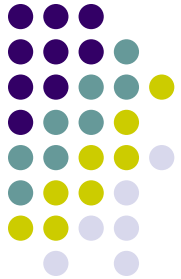
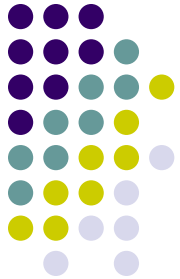


ПРИЗНАКОВАЯ МОДЕЛЬ ТЕКСТА КЛАССИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ ТЕКСТОВ

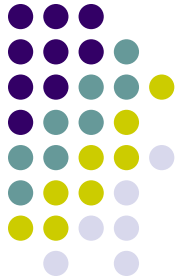
Большакова Елена Игоревна

СОДЕРЖАНИЕ



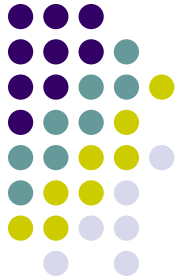
1. Признаковая модель текста:
   - Типы признаков, модель «мешок слов»
   - Веса признаков, показатель *TF-IDF*
2. Признаковая модель в задачах классификации
   - Постановка задачи
   - Методы классификации текстов коллекции
   - Оценки качества классификации, приложения
3. Признаковая модель в задачах кластеризации
   - Постановка задачи, методы
   - Алгоритм k-средних
   - Иерархическая кластеризация
   - Оценки качества кластеризации, приложения
4. Заключение и Домашнее задание № 2

ПРИЗНАКОВАЯ МОДЕЛЬ ТЕКСТА



- Для решения многих прикладных задач КЛ не нужна модель всего языка, достаточно
  модели обрабатываемого текста
  в рамках коллекции (набора) текстов
- Модель текста – абстрактное представление его содержания и формы, свободное от несущественных деталей
- Модель позволяет сравнивать тексты друг с другом и единообразно обрабатывать их коллекции (наборы)
- Такие модели появились в 60-70 гг. в связи с задачами информационного поиска
- Наиболее распространена и практически значима *признаковая модель*:
  текст – неупорядоченный набор (множество)
  информационных признаков (*features*)

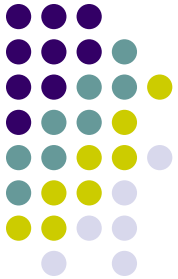
ТИПЫ ПРИЗНАКОВ



Информационные признаки текста

- Лексические
  - слова (*terms*) : обычно значимые, не служебные, реже – словосочетания
  - N-граммы (*шинглы*)
- Статистические признаки текста, в том числе учитывающие лингвистическую информацию:
  - доля различных частей речи в тексте
  - доля сложных предложений
  - средняя длина слова/предложения и т.д.
- Экстралингвистические признаки:
  - тип документа, автор, заголовок
  - дата публикации, источник информации
  - гиперссылки и пр.

ПРИЗНАКОВАЯ МОДЕЛЬ КОЛЛЕКЦИИ ТЕКСТОВ



Пусть имеется N текстов

Каждый текст (документ) – набор признаков:

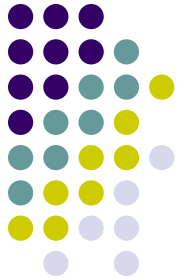
$$d_j = (w_{j1}, \ldots, w_{jm}) \text{ , где}$$

w_{ji} – ***вес*** i-ого признака в j-ом тексте d_j

m – число учитываемых признаков в текстах

- Не учитываются связи признаков
- Вес может отображать только наличие или отсутствие признака ($w_{ji} = 0$ или 1)
- Текст можно рассматривать как вектор в m-мерном пространстве, а коллекцию – как набор векторов

ПРИЗНАКОВАЯ МОДЕЛЬ: «МЕШОК СЛОВ»

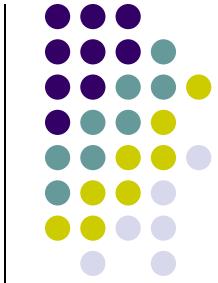


Очень частый, но частный случай признаковой модели:

«мешок слов» (*bag-of-words, BOW*),
термин употребил впервые Z.Harris, 1954 г.

- Признаки – слова текста (*terms*)
- Модель описывает (грубо) содержание текста
- Обычно не учитываются:
  - грамматические формы слов
  - порядок слов в тексте
  - синтаксические связи слов
- Вес признака: чаще всего – частота употребления слова в тексте

«МЕШОК СЛОВ»: ПРИМЕР



1: *Карл у Клары украл кораллы*
2: *Клара у Карла украла кларнет*
3: *Клара у Карла украла кораллы*
4: *Мал золотник, да дорог*

Слова-признаки (леммы):

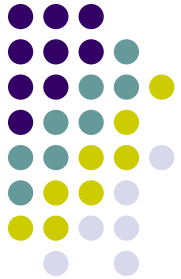
[1]*Карл*, [2]*Клара*, [3]*украсть*, [4]*коралл*,
[5]*кларнет*, [6]*малый*, [7]*золотник*, [8]*дорогой*

Вес: присутствует признак в документе или нет

$d_1=(1,1,1,1,0,0,0,0)$ $d_2=(1,1,1,0,1,0,0,0)$

$d_3=(1,1,1,1,0,0,0,0)$ $d_4=(0,0,0,0,0,1,1,1)$

ЗАДАНИЕ ВЕСОВ ПРИЗНАКОВ



В общем случае:

вес ***i***-ого признака в ***j***-ом тексте задается следующим образом:

$$w_{ji} = l_{ji}\, g_i\, n_j \;,$$ где

l_{ji} – локальный вес признака в тексте

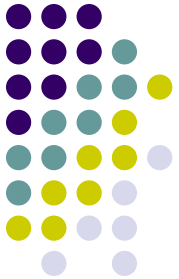
g_i – глобальный вес признака во всей коллекции текстов

n_j – нормирующий множитель для текста

Основой для задания весов обычно служит

f_{ji} – частота (абсолютная или относительная) встречаемости i-ого признака в j-ом тексте

ЗАДАНИЕ ВЕСОВ ПРИЗНАКОВ: ВАРИАНТЫ



Пусть: $l_{ij} = f_{ji}$

- Простая частота признака:

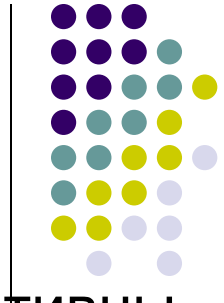
$$w_{ji} = f_{ji} \quad , \text{ а } \quad n_j = 1, \; g_i = 1$$

- Простой косинус (нормализация по длине):

$$w_{ji} = \frac{f_{ji}}{\sqrt{\sum_{i=1}^{m} (f_{ji})^2}} \qquad n_j = \frac{1}{\sqrt{\sum_{i=1}^{m} (l_{ji} g_i)^2}}$$

- Показатель *TF-IDF* (*tf-idf*)

ПОКАЗАТЕЛЬ *IDF*



Показатель *TF-IDF* опирается на предположение, что в коллекции частотные термины менее информативны, чем редкие, поэтому веса частотных признаков должны быть ниже, чем веса редких. Для этого вводятся:

- $df_i = N_i$ (*document frequency*) – число текстов, где есть i-ый признак (показатель его распространенности в коллекции, подокументная частотность)

Редкость признака в коллекции: нужна обратная величина

- idf_i (*inverse document frequency*) – оценка редкости i-го признака (N – число текстов в коллекции)

$$idf_i = \log(\frac{N}{df_i})$$

Логарифм служит для сглаживания больших величин

ПОКАЗАТЕЛЬ *TF-IDF*

Если $l_{ij} = f_{ji} = tf_{ji}$ (*term frequency*)

$g_j = idf_i$, $idf_i = \log(\frac{N}{df_i})$

$n_j = 1$

то $w_{ji} = tf_{ji} idf_i = f_{ji} \log(\frac{N}{N_i})$

Величина *tf-idf* возрастает

- с увеличением числа вхождений термина в документ
- со снижением частоты термина во всей коллекции

- *tf-idf* иногда нормализуют

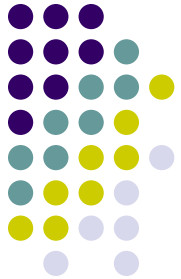
TF-IDF: ПРИМЕР

1: *Мама мыла мылом раму*
2: *Мама мыла, мыла окно*
3: *В магазине мама купила мыло*

	мама	мыть	мыло	рама	окно	магазин	купить
df_i	3	2	2	1	1	1	1
idf_i	0	0,18	0,18	0,47	0,47	0,47	0,47

Документ 1			Документ 2			Документ 3		
слово	tf_{ji}	TF-IDF	слово	tf_{ji}	TF-IDF	слово	tf_{ji}	TF-IDF
рама	1	0,47	окно	1	0,47	магазин	1	0,47
мыло	1	0,18	мыть	2	0,36	купить	1	0,47
мыть	1	0,18	мама	1	0	мыло	1	0,18
мама	1	0				мама	1	0

ДОСТОИНСТВА И НЕДОСТАТКИ ПРИЗНАКОВОЙ МОДЕЛИ



\+ Простота модели

\+ Для векторов удобно вычислять меру близости (например, косинусную меру)

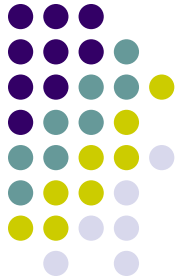
– В векторах много нулевых весов, т.к. в одном тексте редко встречаются сразу все признаки

– Много малоинформативных признаков – тех, которые встречаются только в одном-двух или сразу во всех текстах

Необходимо уменьшать количество признаков

❖ Признаковая модель используется в задачах <u>классификации и кластеризации документов</u>

СОКРАЩЕНИЕ КОЛИЧЕСТВА ПРИЗНАКОВ



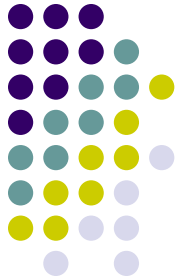
Уменьшение приводит к

- упрощению процедур, повышению их надежности и устойчивости
- обозримости пространства признаков, возможности его содержательного анализа

Используемые методы:

- Методы селекции: из исходного множества признаков отбираются наиболее полезные (*feature engineering ?*)
- Методы трансформации: строятся новые признаки, являющиеся комбинацией исходных, например: признаки, заменяющие группы синонимичных слов

КЛАССИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ ТЕКСТОВЫХ ДОКУМЕНТОВ



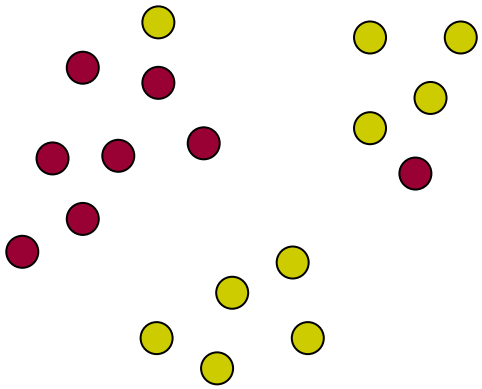
Классификация (рубрицирование) – отнесение документа к заранее известным классам (рубрикам)

- классы: с заданными характеристиками, возможна иерархическая система классов
- часто текстовые документы классифицируют по их содержанию/теме

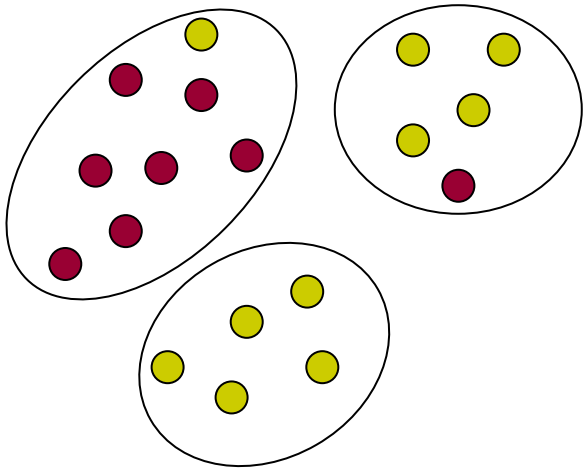
Кластеризация – разбиение заданного множества документов на подмножества (кластеры)

- характеристики, количество, структура кластеров заранее не заданы
- документы в кластерах похожи по смыслу/стилю/теме/структуре/…

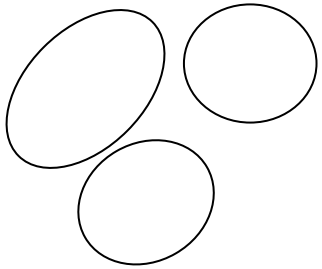
КЛАССИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ: ОТЛИЧИЕ



Классификация: классы изначально предопределены



Кластеризация: кластеры не предопределены, ищем однородные группы

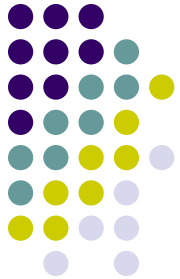


ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ

- Ручной
- Полуавтоматический: написание экспертом правил «если …, то …» и их автоматическое применение
- Автоматический: машинное обучение

Подход	Достоинства	Недостатки
Ручной	высокая точность	- дорого - медленно
Полу-автоматический	приемлемая точность	трудоемкость создания и актуализации правил
Автоматический	простота применения	- не всегда приемлемое качество - для классификации: ручная разметка обучающей выборки

АВТОМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ ПОСТАНОВКА ЗАДАЧИ



- Имеется множество классов/рубрик/категорий

$$C = \{c_1, \ldots, c_{|C|}\}$$

- Имеется множество документов

$$D = \{d_1, \ldots, d_{|D|}\}$$

- Есть неизвестная целевая функция

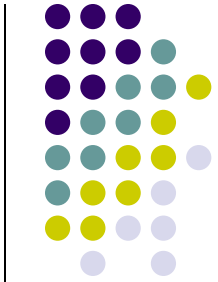
$$\Phi: D \times C \rightarrow \{0,1\}$$

- Необходимо построить классификатор Φ', максимально близкий к Φ

$$\Phi': D \times C \rightarrow \{0,1\}$$ – точный ответ

или $$\Phi': D \times C \rightarrow [0,1]$$ – степень подобия

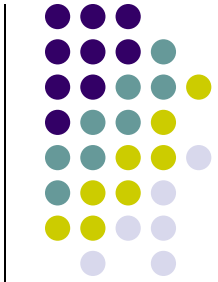
КЛАССИФИКАЦИЯ: ОБУЧАЮЩИЕ ДАННЫЕ



- Обучение с учителем *(supervised)*:
  Имеется множество D' вручную размеченных документов, для которых значения Φ известны
- При этом:
  - либо документы отклассифицированы
  - либо для каждой рубрики есть множество положительных и отрицательных примеров
- Множество документов D' делят на две части:
  - обучающая D (для построения Φ')
  - тестовая (для проверки его работы)

