

# MO2

March 2021

## Part I

## Лекции

### 1 Ядровые методы

Данные:  $x = (x_1, \dots, x_m)$

Базисные функции:  $\phi(x_1, \dots)$

Модель принимает вид:  $a(x) = \sum_{j=1}^m w_j \phi_j(x)$

Для хорошего качества нужно много базисных функций  $\rightarrow$  Ядровые методы позволяют не перебирать большое количество базисных функций

- Быстрое обучение

#### *Ядровые методы*

1. Двойственное представление для линейной регрессии

$$Q(w) = \frac{1}{2} \sum_{i=1}^l (\sum_{j=1}^m (w_j \phi_j(x_i) - y_i)^2 + \frac{\lambda}{2} \|w\|_2^2 = \frac{1}{2} \|\Phi w - y\|_2^2 + \frac{\lambda}{2} \|w\|_2^2$$

$$\Phi = \begin{pmatrix} \phi_1(x_1) & \dots & \phi_m(x_1) \\ \dots & \dots & \dots \\ \phi_1(x_l) & \dots & \phi_m(x_l) \end{pmatrix}$$

$$\nabla_w Q = \Phi^T (\Phi w - y) + \lambda w \rightarrow w = -\frac{1}{\lambda} \Phi^T (\Phi w - y) \rightarrow w = \Phi^T a$$

$w$  является линейной комбинацией строк  $\Phi \rightarrow$  Решение можно искать из  $w = \Phi^T a$

$$Q(a) = \frac{1}{2} \|\Phi \Phi^T a - y\| + \frac{\lambda}{2} a^T \Phi \Phi^T a \rightarrow \min_a$$

$\Phi \Phi^T$  - матрица Грама (попарных скалярных произведений объектов)

Можно записать  $Q(w)$  так, что он зависит только от скалярных произведений объектов

## 2. SVM

$$\begin{cases} \sum_{i=1}^l \lambda_i - \frac{1}{2} \sum_{i,j=1}^l \lambda_i \lambda_j y_i y_j < x_i, x_j > \rightarrow \max_{\lambda} \\ 0 \geq \lambda_i \leq C \\ \sum_{i=1}^l \lambda_i y_i = 0 \end{cases}$$

Такая формулировка задачи зависит от скалярных произведений объектов

## 3. Алгоритм

- (a) Добавляем новые признаки
- (b)  $x, z \in X$
- (c) Делаем это так, что  $\langle \phi(x), \phi(z) \rangle$  выражается через  $\langle x, z \rangle$
- (d) Используем метод, который использует скалярные произведения объектов
- (e) В этом методе  $\langle x, z \rangle \rightarrow \langle \phi(x), \phi(z) \rangle$  (*Kernel trick*)

## 4. Ядро - функция $K(x, z) = \langle \phi(x), \phi(z) \rangle$ , где $\phi : X \rightarrow H$

- (a)  $H$  - спрямляющее пространство
- (b)  $\phi$  - спрямляющее отображение

## 5. Теорема Мерсера

- (a)  $K(x, z)$  - ядро  $\leftrightarrow \begin{cases} K(x, z) = K(z, x) \\ K \text{ неотрицательно определенная} \end{cases}$
- (b)  $\text{НО} \rightarrow \forall l, \forall (x_1, \dots, x_l) \in R^d \rightarrow (K(x_i, x_j))_{i,j=1}^l \text{ НО}$
- (c) На практике теорема Мерсера слишком сложна для применения

## 6. Теорема 1

- (a) Если
  - i.  $K_1(x, z), K_2(x, z)$  - ядра,  $x, z \in X$

- ii.  $f^{(x)}$  - вещественная функция на  $X$
  - iii.  $\phi : X \rightarrow R^n$
  - iv.  $K_3$  - ядро заданное на  $R^n$
- (b) То следующие функции являются ядрами:
- i.  $K(x, z) = K_1(x, z) + K_2(x, z)$
  - ii.  $K(x, z) = \alpha K_1(x, z)$
  - iii.  $K(x, z) = K_1 K_2$
  - iv.  $K(x, z) = f^{(x)} f^{(z)}$
  - v.  $K(x, z) = K(\phi(x), \phi(z))$

## 7. Теорема 2

- (a) Если:
- i.  $K_1(x, z), K_2(x, z), \dots$  - последовательность ядер
  - ii.  $\exists K(x, z) = \lim_{n \rightarrow \infty} K_n(x, z), \forall x, z$
- (b) То:
- i.  $K$  - ядро

## 8. Полиномиальные ядра

- (a)  $p(v)$  - многочлен с неотриц. коэфф
- (b)  $K(x, z) = w_0 + w_1 \langle x, z \rangle + w_2 \langle x, z \rangle^2 + \dots$
- (c) Является ядром по теореме 1
- (d)  $K(x, z) = (\langle x, z \rangle + R)^m = \sum_{i=0}^m C_m^i R^{m-i} \langle x, z \rangle^i$
- i. Если расписать все  $\langle x, z \rangle^i$ , то получим все мономы степени  $i$  от исходных признаков
  - ii. Зачем  $R$ ?  $\rightarrow$  коэффициент при мономе  $= \sqrt{C_m^i R^{m-i}}$
  - iii. Сравним веса при мономах 1 и  $(m-1)$   $\sqrt{\frac{C_m^{m-1} R}{C_m^1 R^{m-1}}} = \sqrt{\frac{1}{R^{m-2}}}$
  - iv.  $R$  больше - мономы высоких степеней имеют низкий вклад
  - v. Конечномерное спрямляющее пространство, но можно сделать линейно разделимое пространство

