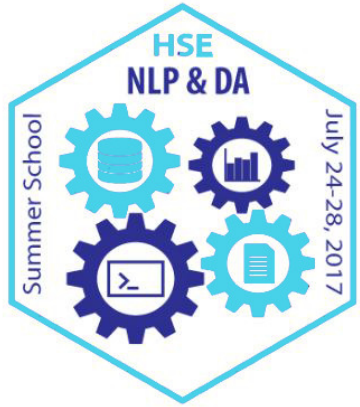
Методы анализа данных: машинное обучение в анализе текстов

Константин Вячеславович Воронцов

ФИЦ ИУ РАН • МФТИ • ВШЭ • МГУ • Яндекс • Форексис • Айтея



Москва • НИУ ВШЭ • 26 июля 2017

Содержание

Машинное обучение — новый двигатель прогресса

«Четвёртая технологическая революция строится на вездесущем и мобильном Интернете, искусственном интеллекте и машинном обучении» (2016)

Клаус Мартин Шваб, президент Всемирного экономического форума

Мир наконец поверил в искусственный интеллект?...
Машинное обучение изменит мир? Или уже меняет?

Бум искусственного интеллекта и нейронных сетей

1997 IBM Deep Blue обыграл чемпиона мира по шахматам

2005 Беспилотный автомобиль: DARPA Grand Challenge

2006 Google Translate – статистический машинный перевод

2011 40 лет DARPA CALO привели к созданию Apple Siri

2011 IBM Watson победил в ТВ-игре «Jeopardy!»

2011–2015 ImageNet: 25% $\to$ 3.5% ошибок против 5% у людей

2012 Google X Lab: распознавание видеокадров с котами

2014 Facebook DeepFace распознаёт лица с точностью 97%

2015 Фонд OpenAI в \$1 млрд. Илона Маска и Сэма Альтмана

2016 DeepMind, OpenAI: динамическое обучение играм Atari

2016 Google DeepMind обыграл чемпиона мира по игре го

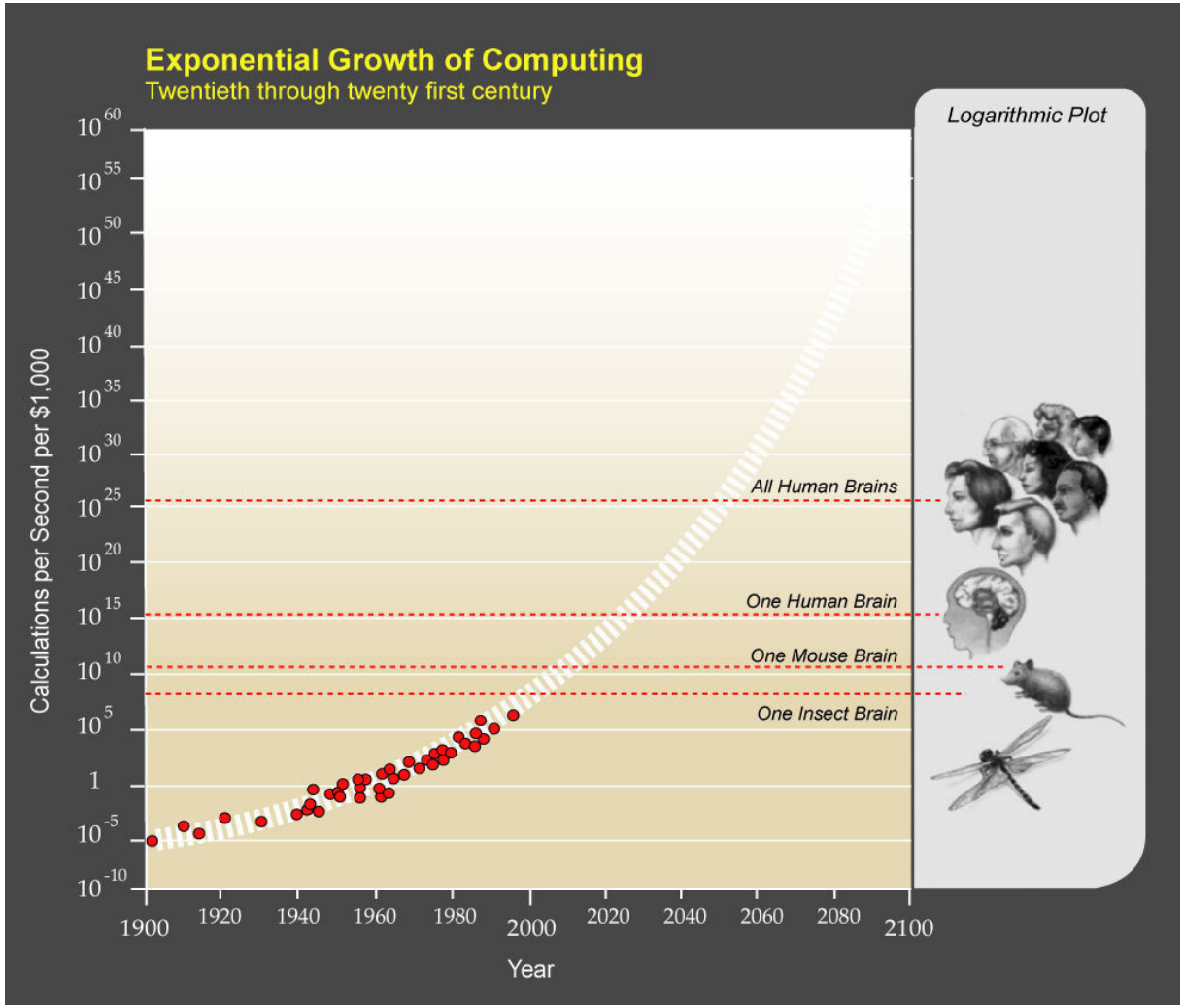
---

http://abv24.com/istoriya-mashinnogo-obucheniya

Три перехода количества в качество в нейронных сетях

1. **Достижения микроэлектроники**
   — процессоры, память, графические карты
   — рост вычислительных мощностей по закону Мура
   — экстраполяция: $80 \cdot 10^9$ нейронов в 2035–2050 гг.
2. **Повсеместное проникновение IT-технологий**
   — доступность средств накопления больших данных
   — краудсорсинг (пример ImageNet)
3. **Развитие методов обучения нейронных сетей**
   — rectified linear unit, ReLU (Nair & Hinton, 2010)
   — быстрые SGD алгоритмы: AdaGrad (Duchi, 2011), RmsProp (Hinton, 2012), AdaDelta (Kingma & Ba 2014)
   — SGD с моментумом Нестерова (Sutskever et al., 2013)
   — dropout (Hinton, 2012)
   — регуляризации

Закон Мура



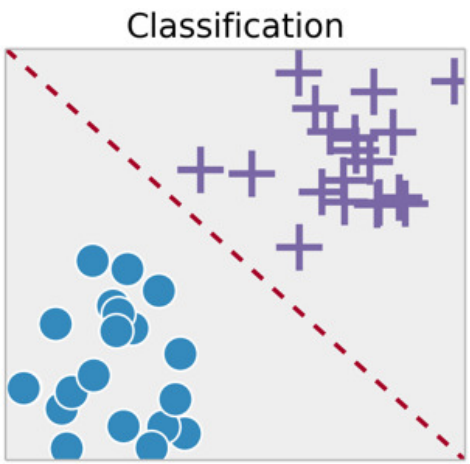
Ray Kurzweil. The singularity is near: When humans transcend biology. 2006.