Предположение:
обучающие и новые данные однородны

КЛАССИФИКАЦИЯ: ЭТАПЫ



1. Лингвистический и статистический анализ текстов коллекции, построение ***образа*** каждого документа, т.е. набора признаков: $\boldsymbol{d_j = (d_{j1}, \ldots, d_{jm})}$

где $\boldsymbol{d_{ji}}$ – вес j-ого признака в i-ом документе $0 \le d_{ji} \le 1$,
$\boldsymbol{m}$ - количество различных признаков

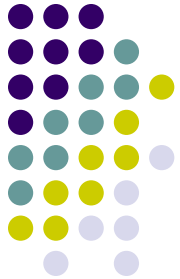
Часто: признаком выступает значимое слово текста(*term*), а вес вычисляется по формуле *tf-idf*

2. Построение (обучение) классификатора выбранным методом машинного обучения

- методы на основе наборов признаков: Байесовский классификатор, Деревья решений
- методы на основе векторов признаков : *kNN*, *SVM*
- методы на основе нейронных сетей

3. Оценка качества построенного классификатора (полнота, точность, F-мера) на тестовых наборах

БАЙЕСОВСКИЙ КЛАССИФИКАТОР



«Наивный Байес» для модели «мешок слов»

- Наивный: предположение, что слова (*terms*) не зависят друг от друга и от их позиций в тексте
- Ищем наилучший класс c^* для документа d_i :

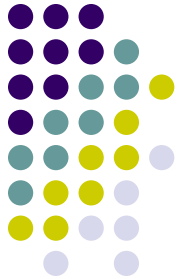
$$c^* = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j \mid d_i) = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j) P(d_i \mid c_j)$$

где $c_j \in C, \quad d_i \in D,$

$P(c_j \mid d_i)$ – условная вероятность, что d_i окажется в c_j

- Вычислить $P(c_j \mid d_i)$ напрямую невозможно, т.к. для этого обучающее множество должно содержать все возможные комбинации классов и документов
- Используем формулу Байеса и предположение

БАЙЕС: ПРАВИЛО КЛАССИФИКАЦИИ



Согласно предположению о независимости, $P(d_i \mid c_j)$ вычисляется как произведение вероятностей встретить слово-термин t_k в документах класса c_j

$$c^* = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j) \prod_{k=1}^{|T_{d_i}|} P(t_k \mid c_j)$$

Оценка вероятностей на обучающем множестве:

$$P(c_j) = \frac{|D_{c_j}|}{|D'|} \qquad P(t_k \mid c_j) = \frac{tf(t_k, c_j)}{\sum_{i=1}^{m} tf(t_i, c_j)}$$

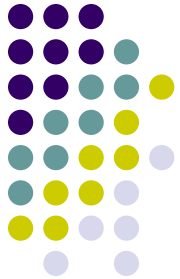
$T = \{t_1, \ldots, t_m\}$ – множество признаков всех документов

T_{di} – множество слов-признаков в документе d_i

D_{cj} – множество документов в классе c_j

$tf(t_k, c_j)$ – частота признака t_k в документах класса c_j

БАЙЕСОВСКИЙ КЛАССИФИКАТОР: ПРИМЕР (2 КЛАССА)



	Термины (слова) в документе	*c*=«Китай»
d_1	китайский пекин китайский	*c*
d_2	китайский китайский шанхай	*c*
d_3	китайский макао	*c*
d_4	токио япония китайский	$\neg c$
d_5	китайский китайский китайский токио япония	?

Обучение: $P(c) = 3/4$, $P(\neg c) = 1/4$

$P(\text{китайский}|c)=5/8$, $P(\text{токио}|c)=P(\text{япония}|c)=0$

$P(\text{китайский}|\neg c)=P(\text{токио}|\neg c)=P(\text{япония}|\neg c)=1/3$

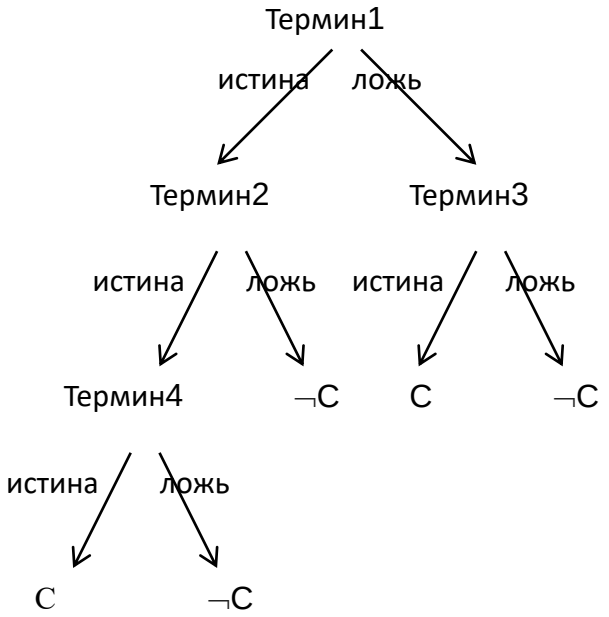
Применение: $P(d_5|c)=3/4*(5/8)^3*0*0=0$

$P(d_5|\neg c)=1/4*(1/3)^3*1/3*1/3\approx 0{,}001$

Следовательно, $c^* = \neg c$ («не Китай»)

МЕТОД ДЕРЕВЬЕВ ПРИНЯТИЯ РЕШЕНИЙ

- Образ документа – множество терминов-слов
- Строим дерево:
  - ✓ узлы – термины документов
  - ✓ листья – метки классов
  - ✓ на ребрах – веса терминов



При обучении ищем термин, обладающий наибольшей различительной способностью – пытаемся максимизировать прирост информации

$$t^* = \underset{t_k \in T}{\operatorname{argmax}}\, I(D, t_k) =$$

$$\underset{t_k \in T}{\operatorname{argmax}} \left(E(D,c) - \left(p(t_k)E(t_k,c) + p(\neg t_k)E(\neg t_k,c)\right)\right)$$

I – кол-во информации
E – энтропия

ДЕРЕВО РЕШЕНИЙ: ПРИМЕР

Обучение (вычисление характеристик):

Исходная энтропия

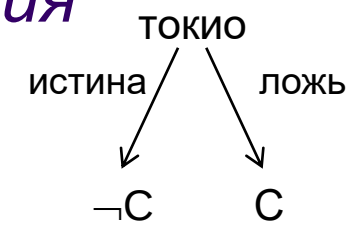
$$E(D,c) = -p(c)\log_2 p(c) - p(\neg c)\log_2 p(\neg c) \approx 0{,}81$$

I(китайский) = 0

I(пекин) = I(шанхай) = I(макао) = 0,12

I(токио) = I(япония) = 0,81

Разделяющий термин: *токио* или *япония*



Применение:

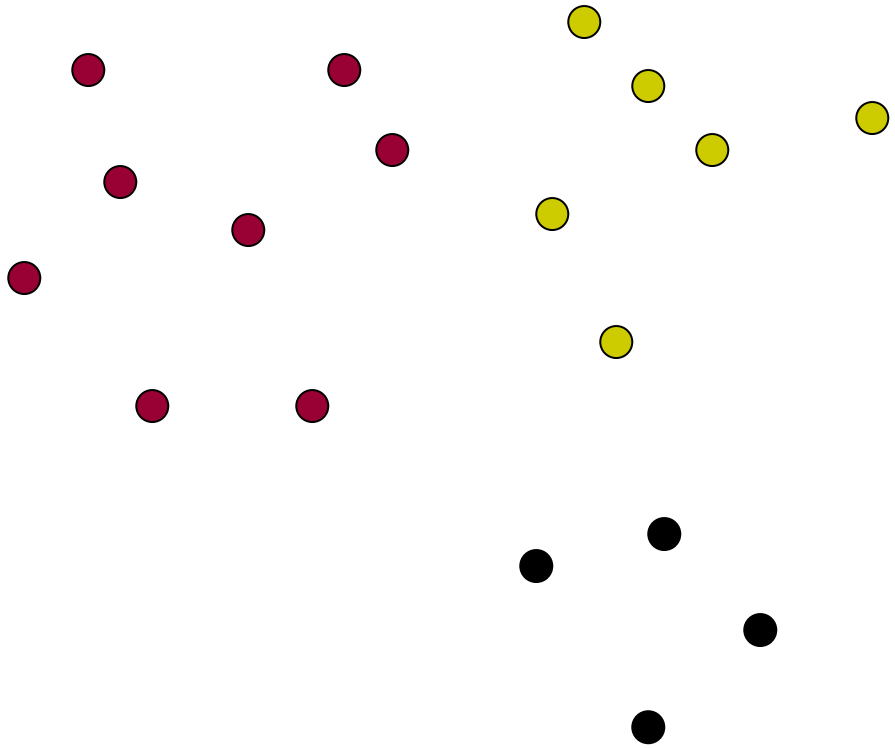
d_5 содержит термин *токио*

Следовательно, $c^* = \neg c$ («не Китай»)

МЕТОДЫ КЛАССИФИКАЦИИ: ДОКУМЕНТ КАК ВЕКТОР В ПРОСТРАНСТВЕ ПРИЗНАКОВ

$d_j = (w_{j1}, \ldots, w_{jm})$, где

w_{ji} – вес i-ого признака в j-ом документе

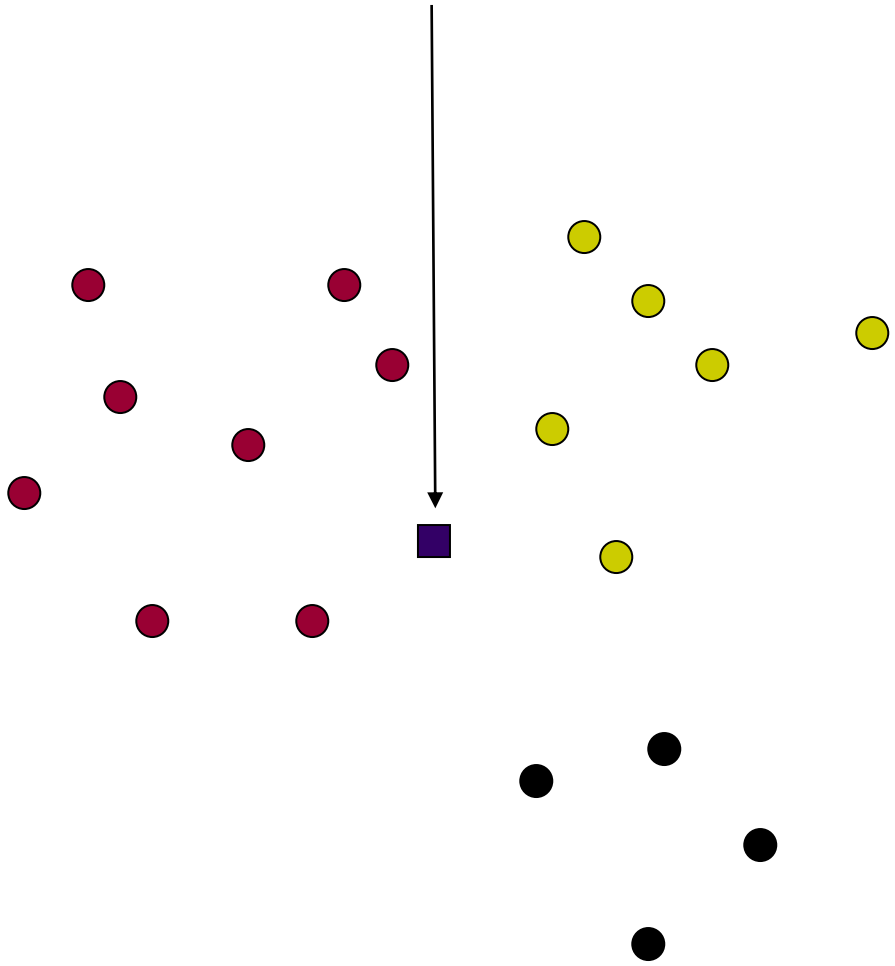


Правительство

Наука

Искусство

КЛАССИФИКАЦИЯ НОВОГО ДОКУМЕНТА



Предположения:

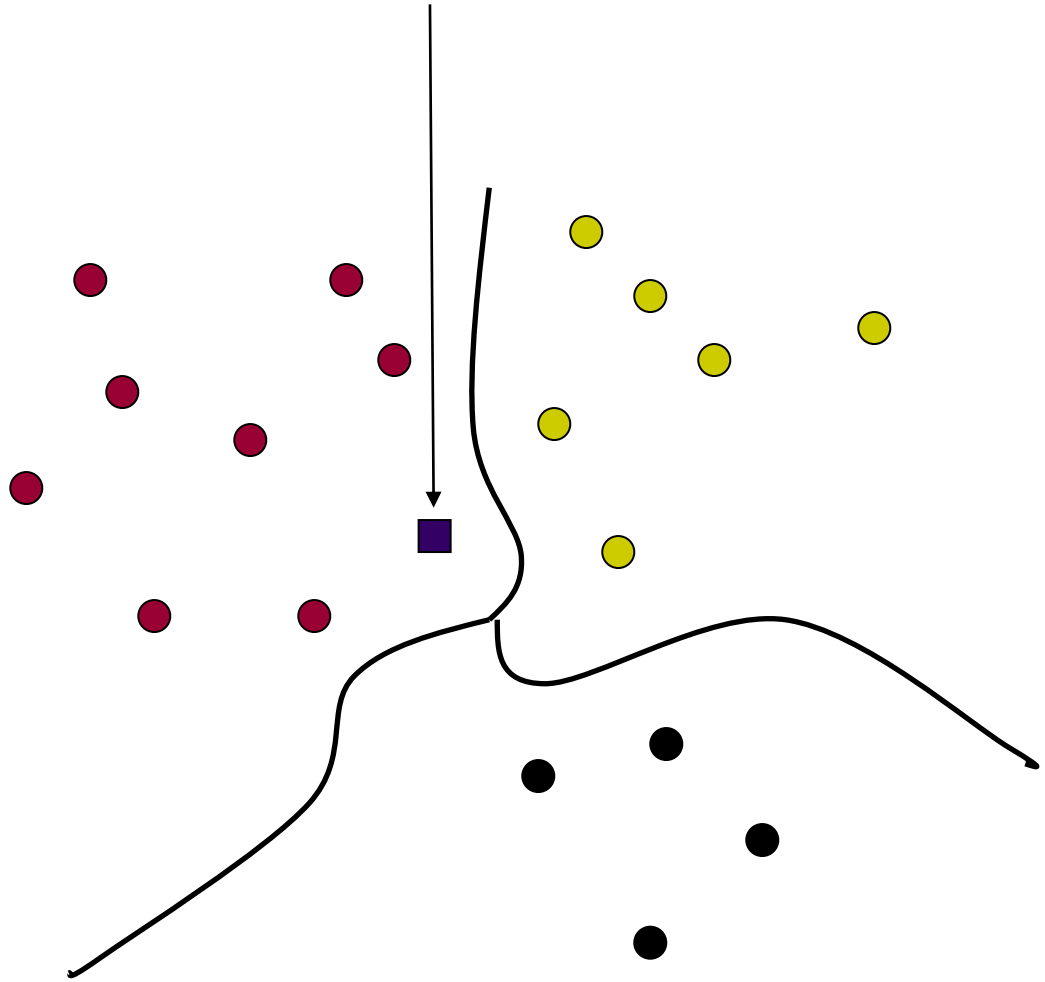
– документы одного класса находятся в одной области пространства

– документы из разных классов находятся непересекающихся областях

Границы классов?