## 9. Гауссовы ядра

- (a) Позволяет перевести в бесконечномерное спрямляющее пространство
- (b)  $K(x, z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$

$$i. \exp(\langle x, z \rangle) = \sum_{k=0}^{\infty} \frac{\langle x, z \rangle^k}{k!}, \forall x, z = \lim_{n \rightarrow \infty} \sum_{k=0}^{\infty} \frac{\langle x, z \rangle^k}{k!}$$

А. Разложение через ряд Тейлора

В. Ядро, как последовательность ядер

$$ii. \frac{\exp(\langle x, z \rangle)}{2\sigma^2} - \text{ядро, аналогично}$$

$$iii. \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\langle x-z, x-z \rangle}{2\sigma^2}\right) = \exp\left(-\frac{\langle x, x \rangle - \langle x, z \rangle - \langle z, z \rangle + \langle x, z \rangle}{2\sigma^2}\right) = \frac{\exp(\langle x, z \rangle / \sigma^2)}{\exp(\|x\|^2 / \sigma^2) \exp(\|z\|^2 / \sigma^2)}$$

$$iv. \exp(\langle x, z \rangle / \sigma^2) = K(x, z) = \langle \phi(x), \phi(z) \rangle$$

$$v. \tilde{\phi}(x) = \frac{\phi(x)}{\|\phi(x)\|} = \frac{\phi(x)}{\sqrt{K(x, x)}}$$

$$vi. \langle \tilde{\phi}(x), \tilde{\phi}(z) \rangle = \frac{\langle \phi(x), \phi(z) \rangle}{\sqrt{K(x, x)K(z, z)}}$$

(c) Какое спрямляющее пространство? - бесконечная сумма всех мономов

(d) Утверждение:  $x_1, \dots, x_l$  - различные векторы из  $\mathbb{R}^d$

Тогда:

$$G = \left(\exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)\right)_{i,j=1}^l - \text{невыврожденная при } \sigma^2 > 0$$

(e)  $x_1, \dots, x_l \in \mathbb{R}^d$  - их матрица Грамма невырождена  $\rightarrow \phi(x_1, \dots, x_l)$   
ЛНЗ  $\rightarrow$  бесконечное количество ЛНЗ векторов  $\rightarrow$  бесконечномерное пространство

## 10. Ядровой SVM

$$(a) \begin{cases} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w,b,\xi} \\ y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

$$L(w, b, \xi, \lambda, \mu) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^d \xi_i - \sum_{i=1}^l \lambda_i (y_i(\langle w, x_i \rangle + b) - 1 + \xi_i) - \sum_{i=1}^l \mu_i \xi_i$$

В точке оптимума  $\nabla_w L = 0$

$$\nabla_w L = w - \sum_{i=1}^l \lambda_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^l \lambda_i y_i x_i$$

$$\nabla_b L = \sum_{i=1}^l \lambda_i y_i = 0$$

$$\nabla_{\xi_i} L = C - \lambda_i - \mu_i = 0 \rightarrow \lambda_i + \mu_i = C$$

Условие дополняющей нежесткости:

$$\lambda_i(y_i(< w, x_i > + b) - 1 + \xi_i) = 0 \rightarrow \lambda_i = 0 \text{ или } (y_i(< w, x_i > + b) - 1 + \xi_i) = 0$$

$$\mu_i \xi_i = 0 \rightarrow \mu_i = 0 \text{ или } \xi_i = 0$$

Свойства лагранжиана:

$$\lambda \geq 0, \mu \geq 0$$

(b) Типы объектов

- i.  $\lambda_i = 0 \rightarrow \mu_i = C \rightarrow \xi_i = 0 \rightarrow x_i$  лежит с правильной стороны от разделяющей гиперплоскости и на достаточном расстоянии от нее.  $w = \sum_{i=1}^l \lambda y_i x_i \rightarrow$  объект не влияет на веса. Называется **периферийный**.
- ii.  $0 < \lambda_i < 1 \rightarrow \mu \neq 0 \rightarrow \xi_0 = 0$ .  $x_i$  не залезает на разделяющую полосу, но  $y_i(< w, x_i > + b) = 1 \rightarrow x_i$  лежит прямо на границе. Дает вклад в  $w$ .  $x_i$  - **опорный граничный**.
- iii.  $\lambda_i = C \rightarrow \xi_i > 0$ .  $x_i$  дает вклад в  $w$ .  $\xi_i > 0 \rightarrow x_i$  нарушает границу - **Опорные нарушители**.

