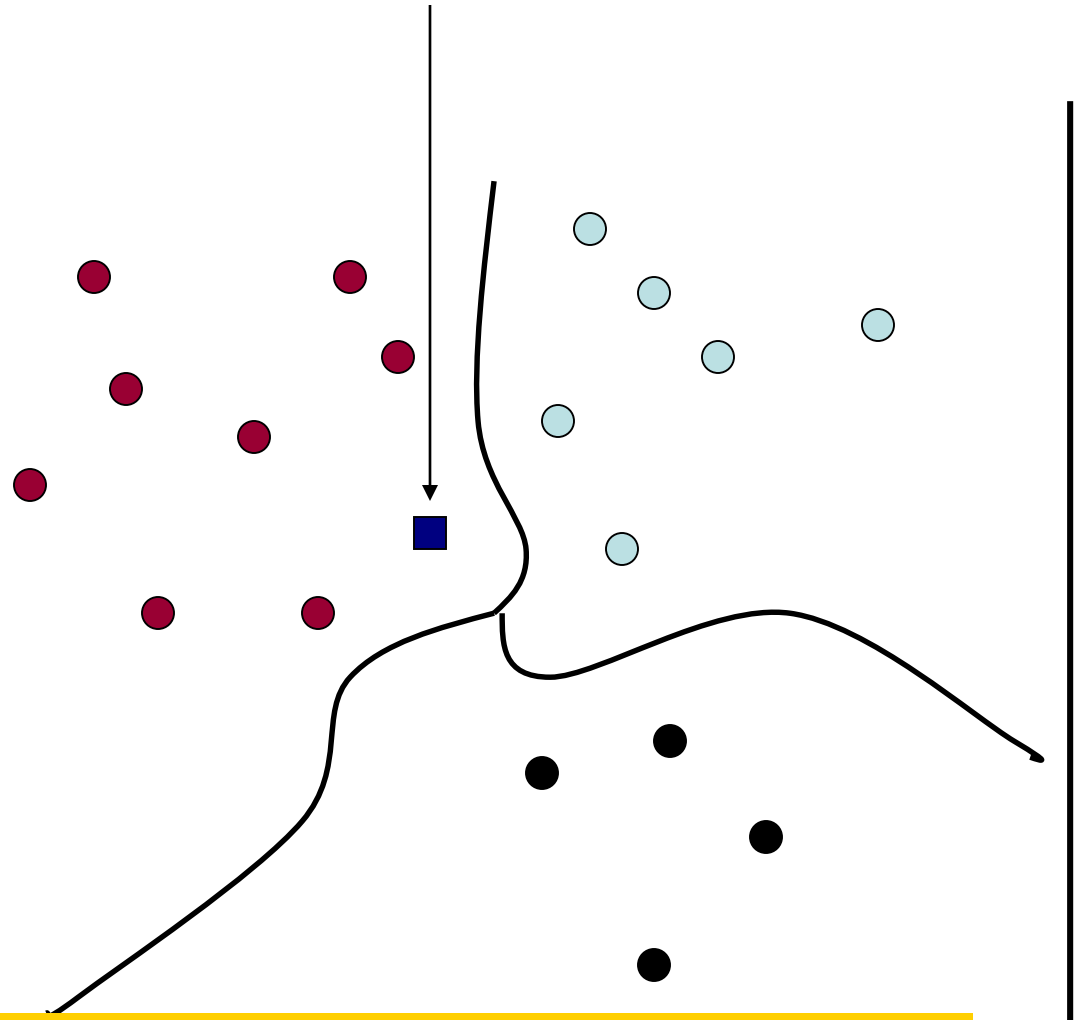
# Документы в векторном пространстве



# Документ относится к какому классу?



# Тема документа - Правительство



Верна ли гипотеза

Government

Science

Arts

Как найти хорошие разделяющие поверхности?

# Метод Rocchio в автоматической рубрикации

Manning et al.

Introduction to information retrieval

Гл. 14

# Rocchio 1971 алгоритм (SMART)

- На практике используется:

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

- $D_r$ = множество известных релевантных doc векторов
- $D_{nr}$ = множество известных нерелевантных doc векторов
  - Отличны от $C_r$ и $C_{nr}$


- $q_m$ = модифицированный вектор запроса; $q_0$ = исходный вектор запроса; $\alpha,\beta,\gamma$: веса
- Новый запрос «сдвигается» по направлению к релевантным документам и «уходит» от нерелевантных документов