- Правительство
- Наука
- Искусство

ТЕМА ДОКУМЕНТА – ПРАВИТЕЛЬСТВО



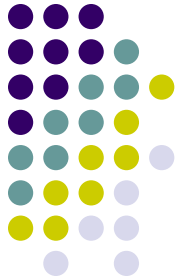
Правильно ли определены границы?

Методы классификации:

- ✓ Роккио
- ✓ *kNN*
- ✓ *SVM*
- ✓ др.

- Правительство
- Наука
- Искусство

КЛАССИФИКАЦИЯ: МЕТОД РОККИО



- Образ документа – вектор признаков
- Метод ищет границы между классами как множества точек, равноудалённых от *центроидов* этих классов (предполагается, что классы имеют форму сфер)

Центроид класса– усреднённый вектор членов класса

$$\mu_{c_j} = \frac{1}{|D_{c_j}|} \sum_{i:d_i \in c_j} d_i$$

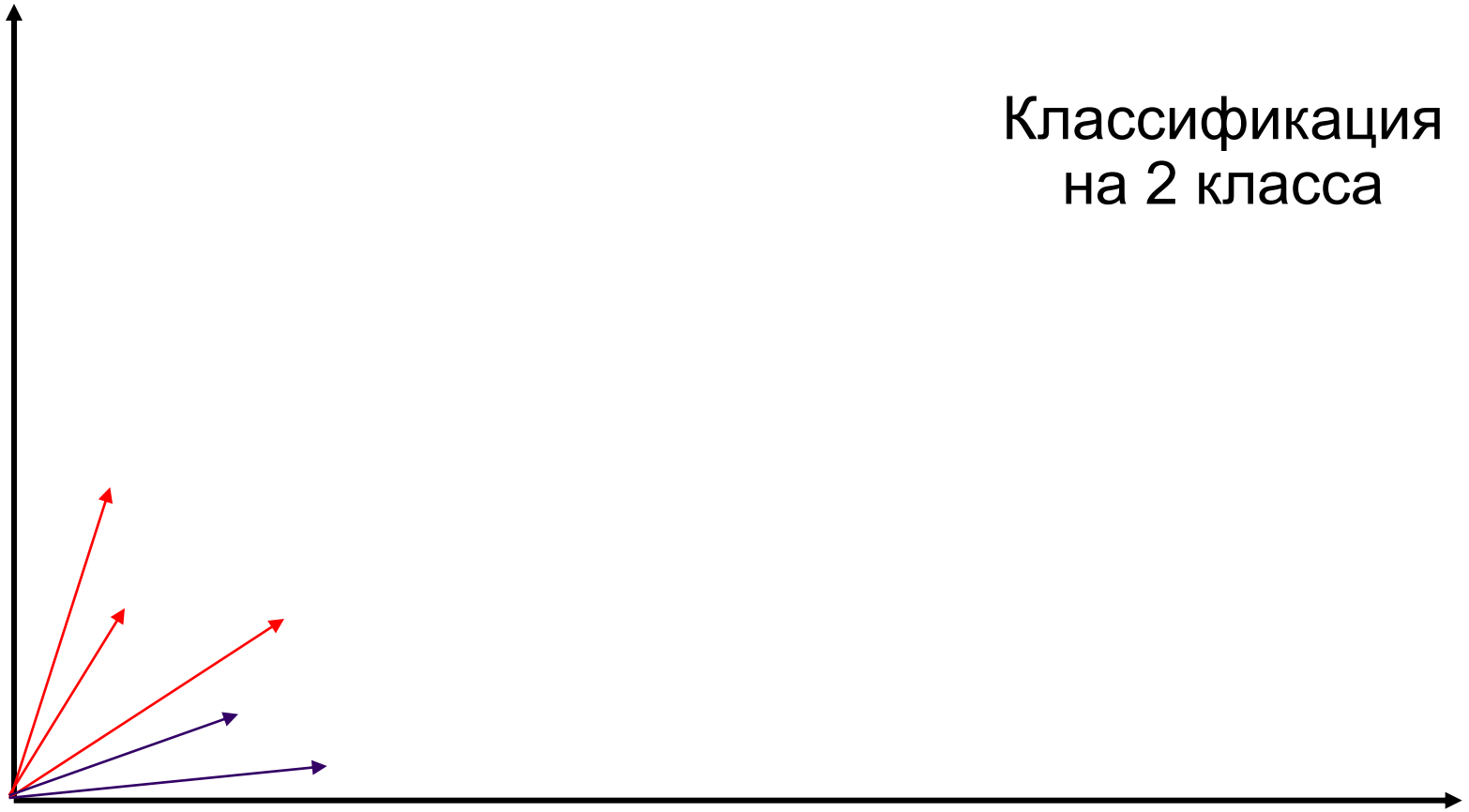
D_{c_j} – множество документов в классе c_j

- Правило классификации: поиск центроида (класса), к которому образ нового документа d ближе всего

$$c^* = \underset{c_j \in C}{\operatorname{argmin}} \left\| \mu_{c_j} - d \right\|$$

- Хорошо работает со «сферическими» классами

ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (1)



Классификация
на 2 класса

ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (2)

ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (3)

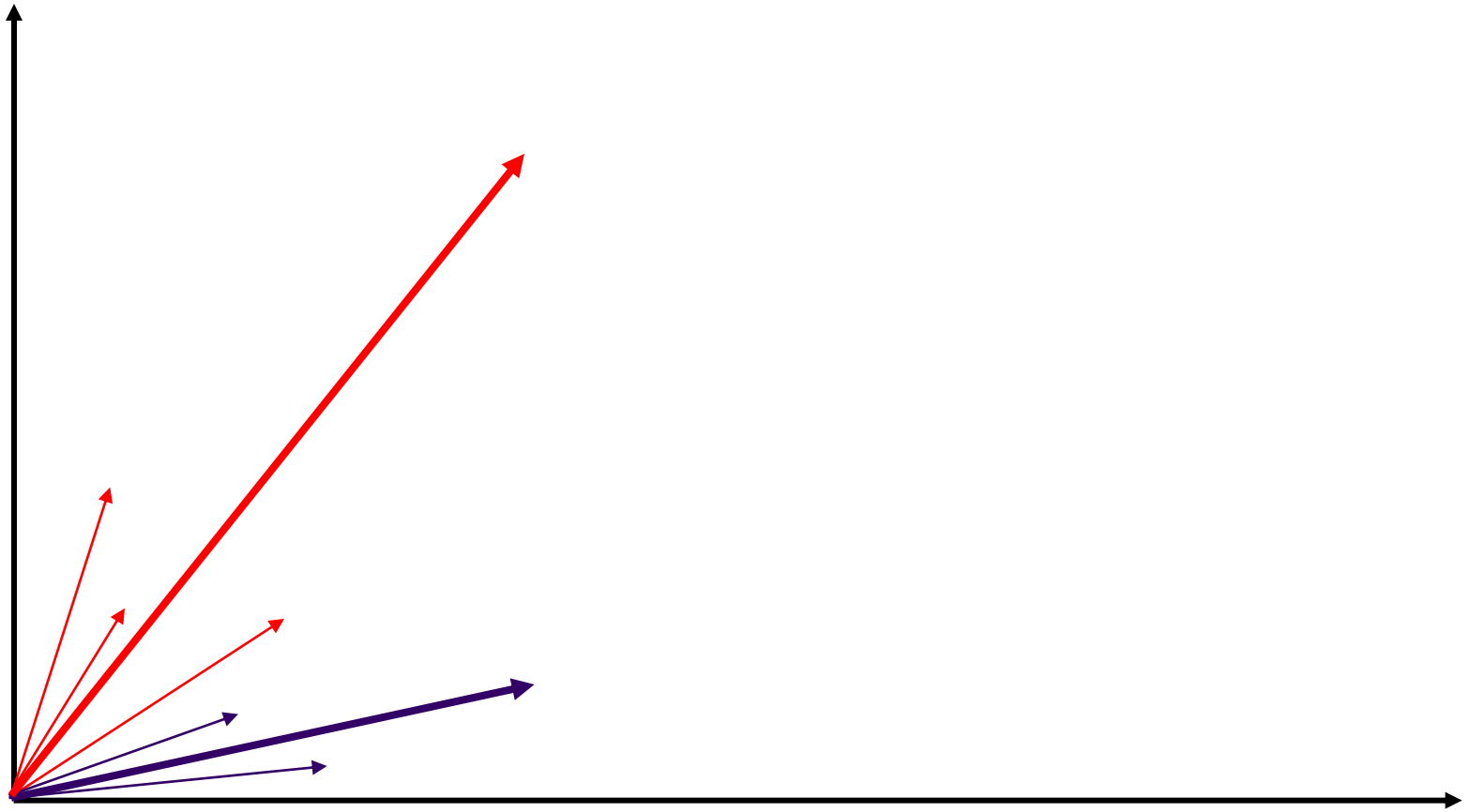


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (4)

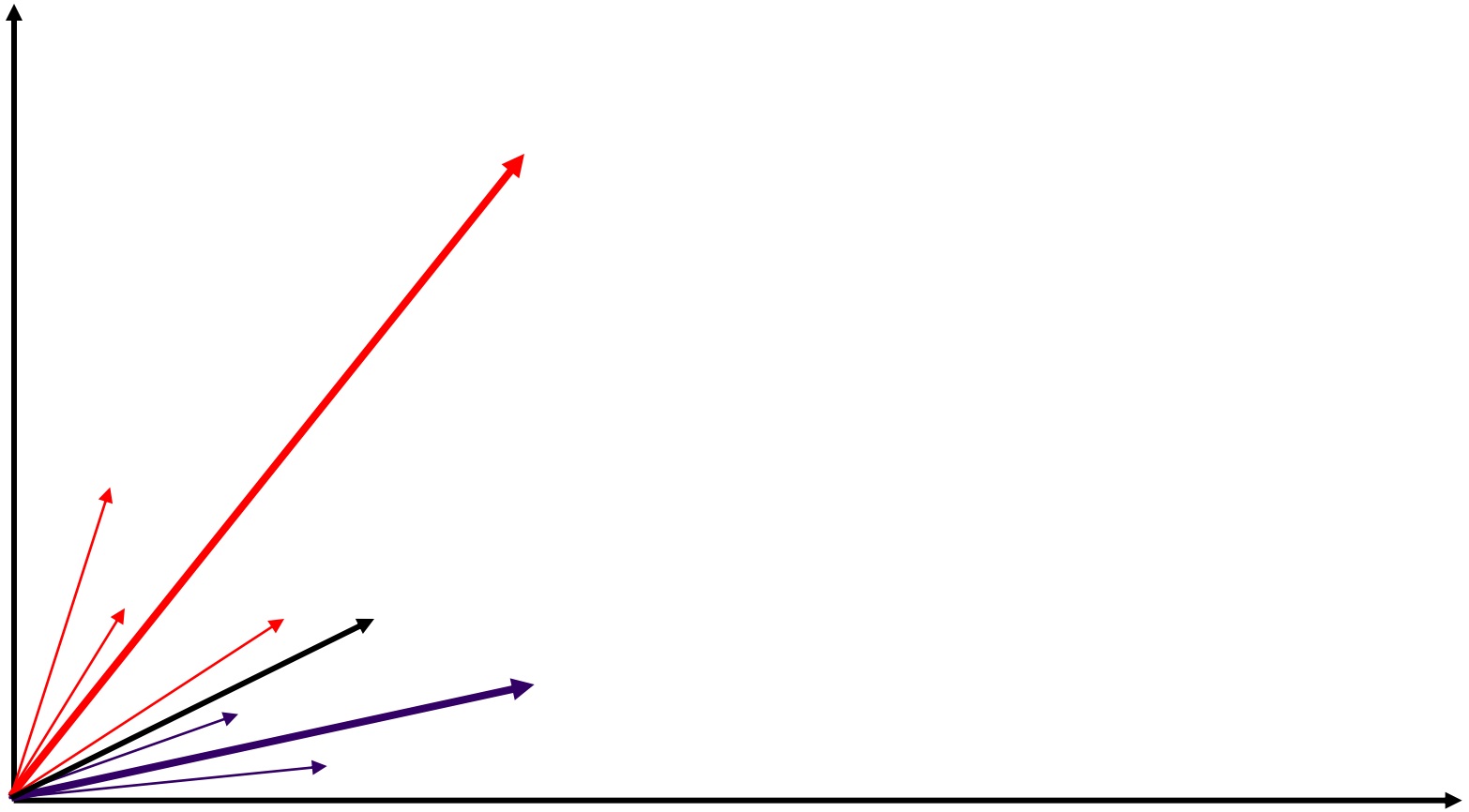


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (5)