(c) Подставляем  $w = \sum_{i=1}^l \lambda y_i x_i$  в лагранжиан, учтем ограничения  $\sum_{i=1}^l \lambda_i y_i = 0$  и  $C - \lambda_i - \mu_i = 0$

**Двойственная задача SVM**

$$\begin{cases} L = \sum_{i=1}^l \lambda_i - \frac{1}{2} \sum_{i,j=1}^l \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle \rightarrow \max_{\lambda} \\ \sum_{i=1}^l \lambda_i y_i = 0 \\ 0 \leq \lambda_i \leq C \end{cases}$$

(d) Если  $\lambda$  - решение, то  $w = \sum_{i=1}^l \lambda_i y_i x_i$  - решение исходной задачи

(e) Задача зависит от объектов только через скалярное произведение  $\rightarrow$  можно заменить его на ядро

(f) Находим b Берем  $x_i : 0 < \lambda_i < C \rightarrow \xi_i = 0 \rightarrow y_i(< w, x_i > + b) = 1 \rightarrow b = y_i - \langle w, x_i \rangle$

(g) Минусы ядрового SVM

- i. Сложно контролировать переобучение
- ii. Необходимо хранить в памяти матрицу Грамма

iii. Нельзя менять функцию потерь

## 11. Применение ядерной модели

$$(a) \ a(x) = \text{sign}(\langle w, x \rangle + b) = \text{sign}(\langle \sum_{i=1}^l \lambda y_i x_i, x \rangle + b) = \text{sign}(\sum_{i=1}^l \lambda_i y_i \langle x_i, x \rangle + b)$$

## 2 Аппроксимации ядер, ЕМ алгоритм

Скалярные произведения тяжело хранить из-за размера матрицы.  
Есть ли возможность построить  $\tilde{\phi}(x) \rightarrow \langle \tilde{\phi}(x_i), \tilde{\phi}(x_j) \rangle \approx K(x_i, x_j)$

### 2.1 Метод случайных признаков Фурье

$$K(x, z) = K(x - z)$$

$K$  - непрерывная функция

*Теорема Бохнера*

$$K(x - z) \rightarrow \exists p(w) \rightarrow K(x - z) = \int_{R^d} p(w) e^{iw^T(x-z)} dw$$

*Используем:*

$$K(x-z) = \int_{R^d} p(w) e^{iw^T(x-z)} dw \xrightarrow{\text{Формула Эйлера}^1} \int_{R^d} p(w) \cos(w^T(x-z)) + i \int_{R^d} p(w) \sin(w^T(x-z))$$

$$\xrightarrow{K(x-z) - \text{веществ.}} \text{Комплексная часть} = 0 \rightarrow K(x-z) = \int_{R^d} p(w) \cos(w^T(x-z)) dw$$

$$\xrightarrow{\text{Монте-Карло}^2} K(x-z) \approx \{w_j \sim p(w)\} : \frac{1}{n} \sum_{i=1}^n \cos w_j^T(x-z)$$

$$= \frac{1}{n} \sum_{i=1}^n \cos w_j^T x \cos w_j^T z + \sin w_j^T x \sin w_j^T z$$

$$\tilde{\phi}(x) = \frac{1}{n} (\cos w_1^T x, \dots, \cos w_n^T x, \sin w_1^T x, \dots, \sin w_n^T x)$$

$$K(x-z) = \langle \tilde{\phi}(x), \tilde{\phi}(z) \rangle$$

---

<sup>1</sup>  $e^{ix} = \cos x + i \sin x$

<sup>2</sup>  $\int_a^b f(x) dx = \frac{b-a}{n} \sum_{i=1}^N f(u_i)$

Для гауссова ядра:

$$p(w) = \mathcal{N}(0, 1)$$

## 2.2 ЕМ алгоритм

Смесь распределений:

$$\begin{cases} p(x) = \sum_{k=0}^K \pi_k p_k(x) \\ \sum \pi_k = 1 \end{cases}$$

Вероятностный эксперимент:

Выбираем  $K$  из  $[\pi_1, \dots, \pi_K]$ , выбираем  $x$  из  $p_k(x)$

$Z$  - скрытые переменные

$$Z = \{0, 1\}^K, \sum Z_k = 1$$

$$p(Z_k = 1) = \pi_k$$

$$p(z) = \prod_{k=1}^K \pi_k^{Z_k}$$

$$p(x | Z_k = 1) = p_k(x)$$

$$p(x | z) = \prod_{k=1}^K (p_k(x)^{Z_k})$$

$$p(x, z) = p(x | z)p(z) = \prod_{k=1}^K (\pi_k p_k(x))^{Z_k}$$

$$p(x) = \sum_{k=1}^K p(x, z = k) = \sum_{k=1}^K \pi_k p_k(x)$$

Вероятностная кластеризация:

$p_k(x)$  - распределение  $k$ -го кластера

$$x \rightarrow (p_1(x), \dots, p_K(x))$$

Хотим описать  $X$  смесью распределений

$$p(x) = \sum_{k=1}^K \pi_k \phi(x | \theta_k), \phi(x | \theta_k) \sim \mathcal{N}(\mu, \Sigma), \theta = (\mu, \Sigma)$$

Неполное правдоподобие:

$$\ln(P(X | \Theta)) = \sum_{i=1}^l \log \sum_{k=1}^K \pi_k \phi(x_i | \theta_k) \rightarrow \max_{\theta}$$