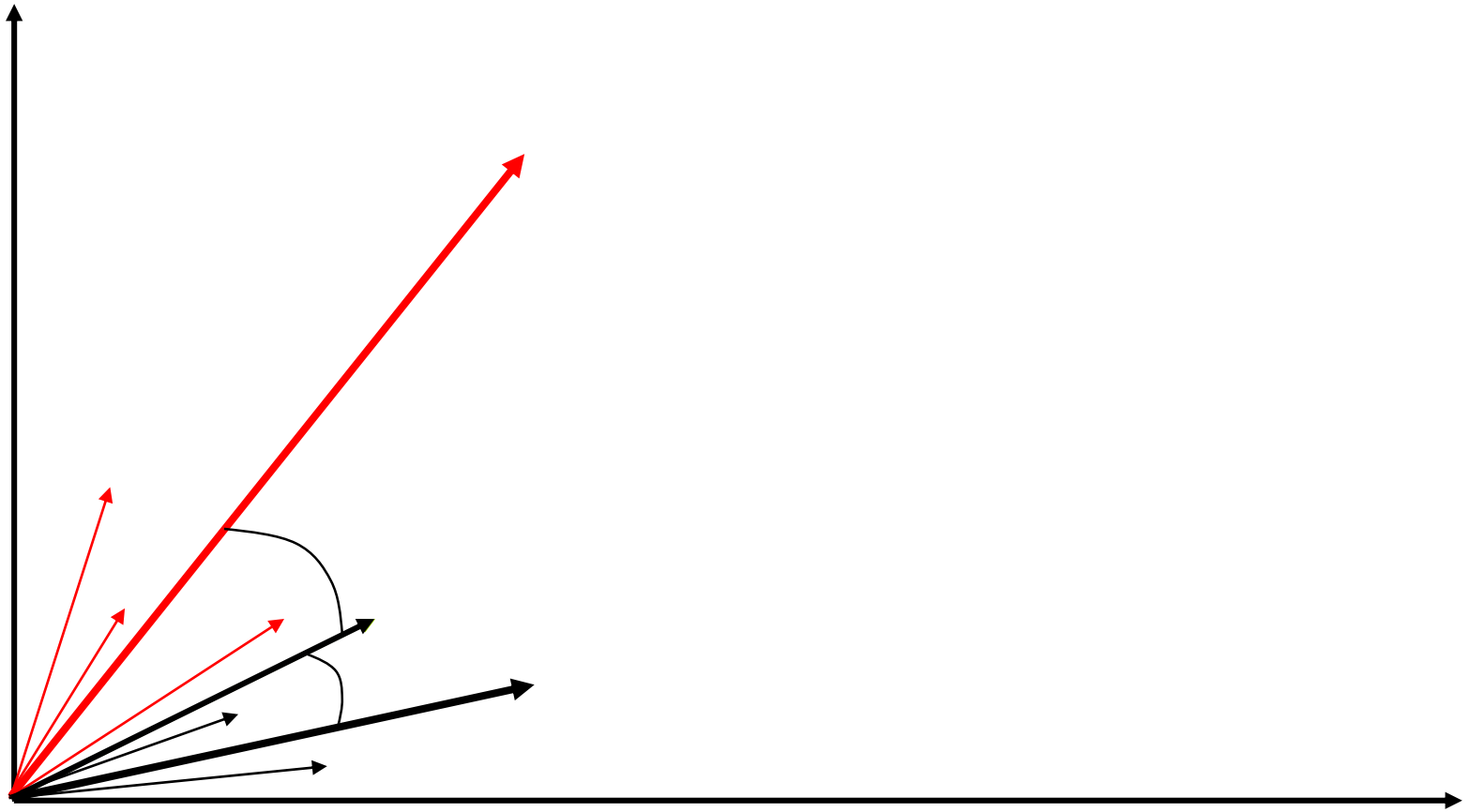
# Использование Rocchio для классификации текстов

- Для документов в каждой категории вычисляем вектор-прототип: суммируем вектора всех примеров документов в категории
  - Прототип = центроид документов категории

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

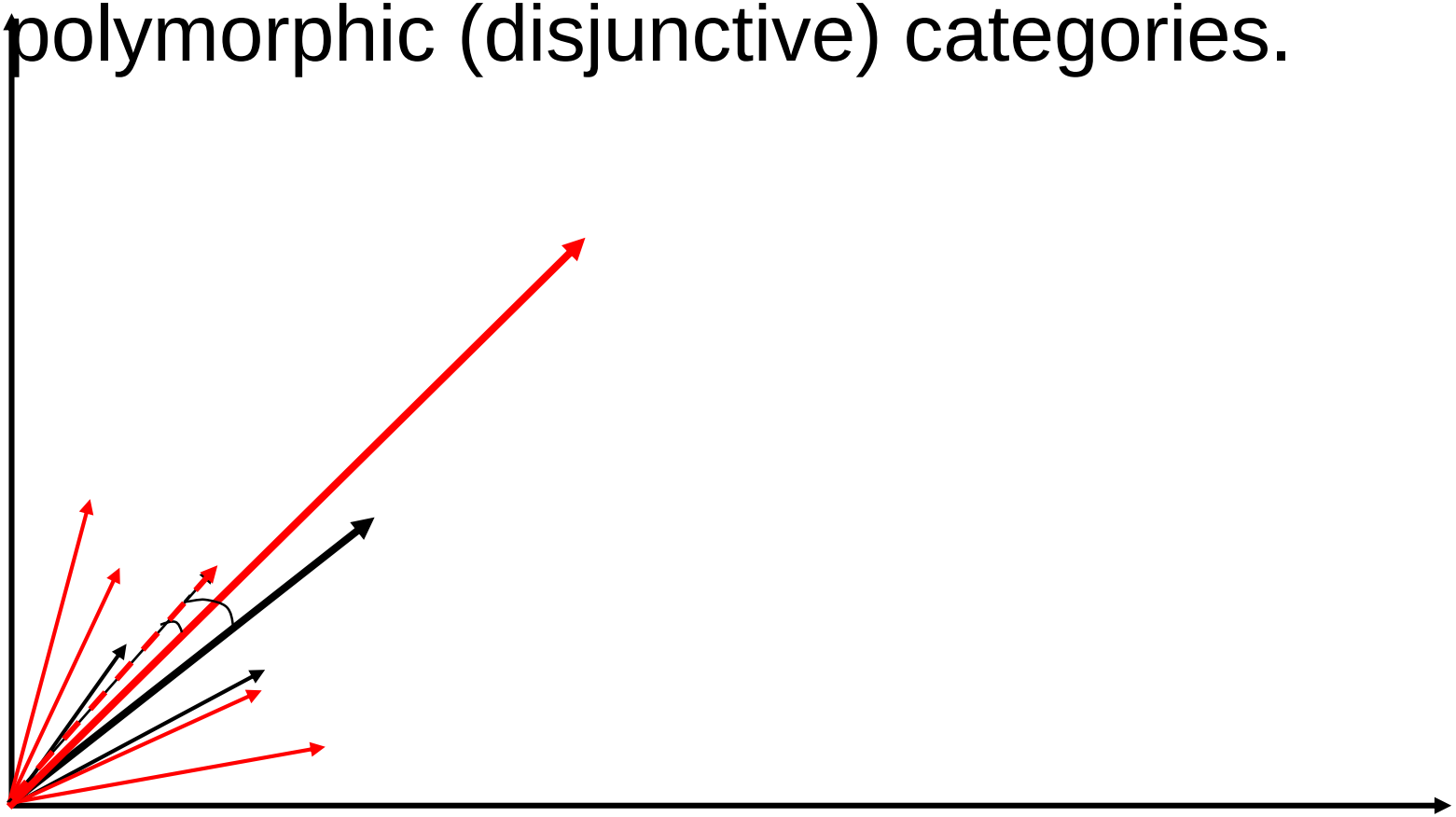
- где $D_c$ – множество документов, отнесенных к категории C, $v(d)$ - векторное представление документа
- Присваиваем тестовым документам категорию ближайшего по косинусной мере вектора-прототипа