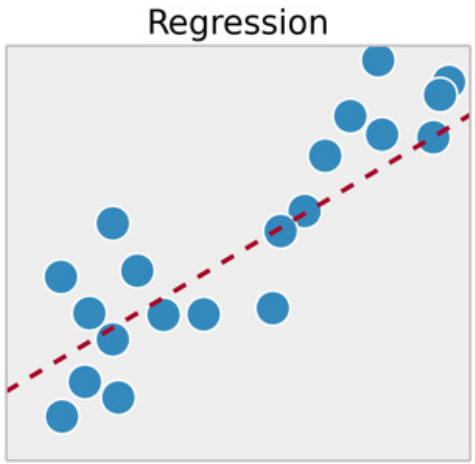
Машинное обучение — это...

- одна из ключевых технологий будущего
- наиболее успешное направление искусственного интеллекта, вытеснившее экспертные системы и инженерию знаний
- математическое моделирование в сложно формализуемых областях, когда данных много, знаний мало
- восстановление функций по заданным точкам в сложно устроенных пространствах
- сплав строгих математических методов, эвристик, IT-технологий и инженерного ремесла на грани искусства
- не только нейросети
- тысячи алгоритмов на стыке математической статистики и численных методов оптимизации
- около 100 000 научных публикаций в год

Типология задач машинного обучения

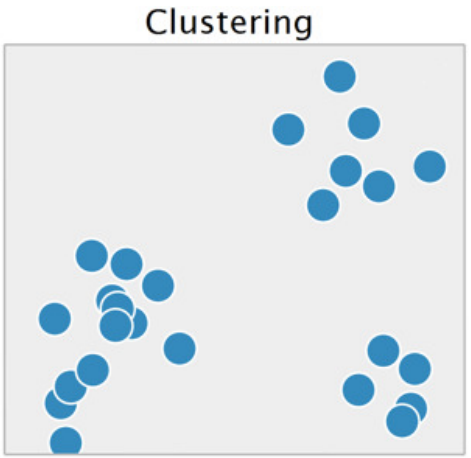
1. Обучение с учителем (supervised learning)
   - классификация (classification)
   - регрессия (regression)
   - прогнозирование (forecasting)
   - ранжирование (learning to rank)





Типология задач машинного обучения

2. Обучение без учителя (unsupervised learning)
   - кластеризация (clustering)
   - поиск ассоциативных правил (association rule learning)
   - восстановление плотности (density estimation)
3. Частичное обучение (semi-supervised learning)
   - трансдуктивное обучение (transductive learning)
   - одноклассовая классификация (one-class classification)
   - обучение с положительными примерами (PU-learning)



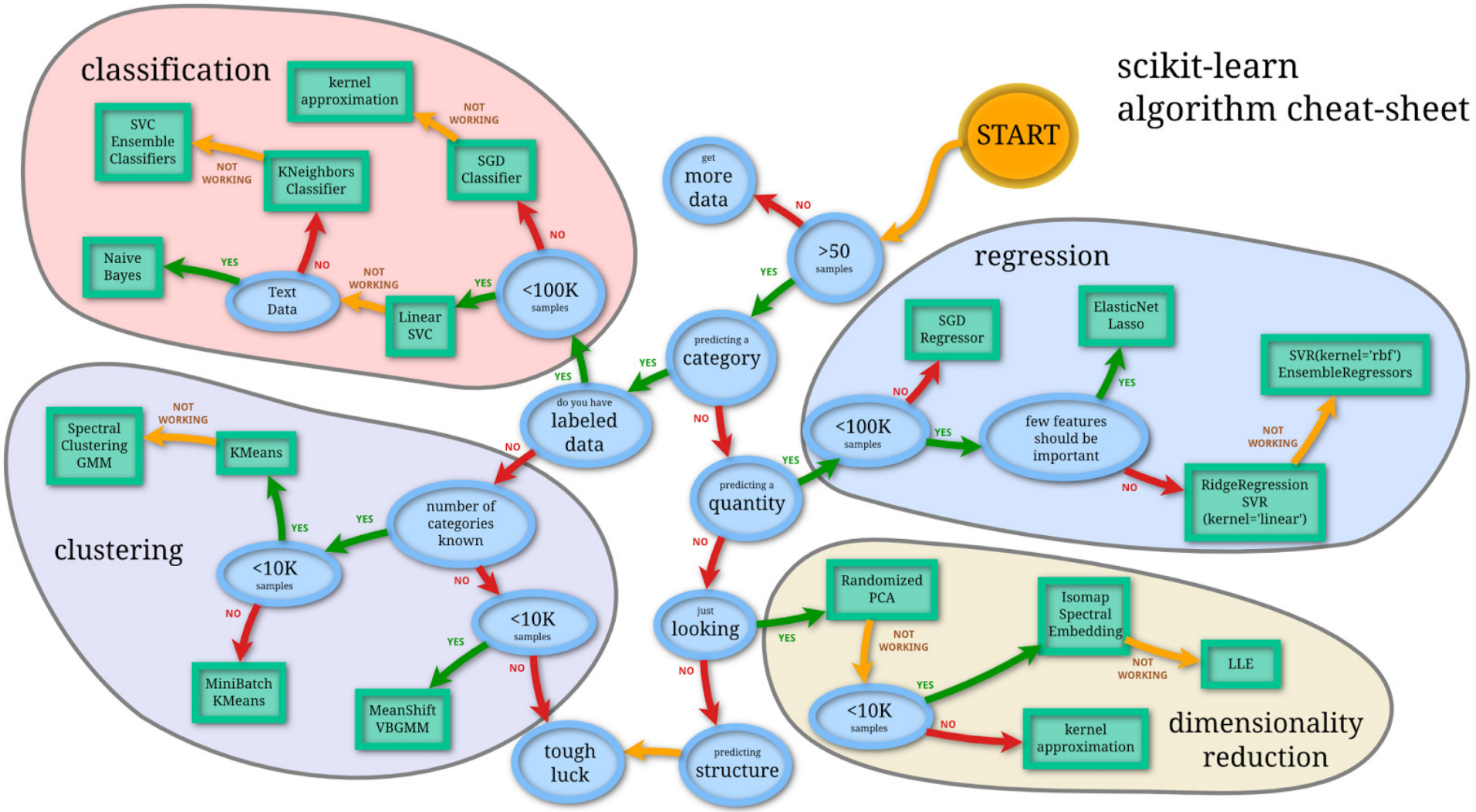
Типология задач машинного обучения

4. Предварительная обработка (data preparation)
   - извлечение признаков (feature extraction)
   - отбор признаков (feature selection)
   - восстановление пропусков (missing values)
   - отсев выбросов (outlier detection)
5. Сокращение размерности (dimensionality reduction)
   - анализ главных компонент (principal component analysis)
   - матричные разложения (matrix factorization)
   - тематическое моделирование (topic modeling)
6. Обучение представлений (representation learning)
   - обучение признаков (feature learning)
   - глубокое обучение (deep learning)
   - обучение многообразий (manifold learning)

Типология задач машинного обучения

7. Обучение функций близости (similarity learning)
8. Обучение выявлению связей (relational learning)
9. Привилегированное обучение (privilege learning)
10. Динамическое обучение (online/incremental learning)
11. Обучение с подкреплением (reinforcement learning)
12. Активное обучение (active learning)
13. Перенос опыта обучения (transfer learning)
14. Многозадачное обучение (multitask learning)
15. Мета-обучение (meta-learning)

Задачи и методы машинного обучения в Python scikit-learn



Задачи обучения с учителем

Восстановление зависимости $y(x)$ по точкам (x_i, y_i), $i = 1, \ldots, \ell$.

Дано: выборка ℓ объектов $x_i = (f_1(x_i), \ldots, f_n(x_i))$ и ответов y_i, $f_j(x)$ — признаки объекта x, $j = 1, \ldots, n$

$$\begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix} \xrightarrow{y} \begin{pmatrix} y_1 \\ \ldots \\ y_\ell \end{pmatrix}$$

Найти: функцию $a(x)$, способную давать правильные ответы на *тестовых объектах* $x'_i = (f_1(x'_i), \ldots, f_n(x'_i))$, $i = 1, \ldots, k$:

$$\begin{pmatrix} f_1(x'_1) & \ldots & f_n(x'_1) \\ \ldots & \ldots & \ldots \\ f_1(x'_k) & \ldots & f_n(x'_k) \end{pmatrix} \xrightarrow{a?} \begin{pmatrix} a(x'_1) \\ \ldots \\ a(x'_k) \end{pmatrix}$$

Задача категоризации текстовых документов

Объект — текстовый документ.

