# Методы реализации расстояний с использованием разнородной обучающей информации

Александр Сенин, 417

21 апреля 2021 г.

# Метрика

Пусть $\mathcal{X}$ — множество объектов произвольной природы.

*Метрикой* (*расстоянием*) будем называть функцию $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\forall x, y, z \in \mathcal{X}$ удовлетворяющую следующим аксиомам:

(a) $\rho(x, x) = 0$

(b) $\rho(x, y) \geqslant 0$ (неотрицательность)

(c) $\rho(x, y) = \rho(y, x)$ (симметричность)

(d) $\rho(x, y) = 0 \Longrightarrow x = y$ (определенность)

(e) $\rho(x, y) \leqslant \rho(x, z) + \rho(z, y)$ (неравенство треугольника)

# Обобщения метрики

Определим функцию различия (dissimilarity function).

*Различием* (*dissimilarity*) будем называть функцию $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\forall x, y \in \mathcal{X}$ удовлетворяющую следующим требованиям:

(D1) $d(x, x) = 0$

(D2) $d(x, y) \geqslant 0$

Рассмотрим множество объектов $x_1, ..., x_N \in \mathcal{X}$.
Известно, что на некоторых парах определена функция различия $d(x_i, x_j)$.
В таком случае заданы тройки $\{(i, j, d_{ij})\}_{(i,j)\in M}$, где $M$ — множество пар, на которых определно различие.

В случае $M = (i, j)_{i,j=1}^{N}$ будем считать, что задана *матрица попарных различий* $D = (d_{ij})_{i,j=1}^{N}$.
В случае, когда известно, что $d$ представляет собой метрику, задана *матрица попарных расстояний* $D$.

# Обобщения метрики

Естественно вслед за различием определить понятие сходства.

*Сходством* (*близостью, similarity*) будем называть функцию $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\forall x \in \mathcal{X}$ удовлетворяющую единственному требованию:

(S1) $s(x, x) > 0$

Будем говорить, что для троек $\{(i, j, d_{ij})\}_{(i,j)\in M}$ существует *реализация расстояния* (*различия*) в пространстве $\mathbb{R}^d$, если существует такой набор (*конфигурация*) $x_1, ..., x_N \in \mathbb{R}^d$, что $\forall (i, j) \in M$

$$\rho_E(x_i, x_j) = ||x_i - x_j|| = d_{ij} \tag{1}$$

На практике точное равенство не всегда достижимо.

# Преобразования сходств и различий

Часто приходится конвертировать сходства в различия или наоборот.
*По сходствам получаем различия:*

- Пусть известен диапазон значений функции сходства $0 \leqslant s(x, y) \leqslant S$, $\forall x, y \in \mathcal{X}$, а также $s(x, x) = S$, $\forall x \in \mathcal{X}$ (среди всевозможных объектов больше всего на себя похож сам объект). Тогда легко показать, что $d(x, y) = S - s(x, y)$ является функцией различия.
- Более общий подход — применение некоторой неотрицательной убывающей функции $f(s(x, y))$, где $f(a) > f(b)$ при $a < b$.
  В общем случае теряем требования (D1) или (D2). Очевидный вариант преобразования $d(x, y) = \frac{1}{s(x,y)}$ лишает требования (D1).

  На практике сохранение всех аксиом функции различия зачастую не важно, выбор функции $f$ позволяет смещать внимание модели на более тщательное воспроизведение близких (или наоборот далеких) объектов.

# Преобразования сходств и различий

*По различиям получаем сходства:*

- Аналогично: применяем некоторую убывающую неотрицательную функцию к функции различия $g(d(x, y))$, где $g(a) > g(b)$ при $a < b$. Слабые требования в определении функции сходства позволяют гарантировать получение строгого сходства при таком преобразовании.

При некоторых дополнительных условиях можно использовать более качественные преобразования с теоретическим гарантиями.

# Многомерное шкалирование

Рассмотрим некоторое множество объектов $\hat{x}_1, ..., \hat{x}_N \in \mathcal{X}$. Известно, что на некоторых парах $M$ этого множества определена функция различия $d(x, y)$: $\{(i, j, d_{ij})\}_{(i,j)\in M}$, где $d_{ij} = d(\hat{x}_i, \hat{x}_j)$.

Неформально, общий подход методов *многомерного шкалирования* (*multidimensional scaling, MDS*) заключается в нахождении (возможно приближенно) конфигурации $x_1, ..., x_N \in \mathbb{R}^d$, реализующей различия $d_{ij}$ в терминах (1):

$$\rho_E(x_i, x_j) = ||x_i - x_j|| = d_{ij}$$

Принято считать, что размерность $d$ полученных векторов $x_1, ..., x_N$ задается предварительно. Однако, существуют теоретические результаты, гарантирующие существование размерности $p$ и конфигурации $x_1, ..., x_N \in \mathbb{R}^p$, точно реализующей матрицу попарных расстояний $D$ в терминах (1), при условии $d(x, y) = \rho_E(x, y)$ — исходная функция различия должна быть расстоянием, причем евклидовым.

# Классическое многомерное шкалирование

Методы многомерного шкалирования распадаются на три класса.