# Иллюстрация метода Rocchio



# Аномалия метода Rocchio

- Prototype models have problems with polymorphic (disjunctive) categories.



# Свойства метода Rocchio

- Формирует простое обобщение примеров в данном классе (прототип).
- Вектор прототипа не нужно нормализовывать по длине, поскольку косинусная близость нечувствительна к длине вектора
- Классификация основана на сходстве с векторами-прототипами
- Не гарантируется, что классификации буду хорошо соответствовать обучабщим данным
- Мало используется вне текстовой классификации
- - но может быть вполне эффективным при классификации текстов

- Дешевый метод для обучения и тестирования классификации

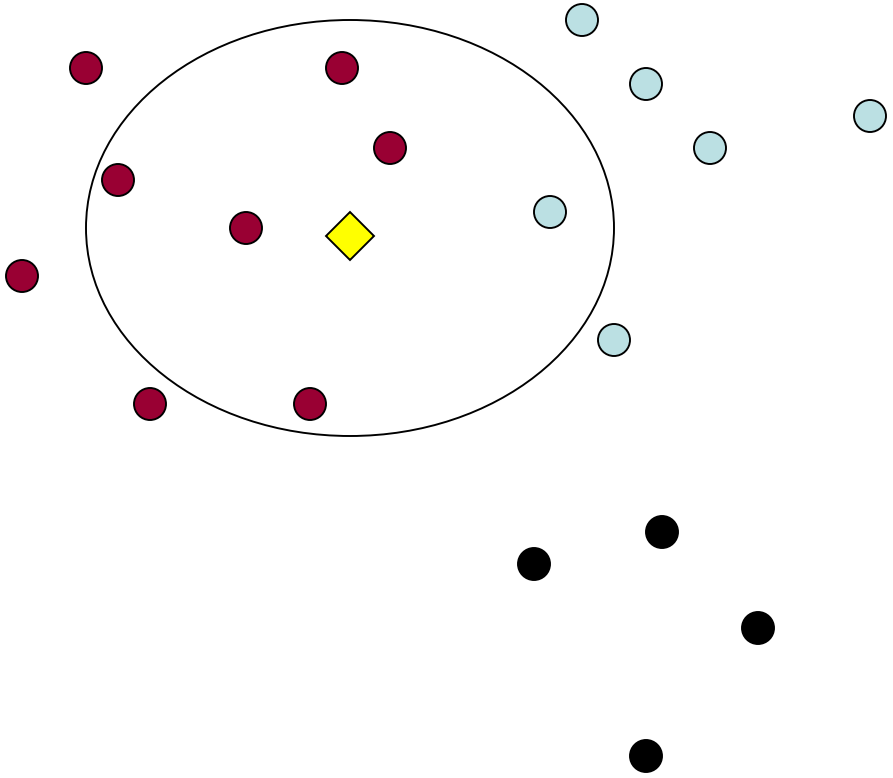
# Метод ближайших соседей (KNN)

# Метод k ближайших соседей

- kNN = k Nearest Neighbor

- Чтобы классифицировать документ в класс c:
- определяем *k*-ближайших соседей документа *d*
- Для каждого класса C вычисляем количество документов i среди соседей, которые принадлежат C
- Оцениваем P(c|*d*) as i/k
- Выбираем класс: $\text{argmax}_c$ P(c|*d*)  [ = majority class]