Классы — рубрики иерархического тематического каталога.

Примеры признаков:

- **количественные:**
  частота слова / n-граммы / коллокации / термина
  в тексте / в заголовках / в аннотации
- **номинальные:** автор, издание, год, и т. д.

Особенности задачи:

- лишь небольшая часть документов имеют метки y_i;
- документ может относиться к нескольким рубрикам;
- в каждом ребре дерева свой классификатор на 2 класса.

Примеры задач регрессии на текстах

MovieReview [Pang, Lee, 2005]

x_i — текст отзыва на фильм

y_i — рейтинг фильма (1..5), поставленный автором отзыва

Salary (kaggle.com: *Adzuna Job Salary Prediction*)

x_i — описание вакансии, предлагаемой работодателем

y_i — годовая зарплата

Yelp (kaggle.com: *Yelp Recruiting Competition*)

x_i — отзыв (на ресторан, отель, сервис и т.п.)

y_i — число голосов «useful», которые получит отзыв

Прогнозирование скачков цен на финансовых рынках

x_i — текст новости

y_i — изменение цены в последующие 10–60 минут

B. Pang, L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales // ACL, 2005.

Конкурс kaggle.com: Avito Context Ad Clicks Prediction

Объект — тройка ⟨пользователь, запрос, объявление⟩.

Предсказать вероятность клика по контекстной рекламе, показанной в ответ на запрос пользователя на avito.ru.

Сырые данные:

- тексты запросов и объявлений
- история показов и кликов пользователей по баннерам
- профиль пользователя (бразуер, устройство и т. д.),
- . . . всего 10 таблиц данных.

Особенности задачи:

- признаки надо придумывать;
- данных много — сотни миллионов показов;
- основной критерий качества — доход рекламной площадки;
- несколько дополнительных критериев и ограничений.

Обучение восстановлению регрессии — задача оптимизации

Задача восстановления регрессионной зависимости, $y_i \in \mathbb{R}$

1. Выбираем *модель регрессии*, например, линейную:
$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^{n} f_j(x) w_j, \qquad x, w \in \mathbb{R}^n$$
2. Выбираем функцию потерь, например, квадратичную:
$$\mathscr{L}(a, y) = (a - y)^2$$
3. Минимизируем потери *методом наименьших квадратов*:
$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \big(a(x_i, w) - y_i\big)^2 \to \min_w$$
4. Проверяем прогностическую (обобщающую) способность:
$$\tilde{Q}(w) = \frac{1}{k} \sum_{i=1}^{k} \big(a(\tilde{x}_i, w) - \tilde{y}_i\big)^2$$

Обучение классификации — тоже задача оптимизации

Задача классификации, $y_i \in \{-1, +1\}$

1. Выбираем *модель классификации*, например, линейную:
$$a(x, w) = \text{sign}\langle x, w\rangle$$
2. Выбираем функцию потерь, например, *бинарную*:
$$\mathscr{L}(a, y) = \big[a(x_i, w)y_i < 0\big]$$
3. Минимизируем *частоту ошибок* на обучающей выборке:
$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}\big[a(x_i, w)y_i < 0\big] \to \min_w$$
4. Проверяем прогностическую (обобщающую) способность:
$$\tilde{Q}(w) = \frac{1}{k}\sum_{i=1}^{k}\big[a(\tilde{x}_i, w)\tilde{y}_i < 0\big]$$

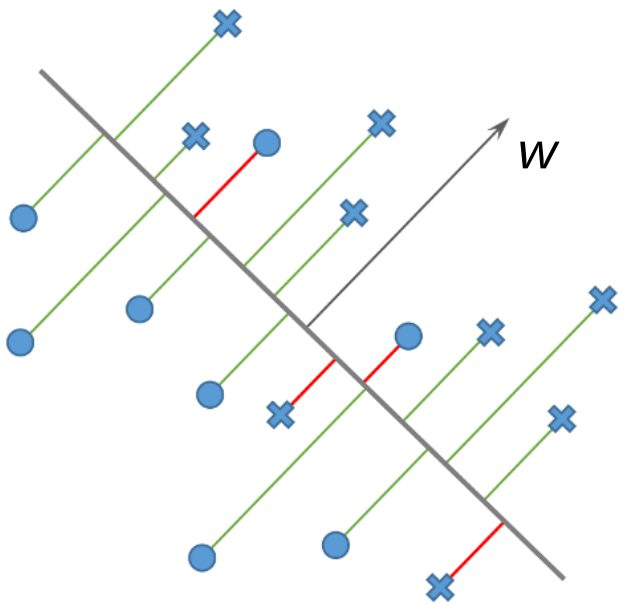
Отступ объекта и его геометрический смысл

w — нормаль (направляющий вектор) разделяющей гиперплоскости, направлена в сторону класса $+1$

$\langle x_i, w \rangle$ — проекция вектора x_i на вектор нормали w

$M_i = \langle x_i, w \rangle y_i$ — *отступ* (margin) объекта x_i

$M_i < 0 \Leftrightarrow$ классификатор ошибается на объекте x_i



- — объекты класса $+1$
- — объекты класса -1
- — нет ошибки, $M_i > 0$
- — ошибка, $M_i < 0$

Чем меньше M_i, тем хуже

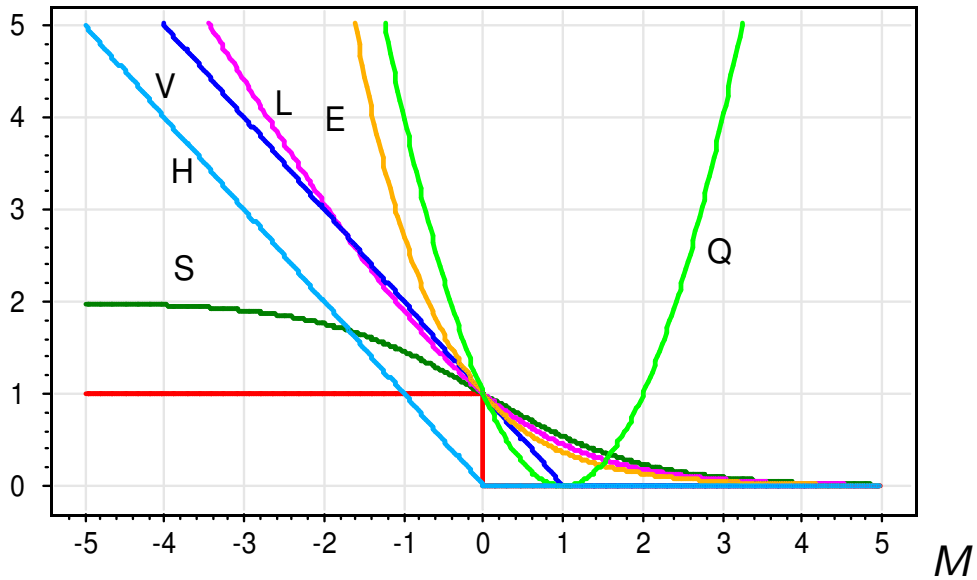
Обучение классификации — оптимизации верхней оценки

Задача классификации, $y_i \in \{-1, +1\}$

1. Выбираем *модель классификации*, например, линейную:
$$a(x, w) = \text{sign}\langle x, w\rangle$$
2. Мажорируем пороговую функцию потерь непрерывной:
$$\big[M_i < 0\big] \leqslant \mathscr{L}\big(M_i\big), \qquad M_i = \langle x_i, w\rangle y_i \;\; — \;\; \text{отступ (margin)}$$
3. Минимизируем *сглаженную частоту ошибок*:
$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) \to \min_w$$
4. Проверяем прогностическую (обобщающую) способность:
$$\tilde{Q}(w) = \frac{1}{k}\sum_{i=1}^{k} \big[\langle \tilde{x}_i, w\rangle \tilde{y}_i < 0\big]$$

Непрерывные аппроксимации пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathscr{L}(M)$:



$V(M) = (1 - M)_+$ — кусочно-линейная (SVM)
$H(M) = (-M)_+$ — кусочно-линейная (Hebb's rule)
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR, Logistic Regression)
$Q(M) = (1 - M)^2$ — квадратичная (Fisher's Linear Discriminant)
$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная (ANN, Artificial Neural Network)
$E(M) = e^{-M}$ — экспоненциальная (AdaBoost)
$[M < 0]$ — пороговая функция потерь.