*Классическое многомерное шкалирование (classical multidimensional scaling, cMDS)*:

Задана матрица попарных различий $D = (d_{ij})_{i,j=1}^{N}$ и выполнено важное предположение: $\mathcal{X}$ является евклидовым пространством, причем функция различия $d$ есть евклидово расстояние
$d(\hat{x}_i, \hat{x}_j) = \rho_E(\hat{x}_i, \hat{x}_j) = \sqrt{\langle \hat{x}_i - \hat{x}_j, \hat{x}_i - \hat{x}_j \rangle}$.

Обозначим *матрицу Грама* (матрицу скалярных произведений)
$B = (b_{ij})_{i,j=1}^{N}$, $b_{ij} = \langle \hat{x}_i, \hat{x}_j \rangle$.
Определим *функционал Strain (натяжения)*:

$$Strain(x_1, ..., x_N) = \sum_{i,j=1}^{N} (b_{ij} - \langle x_i, x_j \rangle)^2, \; x_1, ..., x_N \in \mathbb{R}^d \qquad (2)$$

# Классическое многомерное шкалирование

Часто определяют $Strain$ иначе:

$$Strain(x_1, ..., x_N) = \sqrt{\frac{\sum_{i,j=1}^{N} (b_{ij} - \langle x_i, x_j \rangle)^2}{\sum_{i,j=1}^{N} b_{ij}^2}}, \; x_1, ..., x_N \in \mathbb{R}^d \qquad (3)$$

*Задача классического многомерного шкалирования:*

- По матрице попарных различий $D$ восстановить матрицу Грама $B$.
- Решить оптимизационную задачу:

$$Strain(x_1, ..., x_N) = \sum_{i,j=1}^{N} (b_{ij} - \langle x_i, x_j \rangle)^2 \longrightarrow \min_{x_1, ..., x_N \in \mathbb{R}^d} \qquad (4)$$

# Метрическое многомерное шкалирование

*Метрическое многомерное шкалирование (metric multidimensional scaling, mMDS):*

Обобщение классического подхода, снимаем требования на заданную матрицу попарных различий.

Пусть на некоторых парах $M$ заданы значения функции различия $d(x, y)$: $\{(i, j, d_{ij})\}_{(i,j)\in M}$, где $d_{ij} = d(\hat{x}_i, \hat{x}_j)$.
Определим *функционал Stress (стресса)*:

$$Stress(x_1, ..., x_N) = \sum_{(i,j)\in M} (d_{ij} - ||x_i - x_j||)^2,\ x_1, ..., x_N \in \mathbb{R}^d \qquad (5)$$

Часто определяют $Stress$ иначе:

$$Stress(x_1, ..., x_N) = \sqrt{\frac{\sum_{(i,j)\in M}(d_{ij} - ||x_i - x_j||)^2}{\sum_{(i,j)\in M} d_{ij}^2}},\ x_1, ..., x_N \in \mathbb{R}^d \qquad (6)$$

# Метрическое многомерное шкалирование

*Задача метрического многомерного шкалирования:*

- Решить оптимизационную задачу:

$$Stress(x_1, ..., x_N) = \sum_{(i,j) \in M} (d_{ij} - ||x_i - x_j||)^2 \longrightarrow \min_{x_1, ..., x_N \in \mathbb{R}^d} \quad (7)$$

Аналогично, легко заметить, что для оптимизационной задачи минимизации определения 5 и 6 эквивалентны.

# Неметрическое многомерное шкалирование

*Неметрическое многомерное шкалирование (non-metric multidimensional scaling, nMDS):*

В классической литературе, например, обычно предполагается существование некоторой неизвестной возрастающей функции $f$, такой что $d_{ij} = f(\delta_{ij})$, где $d_{ij}$ — наблюдаемые различия, а $\delta_{ij}$ — истинные. Затем по аналогии с mMDS конструируется функционал стресса. Этот подход обсудим позднее.

Пусть на некоторых парах $M$ заданы значения функции различия $d(x, y)$: $\{(i, j, d_{ij})\}_{(i,j)\in M}$, где $d_{ij} = d(\hat{x}_i, \hat{x}_j)$.

# Неметрическое многомерное шкалирование

*Задача неметрического многомерного шкалирования:*

- По заданным различиям $d_{ij}$ требуется найти конфигурацию $x_1, ..., x_N \in \mathbb{R}^d$, такую что

$$\forall\,(i,j),(k,l) \in M \;\; d_{ij} < d_{kl} \Longleftrightarrow ||x_i - x_j|| < ||x_k - x_l|| \tag{8}$$

- Дополнительно можно потребовать выполнение неравенства с некоторым зазором (отступом) $m > 0$:

$$\forall\,(i,j),(k,l) \in M \;\; d_{ij} < d_{kl} \Longleftrightarrow ||x_i - x_j|| + m < ||x_k - x_l|| \tag{9}$$

# Решение cMDS

*Классическое многомерное шкалирование (classical multidimensional scaling, cMDS):*

При решении cMDS важны два предположения:

- Предположение о том, что функция различия есть евклидово расстояние $d(\hat{x}_i, \hat{x}_j) = \rho_E(\hat{x}_i, \hat{x}_j) = \sqrt{\langle \hat{x}_i - \hat{x}_j, \hat{x}_i - \hat{x}_j \rangle}$
- Предположение о том, что расстояния заданы на всевозможных парах $(i, j)$, то есть задана полноценная матрица попарных расстояний $D = (d_{ij})_{i,j=1}^{N}$.