# Пример: k=6 (6NN)



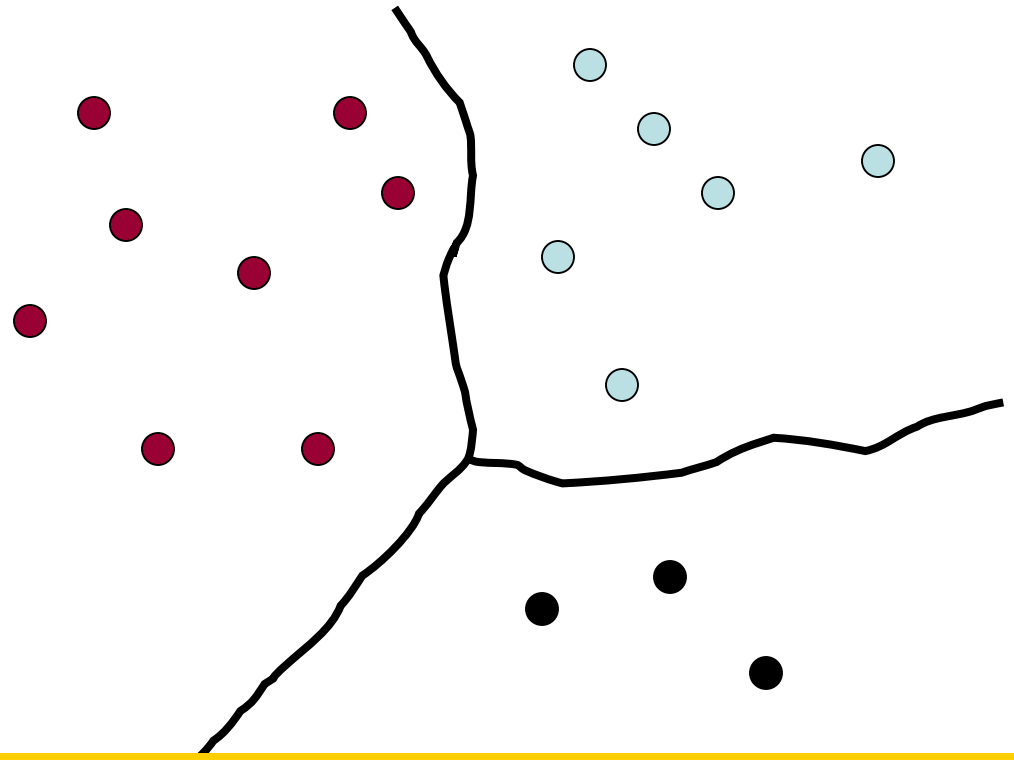
P(science|◆)?

- Government
- Science
- Arts

# Алгоритм: k ближайших соседей

- Использование только одного ближайшего соседа (1NN) ведет к ошибкам из-за:
  - нетипичных примеров
  - ошибок в ручной привязке единственного обучающего примера.
- Более устойчивой альтернативой является *k* наиболее похожих примеров и определение большинства
- Величина *k* is типично нечетная: 3, 5 (наиболее распространенные величины)