Логарифм многооптимальная функция - просто оптимизировать ее сложно  
Используем функцию полного правдоподобия

$$\log(P, X | \Theta) = \sum_{i=1}^l \log \sum_{k=1}^K (\pi_k \phi(x_i | \theta_k))^{Z_k}$$

$$\sum_{i=1}^l \sum_{k=1}^K Z_{ik} (\log \pi_k + \log \phi(x_i | \theta_k)) \rightarrow \max_{\Theta}$$

Известно аналитическое решение для нормального распределения.  
Не знаем  $Z_{ik}$

$$\Theta = (\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_k)$$

Используем метод ALS для поиска  $Z, \Theta$

1. Оптимизация по скрытым переменным

Апостериорное распределение:  $p(Z | X, \Theta) = \frac{P(X, Z | \Theta)}{p(X | \Theta)}$

$$Z^* = \arg \max_Z p(Z | X, \Theta)$$

2. Оптимизировать по  $\Theta$

$$\log p(X, Z^* | \Theta) \rightarrow \max_{\Theta}$$

3. Повторять до сходимости

Можно лучше. Не гарантирует сходимости

ЕМ-алгоритм - метод обучения моделей со скрытыми переменными

**ЕМ-алгоритм**

1. Е-шаг - вычисляем  $p(Z | X, \Theta)$  и запоминаем

2. М-шаг

$$E_{Z \sim p(Z | X, \Theta)} \log p(X, Z | \Theta) = \sum_Z p(Z | X, \Theta) \log p(X, Z | \Theta) \rightarrow \max_{\Theta}$$



Вывод ЕМ-алгоритма

$$\log p(X | \Theta) = Z(q, \Theta) + KL(q || p)$$

$$L(q, \Theta) = \sum_Z q(Z) \log \frac{p(X, Z | \Theta)}{q(Z)}$$

$$KL(q || p) = - \sum_Z q(Z) \log \frac{p(Z | X, \Theta)}{q(Z)}$$

$$\forall q(Z)$$

$L(q, \Theta)$  - нижняя оценка

Берем  $q(Z) = p(Z | X, \Theta)$  - получаем Е-шаг

$L(q, \Theta) = \sum_{Z \sim q(Z)} p(Z) \log(\dots)$  - М-шаг

ЕМ-алгоритм дает гарантии на рост правдоподобия

### 3 ЕМ алгоритм 2

**Свойства**

1.  $\log P(X | \Theta^{new}) \geq \log P(X | \Theta^{old})$
2. Если  $\Theta_i$  не является стационарной точкой l, то  $\Theta_{i+1} \neq \Theta_i$

$$\nabla l(\Theta_i) \neq 0$$

$$\log P(X | \Theta_i) = L(q | \Theta_i) + KL(q(\Theta_i) || p)$$

$$KL = 0 \rightarrow \nabla_{\Theta} KL(q || p) = 0 \rightarrow \nabla L(q | \Theta_i) \neq 0 \rightarrow$$

На М шаге точно сдвинемся и поменяем  $\Theta$

**Теорема**

$$Q(\Theta, \Theta^{Old}) = E_{z \sim p(Z|X, \Theta^{Old})} \log P(Z, X | \Theta^{Old})$$

Пусть Q непрерывна по обоим аргументам

Тогда:

1. Все предельные точки последовательности  $\Theta$  являются стационарными точками  $\log P(X | \Theta)$

2.  $\log P(X \mid \Theta)$  монотонно сходится к  $\log P(X \mid \Theta^*)$  - одной из стационарных точек

Отвлеченная штука:

$X$  - обучающая выборка

Хотим подогнать под нее распределение  $p(x \mid \theta)$

Эмпирическое распределение:

$$\hat{p}(x \mid X) = \frac{1}{l} \sum_{i=1}^l [x = x_i]$$

Минимизировать KL-дивергенцию между эмпирическим и параметрическим распределением.

$$KL(\hat{p}(x \mid X) \parallel p(x \mid \theta)) \rightarrow \min_{\theta}$$

$$\begin{aligned} &= \sum_{i=1}^l \frac{1}{l} \log \frac{1/l}{p(x_i \mid \theta)} = \sum_{i=1}^l \frac{1}{l} \log(1/l) - \log p(x_i \mid \theta) \rightarrow \\ &\sum_{i=1}^l \log p(x_i \mid \theta) \rightarrow \max_{\theta} \end{aligned}$$

## 4 Поиск аномалий

В обучении есть только один класс - неаномальный, надо научиться отделять от него аномалии

### Несбалансированная классификация

1. (Under/over)sampling - взвешенный функционал ошибки
2. Синтетические объекты

(a) SMOTE

- i. Выбираем объекты  $X_1$  из минорного класса, выбираем случайный объект из  $k$  ближайших соседей тоже из минорного класса  $X_2$
- ii. Новый объект:  $X = \alpha X_1 + (1 - \alpha) X_2, \alpha \sim U(0, 1)$
- iii. Предполагает существование объектов между  $X_1, X_2$

(b) Аугментации

## Одноклассовая классификация

Бенчмарк: Классификация X на нормальные и аномальные, стандартные метрики

1. Статистический подход - описываем плотностью  $p(x)$  для новых объектов смотрим на вероятность -  $p(x)$  - novelty score.

