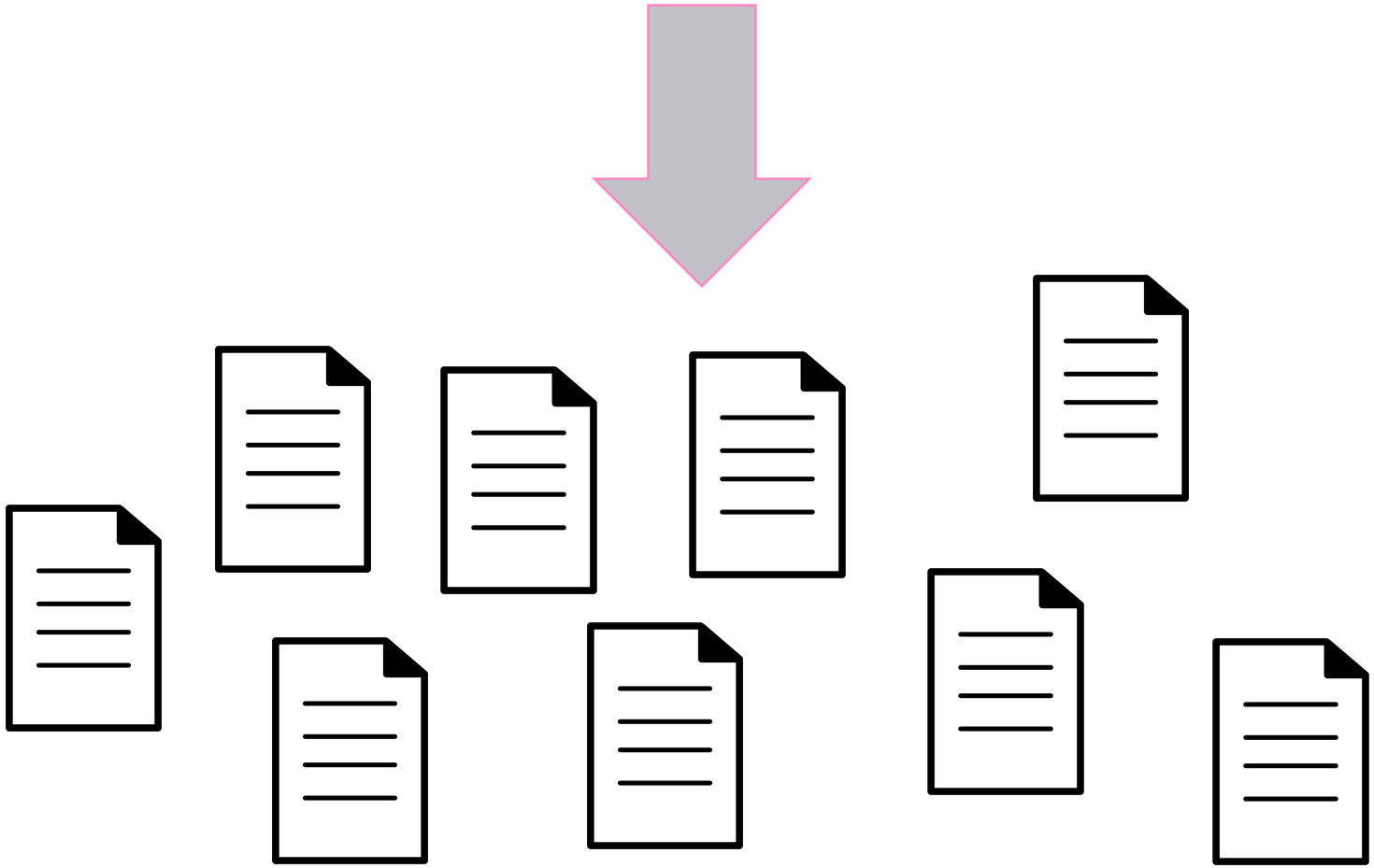
Information retrieval. Metric Learning.