# kNN границы классов



Boundaries are in principle arbitrary surfaces – but usually polyhedra

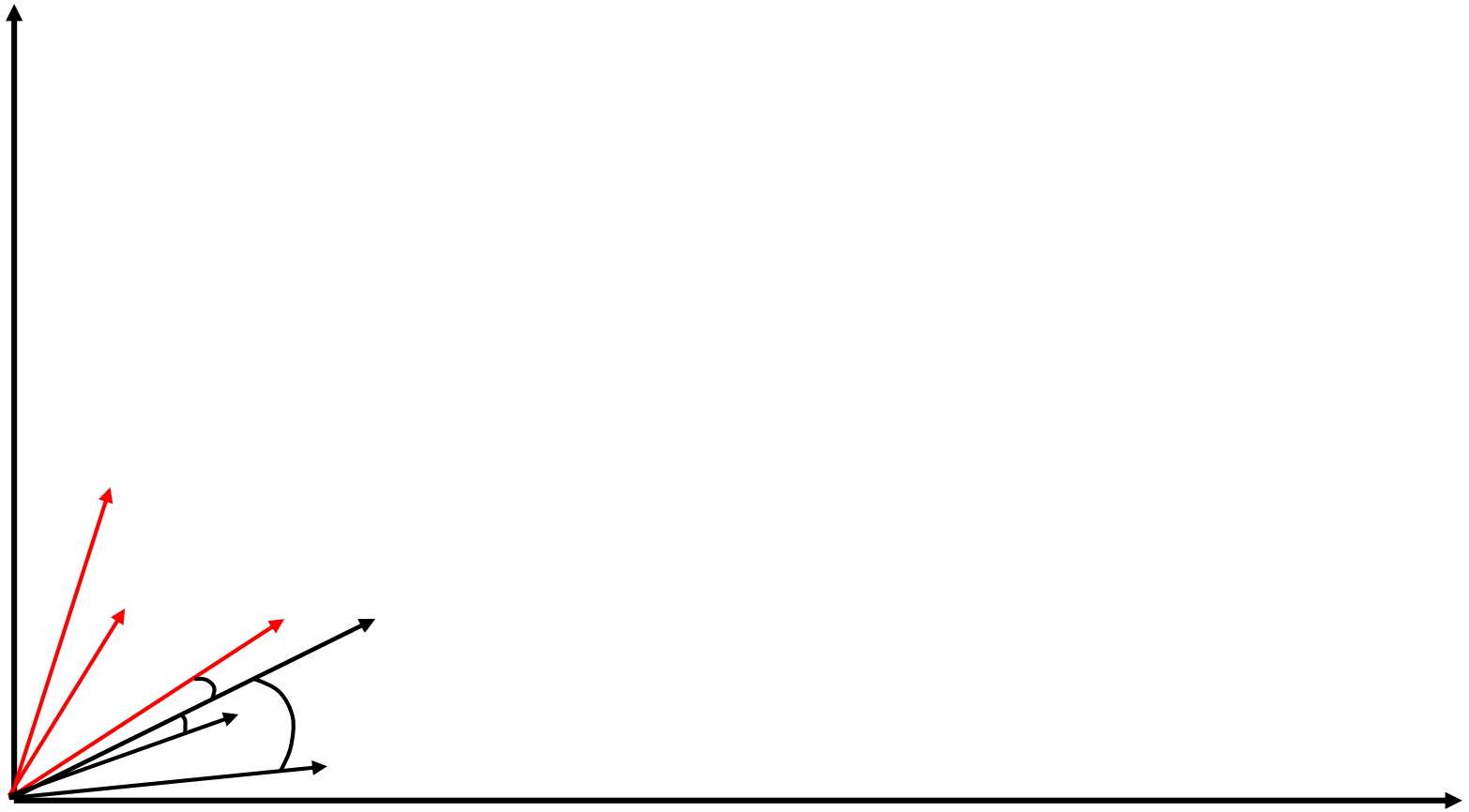
Government

Science

Arts

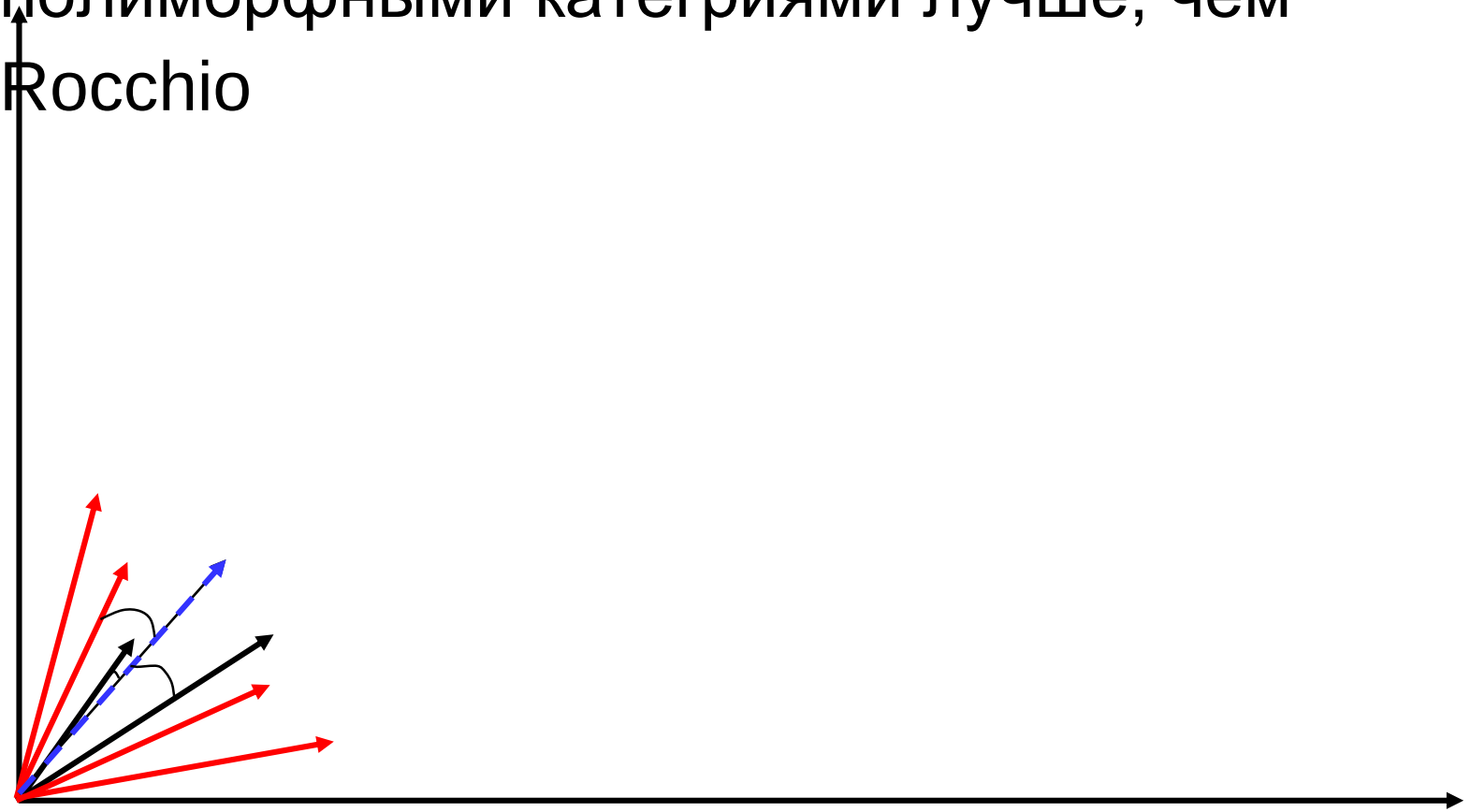
kNN gives locally defined decision boundaries between classes – far away points do not influence each classification decision (unlike in Naïve Bayes, Rocchio, etc.)