Общие подходы к решению оптимизационных задач

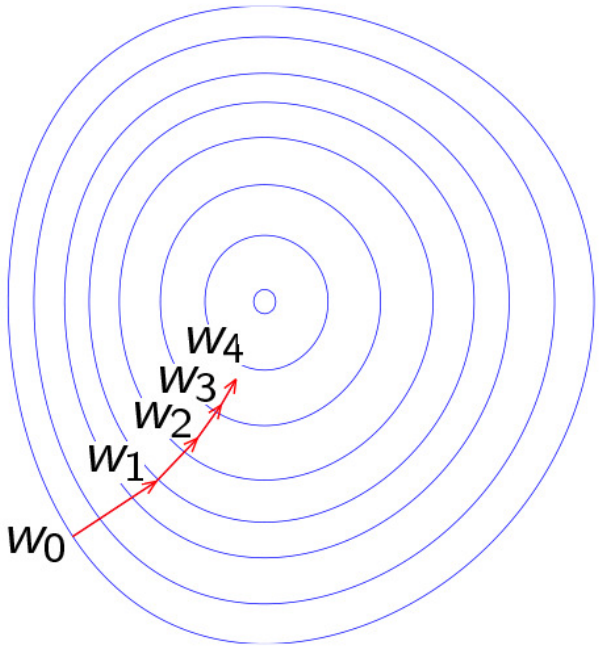
Аналитический подход (напр. метод наименьших квадратов):
Если w — точка минимума *гладкой* функции $Q(w)$, то

$$\frac{\partial Q(w)}{\partial w_j} = 0, \quad j = 1, \ldots, n.$$

Это система n уравнений с n неизвестными.

Численный метод — градиентный спуск:

1 начальное приближение w^0, $t := 0$;
2 **повторять**
3 $\quad w_j^{t+1} := w_j^t - h^t \cdot \frac{\partial Q(w^t)}{\partial w_j}$, $j = 1, \ldots, n$;
4 $\quad t := t + 1$;
5 **пока** процесс не сойдётся;



Причины переобучения линейных моделей

1. слишком мало объектов; слишком много признаков;
2. линейная зависимость (мультиколлинеарность) признаков:

   пусть построен классификатор: $a(x, w) = \text{sign}\langle x, w\rangle$;
   мультиколлинеарность: $\exists v \in \mathbb{R}^n$: $\forall x \;\; \langle x, v\rangle \approx 0$;
   тогда $\forall \gamma \in \mathbb{R} \quad a(x, w) \approx \text{sign}\langle x, w + \gamma v\rangle$

Последствия:

- решение неединственно и неустойчиво;
- появляются слишком большие веса $|w_j| \to \infty$;
- $Q(w)$ на обучении существенно меньше, чем на контроле;

Спасает *регуляризация* — введение дополнительного критерия:

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) + \sum_{j=1}^{n} w_j^2 \;\to\; \min.$$

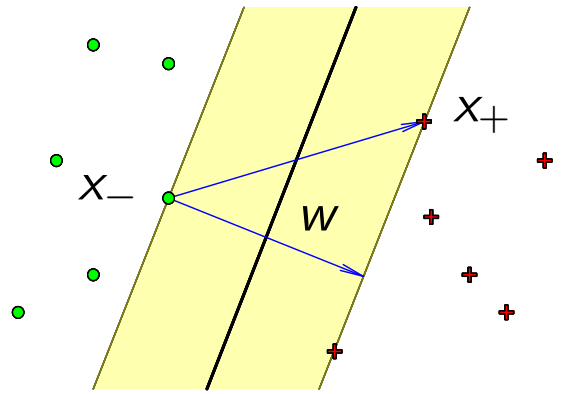
SVM — метод опорных векторов

Линейный классификатор:

$$a(x) = \operatorname{sign}\big(\langle x, w\rangle - w_0\big), \quad w, x \in \mathbb{R}^n,\ w_0 \in \mathbb{R}.$$

Задача обучения SVM:

$$\begin{cases} \frac{1}{2}\sum\limits_{j=1}^{n} w_j^2 + C\sum\limits_{i=1}^{\ell}\xi_i \to \min\limits_{w,w_0,\xi}; \\ M_i \geqslant 1 - \xi_i, \quad i = 1,\ldots,\ell; \\ \xi_i \geqslant 0, \quad i = 1,\ldots,\ell. \end{cases}$$



где $M_i = \big(\langle x_i, w\rangle - w_0\big)y_i$ — отступ объекта x_i.

Эквивалентная задача безусловной минимизации:

$$Q(w, w_0) = \sum_{i=1}^{\ell}\big(1 - M_i\big)_+ + \frac{1}{2C}\sum_{j=1}^{n} w_j^2 \to \min_{w,w_0}.$$

Задача ранжирования: определения и обозначения

X — множество объектов
$X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка
$i \prec j$ — правильный порядок на парах $(i, j) \in \{1, \ldots, \ell\}^2$

Задача:
построить ранжирующую функцию $a\colon X \to \mathbb{R}$ такую, что

$$i \prec j \quad \Rightarrow \quad a(x_i) < a(x_j)$$

Линейная модель ранжирования:

$$a(x; w) = \langle x, w \rangle$$

где $x \mapsto \big(f_1(x), \ldots, f_n(x)\big) \in \mathbb{R}^n$ — вектор признаков объекта x

Задача ранжирования поисковой выдачи

D — коллекция текстовых документов (documents)
Q — множество запросов (queries)
$D_q \subseteq D$ — множество документов, найденных по запросу q
$X = Q \times D$ — объектами являются пары «запрос, документ»:

$$x \equiv (q, d),\ q \in Q,\ d \in D_q$$

Y — упорядоченное множество рейтингов
$y : X \to Y$ — оценки релевантности, поставленные асессорами:
чем выше оценка $y(q, d)$, тем релевантнее документ d запросу q

Правильный порядок определён только между документами, найденными по одному и тому же запросу q:

$$(q, d) \prec (q, d') \quad \Leftrightarrow \quad y(q, d) < y(q, d')$$

Типы признаков в задаче ранжирования поисковой выдачи

- функции только документа d
- функции только запроса q
- функции запроса и документа (q, d)

- текстовые
  — слова запроса q встречаются в d чаще обычного
  — слова запроса q есть в заголовках или выделены в d
- ссылочные
  — на документ d много ссылаются
  — документ d содержит много полезных ссылок
- кликовые
  — на документ d часто кликают
  — на документ d часто кликают по запросу q

TF-IDF(q,d) — мера релевантности документа d запросу q

n_{dw} (term frequency) — число вхождений слова w в текст d;
N_w (document frequency) — число документов, содержащих w;
N — число документов в коллекции D;

N_w/N — оценка вероятности встретить слово w в документе;
$(N_w/N)^{n_{dw}}$ — оценка вероятности встретить его n_{dw} раз;
$P(q,d) = \prod\limits_{w \in q} (N_w/N)^{n_{dw}}$ — оценка вероятности встретить в документе d слова запроса $q = \{w_1, \ldots, w_k\}$ *чисто случайно*;

Оценка релевантности запроса q документу d:

$$-\log P(q,d) = \sum_{w \in q} \underbrace{n_{dw}}_{\mathrm{TF}(w,d)} \underbrace{\log(N/N_w)}_{\mathrm{IDF}(w)} \;\to\; \max.$$

$\mathrm{TF}(w,d) = n_{dw}$ — term frequency;
$\mathrm{IDF}(w) = \log(N/N_w)$ — inverted document frequency.