Откуда брать  $p$

- (a) Параметрический подход:

$$\sum_{i=1}^l \log P(x | \theta) \rightarrow \max_{\theta}$$

- (b) Непараметрический подход:

- i.  $d = 1$ :

$$p(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(\xi \in [x - h, x + h])$$

$$\begin{aligned} \hat{p}(x) &= \frac{1}{2h} \frac{1}{l} \sum_{i=1}^l \mathbb{I}[|x_i - x| < h] = \\ &= \frac{1}{lh} \sum_{i=1}^l \frac{1}{2} \mathbb{I}\left[\frac{|x_i - x|}{h} < 1\right] \end{aligned}$$

Можно заменить на более гладкую плотность:

$$\frac{1}{lh} \sum_{i=1}^l \frac{1}{2} K\left(\frac{x_i - x}{h}\right)$$

- A.  $K(z) = K(-z)$   
B.  $\int_{\mathcal{R}} K(z) dz = 1$   
C.  $K(z) \geq 0$   
D. Не возрастает при  $Z > 0$

- ii.  $d > 1$ :

$$\hat{p}(x) = \frac{1}{lV(h)} \sum_{i=1}^l K\left(\frac{\rho(x_i, x)}{h}\right)$$

$$V(h) = \int K\left(\frac{\rho(x_i, x)}{h}\right) dx$$

$h$  - гиперпараметр

## 2. Метрический подход

$x$  - аномалия, если он далеко от других объектов

Смотреть на количество объектов в  $\epsilon$ -окрестности?

Плохой подход:

Надо смотреть не на единую окрестность, а смотреть на плотность объектов в отдельной точке и на основе нее оценивать окрестность

**Определения:**

(a)  $\rho_k(x)$  - такое минимальное число  $n$ , что:

Для  $\geq k$  объектов из  $X/\{x\}$  выполнено  $\rho(x, z) \leq n$

Для  $\leq k-1$  объектов выполнено  $\rho(x, z) < n$

По сути: расстояние до  $k$ -го ближайшего соседа

(b)  $k$ -окрестность:

$$\mathcal{N}_k(x) = \{z \in X/\{x\} : \rho(x, z) \leq \rho_k(x)\}$$

(c) Reachability Distance:

$$rd_k(x, z) = \max(\rho_k(z), \rho(x, z))$$

Позволяет сгладить расстояние между объектами

(d) Local Reachability Distance

$$lrd_k(x) = \frac{1}{\frac{1}{|\mathcal{N}_k(x)|} \sum_{z \in \mathcal{N}_k(x)} rd_k(x, z)}$$

Обращенное среднее расстояние от  $x$  до ближайших соседей

(e) Local Outlier Factor

$$LOF_k(x) = \frac{\frac{1}{|\mathcal{N}_k(x)|} \sum_{z \in \mathcal{N}_k(x)} lrd_k(z)}{lrd_k(x)}$$

Отлавливаем объекты у которых соседи находятся в плотных областях, но сами они находятся далеко от соседей

## 3. Model-based AD

(a) Есть примеры нормальных объектов

(b) Хотим найти наименьшую область, содержащую все объекты

$$a(x) = \text{sign}(\langle w, x \rangle - \rho)$$

Идея:

Отделяем  $X$  от начала координат с помощью  $a()$

$$\begin{cases} \frac{1}{2} \|w\|^2 + \frac{1}{\nu \ell} \sum \xi_i - \rho \rightarrow \min_{w, \xi, \rho} \\ \langle w, x_i \rangle \geq \rho - \xi_i, \forall i \\ \xi_i \geq 0 \end{cases}$$

$\nu$  - гиперпараметр

$$\sum [a(x) = -1] \leq \nu$$

Требования к решению:

- i. Отделить как можно больше объектов от 0. За это отвечает  $\sum \xi_i$
- ii. Максимизировать отступ. За это отвечает  $\|w\|^2$
- iii. Гиперплоскость как можно дальше от нуля. За это отвечает  $\rho$

$$a(x) = \text{sign}(\langle w, x \rangle - \rho)$$

Можно записать двойственную задачу:

$$\begin{cases} \frac{1}{2} \sum \lambda_i \lambda_j K(x_i, x_j) \rightarrow \min_{\lambda} \\ 0 < \lambda_i \leq \frac{1}{\nu \ell} \\ \sum \lambda_i = 1 \end{cases}$$

С помощью ядра получаем компактную область

#### 4. Random projections

##### (a) Isolation Forest

Строим жадное дерево со случайными предикатами по случайным признакам

Если в каком-то листе оказывается 1 объект - прекращаем разбиение

Аномальные объекты рано получают свой лист

*Обучение:*

Строим лес из  $N$  деревьев, в каждом случайные предикаты.