# Иллюстрация 3NN для текста в векторном пространстве



# 3 NN vs. Rocchio

- Ближайшие соседи справляются с полиморфными категриями лучше, чем Rocchio



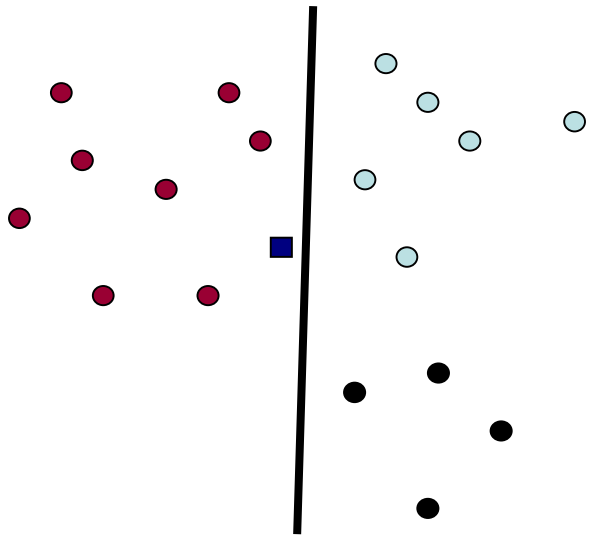
# Линейные классификаторы. Классификатор SVM

# Линейные классификаторы

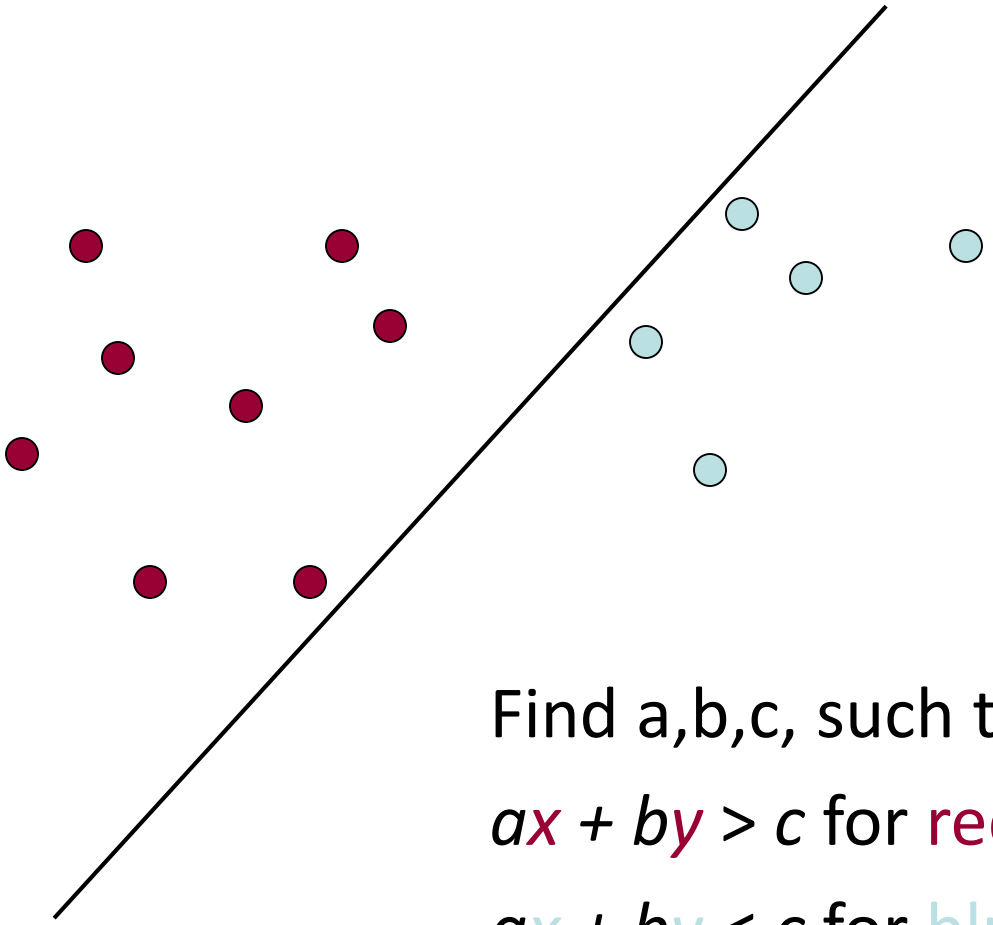
- Проблема разделения документов на 2 класса
  - например, government and non-government
  - one-versus-rest классификация
- Как правильно определить разделяющую поверхность
- Как решить, к какой области относится тестовый документ?

# Разделение гиперплоскостями

- Сильное предположение – линейная разделимость (*linear separability)*:
  - в двух измерения – линия
  - В больших измерениях – гиперплоскость
  - Сепаратор может быть выражен как $ax + by = c$



# Задача линейного программирования



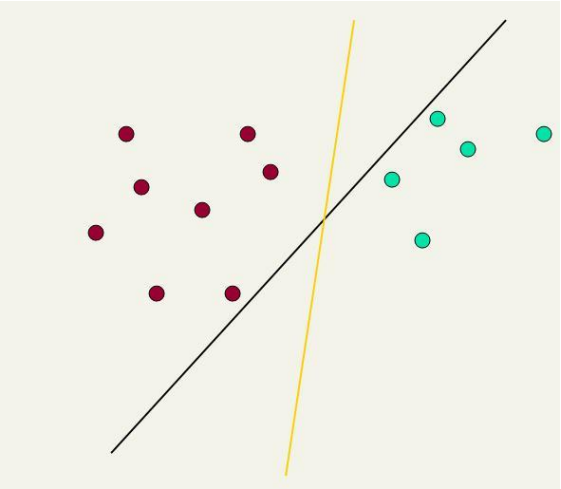
Find a,b,c, such that

$ax + by > c$ for red points

$ax + by < c$ for blue points.