Основные подходы к ранжированию

- Point-wise — поточечный
- Pair-wise — попарный
- List-wise — списочный

Переход к гладкому функционалу качества ранжирования:

$$Q(a) = \sum_{i \prec j} \big[\underbrace{a(x_j) - a(x_i)}_{\text{Margin}(i,j)} < 0\big] \leqslant \sum_{i \prec j} \mathscr{L}\big(a(x_j) - a(x_i)\big) \quad \to \quad \min$$

где $a(x)$ — функция ранжирования;
$\mathscr{L}(M)$ — убывающая непрерывная функция отступа $\text{Margin}(i,j)$:

- $\mathscr{L}(M) = (1 - M)_+$ — RankSVM
- $\mathscr{L}(M) = \exp(-M)$ — RankBoost
- $\mathscr{L}(M) = \log(1 + e^{-M})$ — RankNet

Ranking SVM

Постановка задачи SVM для попарного подхода:

$$Q(a) = C\sum_{i \prec j} \mathscr{L}\big(\underbrace{a(x_j) - a(x_i)}_{\text{margin}(i,j)}\big) + \tfrac{1}{2}\|w\|^2 \quad \to \quad \min_a,$$

где $a(x) = \langle w, x\rangle$ — функция ранжирования,
$\mathscr{L}(M) = (1 - M)_+$ — функция потерь,
$M = \text{margin}(i,j) = \langle w, x_j - x_i\rangle$ — отступ,

Постановка задачи квадратичного программирования:

$$\begin{cases} \frac{1}{2}\sum\limits_{j=1}^{n} w_j^2 + C\sum\limits_{i \prec j} \xi_{ij} \to \min\limits_{w,\xi}; \\ \langle w, x_j - x_i\rangle \geqslant 1 - \xi_{ij}, \quad i \prec j; \\ \xi_{ij} \geqslant 0, \quad i \prec j. \end{cases}$$

Постановка задачи кластеризации

Дано:

$X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка;

$\rho(x_i, x_s)$ — функция расстояния между объектами.

Найти:

$y_i \in Y$ — метки кластеров объектов:

— каждый кластер состоит из близких объектов;

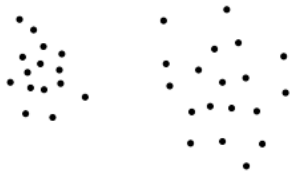
— объекты разных кластеров существенно различны.

Кластеризация — это *обучение без учителя*.

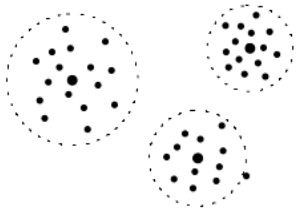
Решение задачи кластеризации принципиально неоднозначно:

- различные критерии качества кластеризации
- различные эвристические методы кластеризации
- различные варианты функции расстоянии ρ

Типы кластерных структур



внутрикластерные расстояния, как правило, меньше межкластерных



кластеры с центром

кластеры могут соединяться перемычками

кластеры могут накладываться на разреженный фон из редко расположенных объектов

Типы кластерных структур

ленточные кластеры

перекрывающиеся кластеры

кластеры могут образовываться не по сходству, а по иным типам регулярностей

кластеры могут вообще отсутствовать

Оценивание близости текстов

Векторные представления текстов $x = \big(f_1(x), \ldots, f_n(x)\big)$**:**

- векторы частот или TF-IDF слов
- сжатые векторные представления слов (word embedding)

Меры сходства текстовых документов:

- косинусная мера

$$\cos(x,u) = \frac{\sum_j f_j(x) f_j(u)}{\big(\sum_j f_j^2(x)\big)^{1/2} \big(\sum_j f_j^2(u)\big)^{1/2}}.$$

- мера Жаккара
- евклидово расстояние
- дивергенция Кульбака–Лейблера
- расстояние Йенсена–Шеннона...

Оптимизационная постановка задачи кластеризации

Дано:

векторы признаков объектов $x_i = \big(f_1(x_i), \ldots, f_n(x_i)\big)$

Найти:

1) кластеризации объектов $y_i \in Y = \{1, \ldots, k\}$;
2) центры k кластеров μ_{yj}, $y \in Y$, $j = 1, \ldots, n$.

Критерий:

минимизация среднего расстояния до центра кластера

$$\sum_{i=1}^{\ell} \underbrace{\sum_{j=1}^{n} \big(f_j(x_i) - \mu_{y_i j}\big)^2}_{\rho^2(x_i, \mu_{y_i})} \to \min_{\{\mu_y\}, \{y_i\}},$$

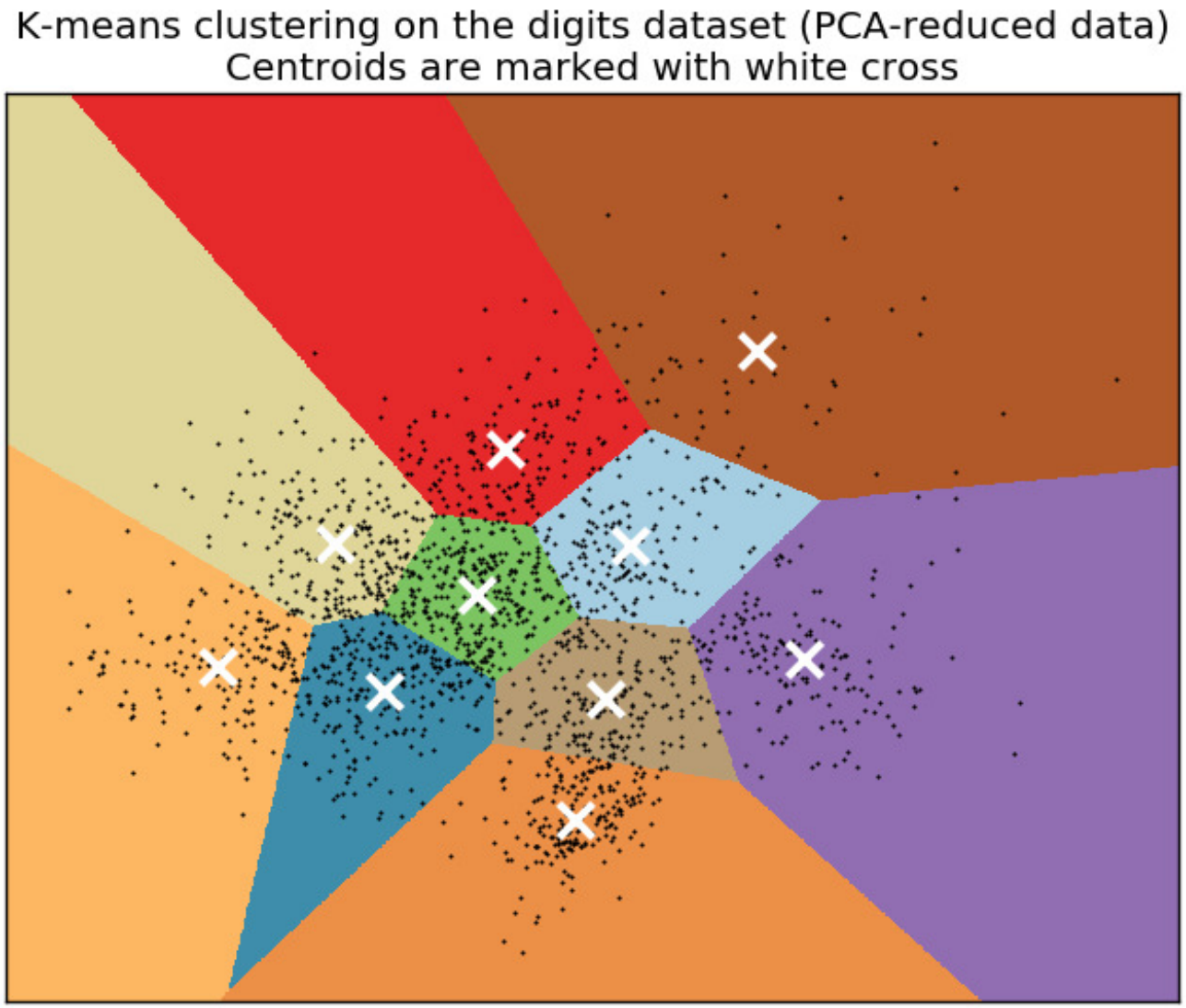
$\rho(x_i, \mu_{y_i})$ — евклидово расстояние между n-мерными векторами.