# Решение cMDS

Основная идея:
$d_{ij}^2 = (x_i - x_j)^T(x_i - x_j) = x_i^T x_i + x_j^T x_j - 2x_i^T x_j = b_{ii} + b_{jj} - 2b_{ij}.$
Дополнительное условие: $\overline{x} = \frac{1}{N}\sum_{i=1}^N x_i = 0.$

- $\overline{x} = 0 \Rightarrow \sum_{i=1}^N b_{ij} = 0$
- $$\frac{1}{N}\sum_{i=1}^N d_{ij}^2 = \frac{1}{N}\sum_{i=1}^N b_{ii} + b_{jj}$$
  $$\frac{1}{N}\sum_{j=1}^N d_{ij}^2 = b_{ii} + \frac{1}{N}\sum_{j=1}^N b_{jj}$$
  $$\frac{1}{N^2}\sum_{i,j=1}^N d_{ij}^2 = \frac{2}{N}\sum_{i=1}^N b_{ii}$$
- $b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i\bullet}^2 - d_{\bullet j}^2 + d_{\bullet\bullet}^2),$
  где $d_{i\bullet}^2 = \frac{1}{N}\sum_{i=1}^N d_{ij}^2,\ d_{\bullet j}^2 = \frac{1}{N}\sum_{j=1}^N d_{ij}^2,\ d_{\bullet\bullet}^2 = \frac{1}{N^2}\sum_{i,j=1}^N d_{ij}^2$

## Решение cMDS

Другими словами,

$$B = C_N A C_N, \tag{10}$$

где $A$ получается из $D$ поэлементным возведением в квадрат и домножением на $-0.5$, $C_N = E - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\mathbf{1}\mathbf{1}^T$ — матрица из единиц.

Пусть $X \in \mathbb{R}^{N \times p}$ — матрица, в которой векторы искомой конфигурации $x_1, ..., x_N$ записаны по строкам.

Тогда по определению матрицы Грама $B = XX^T$, причем

- $B$ — симметричная $(XX^T)^T = XX^T$
- $B$ — неотрицательно определенная $\langle XX^T a, a\rangle = \langle X^T a, X^T a\rangle \geqslant 0$
- $rgB = rgXX^T = rgX = p$

$B$ имеет $p$ положительных собственных значений и $n - p$ нулевых, и может быть представлена в виде:

$$B = \Gamma \Lambda \Gamma^T, \tag{11}$$

где $\Lambda = diag(\lambda_1, ..., \lambda_p)$ и $\Gamma = (\gamma_1, ..., \gamma_p)$ — матрица собственных векторов, отвечающих собственным значениям $\lambda_1, ..., \lambda_p$.

Наконец, отсюда находим искомую конфигурацию

$$X = \Gamma \Lambda^{\frac{1}{2}} \tag{12}$$

# Решение cMDS

Таким образом, решение cMDS заключается в следующем:

1. По матрице попарных расстояний $D$ восстанавливаем матрицу Грама $B$ (10).
2. Находим спектральное разложение матрицы $B$ (11).
3. Восстанавливаем искомую конфигурацию $x_1, ..., x_N$ (12).

- cMDS — базовый подход, предлагает эффективное аналитическое решение, без сложных оптимизационных задач.
- cMDS находит точное решение при поставленных требованиях, но не позволяет выбирать размерность $p$. Для произвольной размерности $d$ оставляем $d$ наибольших собственных значений.
- cMDS можно применять и при неевклидовой функции различия $d(x, y)$. Применяют и для решения задачи mMDS.
- cMDS не может быть применен в случае, когда множество пар $M$ не образует множество всевозможных пар (в матрице попарных различий пропуски).

# Решение mMDS

В метрическом многомерном шкалировании нет столь строгих требований (как в cMDS) на заданные различия $d(x, y)$: $\{(i, j, d_{ij})\}_{(i,j)\in M}$, где $d_{ij} = d(\hat{x}_i, \hat{x}_j)$.

Большинство подходов метрического многомерного шкалирования сводится к оптимизации функционала стресса тем или иным методом.

Самый классический метод — оптимизация градиентным спуском.

Популярный метод — алгоритм оптимизации SMACOF.

# SMACOF

SMACOF — Scaling by MAjorizing a COmplicated Function.
Предлагается оптимизировать стресс с весами (у пропущенных пар $w_{ij} = 0$):

$$Stress(x_1, ..., x_N) = \sum_{i,j=1}^{N} w_{ij}(d_{ij} - ||x_i - x_j||)^2 \longrightarrow \min_{x_1,...,x_N \in \mathbb{R}^d} \quad (13)$$

Идея этого метода состоит в нахождении вариационной верхней оценки на функционал $Stress$ ($g(x, \xi)$ — вариационная верхняя оценка функции $f(x)$, если $f(x) \leqslant g(x, \xi)\ \forall x \in X, \forall \xi \in Z$ и $\forall x \in X\ \exists \xi \in Z$, т.ч. $f(x) = g(x, \xi)$).