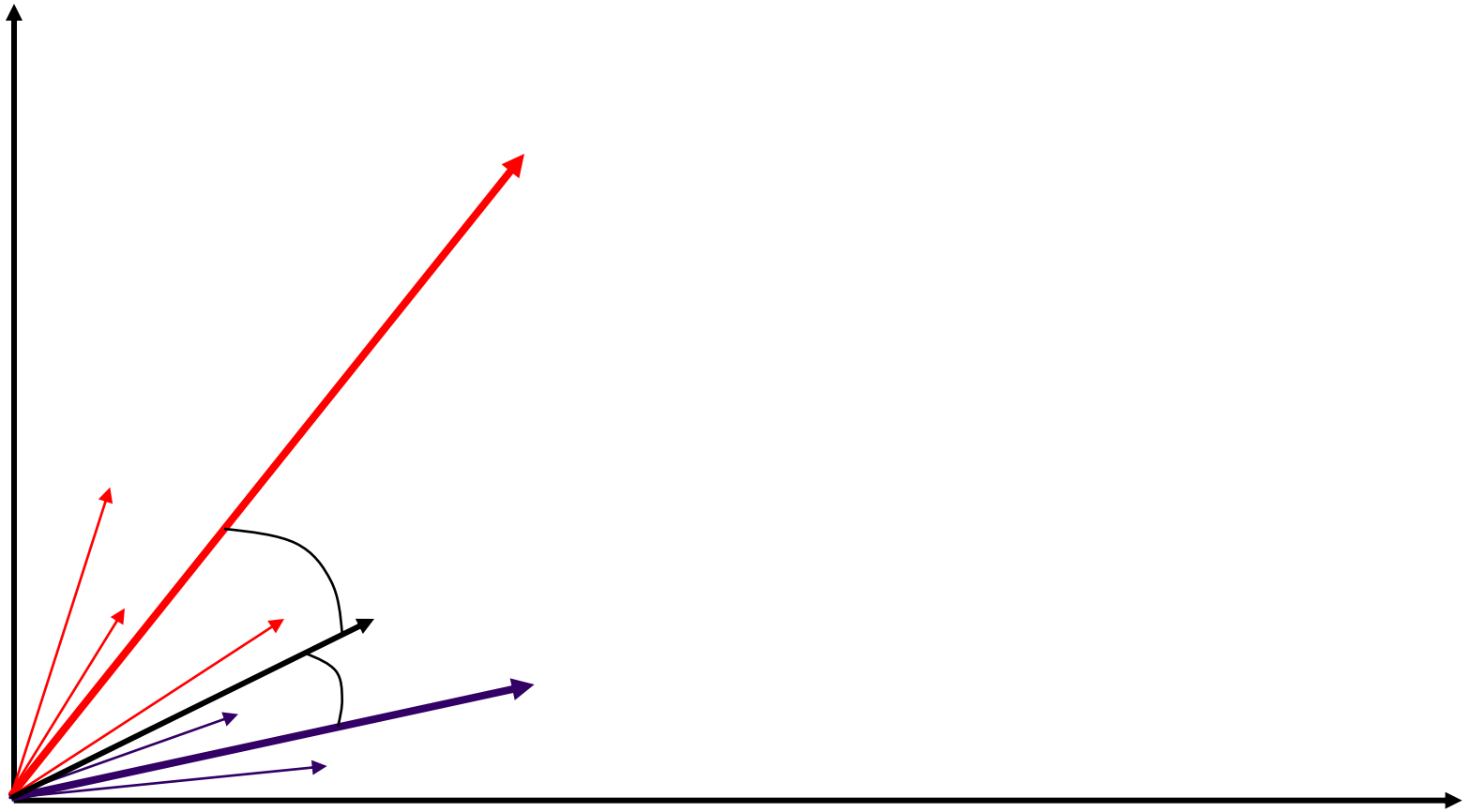
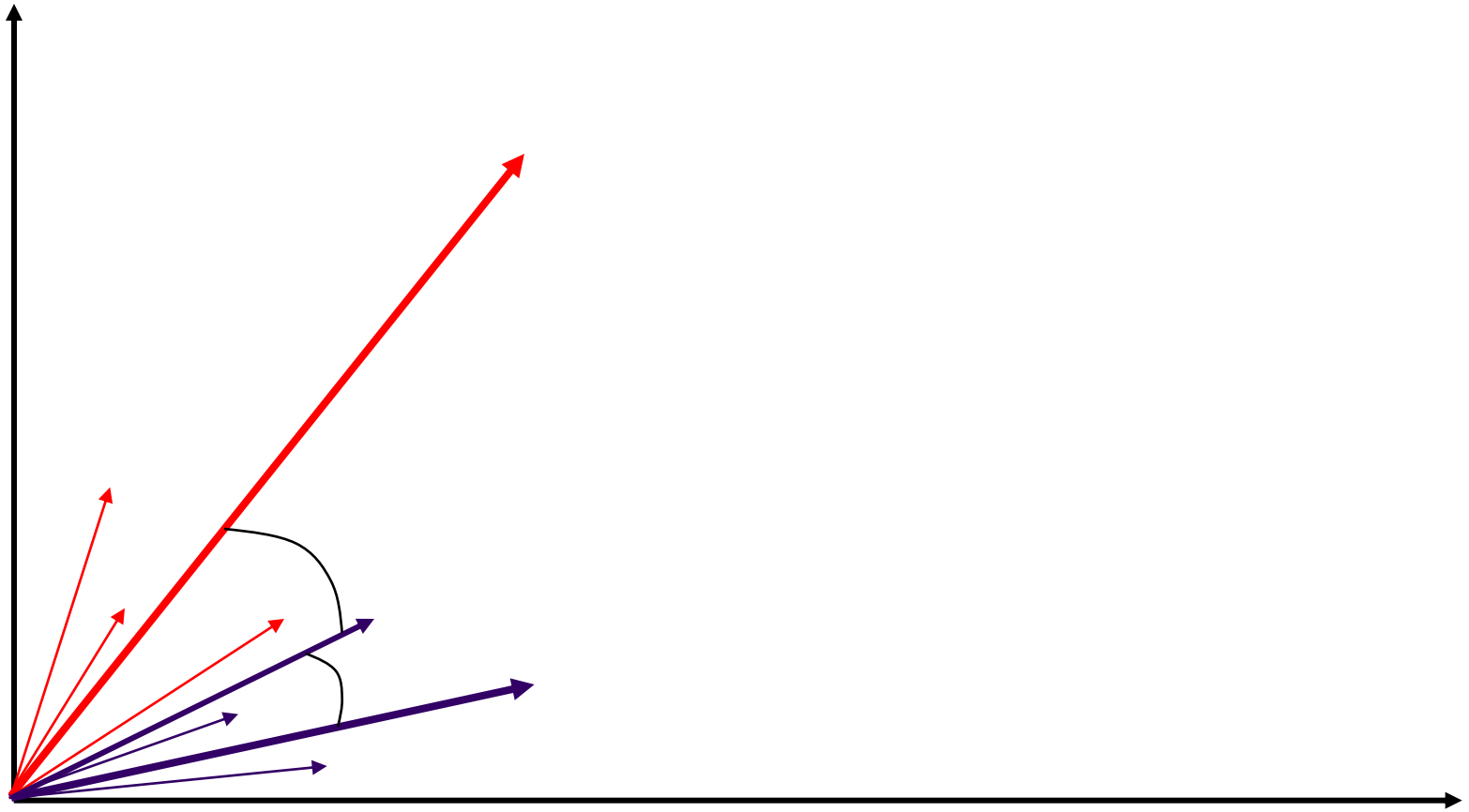


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА РОККИО (6)



МЕТОД РОККИО: ПРИМЕР ПРИМЕНЕНИЯ

Обучение: вычисление центроидов c и $\neg c$

$d_5 = \{0, 0, 0, 0, 0.7, 0.7\}$ (d_{ij} – по формуле *tf-idf*)

	t_1	t_2	t_3	t_4	t_5	t_6
	китайский	*пекин*	*шанхай*	*макао*	*япония*	*токио*
μ_c	0	0,33	0,33	0,33	0	0
$\mu_{\neg c}$	0	0	0	0	0.7	0.7

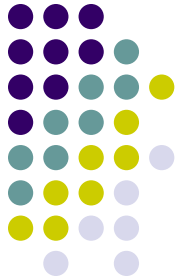
Применение:

$$\|\mu_c - d_5\| = \sqrt{0+0{,}33^2+0{,}33^2+0{,}33^2+0{,}7^2+0{,}7^2} \approx 1{,}14$$

$$\|\mu_{\neg c} - d_5\| = \sqrt{0+0+0+0+0+0} = 0$$

Следовательно, $c^* = \neg c$ («не Китай»)

КЛАССИФИКАЦИЯ: МЕТОД *kNN*



kNN – *k-nearest neighbors* (ближайшие соседи)

- Образ документа – вектор в пространстве признаков
- Предположение: документы одного класса образуют компактную область, причём области разных классов не пересекаются
- Правило классификации: новый документ относится к тому классу, который является наиболее распространённым среди ***k*** его ближайших соседей, классы которых известны
- Неплохо работает с несферическими классами

ПРИМЕР: **Обучение** – выбор ***k*** на основе опыта эксперта и имеющихся знаний о решаемой задаче

Применение, ***k=3***:

$$\|d_5 - d_1\| = \|d_5 - d_2\| = \|d_5 - d_3\| \approx 1{,}92 \qquad \|d_5 - d_4\| = 0$$

Следовательно, $c^* = \neg c$ («не Китай»)

ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА *kNN*, *k*=3 (1)

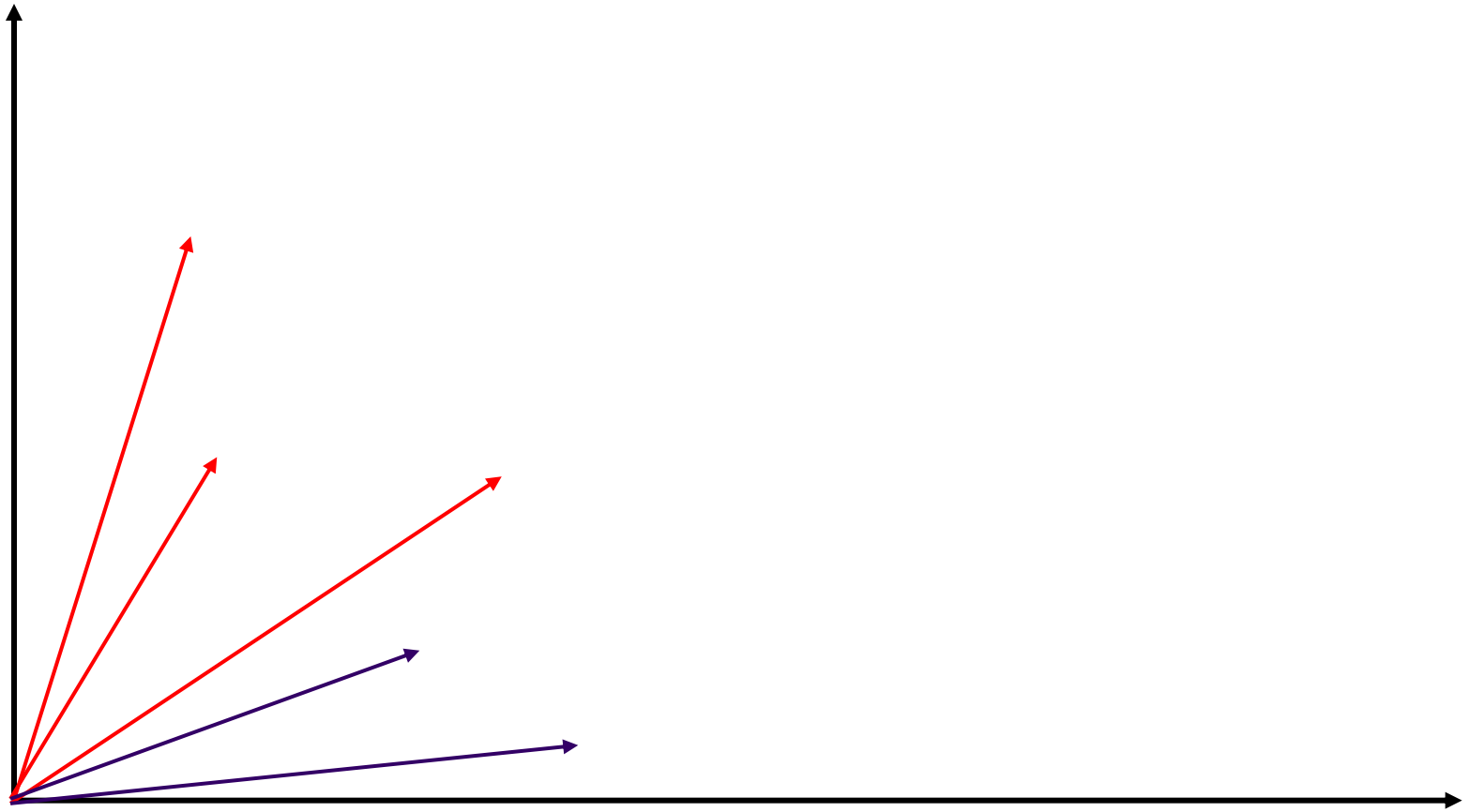


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА *kNN*, *k*=3 (2)