Метод k-средних (k-means, алгоритм Ллойда)

Вход: выборка векторов $x_i = \big(f_1(x_i), \ldots, f_n(x_i)\big)$, параметр k;
Выход: центры кластеров μ_y, $y \in Y$ и кластеризация $y_i \in Y$;

1 начальное приближение центров μ_y, $y \in Y$;
2 **повторять**
3 | отнести каждый x_i к ближайшему центру:
| $y_i := \arg\min\limits_{y \in Y} \rho(x_i, \mu_y)$, $i = 1, \ldots, \ell$;
4 | вычислить новые положения центров:
| $\mu_{yj} := \dfrac{\sum_{i=1}^{\ell} [y_i = y]\, f_j(x_i)}{\sum_{i=1}^{\ell} [y_i = y]}$, $y \in Y$, $j = 1, \ldots, n$;
5 **пока** y_i не перестанут изменяться;

Пример. Результат кластеризации методом k-средних



Агломеративная иерархическая кластеризация

Алгоритм Ланса-Уильямса [1967] основан на оценивании расстояний $R(U,V)$ между парами кластеров U, V.

сначала все кластеры одноэлементные: $C_1 := \{\{x_1\},\ldots,\{x_\ell\}\}$;
расстояния между ними: $R(\{x_i\},\{x_s\}) := \rho(x_i,x_s)$;
для всех $t = 2,\ldots,\ell$ (t — номер итерации):
    найти в C_{t-1} два ближайших кластера:
    $(U,V) := \arg\min\limits_{U\neq V} R(U,V)$;
    $R_t := R(U,V)$;
    слить их в один кластер $W := U \cup V$;
    $C_t := C_{t-1} \cup \{W\} \setminus \{U,V\}$;
    для всех $S \in C_t$
        вычислить $R(W,S)$ по формуле Ланса-Уильямса;

Формула Ланса-Уильямса

Как определить расстояние $R(W, S)$
между кластерами $W = U \cup V$ и S,
зная расстояния $R(U, S)$, $R(V, S)$, $R(U, V)$?

Формула, обобщающая большинство разумных способов определить это расстояние [Ланс, Уильямс, 1967]:

$$\begin{aligned} R(U \cup V, S) = {} & \alpha_U \cdot R(U, S) + \\ & + \alpha_V \cdot R(V, S) + \\ & + \beta \cdot R(U, V) + \\ & + \gamma \cdot \big|R(U, S) - R(V, S)\big|, \end{aligned}$$

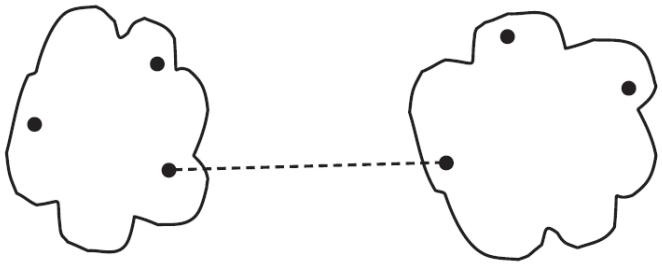
где α_U, α_V, β, γ — числовые параметры.

Частные случаи формулы Ланса-Уильямса

1. Расстояние ближнего соседа:

$$R^{б}(W,S) = \min_{w\in W, s\in S} \rho(w,s);$$

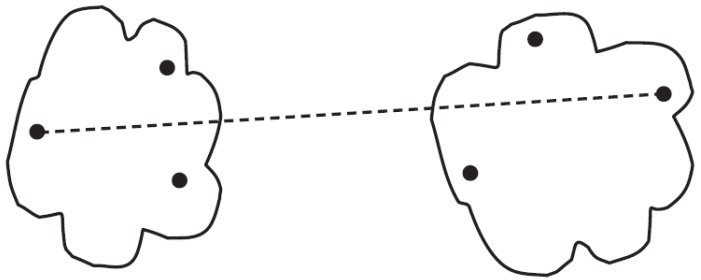
$$\alpha_U = \alpha_V = \tfrac{1}{2},\ \ \beta = 0,\ \ \gamma = -\tfrac{1}{2}.$$



2. Расстояние дальнего соседа:

$$R^{д}(W,S) = \max_{w\in W, s\in S} \rho(w,s);$$

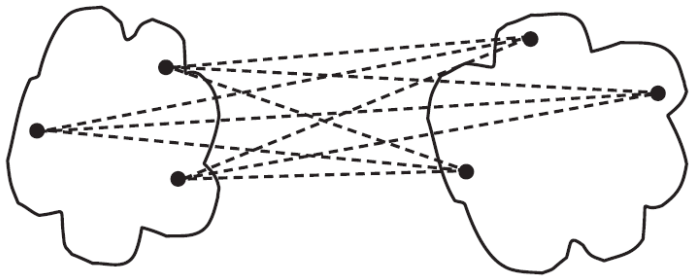
$$\alpha_U = \alpha_V = \tfrac{1}{2},\ \ \beta = 0,\ \ \gamma = \tfrac{1}{2}.$$



3. Групповое среднее расстояние:

$$R^{г}(W,S) = \frac{1}{|W||S|} \sum_{w\in W} \sum_{s\in S} \rho(w,s);$$

$$\alpha_U = \frac{|U|}{|W|},\ \ \alpha_V = \frac{|V|}{|W|},\ \ \beta = \gamma = 0.$$



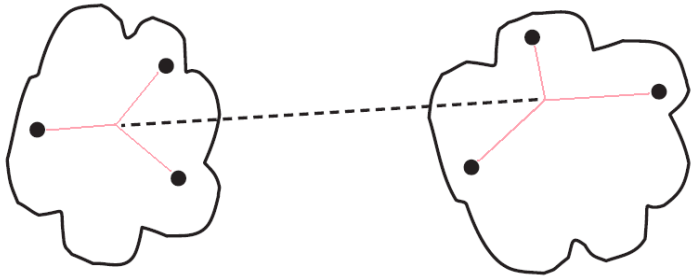
Частные случаи формулы Ланса-Уильямса

4. Расстояние между центрами:

$$R^{ц}(W,S) = \rho^2\Big(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\Big);$$