# Какая гиперплоскость?

- Какие точки влияют на оптимальность
  - Все точки
    - Линейная регрессия
    - Байесовский классификаторNaïve Bayes
  - Только «трудные» точки, близкие к краям класса
    - Метод опорных векторов



# Линейный классификатор: Пример

- Класс: “interest rate” (Процентная ставка)
- Веса слов линейного классификатора
- $w_i$ $t_i$ $w_i$ $t_i$

| | |
|---|---|
| 0.70 prime | −0.71 dlrs |
| 0.67 rate | −0.35 world |
| 0.63 interest | −0.33 sees |
| 0.60 rates | −0.25 year |
| 0.46 discount | −0.24 group |
| 0.43 bundesbank | −0.24 dlr |

- Чтобы классифицировать в тексте, нужно вектор документа и найти скалярное произведение с вектором классификатора (wi)

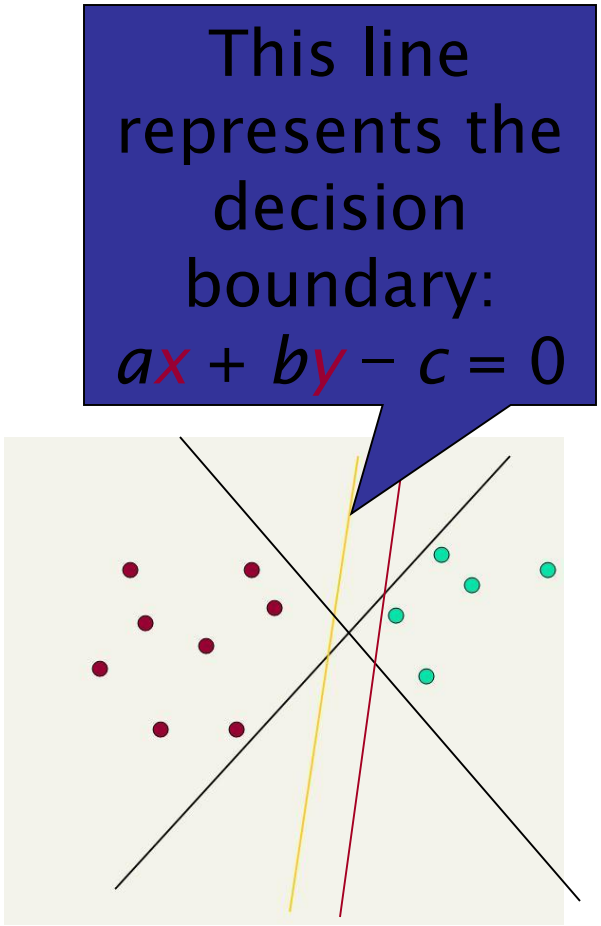
# Линейные классификаторы

- Многие известные классификаторы – линейные классификаторы:
  - Байесовский классификатор
  - Персепторн
  - Роккио
  - Линейная регрессия
  - Логистическая регрессия
  - Метод опорных векторов (с линейным ядром)
- Вопреки сходству имеются значительные различия
  - Можно построить бесконечное число разделяющтх гиперплоскостей. Какую выбрать?
  - Различные методы обучения выбирают различные гиперповерхности
  - Классификаторы, которые более мощные, чем линейные, не всегда работают лучше. Почему?

# Классификатор SVM

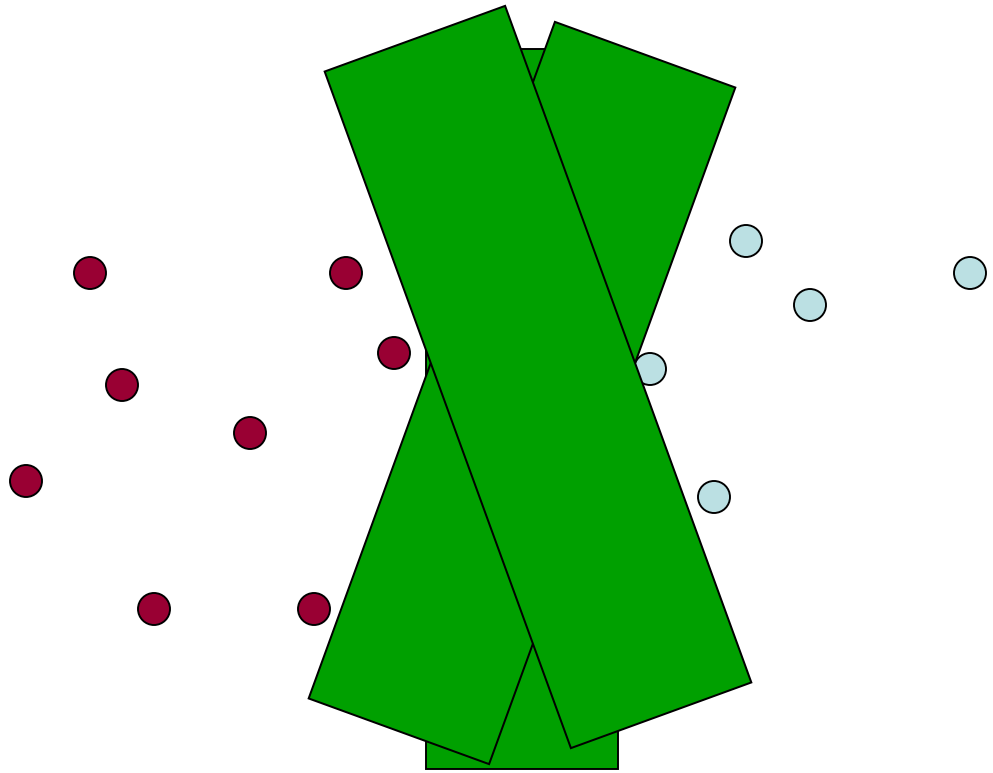
# Линейные классификаторы: Какая гиперплоскость?