Сбертех, МФТИ

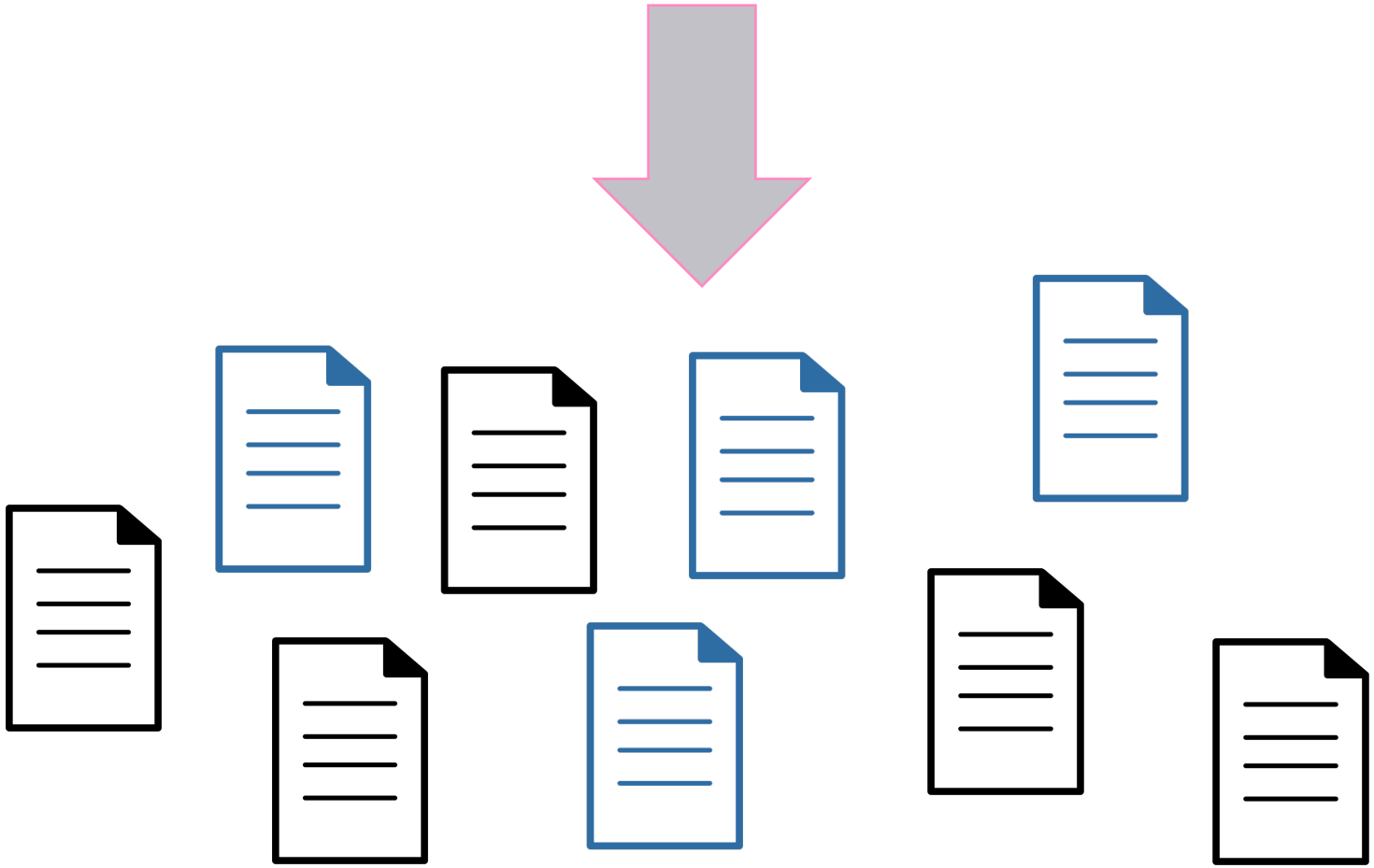
Задача информационного поиска

Кто снял фильм Титаник?



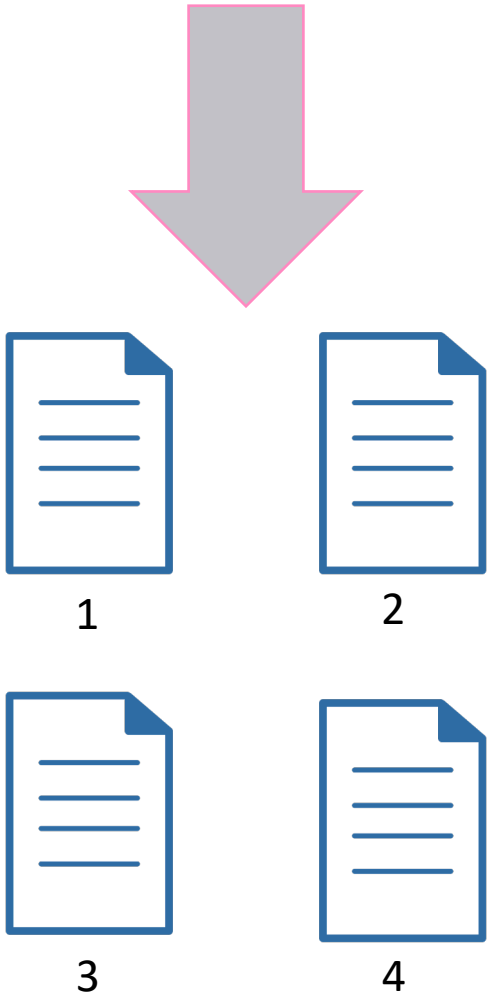
Задача информационного поиска – Поиск(retrieval)

Кто снял фильм Титаник?



Задача информационного поиска – Ранжирование(ranking)

Кто снял фильм Титаник?



Retrieval модели

Поисковики

Найди мне спортивный магазин в Москве, где продаются коньки.

Диалоговые модели

А что будет если мы не вспомним о Шафутинском на 3е сентября?

Маркетплейсы

Носки теплые махровые бардовые в крапинку

Вопросно-ответные системы

Почему небо голубое?

Карты

ул. Ивана Сбертехова, дом 5, строение 1

Системы принятия решений

Покажи мне сотрудников с самым маленьким KPI в прошлом месяце.

Инструменты поиска

Поиск по пересечению токенов → Поиск по семантике → Отбор кандидатов → Ранжирование кандидатов

Нечеткий поиск

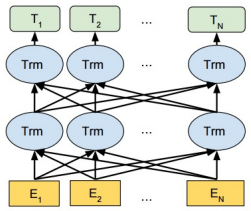
1. Алгоритмы нечеткого поиска
   1. Мера Левенштейна
   2. Расстояние Хэмминга
   3. Расстояние Monge-Elkan
   4. N-gram TF-IDF
2. Индексы хранения - Elastic, Lucene

Семантический поиск

1. Контекстно-независимые эмбеддинги
   1. Fast-text
   2. Word2Vec
2. Контекстно-зависимые эмбеддинги
   1. ELMO
   2. BERT
3. Индексы приближенного поиска

CatBoost

LightGBM



fastText

Формулировка задачи

Для данного запроса q среди коллекции D найти N релевантных документов $d_1, d_2, d_3, \dots, d_N$.

Задача многоклассовой классификации – среди множество документов коллекции D найти N документов объединенных тематикой запроса.

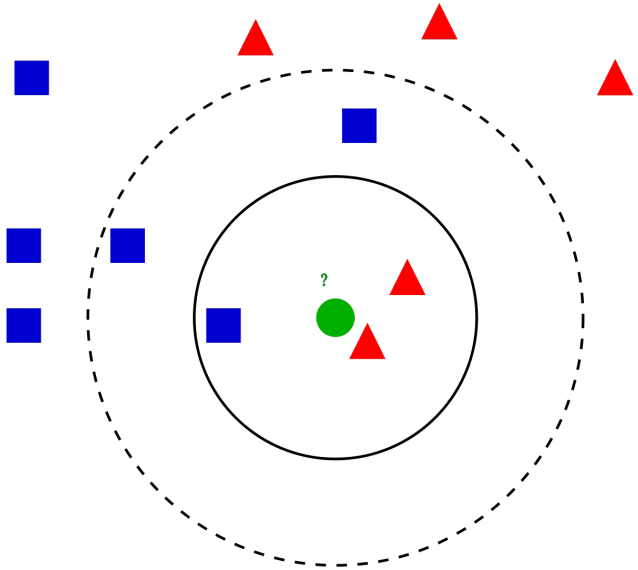
Запрос	Документ
Запрос#1	**Документ#1**
Запрос#1	**Документ#2**
Запрос#2	**Документ#1**
Запрос#2	**Документ#2**
Запрос#2	**Документ#3**