$$\alpha_U = \frac{|U|}{|W|}, \quad \alpha_V = \frac{|V|}{|W|},$$

$$\beta = -\alpha_U\alpha_V, \quad \gamma = 0.$$



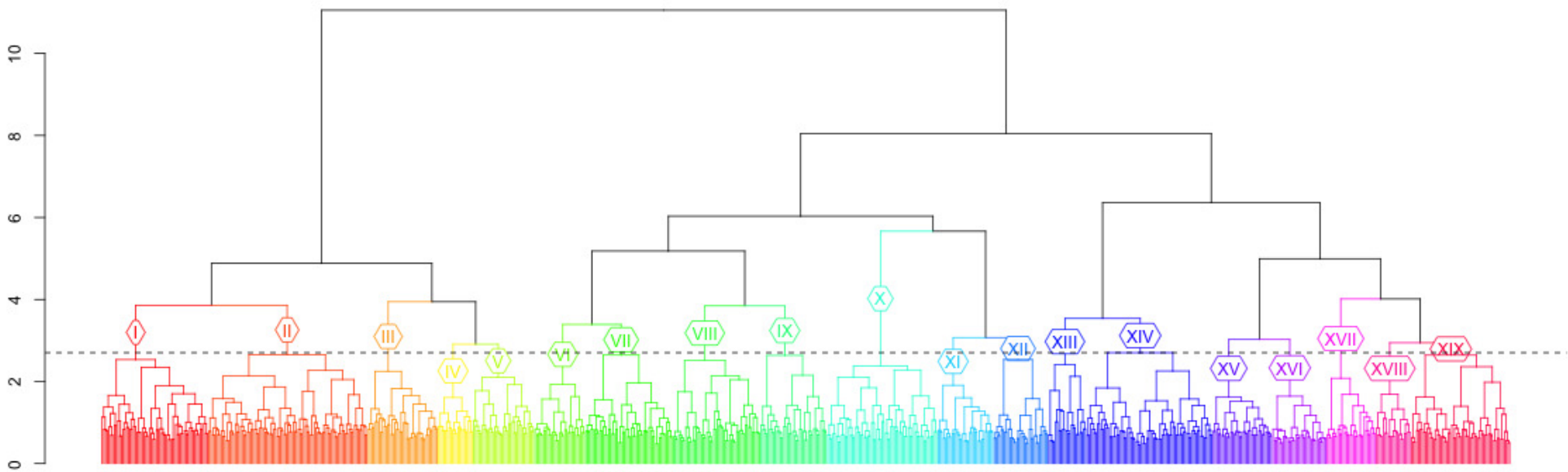
5. Расстояние Уорда — рекомендуется для использования

$$R^{у}(W,S) = \frac{|S||W|}{|S|+|W|}\,\rho^2\Big(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\Big);$$

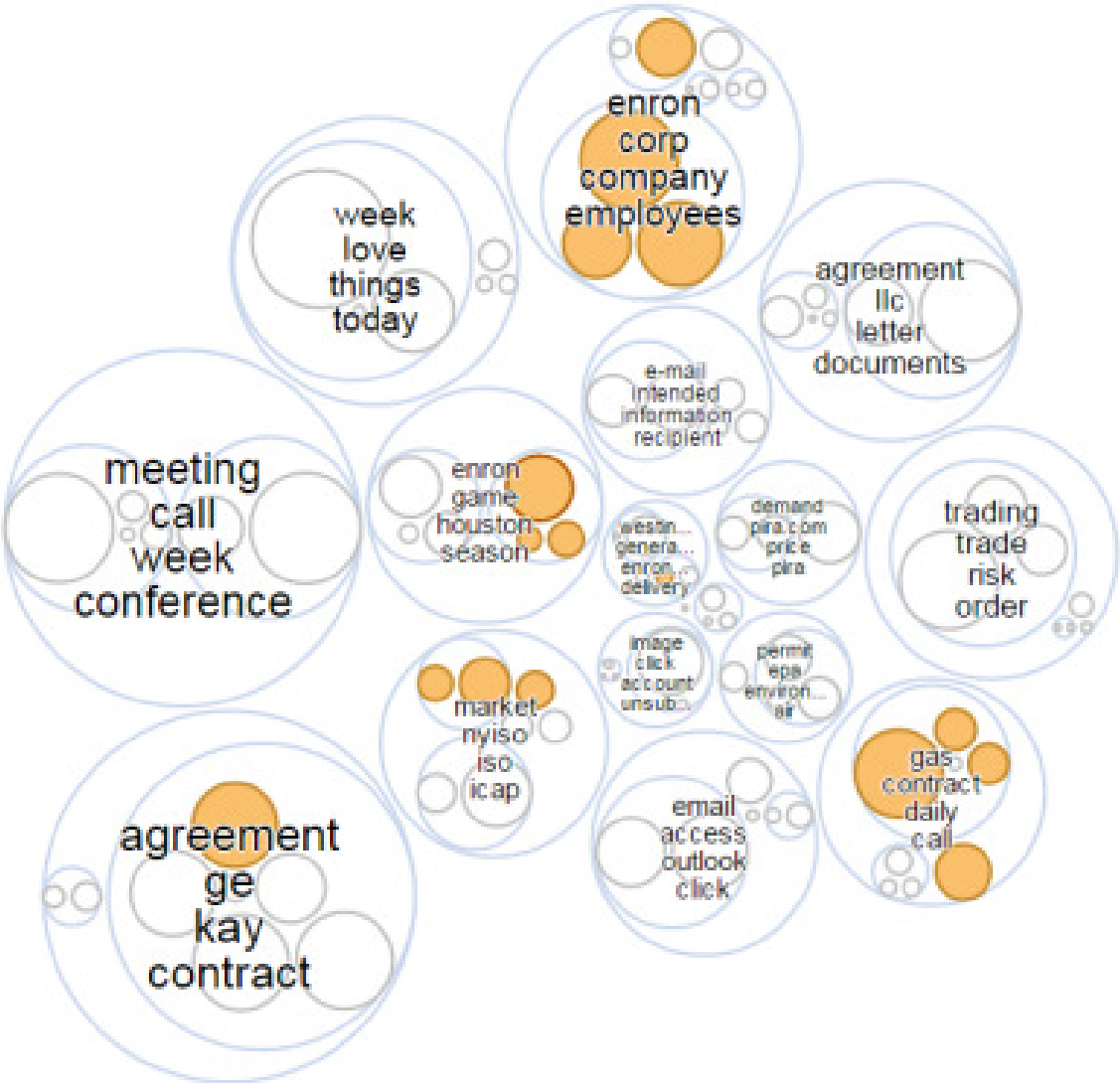
$$\alpha_U = \frac{|S|+|U|}{|S|+|W|}, \quad \alpha_V = \frac{|S|+|V|}{|S|+|W|}, \quad \beta = \frac{-|S|}{|S|+|W|}, \quad \gamma = 0.$$

Дендрограмма — визуализация иерархической кластеризации

- По вертикальной оси откладываются расстояния R_t
- Уровень отсечения определяет число кластеров
- Дендрограмма не имеет самопересечений и группирует объекты в кластеры вдоль горизонтальной оси



Ещё один способ визуализации иерархической кластеризации



Задача автоматического выделения терминов

Термин — фраза (*n*-грамма) со следующим набором свойств:

1. *высокая частотность* (frequency):
   много раз встречается в коллекции;
2. *совстречаемость слов* (collocation):
   состоит из слов, неслучайно часто встречающихся вместе;
3. *полнота* (completeness):
   является максимальной по включению цепочкой слов;
4. *синтаксическая связность* (syntactic connectedness):
   является грамматически корректным словосочетанием;
5. *тематичность* (topicality):
   часто встречается в небольшом числе тем.

Сумма технологий для ATE (Authomatic Term Extraction):
TopMine 1 2 3 + SyntaxNet 4 + BigARTM 5

Алгоритм TopMine: определения и основные идеи

- Хэш-таблица $C(a_1, \ldots, a_k)$ счётчиков частых k-грамм, инициализируется для всех униграмм a с частотой $n_a \geqslant \varepsilon_1$
- Свойство антимонотонности:

$$C(a_1, \ldots, a_k) \geqslant C(a_1, \ldots, a_k, a_{k+1}).$$

- $A_{d,k}$ — множество позиций i в документе d таких, что

$$C(w_{d,i}, \ldots, w_{d,i+k-1}) \geqslant \varepsilon_k,$$

инициализируется для всех частых униграмм.