- Множество возможностей для *a, b, c.*
- Некоторые методы ищут разделяющую гиперплоскость, но не оптимально
- Метод опорных векторов (SVM) находит оптимальное решение
  - Максимизирует расстояние между гиперплоскостью и трудными точками, близкими к границе раздела
  - Интуитивно: если нет точек около границы раздела, то нет и сложных (неопределенных) примеров



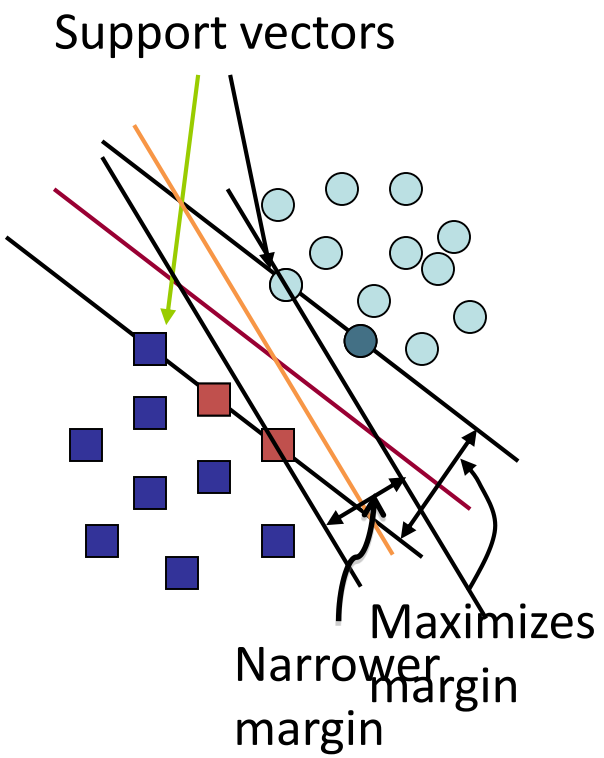
# Другая интуиция

- С «толстым сепаратором» меньше вариантов поворота



# Метод опорных векторов (SVM)

- SVM максимизирует зазор между классами
- Решение полностью определяется набором примером, называемых опорными векторами
- Решение – задача квадратичного программирования
- В настоящее время, считается лучшим методом автоматической классификации текстов



# Метод опорных векторов

- Предполагается, что точки имеют вид
- {(x1,c1),… (xn,cn)}, где где *ci* принимает значение 1 или −1, в зависимости от того, какому классу принадлежит точка
- Каждое xi - это p-мерный вещественный вектор, обычно нормализованный значениями [0,1] или [-1,1].
- Нужно найти разделяющую гиперплоскость, которая имеет вид:
- w*x-b=0, где w – перпендикуляр к разделяющей гиперплоскости

# Метод опорных векторов

- Интересует максимальное разделение, т.е. две параллельные гиперплоскости. Можно показать, что они могут быть описаны равенствами
- w*x-b=1
- w*x-b=-1
- Нужно максимизировать расстояние между плоскостями, которое d=2/|w|, т.е. минимизировать |w|
- В итоге задача квадратичной оптимизации:
- |w|-> min, при условиях
- w*x-b>>1, ci=1
- w*x-b<<-1,ci=-1

Имеются обобщения на случаи невыпуклых множеств, нелинейной отделимости

# Результаты на коллекции Reuters

(a)

| | NB | Rocchio | kNN | SVM |
|---|---|---|---|---|
| micro-avg-L (90 classes) | 80 | 85 | 86 | 89 |
| macro-avg (90 classes) | 47 | 59 | 60 | 60 |

(b)

| | NB | Rocchio | kNN | trees | SVM |
|---|---|---|---|---|---|
| earn | 96 | 93 | 97 | 98 | 98 |
| acq | 88 | 65 | 92 | 90 | 94 |
| money-fx | 57 | 47 | 78 | 66 | 75 |
| grain | 79 | 68 | 82 | 85 | 95 |
| crude | 80 | 70 | 86 | 85 | 89 |
| trade | 64 | 65 | 77 | 73 | 76 |
| interest | 65 | 63 | 74 | 67 | 78 |
| ship | 85 | 49 | 79 | 74 | 86 |
| wheat | 70 | 69 | 77 | 93 | 92 |
| corn | 65 | 48 | 78 | 92 | 90 |
| micro-avg (top 10) | 82 | 65 | 82 | 88 | 92 |
| micro-avg-D (118 classes) | 75 | 62 | n/a | n/a | 87 |

Evaluation measure: $F_1$