- Большое количество классов
- Постоянное добавление новых документов, как следствие расширение классов
- Большое признаковое пространство

Метод ближайших соседей

KNN в поиске

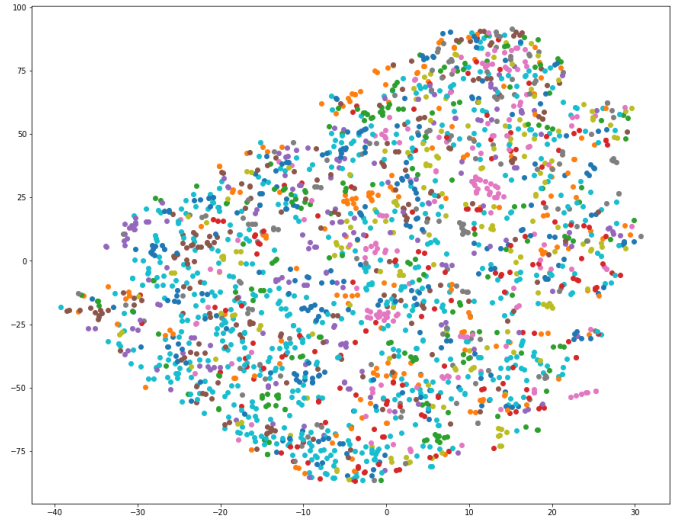
Суть алгоритма KNN – классификация объектов на основании классов ближайших объектов с помощью заданной функции расстояния.



+ Нет необходимости в переобучении при добавлении новых объектов
+ Быстрый поиск(ANN, MIPS)
+ Прост для понимания

- Не имеет собственных методов определения “похожести” объектов – просто ищет ближайшие объекты по заданной метрике.
- Плохо работает, если признаки имеют разный масштаб.

Какое будет качество KNN тут?



KNN в поиске

Id	Пол	Возраст	Вес	Доход(Руб)
1	1	34	82	30000
2	0	54	65	100000
3	0	45	76	50000
4	1	39	80	10000

- Признаки могут быть разного масштаба и могут быть коррелированы.
  - Традиционный подход – стандартизация, нормализация.
- Евклидово расстояние менее полезно в больших размерностях.

Метрика расстояния Махаланобиса – **l2 расстояние + стандартизация**

$$d(x_i, x_j) = \sqrt{(x_i - x_j)\sum_{X}^{-1}(x_i - x_j)}$$

$\sum_X^{-1}$ - обратная матрица ковариации

Метрика расстояния Малаханобиса

$$d(x_i, x_j) = \sqrt{(x_i - x_j)\sum_{X}^{-1}(x_i - x_j)}$$

$\sum_X^{-1}$ - эквивалентно некоторому линейному преобразованию $V^T V$

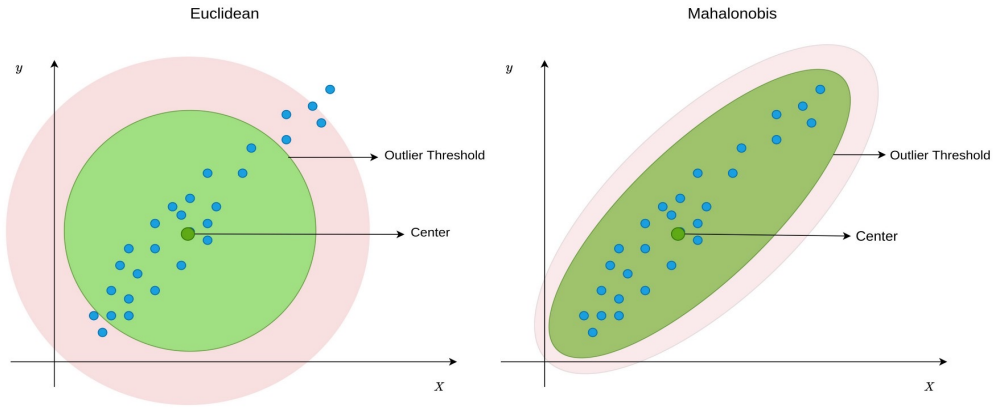
V – обучаемая матрица весов

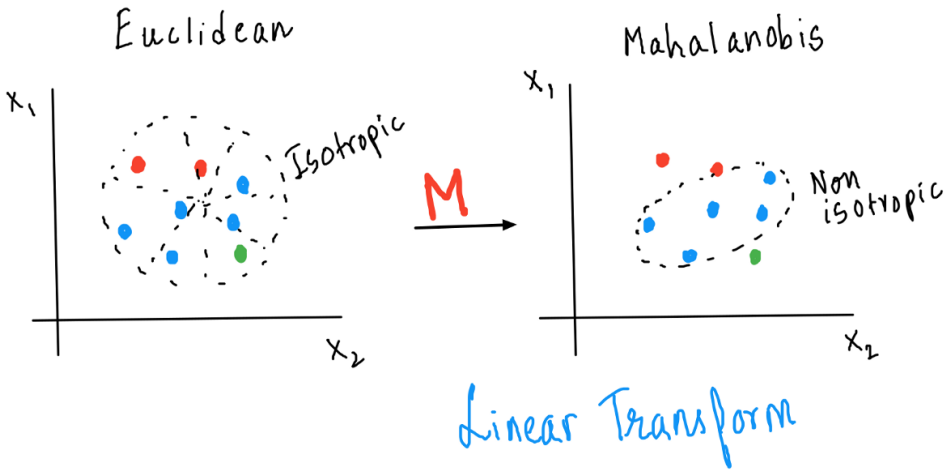
$$d(x_i, x_j) = \sqrt{(x_i - x_j)\sum_{X}^{-1}(x_i - x_j)}$$