- Основной шаг алгоритма: для всех $i = 1, \ldots, n_d$

если $(i \in A_{d,k})$ **и** $(i+1 \in A_{d,k})$ **то** $++C(w_{d,i}, \ldots, w_{d,i+k})$.

Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R. Voss, Jiawei Han. Scalable Topical Phrase Mining from Text Corpora // VLDB, 2015.

Алгоритм TopMine: быстрый поиск высокочастотных k-грамм

Вход: коллекция D, пороги ε_k;
Выход: хэш-таблица частот $C(a_1, \ldots, a_k)$, $k = 1, \ldots, k_{\max}$;

$A_{d,1} := \{1, \ldots, n_d\}$;
$C(w) := n_w$ для всех $w \in W$ таких, что $n_w \geqslant \varepsilon_1$;
для $k := 2, \ldots, k_{\max}$ **пока** $D \neq \varnothing$
  для всех $d \in D$
    $A_{d,k} := \left\{ i \in A_{d,k-1} \;\middle|\; C(w_{d,i}, \ldots, w_{d,i+k-2}) \geqslant \varepsilon_k \right\}$;
    если $A_{d,k} = \varnothing$ **то** $D := D \backslash \{d\}$;
    для всех $i \in A_{d,k}$
      если $i + 1 \in A_{d,k}$ **то** $++C(w_{d,i}, \ldots, w_{d,i+k-1})$;
  оставить только частые k-граммы: $C(a_1, \ldots, a_k) \geqslant \varepsilon_k$;

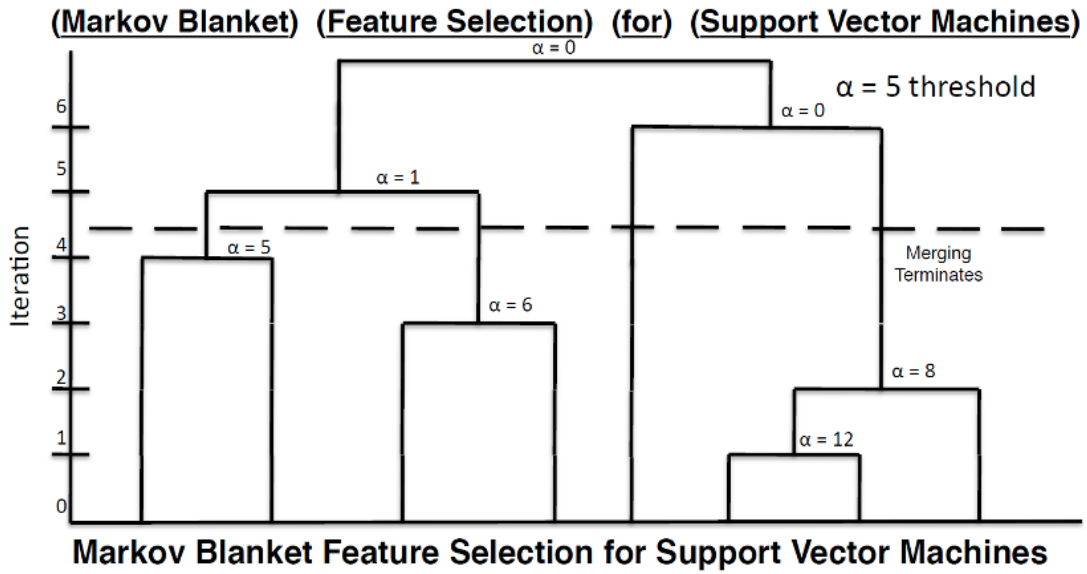
Преимущество алгоритма: линейная память и скорость.

Алгоритм TopMine: отбор фраз по совстречаемости и полноте

Итеративное слияние фраз с понижением значимости α.

p_u — оценка вероятности встретить фразу u
p_{uv} — оценка вероятности встретить фразу uv

Критерии: $\text{SignificanceScore} = \dfrac{p_{uv} - p_u p_v}{\sqrt{p_{uv}}}$



Метод главных компонент (Principal Component Analysis, PCA)

Матрицы «объекты–признаки», старая и новая:

$$\underset{\ell\times n}{F} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}; \quad \underset{\ell\times m}{G} = \begin{pmatrix} g_1(x_1) & \ldots & g_m(x_1) \\ \ldots & \ldots & \ldots \\ g_1(x_\ell) & \ldots & g_m(x_\ell) \end{pmatrix}.$$

Старые признаки должны линейно восстанавливаться по новым:

$$\hat{f}_j(x) = \sum_{t=1}^{m} g_t(x) u_{jt}, \quad j = 1, \ldots, n, \quad \forall x \in X,$$

Найти: низкоранговое матричное разложение:

$$\sum_{i=1}^{\ell}\sum_{j=1}^{n}\big(\hat{f}_j(x_i) - f_j(x_i)\big)^2 = \big\|GU^{\mathsf{T}} - F\big\|^2 \to \min_{G,U}$$

Решение — через сингулярное разложение $F = V\Lambda U^{\mathsf{T}}$.

Разреженный метод главных компонент

Если x_i — документы, f_j — частоты слов, F разрежена.
Вместо сингулярного разложения решают другую задачу:

$$\sum_{(i,j)} \Big(\sum_{t \in T} g_{it} u_{jt} - f_j(x_i) \Big)^2 \to \min_{G,U}.$$

Решив её, получим в пространстве низкой размерности m:
u_j — векторные представления слов $j = 1, \ldots, n$,
g_i — векторные представления документов $i = 1, \ldots, \ell$.

К сожалению, координаты этих векторов не интерпретируемы.
Возможно ли наделить их семантикой «тем» или «интересов»?
Да, если наложить на них некоторые ограничения...

Tacáks G., Pilászy I., Németh B., Tikk D. Scalable collaborative filtering approaches for large recommendation systems // JMLR, 2009.

Резюме

- Задачи машинного обучения разнообразны
- У каждой задачи есть ДНК, «Дано–Найти–Критерий»
- Большинство задач сводятся к оптимизации
- Для их решения применяются численные методы
- SVM — сильный метод классификации текстов
- PCA переводит решение в более удобное пространство
- Не успели поговорить про качество и переобучение :(
- Умышленно не поговорили про ANN, WE и TM

Полезные ссылки

- www.MachineLearning.ru — русскоязычная вики
- www.kdnuggets.com — главный сайт датамайнеров
- www.datasciencecentral.com — 72 000 датамайнеров
- www.kaggle.com — конкурсы анализа данных
- DataRing.ru — отечественная конкурсная платформа
- archive.ics.uci.edu/ml — UCI ML Repository (349 datasets)
- ru.coursera.org/learn/machine-learning — курс Эндрю Ына
- ru.coursera.org/learn/vvedenie-mashinnoe-obuchenie — курс Воронцова от ВШЭ и ШАД Яндекс
- ru.coursera.org/specializations/machine-learning-data-analysis — специализация от МФТИ и ШАД Яндекс

Рекомендуемая литература

- *Домингос П.* Верховный алгоритм. 2016. 336 с.
- *Коэльо Л. П., Ричарт В.* Построение систем машинного обучения на языке Python. 2016. 302 с.
- Машинное обучение (курс лекций, К. В. Воронцов). www.MachineLearning.ru. 2004–2017.
- *Мерков А. Б.* Распознавание образов. Введение в методы статистического обучения. 2011. 256 с.
- *Мерков А. Б.* Распознавание образов. Построение и обучение вероятностных моделей. 2014. 238 с.
- *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning. Springer, 2014. 739 p.
- *Bishop C. M.* Pattern recognition and machine learning. Springer, 2006. 738 p.