Максимальная глубина  $D = \log_2 \ell$

*Применение:*

$h_n(x)$  - оценка аномальности  $x$  с точки зрения  $n$  дерева

$K_n(x)$  - глубина листа в который попадает  $x$  в  $n$  дереве

Нужно сделать поправку на количество объектов в листе

$$h_n(x) = K_n(x) + C(m_n(x))$$

$$C(m) = 2H(m-1) - 2\frac{m-1}{m}$$

$$H(i) \approx \ln i + 0.577$$

Можно использовать и  $\log_2(m)$

$$a(x) = 2^{-\frac{\frac{1}{N} \sum_{n=1}^N h_n(x)}{C(l)}}$$

$C(l)$  - средняя длина пути

(b) Extra Random Trees

Берем индикатор попадания в листья и строим линейную модель

*Как измерять качество:*

1. Anomaly detection

Есть примеры аномалий, но мало данных

Смотрим какое количество аномалий модель угадывает

2. Novelty detection

Аномалии не даны, качество модели оценивается глазами

## 5 Обучение без учителя

1. Кластеризация: DBScan, Спектральная классификация, Affinity Propagation

2. Внешние метрики качества кластеризации

3. Тематическое моделирование

**K-means:**

Основная проблема - ищет сферические кластеры

## 5.1 DBScan

Типы объектов:

1. Ядровые:  
В  $\epsilon$ -окрестности находится N объектов
2. Пограничные объекты:  
Достижимы из ядровых, находится в  $\epsilon$ -окрестности ядрового
3. Выбросы:  
Все остальные

Псевдокод:

```
K = 0 #Num clusters
rho #metric
epsilon , N #hyperparam
for i = 1 ... l:
    if label(x[i]) != 0:
        continue
    #point neighborhood
    U = {x in X | rho(x[i], x[j]) <= epsilon}
    if |U| < N:
        label(x[i]) = noise
        continue
    K++ # found new cluster if |U| > N
    label(x[i]) = K
    U = U \ {x[i]}
    for x[j] in U:
        if label(x[j]) = noise:
            label(x[j]) = K
        if label(x[j]) != 0:
            continue
        label(x[j]) = K
    #point neighborhood
    R = {xm in X | rho(x[m], x[j]) < epsilon}
    if |R| >= N:
        #new core object neighborhood included in U
        U = U & R
```

Преимущества:

1. Находит кластеры сложной формы
2. Находит выбросы
3. Быстрый
4. Не надо задавать число кластеров

Недостатки:

1. Проблемы если кластеры разной плотности
2. Проблемы с точками на краях
3. Не работает если кластеры характеризуются неплотность
4. Не параллелится

## 5.2 Иерархическая кластеризация

Цель: Найти кластерную структуру

Визуализировать разную структуру кластеров при разном их количестве

*Агломеративная кластеризация*

Начинаем с того, что каждый объект является кластером

Псевдокод:

```

C = {{x[1]}, {x[2]}, ..., {x[l]}}
for n = 2, ..., l:
    G, H = argmin rho(G, H) #find nearest clusters
    C = (C \ {G, H}) U {G U H}

```

Функция расстояния между кластерами  $\rho$ :

1. Single Linkage:

$$\rho_{sl}(G, H) = \min_{x_i \in G, x_j \in H} \rho(x_i, x_j)$$

Чувствителен к выбросам

Главная проблема: Chaining

Алгоритм подцепляет отдельные объекты, а не кластеры

Дендрограмма - картинка присоединения объектов



## 2. Complete linkage

$$\rho_{cl}(G, H) = \max_{x_i \in G, x_j \in H} \rho(x_i, x_j)$$

Кластеры не будут компактными

## 3. Group Average

$$\rho_{ga}(G, H) = \frac{1}{|G| |H|} \sum_{x_i \in G, x_j \in H} \rho(x_i, x_j)$$

## 5.3 Графовая кластеризация

$G = (X, E)$

$E$  - ребра:

### 1. Полный граф - все вершины связаны

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

### 2. KNN-граф:

$$w_{ij} \neq 0 \Leftrightarrow x_i, x_j \text{ ближайшие соседи}$$

### 3. $\epsilon$ -граф:

$$w_{ij} \neq 0 \Leftrightarrow \rho(x_i, x_j) \leq \epsilon$$

*Поиск решение*

### 1. Найти связные компоненты (для Зего варианта)

Тупой метод

### 2. Минимальное остовное дерево (Алгоритм Краскала)

(a) Начинаем с отдельных вершин

(b) Сливаем два кластера с максимальным ребром между ними

(c) Повторить пока не будет  $K$  кластеров

(d) Это агломеративная кластеризация с sl

(e) Решает задачу:

$$\max \min_{x_i \in G, x_j \in H} \rho(x_i, x_j)$$

### 3. Спектральная кластеризация

$$A, B \subset X, A \cap B = \emptyset$$

$$W(A, B) = \sum_{x_i \in A, x_j \in B} w_{ij}$$

$$X = A_1 \cup A_2 \cup \dots \cup A_k$$

Ошибка кластеризации:

$$\text{Ratio Cut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^K \frac{w(A_i, \bar{A}_i)}{|A_i|} \rightarrow \min_{A_1, \dots, A_k} (\star)$$

$$\bar{A}_i = X \setminus A_i$$

Хотим, чтобы ребра между кластерами были как можно менее значимы  $\rightarrow$  Каждый кластер должен быть изолированным

$K = 2 \rightarrow$  Задача поиска максимального потока

$K > 2 \rightarrow$  NP-полная задача

---

$$G = (X, E)$$

$d_i = \sum_{j=1} dw_{ij}$  - сумма ребер, которые с ней связаны

$$D = \text{diag}(d_1, \dots, d_l)$$

$L = D - W$ ,  $W$  - матрица смежности,  $L$  - **лаплассиан**

*Свойства лаплассиана*

(a)

$$f \in R^d$$
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^l w_{ij} (f_i - f_j)^2$$