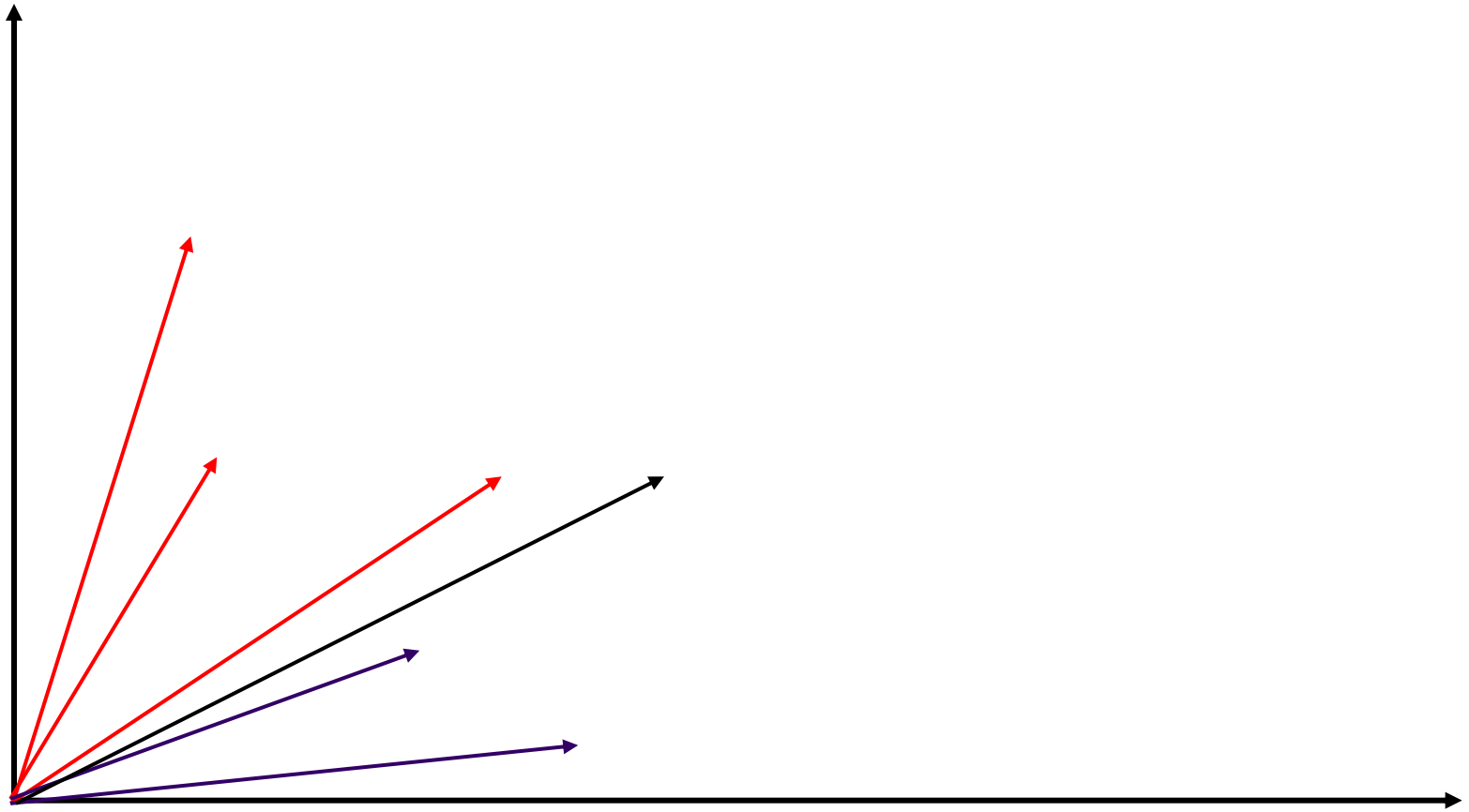


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА *kNN*, *k=3* (3)

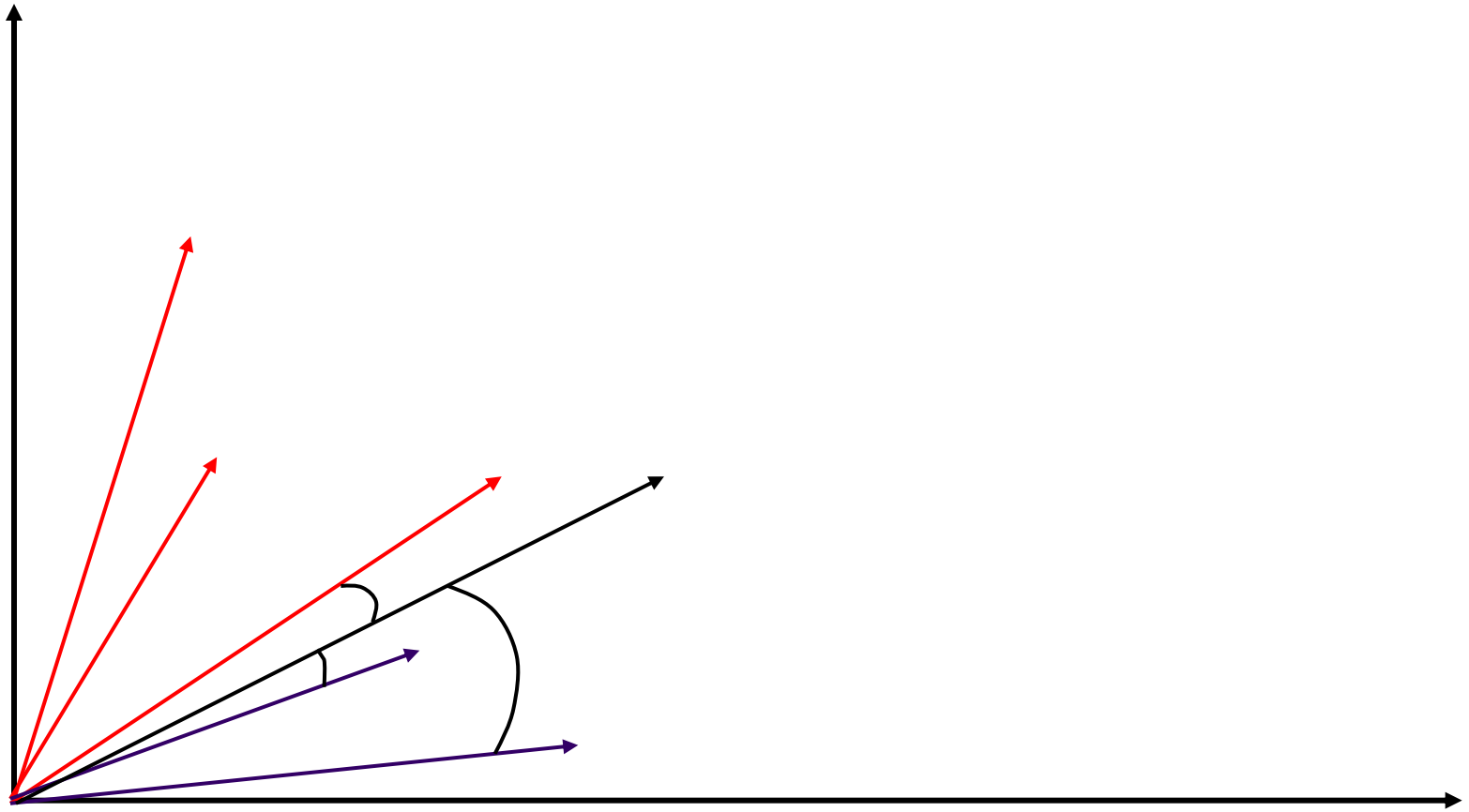
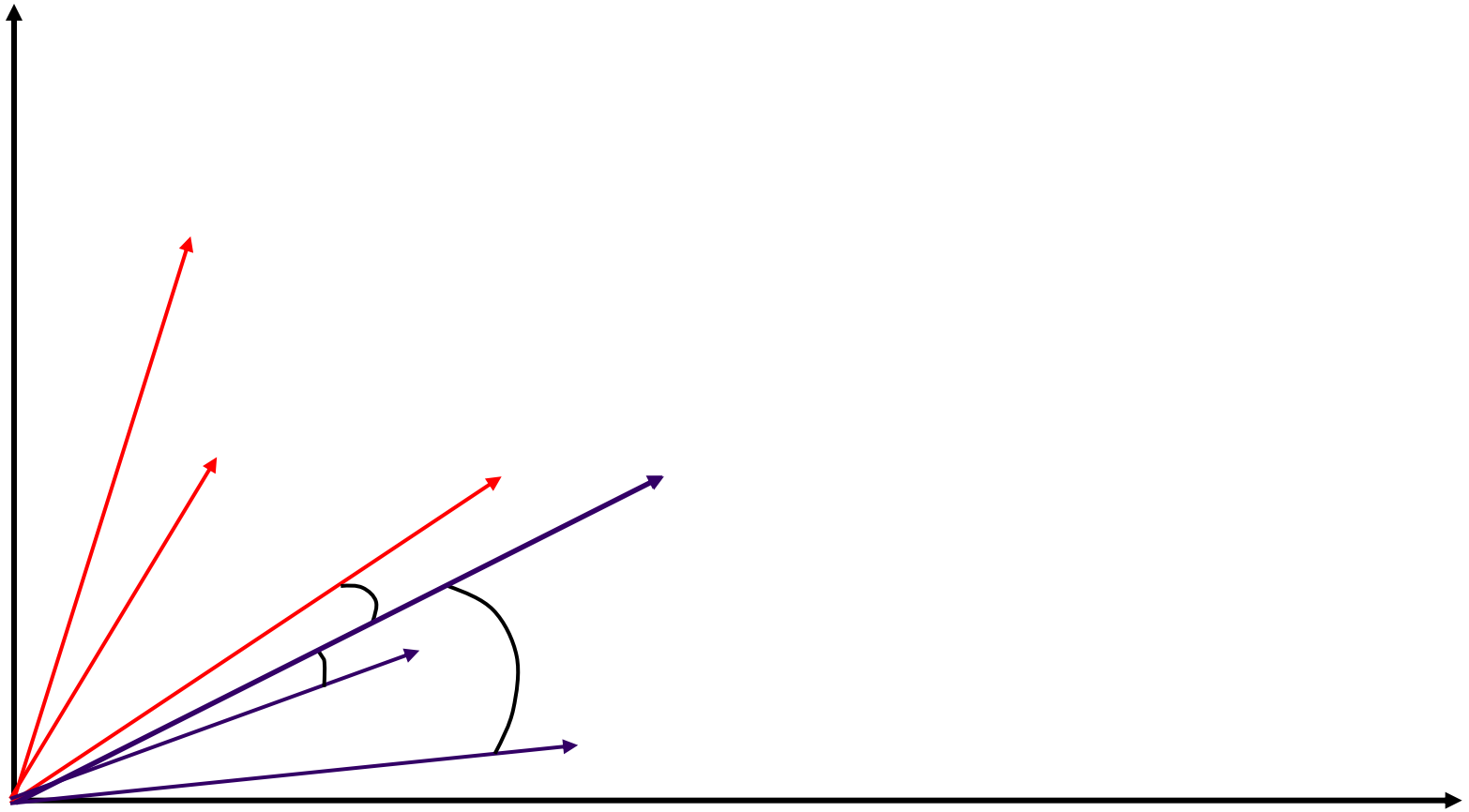


ИЛЛЮСТРАЦИЯ РАБОТЫ МЕТОДА *kNN*, *k*=3 (4)

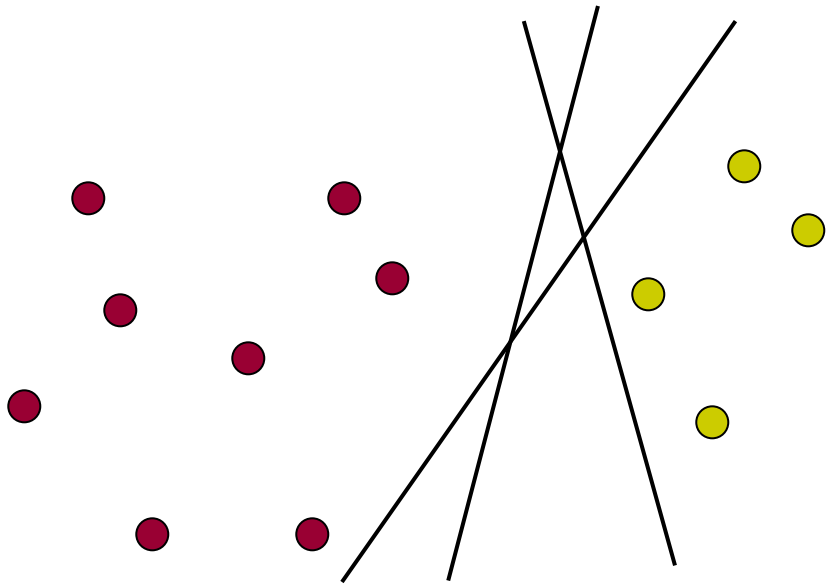


МЕТОД ОПОРНЫХ ВЕКТОРОВ

Support Vector Machine (SVM)

Документы нужно разделить на два класса («Китай»-«не Китай»)

Как правильно определить разделяющую поверхность (линию, гиперплоскость)?



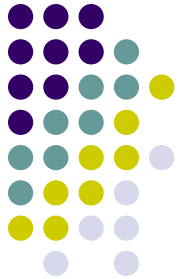
Найти a, b, c, такие, что

$ax + by > c$ для красных

$ax + by < c$ для желтых

Решений бесконечно много, нужно искать оптимальное

ИДЕЯ МЕТОДА *SVM*



- Предполагается, что обучающая выборка имеет вид $\{(d_1,m_1),\ldots (d_n,m_n)\}$, где

  $m_i = 1$, если $d_i \in c$, и $m_i = -1$, если $d_i \in \neg c$

- Нужно найти разделяющую гиперплоскость:

  $$w\, d - b = 0\ ,$$ где

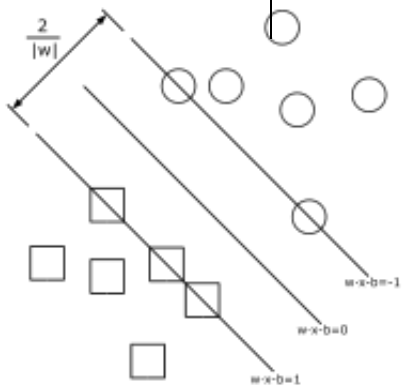
  w – перпендикуляр к гиперплоскости, b – константа

- Нас интересует оптимальное разделение:
  - берем гиперплоскости, параллельные оптимальной, и ближайшие к ним точки – опорные вектора (*support vectors*)
  - максимизируем расстояние между гиперплоскостью и опорными векторами

ОПТИМАЛЬНАЯ ГИПЕРПЛОСКОСТЬ

Параллельные гиперплоскости имеют вид:

$$w\,d - b = 1 \qquad w\,d - b = -1$$



Нужно выбрать w и b такие, что:

- Между гиперплоскостями не лежат точки обучающей выборки
- Расстояние между ними максимально

Потребуем: $w\,d_i - b \geq 1$ для $d_i \in c$,

$w\,d_i - b \leq -1$ для $d_i \in \neg\, c$

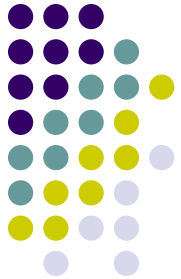
Расстояние: $2/\| w \|$, т.е. нужно минимизировать w