Вместо оптимизации функционала $Stress$ (13) предлагается итерационно оптимизировать его вариационную верхнюю оценку.

# Решение nMDS

Классический подход, предложенный Kruskal, работает с входными различиями $\delta_{ij}$. Вводится функционал стресса:

$$S(x_1, ..., x_N) = \sqrt{\frac{\sum (f(\delta_{ij}) - ||x_i - x_j||)^2}{\sum ||x_i - x_j||^2}}, \; x_1, ..., x_N \in \mathbb{R}^d \qquad (14)$$

В такой постановке задачи требуется найти конфигурацию, минимизирующую функционал $S$, а также монотонную функцию $f$ такую, что $f(\delta_{ij}) \leqslant f(\delta_{kl})$ в случае $\delta_{ij} < \delta_{kl}$.

В общем случае нет точного решения. Суть задачи сводится к сохранению порядка на значениях функции различия.

Дополнительная трудность в нахождении $f$.

# Решение nMDS

В классике проблему отыскания функции $f$ предлагается решить методами *изотонической регрессии*: найти $\hat{d}_{ij}$ такие, что в случае $\delta_{i_1,j_1} \leqslant \delta_{i_2,j_2} \leqslant ... \leqslant \delta_{i_m,j_m}$ выполняется $\hat{d}_{i_1,j_1} \leqslant \hat{d}_{i_2,j_2} \leqslant ... \leqslant \hat{d}_{i_m,j_m}$.
После этого вновь конструируется уже привычный функционал стресса:

$$S(x_1, ..., x_N) = \sqrt{\frac{\sum(\hat{d}_{i,j} - ||x_i - x_j||)^2}{\sum ||x_i - x_j||^2}}, \; x_1, ..., x_N \in \mathbb{R}^d \qquad (15)$$

При масштабировании конфигурации в $k$ раз ($x_i \to kx_i$) различия, в том числе и $\hat{d}_{i,j}$, увеличатся в $k$ раз, а весь числитель под корнем в $k^2$ раз. Чтобы это скомпенсировать, добавляется нормировочный знаменатель.

# Описание подхода

Наша цель: предложить универсальную модель, в зависимости от модификации решающую задачи метрического и неметрического многомерного шкалирования, при этом допускающую расширения на случай разнородности данных.

За основу возьмем следующую модель: Пусть отображение объектов в целевое пространство $l_w : \{1, ..., N\} \to \mathbb{R}^d$ осуществляется одним полносвязным слоем нейронной сети с весами $w$. Для объекта с номером $i$ сеть будет хранить $d$ чисел — веса, соответствующие $i$-му входу.

Используем логику работы *сиамских сетей*. Представление, полученное сетью для $i$-го объекта, будем обозначать $l_w(i)$. Обучать сеть $l_w$ будем обратным распространением ошибки.

# Решение mMDS предлагаемым подходом

Для задачи *метрического многомерного шкалирования* будем оптимизировать функцию потерь:

$$L_{mMDS} = \sum_{(i,j)\in M} (d_{ij} - ||l_w(i) - l_w(j)||)^2 \tag{16}$$

# Решение nMDS предлагаемым подходом

В случае задачи *неметрического многомерного шкалирования* отойдем от классической постановки функционала $Stress$ (Kruskal) к нашей постановке задачи (9):

$$\forall\,(i,j),(k,l) \in M \;\; d_{ij} < d_{kl} \Longleftrightarrow ||x_i - x_j|| + m < ||x_k - x_l||$$

В такой постановке задачи хорошо прослеживается суть — сохранить отношение порядка на различиях. Мы предложим два подхода к решению задачи с использованием нашего полносвязного слоя $l_w$.

# Triplet Loss

Рассмотрим понятие *triplet loss*. Пусть по входным данным сформирована тройка — точка интереса (anchor point), точка того же класса, что точка интереса (anchor-positive), точка другого класса (anchor-negative). Тогда triplet loss выглядит так:

$$L = max(0, d_{ap} - d_{an} + m) \tag{17}$$

Здесь $d_{ap}$ — расстояние от представления anchor point до представления anchor-positive, $d_{ap}$ — до представления anchor-negative соответственно, $m$ — отступ, гарантирующий выполнение $|d_{ap} - d_{an}| \geq m$.

В нашей задаче отсутствуют метки anchor-positive и anchor-negative на объектах. Однако, мы можем использовать в качестве anchor-positive и anchor-negative для объекта с номером $i$ такие объекты с номерами $j$ и $k$ соответственно, что $d_{ij} < d_{ik}$.

# Решение nMDS предлагаемым подходом

Таким образом, пусть пары $(i,j),(i,k) \in M$.

$$L_{nMDS}(x_i,x_j,x_k) = \begin{cases} max(0,||x_i - x_j|| - ||x_i - x_k|| + m), & \text{если } d_{ij} < d_{ik} \\ max(0,||x_i - x_k|| - ||x_i - x_j|| + m), & \text{если } d_{ik} < d_{ij} \end{cases} \tag{18}$$

Тогда функция потерь будет выглядеть так:

$$L_{nMDS} = \sum_{(i,j),(i,k)\in M} L_{nMDS}(l_w(i),l_w(j),l_w(k)) \tag{19}$$

# Решение nMDS предлагаемым подходом

Естественнее работать с четверками.