$$= \sqrt{(x_i - x_j)V^T V(x_i - x_j)}$$

$$= \sqrt{(Vx_i - Vx_j)^T (Vx_i - Vx_j)}$$

Эллипсоидная функция расстояния





Линейный дискриминант Фишера

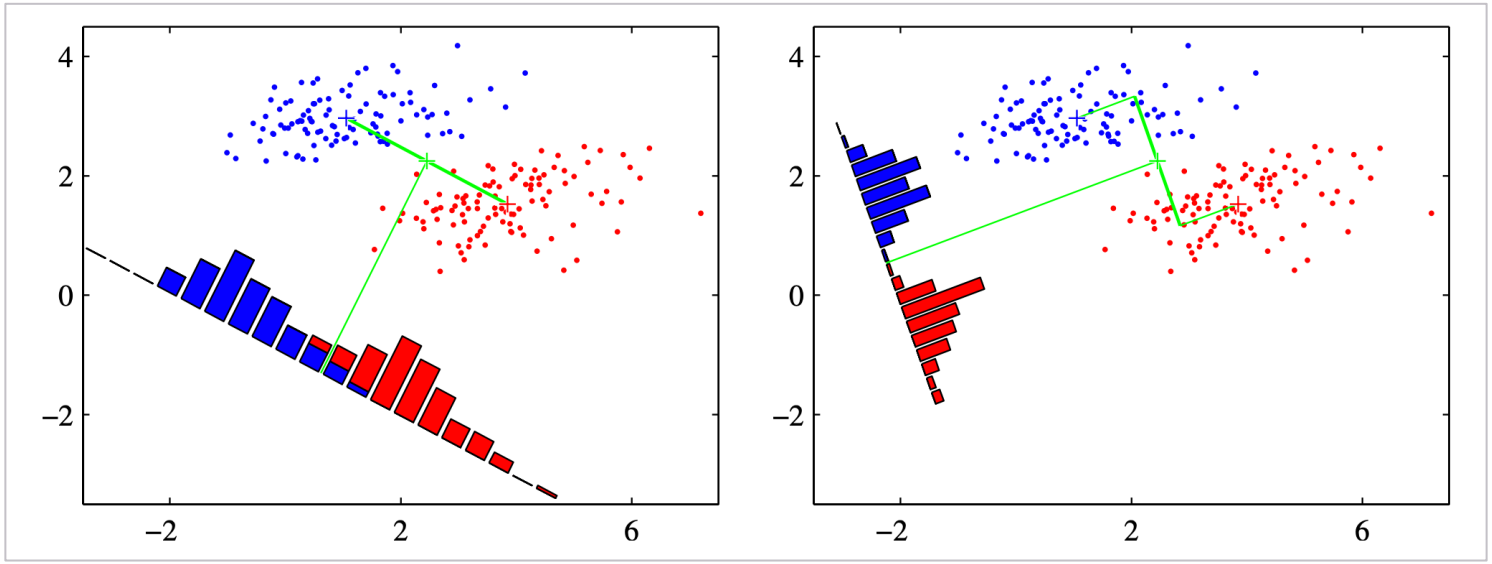
$$d(x_i, x_j) = \sqrt{(Vx_i - Vx_j)^T (Vx_i - Vx_j)}$$

$$x_i \in C_1, x_j \in C_2$$

Как научить модель наилучшим образом разделять объектов разных классов?

С помощью вектора весов V перевести объекты в размерность меньшего пространства(для работы l2), так что

- *Расстояние между объектами разных классов максимально*
- *Расстояние между объектами одних классов минимально*



Линейный дискриминант Фишера

Определим средние классов:

$$m_1 = \frac{1}{N}\sum_{n\in C_1} x_n \qquad m_2 = \frac{1}{N}\sum_{n\in C_2} x_n$$

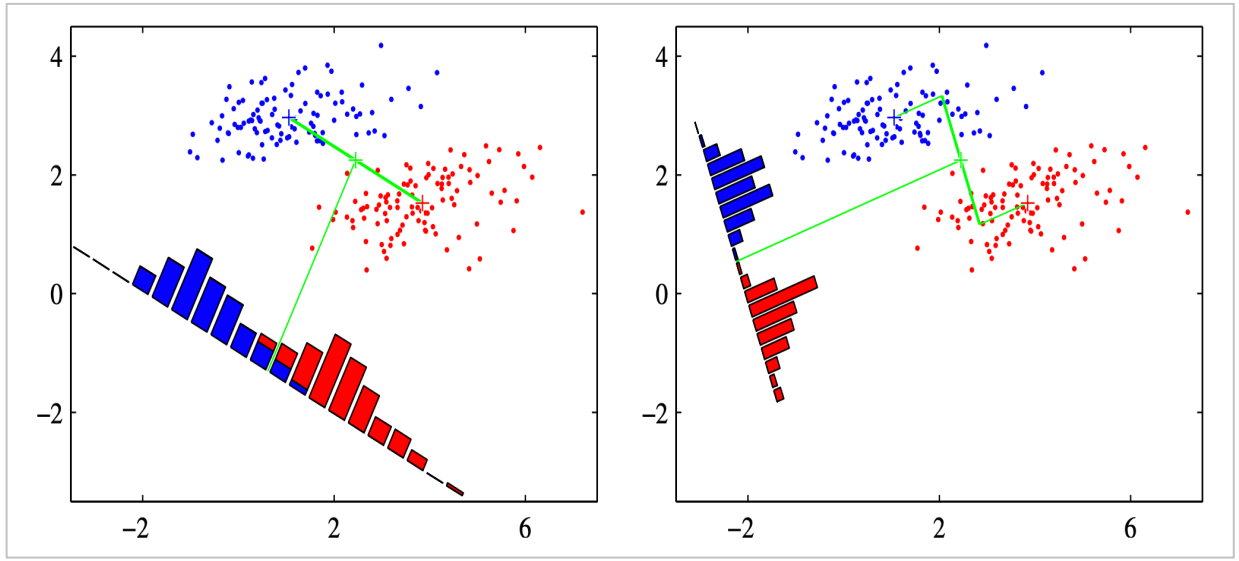
Надо подобрать такой вектор параметров V, что разница между средними классов должна быть большой:

$$\max(V^T m_1 - V^T m_2)$$

А внутриклассовое расстояние(внутриклассовая дисперсия) должно быть маленьким:

$$s_1 = \frac{1}{N}\sum_{n\in C_1} (V^T x_n - m_1) \quad s_2 = \frac{1}{N}\sum_{n\in C_2} (V^T x_n - m_2)$$

$$\min(s_1 + s_2)$$



Максимизация только расстояния между классами

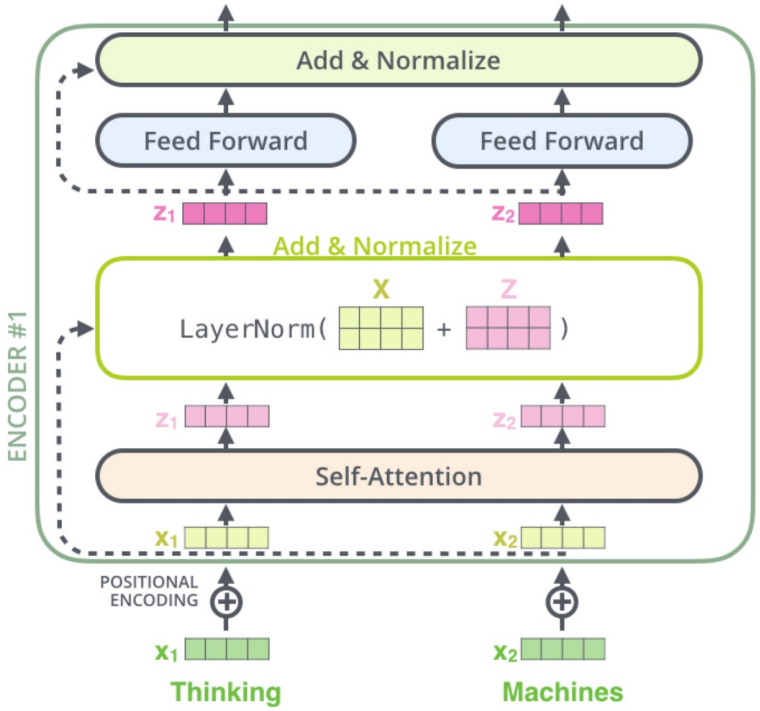
Максимизация расстояния между классами и минимизация внутриклассовой дисперссии

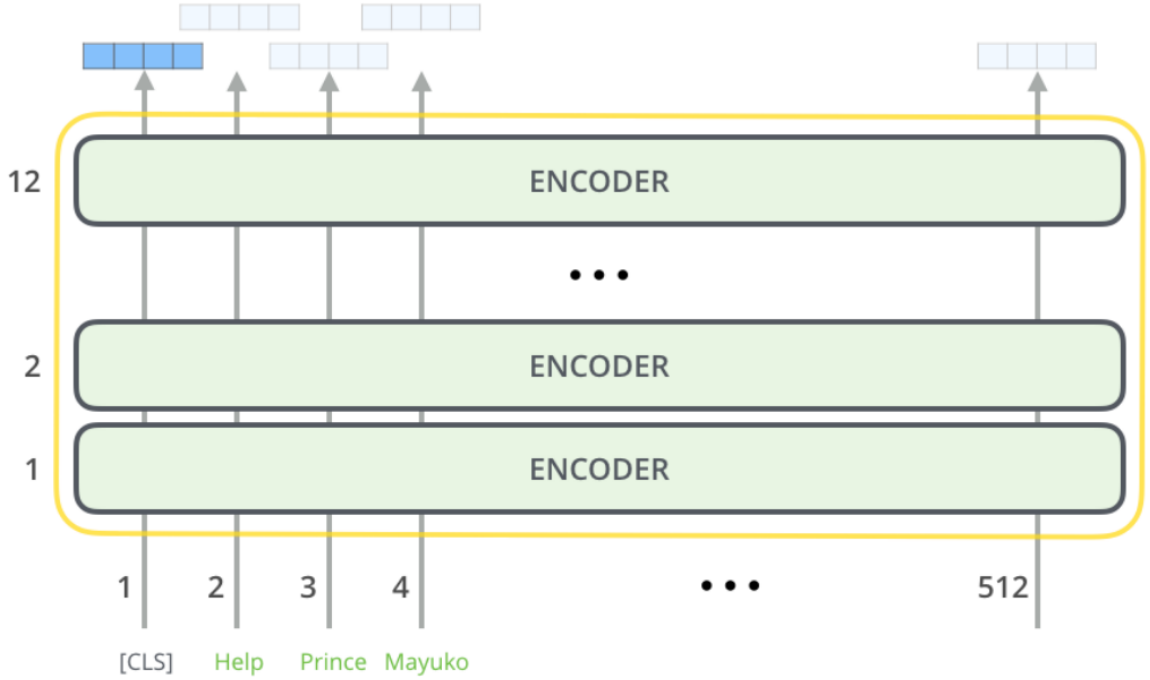
Задача – максимизировать отношение с помощью ММП методом оценки параметров.