Решение задачи минимизации – Решающее правило:

$$f(d) = sign(\sum_{i=1}^{n} \alpha_i m_i d_i * d - b)$$

$\alpha_i \neq 0$ для опорных векторов

МЕТОД *SVM:* ПРИМЕР ПРИМЕНЕНИЯ



Обучение: получение значений α_i и m_i

(α_i получают из статистических программных пакетов)

$$\alpha_1 \approx 0{,}31 \quad \alpha_2 \approx 0{,}23 \quad \alpha_3 \approx 0{,}23 \quad \alpha_4 \approx 0{,}78$$

$$m_1 = -1 \quad m_2 = -1 \quad m_3 = -1 \quad m_4 = -1$$

$$d_5 = (0,\ 0,\ 0,\ 0{,}7,\ 0{,}7)$$

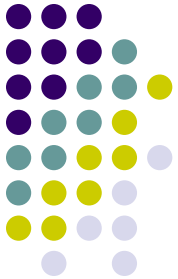
$$w = \sum_{i=1}^{n} \alpha_i m_i d_i = (0,\ -0{,}31,\ -0{,}23,\ 0{,}23,\ 0{,}55,\ 0{,}55)$$

$$b = m_i - w\, d_i = -0{,}5$$

Применение (определение, к какой полуплоскости относится новый документ): $f(d_5) = -1$

Следовательно, $c^* = \neg c$ («не Китай»)

НЕЙРОННЫЕ СЕТИ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТОВ

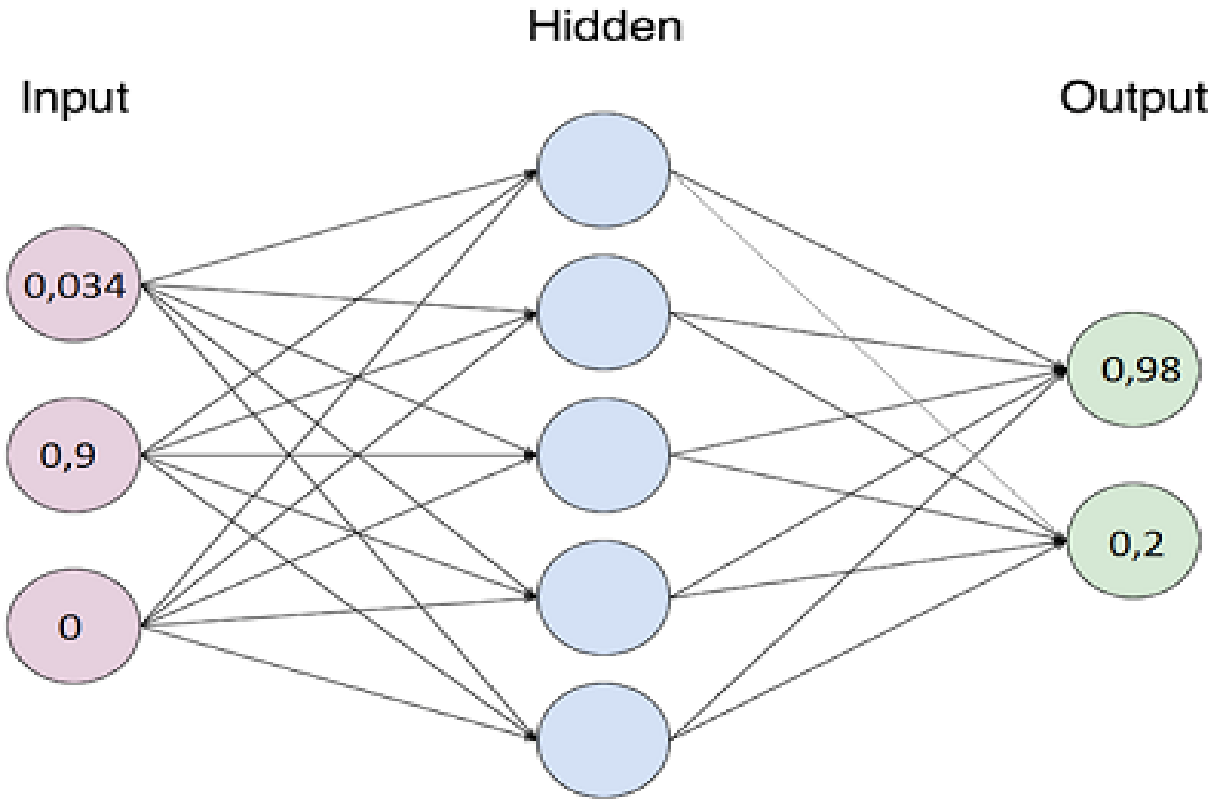


- НС различаются по архитектуре (сложности), для задачи классификации текстов коллекции – *персептрон*
- *Многослойный персептрон*:
  - входной слой нейронов + выходной + несколько промежуточных (скрытых)
  - обычно 1-4 промежуточных слоя, с уменьшающимся числом нейронов от входа к выходу
- Обучение НС состоит из нескольких эпох:
  каждая эпоха – это проход от (заданного) входа к выходу и обратно (от заданного выхода в входу) с целью минимизации несоответствия
- При проходе обычно применяется градиентный спуск для минимизации ошибки на обучающих примерах

НЕЙРОННАЯ СЕТЬ С ОДНИМ СКРЫТЫМ СЛОЕМ

Все признаки классифицируемого документа должны быть представлены числами, как и выходные данные

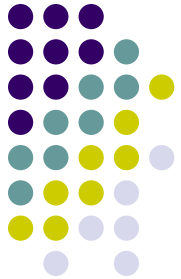
Полносвязный персептрон одним скрытым слоем:



КЛАССИФИКАЦИЯ НА БАЗЕ НС И ПРИЗНАКОВ ТЕКСТА

- Входной слой НС соответствует признакам документа (каждый нейрон – очередной признак), т.е. на вход поступает числовой вектор признаков текста
  - часто входной слой строится на базе векторов слов (*embeddings*) из предсказательных языковых моделей
- Выходной слой НС соответствует выявленным классам,
  - обычно число нейронов – количество классов
  - каждый нейрон выходного слоя, как правило, соответствует некоторому классу и дает
    – либо число 0 или 1: принадлежность текста к классу
    – либо число (0,1) : вероятность принадлежности

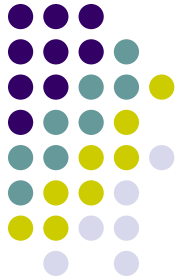
ОСОБЕННОСТИ МЕТОДОВ ДЛЯ КЛАССИФИКАЦИИ



- Алгоритм «наивной» байесовской классификации устойчив к шуму и неоднородным данным (в отличие от алгоритма деревьев принятия решений)
- Алгоритм Роккио хорошо работает для классов, близких к сферическим
- Алгоритм k-ближайших соседей неплохо справляется с несферическими и несвязанными классами
- Алгоритм опорных векторов обычно используют для разбиения на два непересекающихся класса
- Нейронные сети могут по требовать достаточно большого объема обучающих данных

❖ Качество классификации может существенно различаться: 60-99%,
метод подбирается к конкретной задаче

ОЦЕНКИ КАЧЕСТВА КЛАССИФИКАЦИИ



Поскольку обучающее множество может некорректно отражать реальные данные, необходимы оценки.

Основные показатели:

- *Точность* – доля правильных решений (объектов класса) среди найденных объектов
- *Полнота* – доля правильно найденных по отношению к общему числу объектов класса
- *Ошибка* – доля ложных решений
- *Аккуратность* – доля всех правильно принятых решений к общему числу решений
- и др.

$$\text{Precision} = \frac{tp}{tp + fp}$$

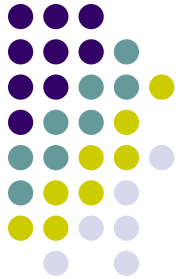
$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{True negative rate} = \frac{tn}{tn + fp}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$t,\ f,\ p,\ n$ – *true, false, positive, negative*

КЛАССИФИКАЦИЯ: КОМБИНИРОВАННАЯ МЕРА



- Обычно чем лучше точность, тем хуже полнота и наоборот
- *F-мера* – интегральный показатель
- Часто применяется сбалансированная, *F1-мера* – среднее гармоническое между полнотой и точностью:

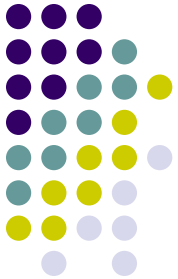
  $\beta=1$ или $\alpha=1/2$

$$F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P+R}$$

$$F1 = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R}$$

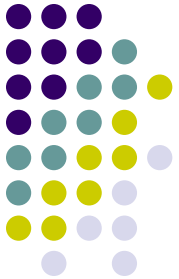
- В графическом виде: *ROC-кривая* ошибок (для бинарной классификации) отображает соотношение между долей TP-объектов от общего числа объектов (*чувствительность*) и долей FP-объектов (1-FP, *специфичность* метода)

ОЦЕНКИ КАЧЕСТВА ДЛЯ НЕСКОЛЬКИХ КЛАССОВ



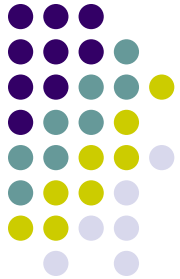
- Используются полнота, точность, F-мера, ошибка классификатора для каждого класса
- Если классов больше двух, то как объединять рассмотренные оценки для каждого класса?
- Используется
  - макроусреднение: составляются таблицы принятия решений для каждого класса по-отдельности, вычисляются меры, берется среднее по всем классам
  - микроусреднение: составляется единая таблица для всех классов, затем по этой таблице вычисляют меры

КЛАССИФИКАЦИЯ: ПРИКЛАДНЫЕ ЗАДАЧИ



- Упорядочивание набора документов
- Навигация по набору документов
  – составление интернет-каталогов
- Ограничение области поиска (в поисковых системах)
- Фильтрация потока документов:
  – фильтрация спама
- Персонализированный/тематический подбор информации:
  – контекстная реклама, новости и т.п.

АВТОМАТИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ ПОСТАНОВКА ЗАДАЧИ



- Обучение без учителя (*unsupervised*)
- Имеется множество документов

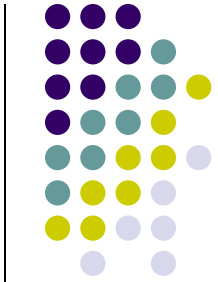
$$D = \{d_1, \ldots, d_{|D|}\}$$

- Необходимо их разбить на подмножества – **кластеры** похожих документов

$$C = \{c_1, \ldots, c_{|C|}\}$$

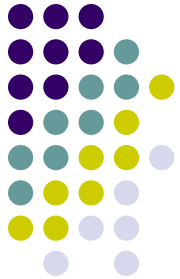
- Алгоритм должен самостоятельно принимать решение о количестве и составе кластеров
- Используется понятие схожести документов
  - в идеале: семантическое сходство
  - на практике: документы – вектора в пространстве признаков, важно расстояние между ними

МЕТОДЫ КЛАСТЕРИЗАЦИИ



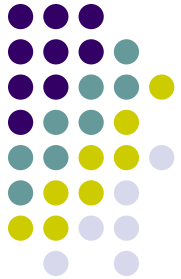
- **Плоские алгоритмы** создают неструктурированное множество кластеров
  - алгоритм k-средних
  - нечеткий алгоритм c-средних
  - плотностный алгоритм *DBSCAN (Density Based Spatial Clustering of Applications with Noise)*
  - алгоритм *SOM (Self Organization Map)*
  - алгоритм *C^2ICM (Cover-Coefficient-based Incremental Clustering Methodology)*
- **Иерархические алгоритмы** создают структурированное множество кластеров:
  - восходящие (*агломеративные*)
  - нисходящие (*дивизимные*)

ПЛОСКИЕ АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ



- Алгоритм *k*-средних и нечеткий алгоритм *c*-средних:
  - опора на центроиды кластеров
  - в алгоритме *c*-средних документ может быть отнесен к нескольким кластерам
- Плотностный алгоритм *DBSCAN*:
  - плотность внутри кластера выше, чем снаружи
  - учитывает кластеры произвольной формы
- Алгоритм C^2ICM:
  - опора на «затравки» кластеров (документы, признаки которых «покрывают» соседние документы)
  - позволяет изменять кластерную структуру без проведения перекластеризации всех данных

АЛГОРИТМ *k*-СРЕДНИХ



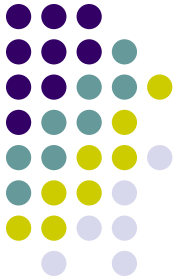
Входные данные:

- количество кластеров *k*
- множество документов как векторов

Выполнение алгоритма:

1. Выбираем *k* начальных центроидов кластеров
2. Каждый документ относим к тому кластеру, чей центроид является наиболее близким
3. Выполняем повторное вычисление центроидов каждого кластера
4. Повторяем, пока не достигнем условия остановки:
   - выполнено пороговое число итераций
   - центроиды кластеров больше не изменяются
   - достигнуто пороговое значение целевой функции

ЦЕЛЕВАЯ ФУНКЦИЯ АЛГОРИТМА *k*-СРЕДНИХ



Алгоритм минимизирует целевую функцию (среднеквадратичную ошибку кластеризации)
как среднеквадратичное расстояние между документами и центрами их кластеров

$$e(D,C) = \sum_{j=1}^{k} \sum_{i:d_i \in c_j} \left\| d_i - \mu_j \right\|^2 \quad \text{, где}$$

μ_i – центроид кластера c_j, вычисляется (и перевычисляется) как

$$\mu_j = \frac{1}{|c_j|} \sum_{i:d_i \in c_j} d_i$$

Идеальный кластер –
сфера с центроидом в ее центре

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (0)