В нашей постановке задачи: $\forall\,(i,j),(k,l) \in M \;\; d_{ij} < d_{kl} \iff$
$||x_i - x_j|| + m < ||x_k - x_l|| \iff ||x_i - x_j|| - ||x_k - x_l|| + m < 0$

Отсюда приходим к идее quadruplet loss.

Пусть пары $(i,j),(k,l) \in M$.

$$L_{nMDS}(x_i, x_j, x_k, x_l) = \begin{cases} max(0, ||x_i - x_j|| - ||x_k - x_l|| + m), & d_{ij} < d_{kl} \\ max(0, ||x_i - x_k|| - ||x_k - x_l|| + m), & d_{kl} < d_{ij} \end{cases} \tag{20}$$

Тогда функция потерь будет выглядеть так:

$$L_{nMDS} = \sum_{(i,j),(k,\hat{l}) \in M} L_{nMDS}(l_w(i), l_w(j), l_w(k), l_w(\hat{l})) \tag{21}$$

# Сравнительный анализ предлагаемого подхода

Как оценивать качество построенной конфигурации $x_1, ..., x_N$?

Исторический, ключевой функционал качества — величина нормированного стресса. Мы предлагаем для mMDS нормировать на $\sum d_{ij}^2$, Kruskal для nMDS предлагает нормировать на $\sum ||x_i - x_j||^2$ (Kruskal Stress).

Мы предлагаем дополнительно использовать ранговую корреляцию Спирмена — показатель того, насколько хорошо сохраняется порядок на различиях.

Третий путь — смотреть на показатели прикладной области.

# Сравнительный анализ предлагаемого подхода

Параметры эксперимента:
Реализация на PyTorch. Оптимизация Adam.
Сгенерирован набор данных $\hat{x}_1, ..., \hat{x}_N \in \mathbb{R}^D$ при $N = 100, D = 3$.
Функция различия $d(\hat{x}_i, \hat{x}_j) = \rho_E(\hat{x}_i, \hat{x}_j)$ — евклидово расстояние.
Размерность искомой конфигурации $d = 2, 3$.

*Решение задачи метрического многомерного шкалирования:*

| Размерность конфигурации d = 3 | | |
|---|---|---|
| Метод | Stress | Корреляция |
| cMDS | $10^{-16}$ | 1.0 |
| SMACOF | $10^{-3}$ | 1.0 |
| mMDS | $10^{-3}$ | 1.0 |

| Размерность конфигурации d = 2 | | |
|---|---|---|
| Метод | Stress | Корреляция |
| cMDS | $0.277$ | 0.811 |
| SMACOF | $0.218$ | 0.849 |
| mMDS | $0.222$ | 0.845 |

# Сравнительный анализ предлагаемого подхода

Параметры эксперимента:
Реализация на PyTorch.
Сгенерирован набор данных $\hat{x}_1, ..., \hat{x}_N \in \mathbb{R}^D$ при $N = 100, D = 3$.
Функция различия $d(\hat{x}_i, \hat{x}_j) = \rho_E(\hat{x}_i, \hat{x}_j)$ — евклидово расстояние.
Размерность искомой конфигурации $d = 2$.

*Решение задачи неметрического многомерного шкалирования:*

| Размерность конфигурации d = 2 | | |
|---|---|---|
| Метод | Kruskal Stress | Корреляция |
| cMDS | 0.221 | 0.811 |
| sklearn nMDS | 20.953 | 0.017 |
| triplet nMDS | 31.482 | 0.848 |
| quadruplet nMDS | 40.321 | 0.851 |

Kruskal Stress здесь сторонний показатель, мы оптимизировали не его, показательна корреляция.

По времени работы: mMDS вышел на корреляцию 0.8 за 10 минут, triplet nMDS за 10 секунд.

# Задача построения векторных представлений слов

Будем работать со словами естественного языка. Пусть задан *словарь* $V$ — множество слов.

Задача построения векторных представлений заключается в нахождении и сопоставлении конфигурации $x_1, ..., x_N \in \mathbb{R}^d$ всем словам из словаря $V$ таким образом, что $x_1, ..., x_N$ отвечают некоторым критериям качества.

Найденные векторы $x_i$ конфигурации будем называть *векторными представлениями слов*.

# Оценивание векторных представлений слов

Что считать критериями качества? Как оценивать построенные векторные представления?

Методы оценивания поделим на два класса:

1. *Внешнее оценивание (extrinsic evaluation)*

   Оцениваем при решении «внешних» задач с помощью сторонних методов машинного обучения и обработки естественного языка. Например, используем представления в решении задачи анализа тональности текста или задачи частеречной разметки.

2. *Внутреннее оценивание (intrinsic evaluation)*

   Оцениваем вне контекста задач обработки естественного языка. Часто опираются на представления людей о взаимосвязях слов.

Остановимся на внутреннем оценивании.

# Оценивание векторных представлений слов

При внутреннем оценивании принято выделять три показателя:

- Сохранение семантической близости.
- Сохранение аналогий.
- Категоризация.

Разберем каждый из них.

# Семантическая близость

Неформально, свойство семантической близости — близость в математическом смысле у векторных представлений для близких по смыслу слов.