$$\max\left(\frac{V^T m_1 - V^T m_2}{s_1 + s_2}\right)$$

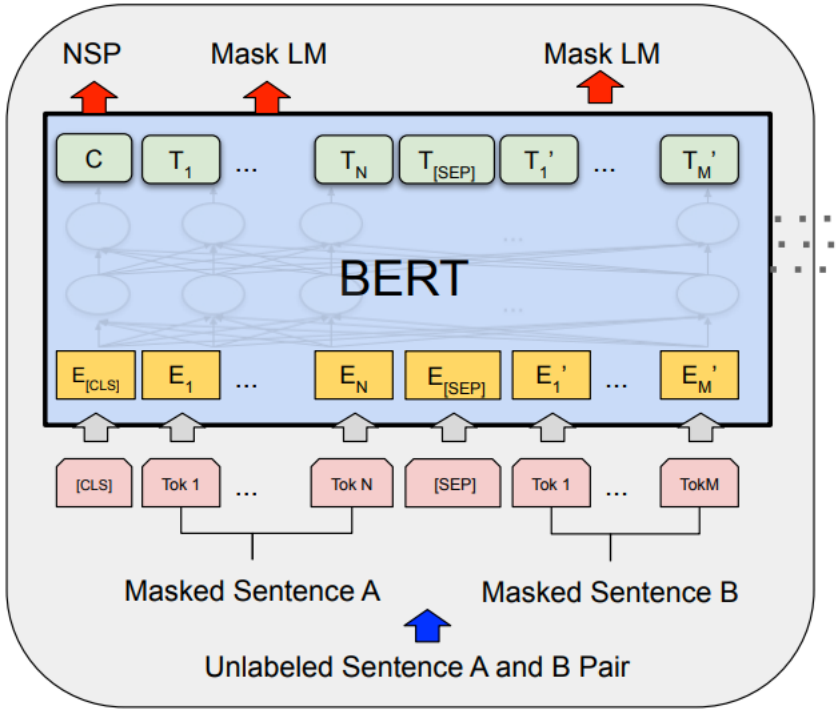
Aaaand we back…

BERT

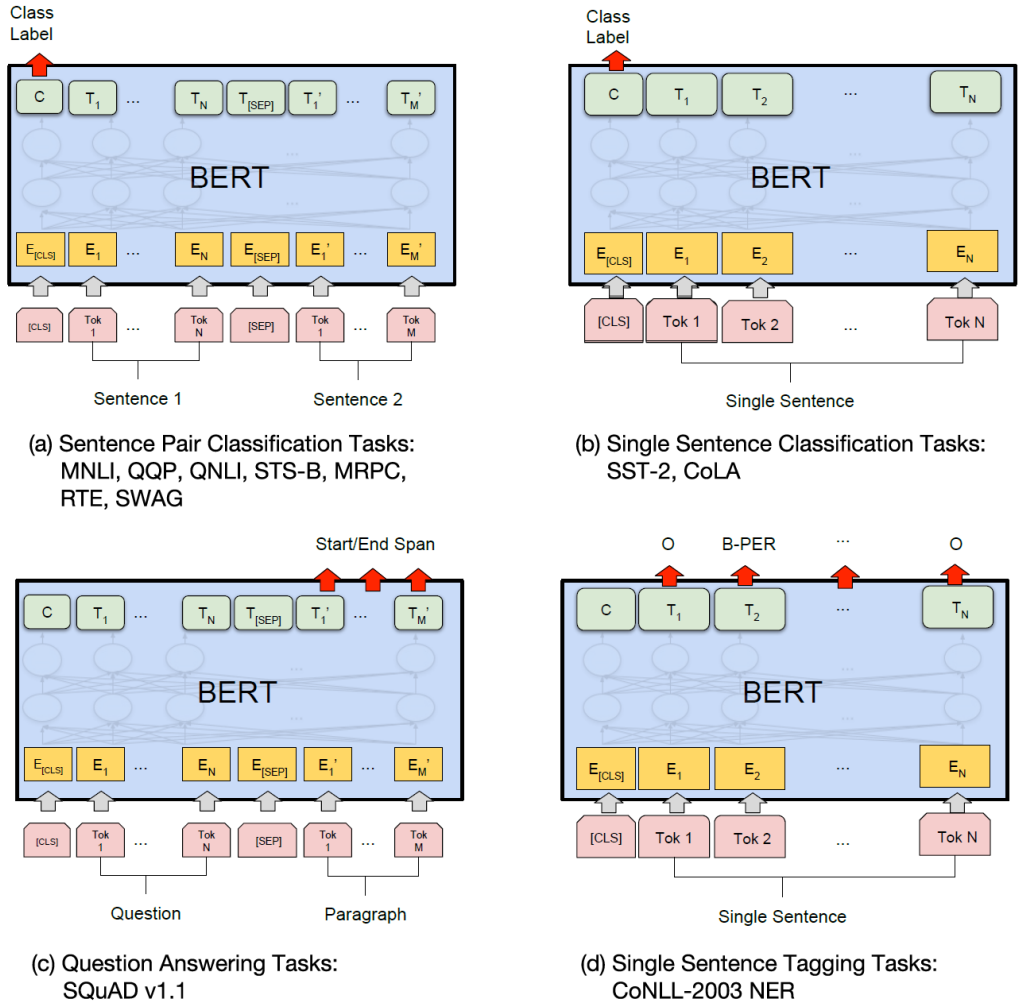

Add & Normalize
Feed Forward
Feed Forward
z1
z2
Add & Normalize
X
Z
LayerNorm(+)
ENCODER #1
z1
z2
Self-Attention
x1
x2
POSITIONAL ENCODING
x1
x2
Thinking
Machines