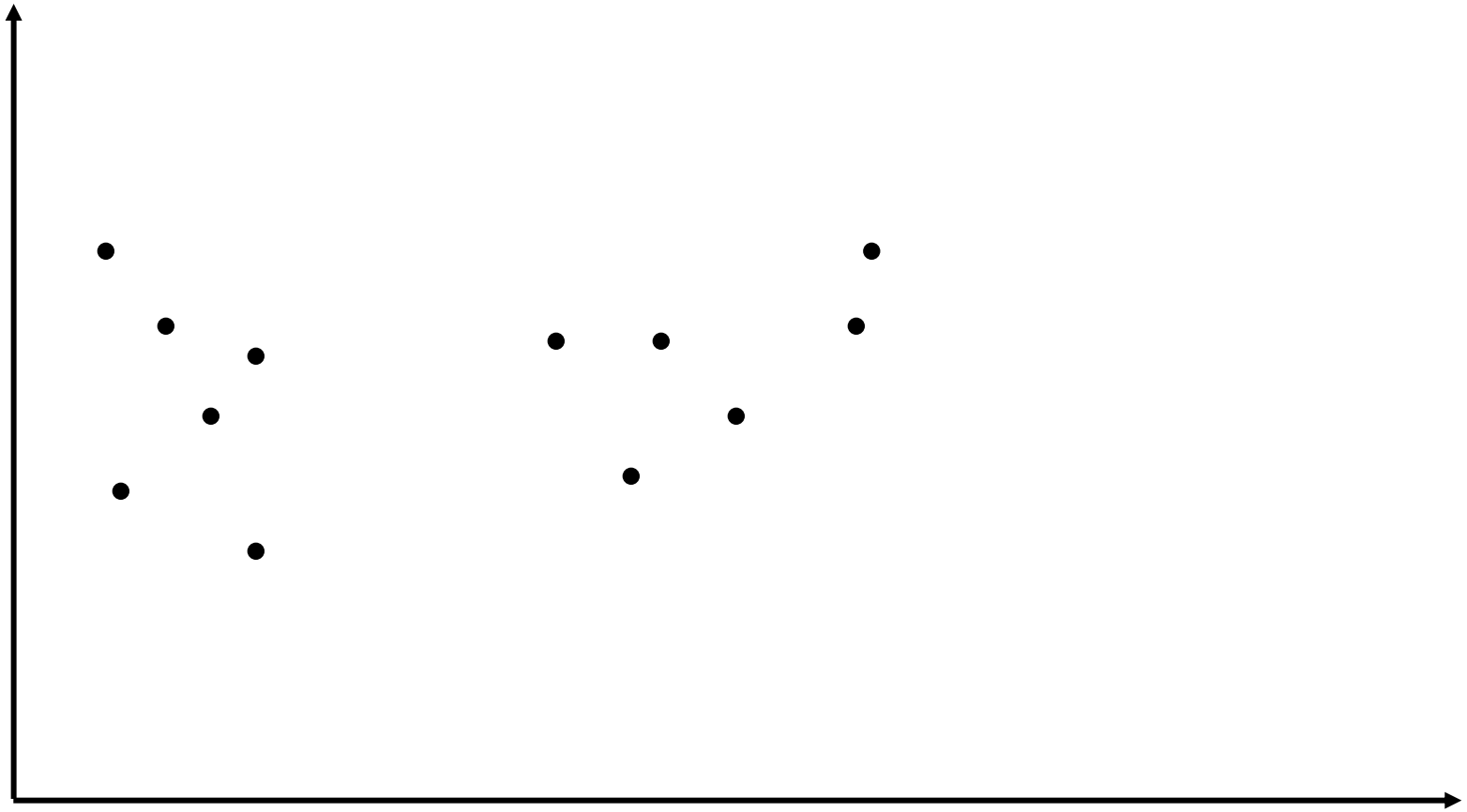
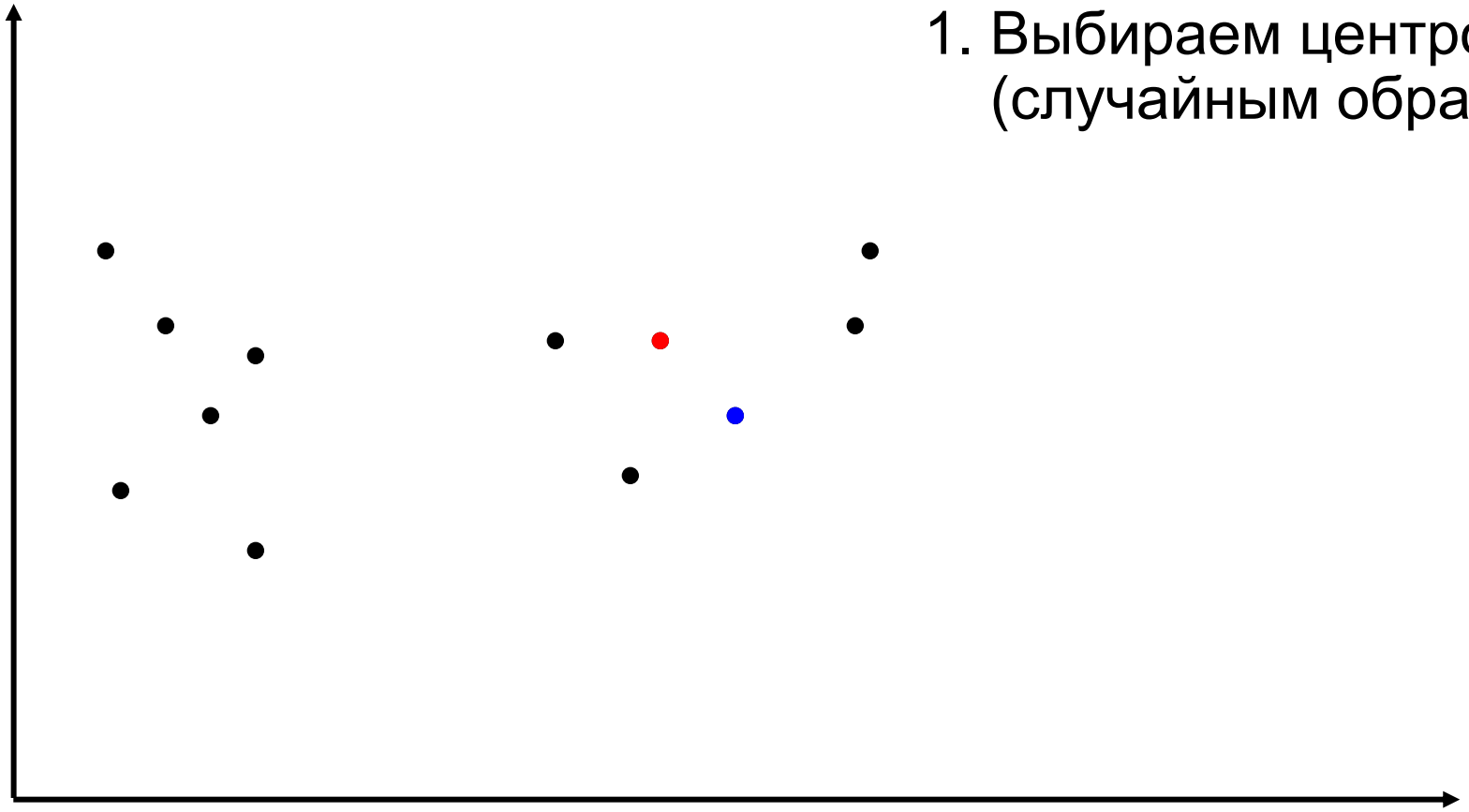
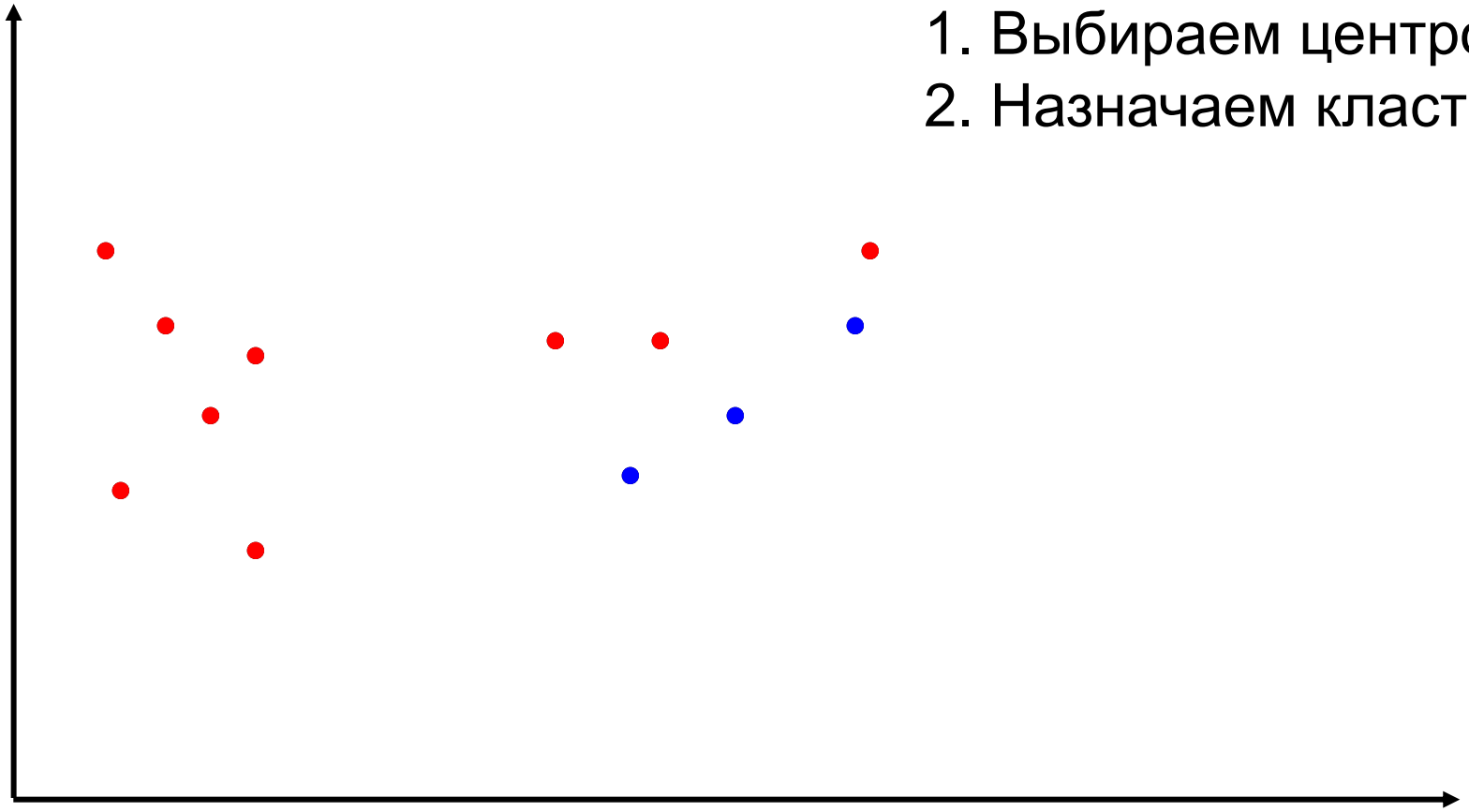


ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (1)



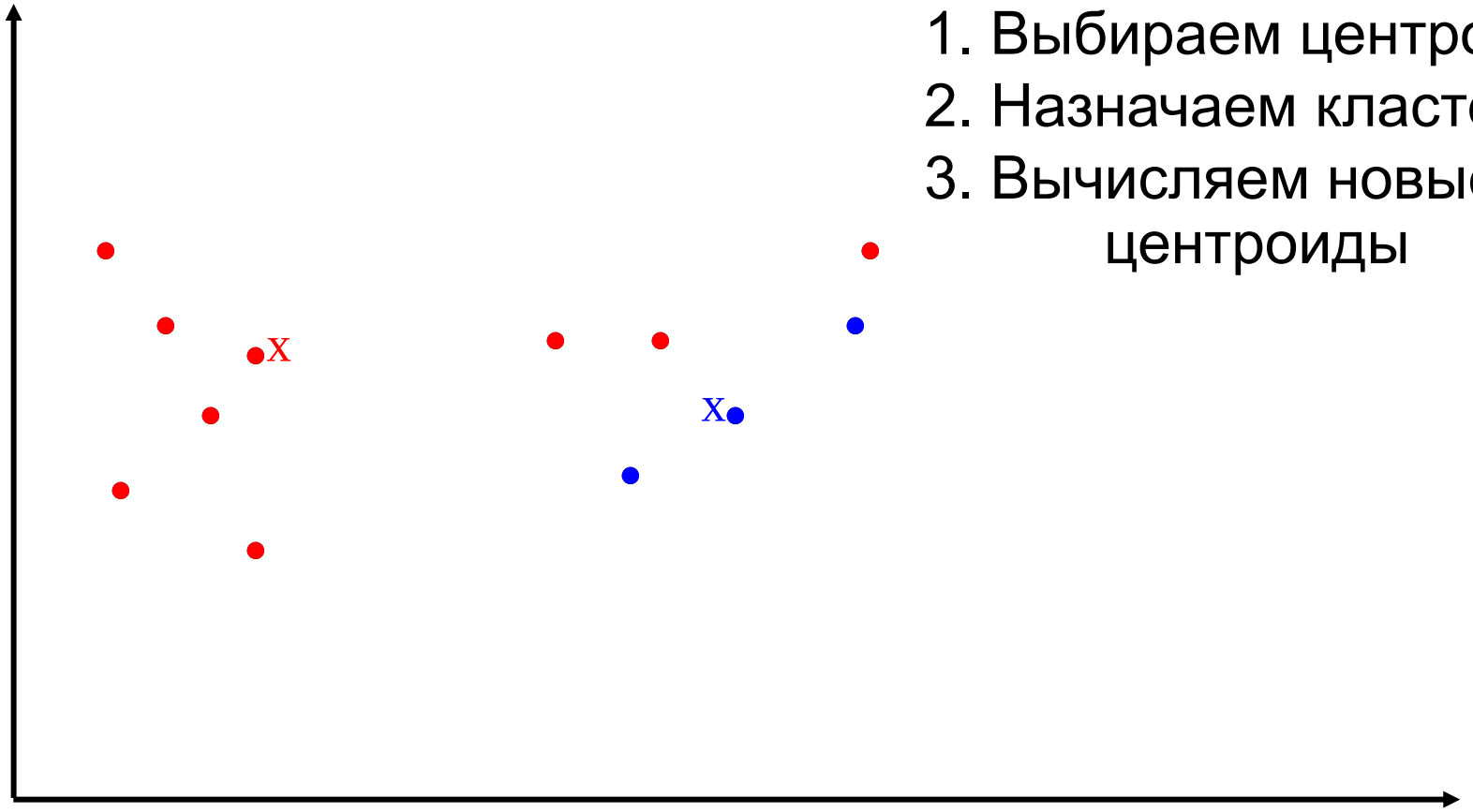
1. Выбираем центроиды (случайным образом)

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (2)



1. Выбираем центроиды
2. Назначаем кластеры

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (3)