Как определять близость слов по смыслу?
Принято выделять два понятия:

- Близость в смысле сходства, похожести сущностей, которые эти слова описывают (similarity)
- Близость в смысле связанности сущностей, которые слова описывают (relatedness)

Слова car и crash не похожи, но связаны. Слова car и train похожи, описывают близкие сущности. Слова Freud и psychology не похожи, но связаны.

# Сохранение аналогии

Основная идея свойства сохранении аналогий (word analogy task):
Задаются два связанных слова $a$ и $b$, и некоторое другое слово $c$.

По полученным представлениям тем или иным образом строится новое слово $d$, связанное с $c$ также, как $a$ связано с $b$. Сохранение аналогий — способность по векторным представлениям строить слова «по аналогии».

Например, если задать в качестве слова $a =$ «man», а слова $b =$ «woman», то для слова $c =$ «uncle» естественно ожидать построения слова «aunt».

# Категоризация

Задан некоторый набор данных, состоящий из $M$ слов и $K$ кластеров (заведомо дано разбиение по кластерам).

Идея состоит в попытке восстановить некоторым алгоритмом кластеризации эти кластеры, но в пространстве векторных представлений слов.

Оценкой качества служит среднее или суммарное расстояние между восстановленными кластерами и заданными.

## Оценивание свойства семантической близости

Обычно выполнение свойства семантической близости проверяют с помощью заранее размеченных экспертами наборов данных — «золотых стандартов» семантической близости. Общий подход построения таких наборов — оценить семантическую близость на парах слов, например, усредняя мнение экспертов.

Пусть известны степени семантической близости на парах слов $M$: $\{(i, j, s_{ij})\}_{(i,j)\in M}$. Два пути: определить функцию близости $s(x, y)$ (часто косинусная близость) на построенных векторных представлениях, либо преобразовать близости $s_{ij}$ в различия $d_{ij}$.

Переходим к рангам в вариационных рядах и вычисляем коэффициент корреляции Спирмена:

$$\rho = 1 - \frac{6}{n(n-1)(n+1)} \sum_{k=1}^{|M|} (R_k - S_k)^2, \tag{22}$$

где $R_k, S_k$ - ранги $s_{ij}$ и $s(x_i, x_j)$ ($d_{ij}$ и $d(x_i, x_j)$) в соответствующих вариационных рядах.

# Анализ «золотых стандартов» семантической близости

Выбрано три самых популярных «золотых стандарта».

| Название | Язык | Количество пар | Диапазон оценки | Значения оценки |
|---|---|---|---|---|
| WordSim353 | Английский | 353 | [0; 10] | Вещественные |
| SimLex-999 | Английский | 999 | [0; 10] | Вещественные |
| The MEN Test Collection | Английский | 3000 | [0; 50] | Целые |

Таблица: Общие характеристики «золотых стандартов».

| Название | Что оценивалось? | Число разметчиков | Распределение по частям речи |
|---|---|---|---|
| WordSim353 | Relatedness | 13 на 153 пары/16 на 200 пар | 100% n |
| The MEN Test Collection | Relatedness | 1 на каждую пару | 81.5% n + 12.76% j + 5.73% v |
| SimLex-999 | Similarity | 500 | 66.6% n + 11.1% j + 22.2% v |

Таблица: Особенности «золотых стандартов».

# Преимущества и недостатки «золотых стандартов»

1. WordSim353
   - Только существительные.
   - Экспертам не объяснялась разница между similar и related.
   - Словарь больше числа пар (437 слов при 351 паре)
2. SimLex999
   - Нет пар из слов разных частей речи.
   - Много слов встречаются только в одной паре.
   - Слишком большой словарь (1028 слов при 998 парах)
   - Экспертам объяснялась разница между similar и related.
3. MEN
   - Много пар, небольшой словарь (751 слово при 3000 пар)
   - Редки слова, которые встречаются только в одной паре.
   - Есть пары из слов разных частей речи.
   - Экспертам не объяснялась разница между similar и related.

Для работы с «золотыми стандартами» реализована небольшая библиотека на Python.

# Задача аналогии

*Задачей аналогии* будем называть следующую задачу:
По известной аналогии $a : a' :: b : b'$ строится тройка $a : a' :: b$.
Требуется по векторным представлениям $x_1, ..., x_N$ слов словаря $V$ для тройки слов $a : a' :: b$ найти слово $\beta \in V$ так, что четверка $a : a' :: b : \beta$ образует известную аналогию, то есть $\beta = b'$.

Обычно задается набор аналогий $A$, а качество векторных представлений оценивается как точность — доля верно решенных задач аналогии.

# Решение задачи аналогии

Как находить слово $b'$? Как решать задачу аналогии?

Определим косинус между двумя векторами (косинусная близость) как $\cos(x, y) = \frac{\langle x, y \rangle}{||x|| ||y||}$. Далее будем понимать под $a$ векторное представление $x(a)$ соответствующего слова.