12
ENCODER
2
ENCODER
1
ENCODER
1
2
3
4
512
[CLS]
Help
Prince
Mayuko

Обучение BERT



Pretrain phase



Finetune phase

Metric Learning in Deep Learning

Before

After

Ключевые моменты обучения

Архитектура

Метрики валидации и теста

Выбор функции ошибки

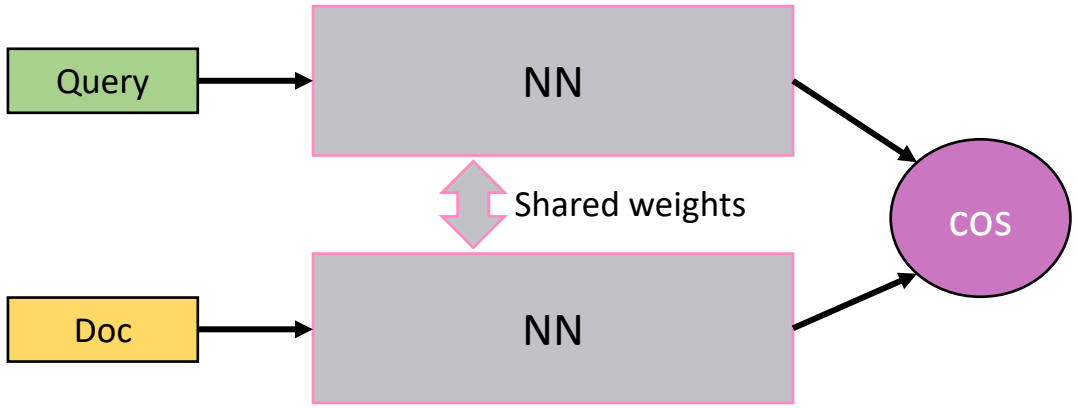
Sampling

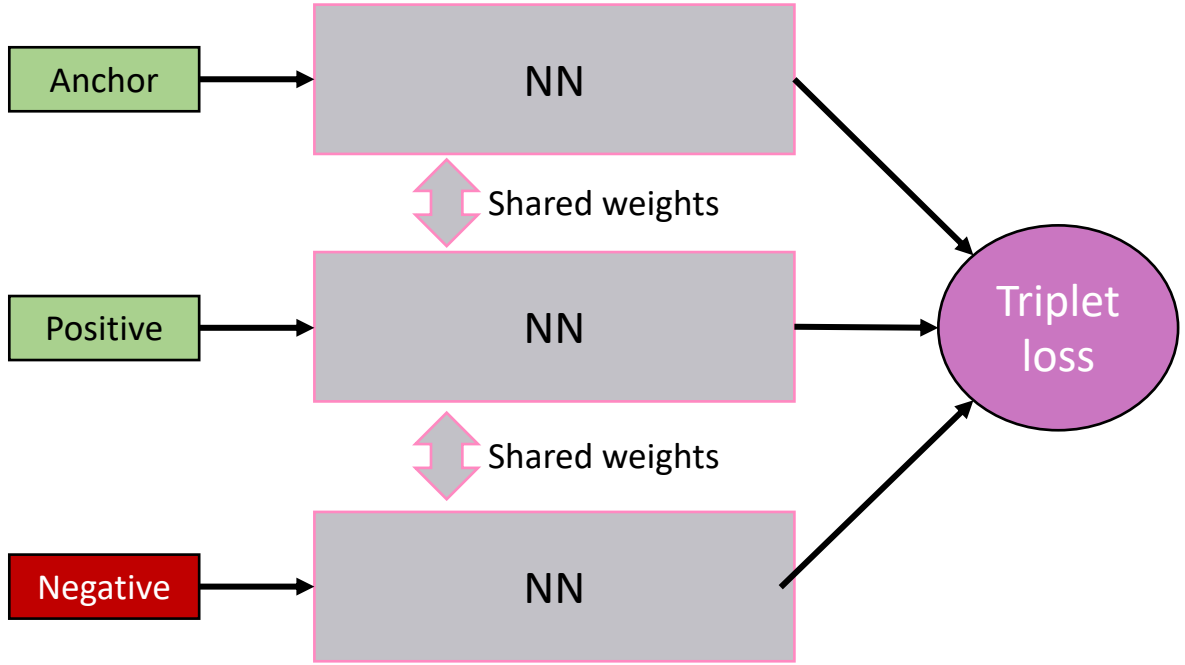
Подготовка данных

Train/Val Split

Архитектура

Deep Structured Semantic Model(DSSM)





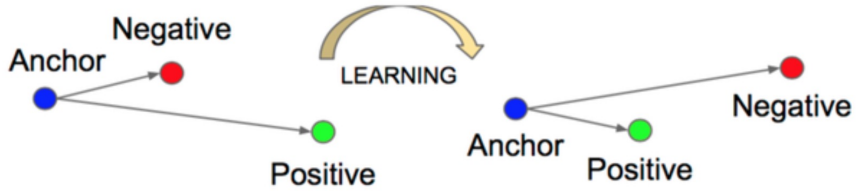
Функция ошибки

$$y_{pred} = dist(x_i, x_j)$$

$$y_{target} = \begin{cases} 1, if\ C_i = C_j \\ 0, otherwise \end{cases}$$

$$L = y_{target} * y_{pred} + (1 - y_{target}) * max(0, \delta - y_{pred})$$

Contrastive Loss



$$L = \max(0, dist(anchor, pos) + \delta - dist(anchor, neg))$$

$$if\ dist(anchor, pos) + \delta > dist(anchor, neg) \rightarrow \boldsymbol{optimize}$$