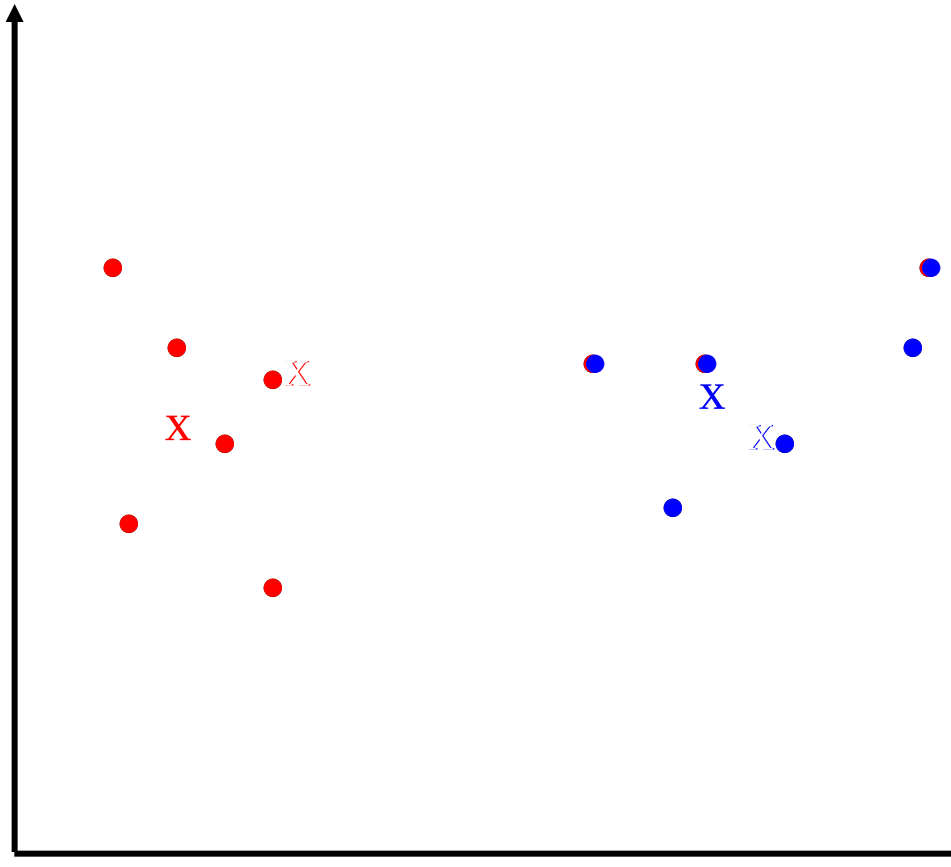


1. Выбираем центроиды
2. Назначаем кластеры
3. Вычисляем новые центроиды

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (4)

1. Выбираем центроиды
2. Назначаем кластеры
3. Вычисляем новые центроиды
4. Переназначаем кластеры

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (5)



1. Выбираем центроиды
2. Назначаем кластеры
3. Вычисляем новые центроиды
4. Переназначаем кластеры
5. Вычисляем новые центроиды

ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (6)

1. Выбираем центроиды
2. Назначаем кластеры
3. Вычисляем новые центроиды
4. Переназначаем кластеры
5. Вычисляем новые центроиды
6. Переназначаем кластеры

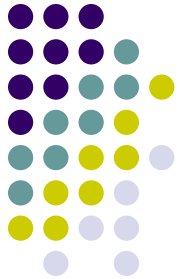
ИЛЛЮСТРАЦИЯ РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ (7)

Центроиды классов не изменились ➔ завершение работы

АЛГОРИТМ *k*-СРЕДНИХ: ПРИМЕР

№	Термины в документе	c=«Китай»
1	китайский пекин китайский	*c*
2	китайский китайский шанхай	*c*
3	китайский макао	*c*
4	токио япония китайский	$\neg c$
5	китайский китайский китайский токио япония	$\neg c$
6	токио пекин	

ПРИМЕР РАБОТЫ АЛГОРИТМА *k*-СРЕДНИХ



Итерация 1. Случайным образом инициализированы μ_i:

μ_1=[0,96 0,80 0,42 0,79 0,66 0,85] μ_2=[0,49 0,14 0,91 0,96 0,04 0,93]

dist	d_1	d_2	d_3	d_4	d_5	d_6
μ_1	**1,55**	1,81	1,66	**1,51**	**1,38**	**0,85**
μ_2	1,82	**1,38**	**1,37**	1,74	1,59	0,93

➔ c_1:={d_1, d_4, d_5, d_6}, c_2:={d_2, d_3}

Итерация 2.

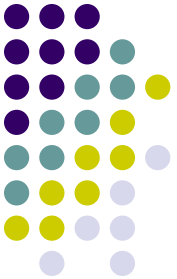
μ_1=[0,24 0,45 0 0 0,43 0,35] μ_2=[0,16 0 0,49 0,49 0 0]

dist	d_1	d_2	d_3	d_4	d_5	d_6
μ_1	**0,74**	1,21	1,22	**0,68**	**0,61**	**0,67**
μ_2	1,18	**0,69**	**0,69**	1,21	1,18	1,24

➔ c_1:={d_1, d_4, d_5, d_6}, c_2:={d_2, d_3}

Разбиение не изменилось, условие остановки выполнено

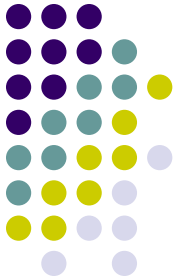
ИЕРАРХИЧЕСКИЕ АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ



- Восходящие: построение кластеров снизу вверх
  - Начало: один документ – один кластер
  - Последовательно объединяем пары кластеров
  - В итоге: один кластер – все документы
- Нисходящие: построение кластеров сверху вниз
  - Начало: все документы – один кластер
  - Рекурсивно делим кластеры пополам (с помощью алгоритма плоской кластеризации)
  - В итоге: один кластер – один документ

Создается структурированное множество кластеров: история объединения/деления кластеров дает их иерархию (бинарное дерево)

ВОСХОДЯЩИЕ АЛГОРИТМЫ: КРИТЕРИИ ОБЪЕДИНЕНИЯ



Сходство двух кластеров есть:

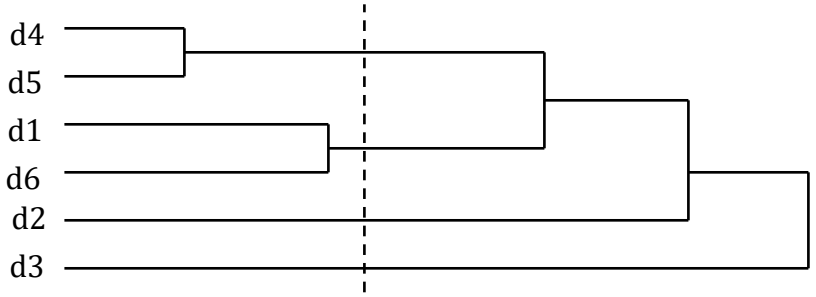
- сходство между их наиболее похожими документами (одиночная связь)
  - ✓ создаются протяженные кластеры
  - ✓ не учитывается вся структура кластера
- сходство между их наиболее непохожими документами (полная связь)
  - ✓ создаются компактные кластеры
  - ✓ учитывается вся структура кластера
- среднее сходство всех пар документов (групповое усреднение)
- сходство между их центроидами

ПРИМЕР ПРИМЕНЕНИЯ ВОСХОДЯЩИХ АЛГОРИТМОВ

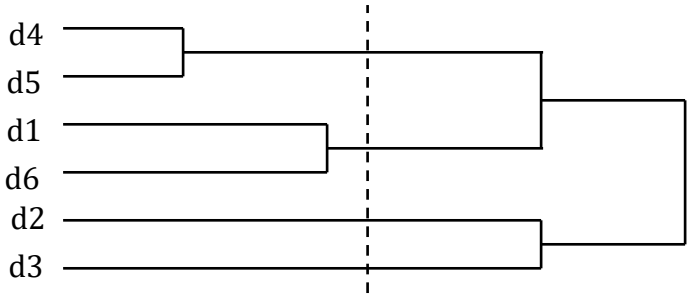
Матрица расстояний:

sim	d_1	d_2	d_3	d_4	d_5	d_6
d_1	0					
d_2	1,36	0				
d_3	1,37	1,39	0			
d_4	1,36	1,39	1,40	0		
d_5	1,32	1,36	1,38	0,27	0	
d_6	0,66	1,43	1,41	1,30	1,21	0

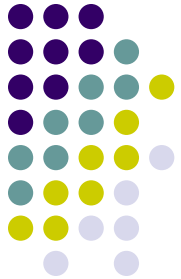
Одиночная связь



Полная связь



ОСОБЕННОСТИ ЗАДАЧИ КЛАСТЕРИЗАЦИИ

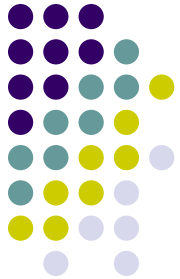


Решение задачи кластеризации принципиально неоднозначно:

- Не существует однозначно наилучшего критерия качества кластеризации
- Часто количество кластеров заранее неизвестно
- Результат кластеризации существенно зависит от того, как определяется схожесть
- Нет общепризнанного оптимального алгоритма
- Нет общепризнанных тестовых данных

Главное основание для выбора алгоритма – знание теоретических характеристик метода и оценка пригодности для решения поставленной задачи

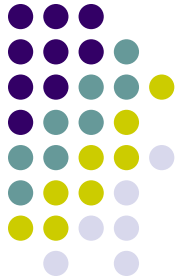
ОЦЕНКА КАЧЕСТВА КЛАСТЕРИЗАЦИИ



Вычисляются меры двух видов:

- Внешние меры: сравнение созданного разбиения с «эталонным»
  - анализируется сходство предсказаний экспертов и предсказаний системы относительно принадлежности каждой пары документов одному или разным кластерам
- Внутренние меры: анализ внутренних свойств
  - компактность: члены одного кластера должны быть близки друг другу
  - отделимость: кластеры должны достаточно далеко отстоять друг от друга

КАЧЕСТВО КЛАСТЕРИЗАЦИИ: ВНЕШНИЕ МЕРЫ



- *Rand Index* оценивает, насколько много из тех пар объектов, которые были в одном классе, и тех пар объектов, которые находились в разных классах, сохранили это состояние после кластеризации алгоритмом:

$$RandIndex = (TP+TN) / (TP+TN+FP+FN)$$

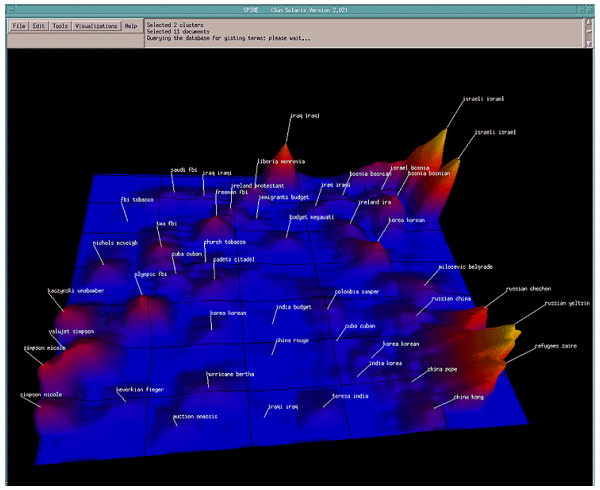
- Индекс Жаккара (*Jaccard Index*) похож на *Rand Index*, только не учитывает пары элементов находящиеся в разные классах и разных кластерах (TN)

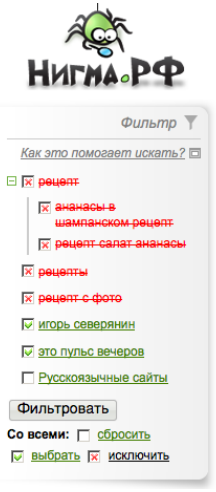
$$JaccardInd = (TP+TN) / (TP+FP+FN)$$

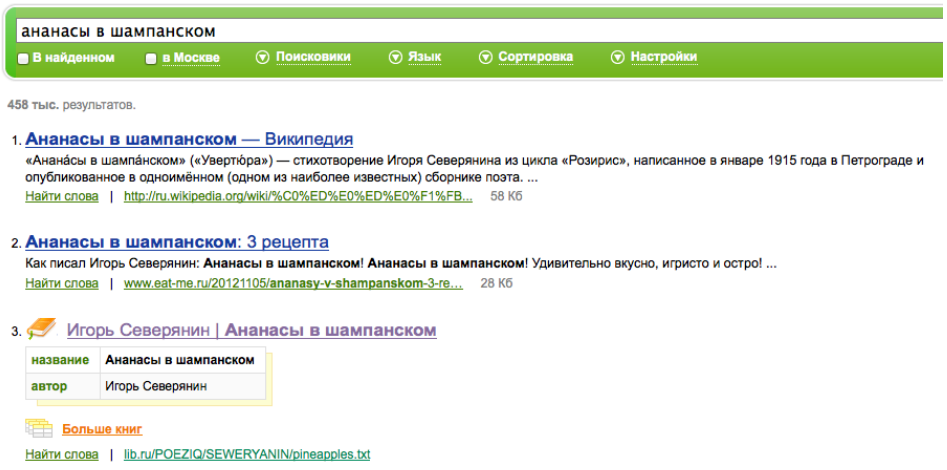
- Значения обоих мер − от 0 до 1, где 1 означает полное совпадение кластеров с заданными классами, а 0 − отсутствие совпадений

КЛАСТЕРИЗАЦИЯ: ПРИКЛАДНЫЕ ЗАДАЧИ

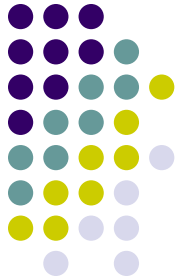
- Упорядочивание набора документов
- Навигация по набору документов
- Информационный поиск:
  – улучшение результатов поиска
  – «интеллектуальная» группировка результатов







ЗАКЛЮЧЕНИЕ: ВОПРОСЫ



Методы классификации и кластеризации текстов – одна из наиболее разработанных областей КЛ и информационного поиска

- *Какие еще могут быть признаки текстов?*
- *В чем отличие области применимости рассмотренных методов классификации от методов HMM и CRF*

СПАСИБО ЗА ВНИМАНИЕ

ДОМАШНЕЕ ЗАДАНИЕ № 2

На выбор 5 вариантов:

A. На базе интерфейса НКРЯ, словарей и др. ресурсов исследовать временные изменения смысла выбранного слова/слов и частоты его/их употребления
B. Для уже существующей N-граммной модели рассчитать вероятности нескольких фраз и перплексию
C. Построить свою N-граммную модель и вычислить по ней вероятности нескольких фраз и перплексию (дополни-тельно: сравнение двух N-граммных моделей)
D. Провести исследование явления "разреженности данных" в коллекциях/корпусах текстов
E. Провести анализ того, как явление "разреженности данных" в коллекциях/корпусах текстов отражается в предсказательных языковых моделях типа *Word2Vec*

Срок выполнения – до 12 марта включительно