Однозначного способа решать задачу аналогии не существует. Рассмотрим основные подходы:

- Подход *3CosAdd* рассматривает аналогию как параллелограмм:

$$b' = \arg\max_{\beta \in V} \cos(\beta, b + a - a') \tag{23}$$

- Аналогично *PairDistance*:

$$b' = \arg\max_{\beta \in V} \cos(\beta - b, a' - a) \tag{24}$$

## Решение задачи аналогии

- Подход *3CosMul* предлагает в предположении нормированности векторов воспользоваться линейностью скалярного произведения $\cos(\beta, b + a - a') = \cos(\beta, b) + \cos(\beta, a) - \cos(\beta, a')$, а затем перейти от сложения к умножению:

$$b' = \arg\max_{\beta \in V} \frac{\cos(\beta, b)\cos(\beta, a')}{\cos(\beta, a) + \epsilon} \tag{25}$$

- Подход *3CosAvg* предлагает по всему набору аналогий $A$ найти средний сдвиг $s = \frac{\sum_{i=0}^{m} a}{m} - \frac{\sum_{i=0}^{n} a'}{n}$, а затем добавить его к $b$:

$$b' = \arg\max_{\beta \in V} \cos(\beta, b + s) \tag{26}$$

Утверждается, что такой базовый подход повышает устойчивость к шуму в векторных представлениях, кроме помогает решать задачу аналогии в моделях с полисемией (учитывается несколько смыслов у слов).

## Решение задачи аналогии

- Подход *LRCos* предполагает некоторую структуру в наборе аналогий $A$.

  Например, для аналогии France : Paris :: Japan : Tokyo естественно считать слова $a, b$ принадлежащими исходному классу $source$ — страны, а слова $a', b'$ целевому $target$ — столицы.

  Идея: переформулируем вопрос «Что связано с Францией так же, как Токио связано с Японией?» в «Какое слово лежит в том же классе, что Токио, и ближайшее к слову Франция?».

  Предлагается обучить логистическую регрессию для распознавания слов целевого класса — в обучающую выборку для одного класса попадают все слова из $target$, для второго класса все слова из $source$ и прочие случайные слова. Окончательно вектор $b'$ находится следующим образом:

$$b' = \arg\max_{\beta \in V} P(\beta \in target) \cos(\beta, b) \tag{27}$$

## «Золотые стандарты» аналогий

Аналогично «золотым стандартам» семантической близости существуют общепринятые наборы аналогий $A$. Перечислим основные:

1. Google Analogy. Это самый классический большой набор аналогий из 19000 четверок. Типы связи в аналогиях: страны и столицы, валюты, морфология (формы слов).
2. MSR. Набор из 8000 четверок, тип связи: морфология.
3. BATS. Самый современный и совершенный набор аналогий. Структурирован в формате пар $(source, target)$. Сбалансирован, содержит связи следующих типов: морфология, энциклопедическая семантика (страны — столицы, вещи — цвета, мужское — женское, животные — жилища и т.д.), лексикографическая семантика (антонимы, несколько классов синонимов, гиперонимы, меронимы и т.д.)

# Построение векторных представлений на основе «золотых стандартов» семантической близости и аналогий.

*Дано*: размеченные экспертами наборы семантической близости на парах слов $M$: $\{s_{ij}\}$, $(i,j) \in M$, набор аналогий $A$, словари наборов пересекаются.

*Задача*: предложить способ построения векторных представлений слов методами многомерного шкалирования.

# Построение векторных представлений на основе «золотых стандартов» семантической близости и аналогий.

Набор экспертных близостей $\{s_{ij}\}$ преобразуем в различия $\{d_{ij}\}$. Будем использовать предложенную нами модель, реализующую метрическое и неметрическое многомерное шкалирование с помощью полносвязной однослойной нейронной сети $l_w$.

Таким образом, используя функцию потерь:

$$L_{mMDS} = (d_{ij} - ||l_w(i) - l_w(j)||)^2 \quad (28)$$

в случае метрического, и функцию потерь quadruplet loss:

$$L_{nMDS} = max(0, ||l_w(i) - l_w(j)|| - ||l_w(k) - l_w(\hat{l})|| + m), \text{при } d_{ik} < d_{kl} \quad (29)$$

в случае неметрического многомерного шкалирования можем решать задачу нахождения векторных представлений по семантическим близостям.

# Построение векторных представлений на основе «золотых стандартов» семантической близости и аналогий.

Для привлечения наборов аналогий естественно усовершенствовать модель, уже работающую с четверками.

Необходимо сконструировать функцию потерь на четверке векторных представлений, при этом про отвечающие им слова известно, что они образуют аналогию.
Обратимся к подходу *PairDistance*: $b' = \arg\max_{\beta \in V} \cos(\beta - b, a' - a)$.
Продолжая логику подхода естественно считать, что на четверке аналогии $a : a' :: b : b'$ достигается максимум расматриваемой косинусной близости: $\cos(b' - b, a' - a) \approx 1$. Отсюда получаем функцию потерь:

$$L_{analogy} = -\cos(l_w(i_{b'}) - l_w(i_b), l_w(i_{a'}) - l_w(i_a)) \tag{30}$$

Тогда искомый способ построения векторных построений заключается в нахождении конфигурации $x_1, ..., x_N$ путем минимизации функции потерь:

$$L = \sum_{\substack{(i,j),(k,l) \in M \\ (i,j,k,l) \in A}} (L_{nMDS} + L_{analogy}) \tag{31}$$

# Сравнительный анализ полученного решения

В качестве набора семантической близости $\{s_{ij}\}$ выбран MEN, как набор с наиболее удачным соотношением размера словаря и числа пар слов. Дополнительно, в нем есть пары разных частей речи.