Distance metric – cosine & l2

Triplet Loss

Целое семейство функций ошибок для Metric Learning

SphereFace

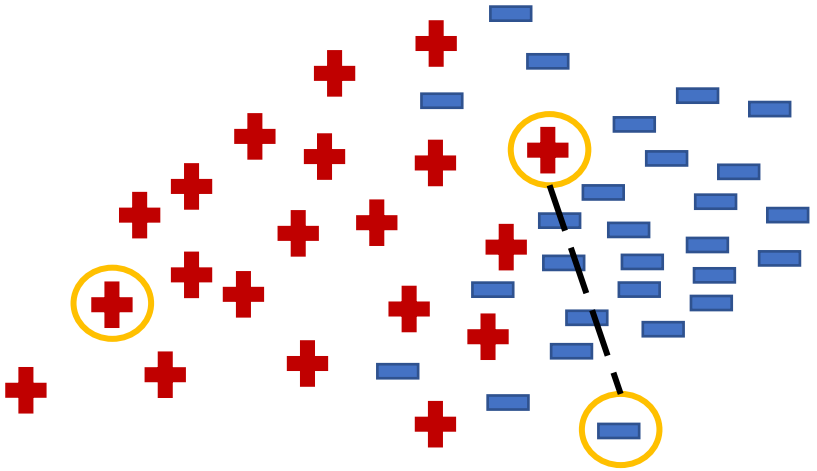
CosFace

ArcFace

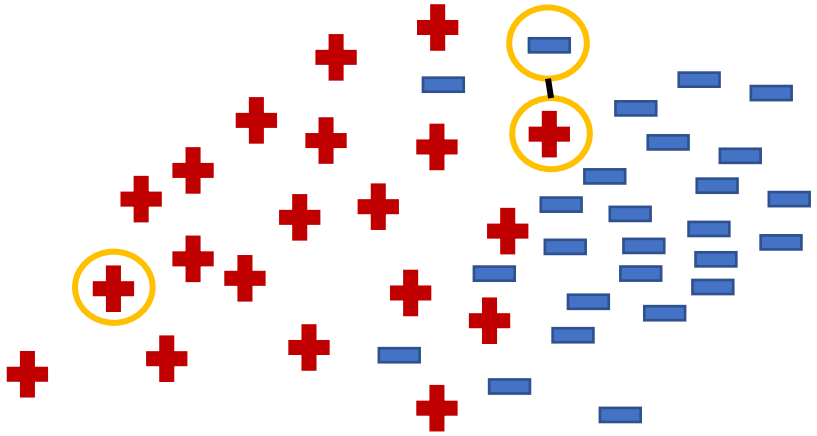
Softmax Cross-Entropy Loss

Hard Negative Mining

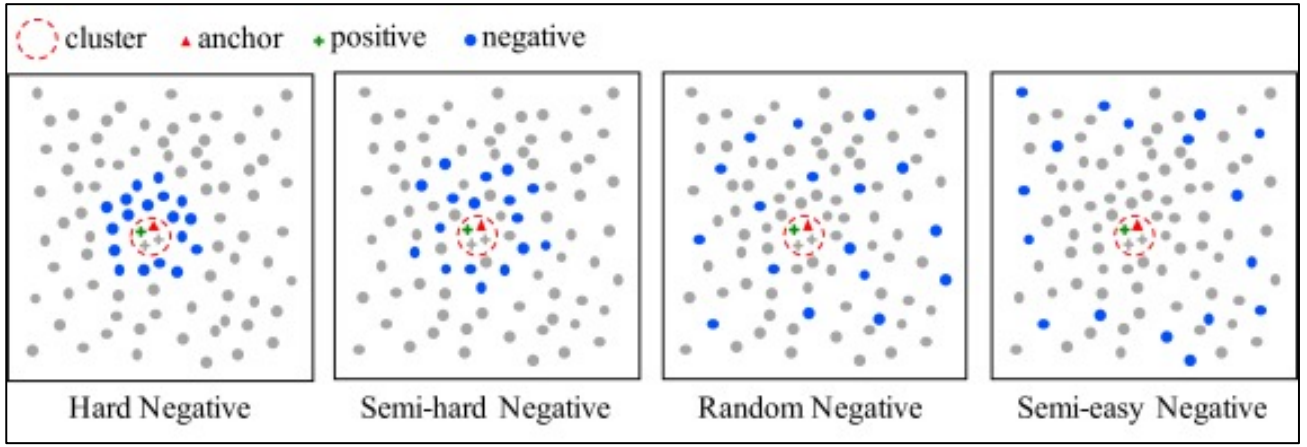
1) Random Mining

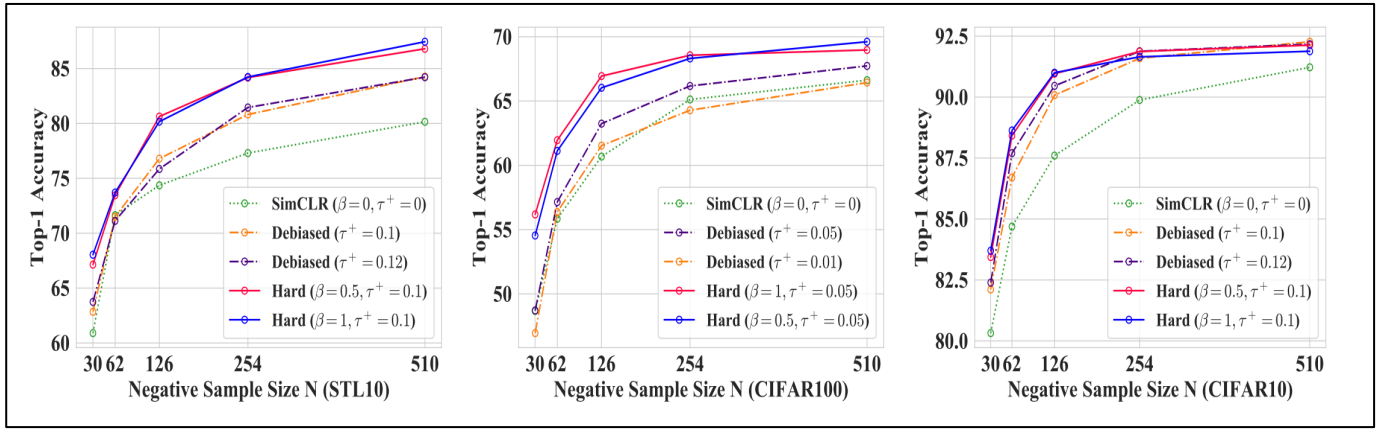


2) Hard Negative Mining



$$dist(pos, hard\ neg) \ll dist(pos, semi-hard) \ll dist(pos, easy-neg)$$





Hard Negative Sampling of CIFAR100

Разбиение и оценка качества

Разбиение данных - обучение на части классов и валидация на других классов – оценка способности обобщения модели на новые данные.

- Данные часто обновляется и каждый раз при добавлении нельзя дообучать систему.

При обучении

Оценка качества разделения объектов

При оценке

Оценка качества распознавания

1) **Separation Accuracy** – насколько хорошо мы разделяем объекты разных классов во время обучения.

$$R = \begin{cases} 1, if\ dist(x_{C_i}, y_{C_i}) < dist(x_{C_i}, y_{C_j}) \\ 0, if\ dist(x_{C_i}, y_{C_i}) \geq\ dist(x_{C_i}, y_{C_j}) \end{cases}$$

2) **Silhouette coefficient** – метрика качества разделения кластеров($-1 \leq s \leq 1$).

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j) \rightarrow min \qquad b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \rightarrow max$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

1) **Recall** – количество правильно подобранных документов в N кандидатах выдачи.

$$R = \frac{\#Relevant\ Docs}{N}$$

2) **Micro-recall** – взвешенный Recall относительно количества объектов в одном классе.

$$R = \frac{\sum_i TP_{C_i}}{\sum_i N_{C_i}}$$