(b)  $L$  - симметричная

(c)  $L$  - неотрицательно определенная

**Теорема**

(a) Кратность  $\lambda = 0$  у  $L$  равна числу компонент связности графа  
Кратность:

- i. Собственные значения:  $Ax = \lambda x$
- ii.  $\det(A - \lambda I) = 0 \rightarrow \lambda_i$
- iii.  $\lambda_i$  - решение
- iv.  $\lambda_i, \forall i$  - спектр графа
- v. Характеристическое уравнение выражаем в виде характеристического многочлена и раскладываем его в виде решений

$$P_A(\lambda) = (-1)^n \prod_{i=1}^n (\lambda - \lambda_i) = (-1)^n (\lambda - \lambda_1)^{k_1} \dots$$

$k_1$  называется кратностью для  $\lambda_1$

(b)  $A_1, \dots, A_k$

Вектор индикатор:  $f_i = ([x_j \in A_i])_{j=1}^\ell$

$f_1, \dots, f_k$  - собственные векторы для  $\lambda = 0$

Доказательство:

$K = 1$ :

(a) Является ли  $\lambda = 0$  собственным значением

$$f = (1, \dots, 1)$$

$$Lf = Df - Wf = \begin{pmatrix} d_1 \\ \vdots \\ d_\ell \end{pmatrix} - \begin{pmatrix} \sum w_{1,j} \\ \vdots \\ \sum w_{\ell,j} \end{pmatrix} = 0$$

(b) Кратность  $\lambda = 0 = 1 \rightarrow$  нет других собственных векторов

Допустим:

$$\exists f' \in R^\ell : \exists p \neq q, f'p \neq f'q, Lf' = 0$$

$$(f')^T Lf' = \frac{1}{2} \sum_{i,j=1}^\ell w_{ij} (f'_i - f'_j)^2 = 0$$

$$\forall i, j : \begin{cases} w_{ij} = 0 & \text{Нет ребра} \\ f'_i = f'_j \end{cases}$$

Граф G связный  $\rightarrow$  Существует путь из p в q  $\rightarrow$

Путь:  $w \rightarrow i_1 \rightarrow \dots \rightarrow p$

$$w_{pi_1} \neq 0 \rightarrow f'p = f'_{i_1} \rightarrow w_{i_1 \dots} \neq 0 \rightarrow \dots \rightarrow f'_p = f'_{i_1} = \dots = f'_q$$

$\perp$

$K > 1$ :

Можно упорядочить объекты так, чтобы  $L$  был блочно-диагональным

$$L = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ & \ddots & \\ 0 & 0 & L_K \end{pmatrix}$$

Спектр блочно-диагональной матрицы = объединение спектров отдельных блоков

$$L_i \rightarrow f_i = ([x_j \in A_i])_{j=1}^\ell$$

Кратность  $\lambda = 0$  равна  $K$

---

Гипотеза:  $x_i, x_j$  - похожие объекты  $\Rightarrow$  у собственного вектора  $f_i$  для  $\lambda_i \approx 0$ , будет  $f_{ij} \approx f_{ik}$

Для связанных графов не берем  $\lambda = 0$ , т.к. тогда будет одна компонентна

**Алгоритм:**

- (a) Строим лапласиан
- (b) Находим  $m$  (гиперпараметр) нормированных собственных векторов  $u_1, \dots, u_m$ , соотв. наименьшим собственным значениям.  
Сложность:  $O(l^3)$
- (c)  $U = (u_1 \mid \dots \mid u_m) \in R^{\ell \times m}$
- (d) Новые признаки близки для объектов в одной плотной области
- (e) K-means

**Как это связано с задачей ( $\star$ ):**

Если эту задачу релаксировать и искать не жесткое приписывание к классам, а распределение, то ее решение  $U$ .

## 6 Метрики качества классификации, тематическое моделирование

### 6.1 Affinity Propagation

Цель - найти типовые объекты и на их основе выделять кластеры

Сходство вершин:

$$s(i, k) = - \|x_i - x_k\|^2$$

$r(i, k)$  - Насколько  $x_k$  является типовым объектом для  $x_i$

$a(i, k)$  - Насколько у  $x_i$  важный голос для типового объекта

Инициализируем 0 и итеративно рассчитываем показатели:

$$r(i, k) = s(i, k) - \max_{k' \neq k} (a(i, k') + s(i, k'))$$

Если рядом есть более близкие объекты, чем  $k$ , то  $k$  не очень хороший представитель.

$$a(i, k) = \min(0, r(k, k) + \sum_{i' \neq i, i' \neq k} \max(r(i', u)))$$

$$\alpha(x_i) = \arg \max_{k \in \{1, \dots, \ell\}} (r(i, k) + a(i, k))$$

## 6.2 Оценка качества кластеризации

### 1. Внутренние

- (a) Без использование лейблов
- (b) Внутрикластерное расстояние
- (c) Межкластерное расстояние