В качестве набора аналогий $A$ выбран BATS, в нем большое разнообразие типов связи, есть большое пересечение словаря с MEN (словари остальных наборов аналогий почти не пересекаются со словарями наборов семантических близостей).

Аналогии решаем методом PairDistance. Реализация на PyTorch.
Оптимизация Adam.
Целевая размерность представлений $d = 2$.

# Сравнительный анализ полученного решения

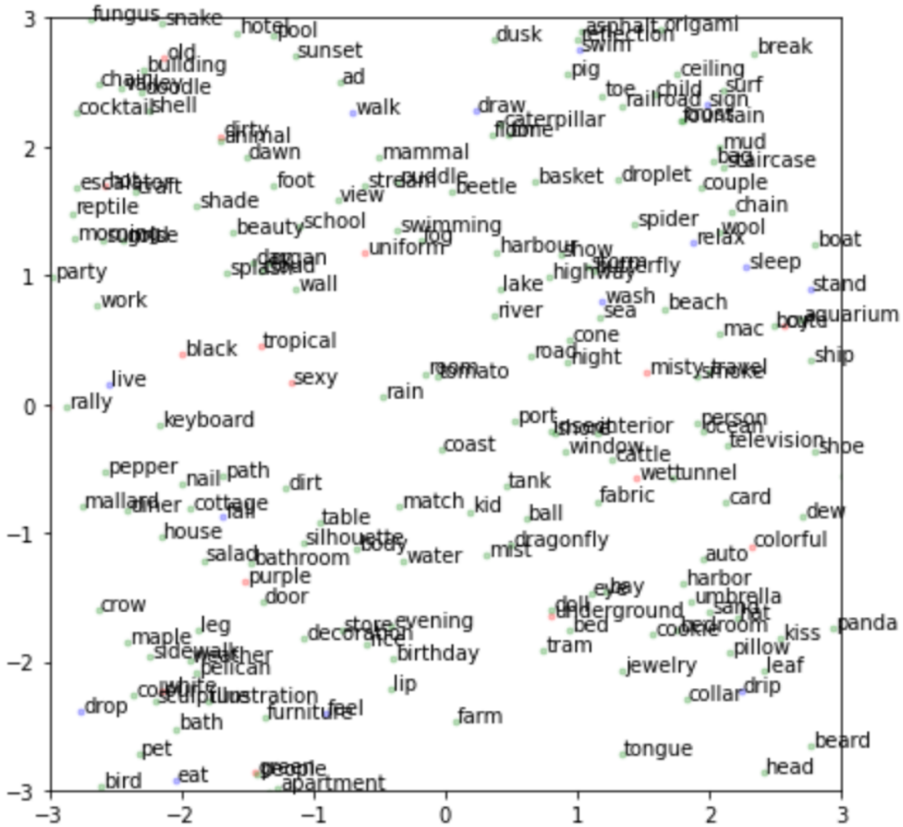
| Метод | Stress | Корреляция с MEN | Корреляция с WordSim353 | Корреляция с SimLex-999 | Точность на BATS |
|---|---|---|---|---|---|
| mMDS | 0.19 | 0.90 | 0.38 | 0.23 | 7.7% |
| nMDS + analogy | 0.24 | 0.88 | 0.37 | 0.21 | 15.5% |

Диапазоны значений показателей у «взрослых» методов построения по большим корпусам текстов:
У word2vec в зависимости от размерности представлений и входных данных на wordsim порядка 0.6-0.7, на simlex порядка 0.2-0.3.
BATS — «сложный» набор для методов построения векторных представлений. В зависимости от типа связи, размерности и входных данных корреляция колеблется от нескольких долей процента до 80-90%.

Полученные оценки смещены! Они вычисляются не на полных наборах данных, а только на словах, которые есть в MEN.

Правильнее будет смотреть на изменение корреляции с MEN и точности на BATS.

# Визуализация




Удачный пример: кластер слов swimming - lake - river - sea - beach - coast - water - boat - ship

Нудачный пример: соседние слова tram - jewelry - panda - kiss

# Возможные улучшения предложенного решения

- В данных могут быть «висячие вершины» — слова, которые встречаются только в одной паре. В MEN из 700 слов словаря таких слов 42. Очевидно, представление для таких слов можно достраивать отдельно (подойдет любая точка на сфере нужного радиуса).
- Проблема предложенного подхода — слишком много пропусков в наборах семантической близости. Можно решить, если предложить способ построения близостей (различий) по корпусам текстов.

# Результаты

- Были введены основные понятия и сформулированы задачи многомерного шкалирования. Проведен обзор существующих подходов к решению сформулированных задач.
- Разработан и реализован подход к решению задач метрического и неметрического шкалирования. Проведен сравнительный анализ с существующими подходами.
- Рассмотрена задача построения векторных представлений слов. Проведен обзор методов оценивания векторных представлений. Для работы с «золотыми стандартами» разработана и реализована библиотека на Python.
- Проведен обзор методов решения задачи аналогий.
- Разработан и реализован подход к построению векторных представлений слов на основе «золотых стандартов» семантической близости и аналогий.