### 2. Внешние

- (a) Знаем истинные номера кластеров  $y_1, \dots, y_\ell$
- (b) Номера кластеров нельзя сравнивать с истинными
- (c) Посчитать  $K!$  перестановок и найти классы?
- (d) **Требования к метрике**

#### i. Гомогенность

Значение метрики качества должно уменьшаться при объединении в один кластер двух эталонных

#### ii. Полнота

Значение метрики качества должно уменьшаться при разделении эталонного кластера на части

iii. Rag bag

Значение метрики качества должно быть выше у той версии кластеризации, которая помещает новый нерелевантный обоим кластерам элемент в шумный кластер, по сравнению с версией, которая помещает этот элемент в чистый кластер

iv. Cluster size vs quantity

Значительное ухудшение кластеризации большого числа небольших кластеров должно обходиться дороже небольшого ухудшения кластеризации в крупном кластере.

(e) **Метрики**

i. Bcubed

$y(x)$  - номер кластера в истинной разметке

$a(x)$  - выход кластеризации

Correctness:

$$C(x_i, x_j) = \begin{cases} 1, & y(x_i) = y(x_j) \quad a(x_i) = a(x_j) \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Precision-Bcubed} = \text{Avg}_{x_i}(\text{Avg}_{x_j, a(x_i)=a(x_j)}(C(x_i, x_j)))$$

$$\text{Recall-Bcubed} = \text{Avg}_{x_i}(\text{Avg}_{x_j, y(x_j)=y(x_i)}C(x_i, x_j))$$

$$F_{\text{Bcubed}} = 2 \frac{\text{PrecisionRecall}}{\text{Precision} + \text{Recall}}$$

## 6.3 Подбор метрик для продукта

Вводим набор ухудшений

## 7 Тематическое моделирование

Методы кластеризации для текстов

Есть  $T$  тематик

$x_d$  - документ

$\theta_d \in R^T$  - распределение тематик для документа

$\phi_t \in R^W$  - тема описывается распределением на словах

$W$  размер словаря

## 7.1 LSA (Latent Semantic Analysis)

$$X \in R^{d \times W}$$

$X_{dw}$  - сколько раз слово  $w$  входит в документ  $d$

$$X = \Theta \times \Phi, \Theta \in R^{d \times T}, \Phi \in R^{T \times W}$$

$$x_{dw} = \sum_{t=1}^T \theta_{dt} \times \phi_{wt}$$

Делаем SVD для разложения

## 7.2 PLSA

Хотим ввести вероятности

$$p(t \mid d) = \theta_{td}$$

$$p(w \mid t) = \phi_{wt}$$

Генерация текста  $x_d$

1. Сэмплируем тему  $t \sim p(t \mid d)$
2. Сэмплируем слово  $w \sim p(w \mid t)$
3. Добавляем слово в текст
4. Повторяем до нужной длины

$\theta_{dt}, \phi_{tw}$  - параметры модели

Неполное правдоподобие данных:

$$\begin{cases} \sum_{d=1}^D \sum_{t=1}^T \sum_{w=1}^W [t_{dj} = t] \log \phi_{w_{dj}t} \theta_{td} \\ \theta_{td}, \phi_{wt} \geq 0 \\ \sum \theta, \sum \phi = 1 \end{cases}$$

Тема является скрытой переменной

Можно построить ЕМ-алгоритм по этой задаче:

Е-шаг:  $p(t_{dj} \mid d, w_{dj})$

М-шаг:  $\phi_{wt}, \theta_{td} = ?$

### 7.3 LDA (Latent Dirichlet Allocation)

В PLSA не требуем невырожденности распределений

Дополнительно вводим распределения параметров:

$$\Phi_t = Dir(\alpha)$$

$$\Theta_d = Dir(\beta)$$

$$Dir(x_1, \dots, x_n, \alpha) = \frac{\Gamma(\alpha n)}{\Gamma^n(\alpha)} \prod_{i=1}^n x_i^{\alpha-1}$$

Распределение на дискретных распределениях с n исходами

## 8 Частичное обучение (semisupervised)

Используется в случае:

Есть обучающая выборка  $X^l = \{(x_i, y_i)\}_{i=1}^{\ell}$

Есть неразмеченная выборка:  $X^u = \{(x_i)\}_{i=\ell+1}^n$

Самая большая ценность состоит в необычных объектах

Мотивация: Неразмеченные данные собрать проще, чем размеченные

Категории задачи:

1. Semisupervised learning:  $X_\ell \cup X_u \rightarrow a(x)$
2. Transductive learning  $X_\ell \cup X_u \rightarrow$  Найти метки для  $X_u$

**Методы:**

1. Self-training
  - (a) Обучить  $a(x)$  на  $X^\ell$
  - (b) Применить на  $X^u$
  - (c) Добавить  $(x_i, a(x_i))$  в  $X^\ell$  объекты на которых модель наиболее уверенаКритерий:
  - i. Для классификации: самые большие уверенности
  - ii. Брать по порогу расстояния к обучающим
  - iii. Всю  $X^u$  с весами на основе уверенности моделиВзвешиваем лосс
- (d) Повторить



## 2. Генеративные модели

Описываем каждый класс нормальным распределением

Если бы были только  $X^\ell$ :

Правдоподобие и максимизация по параметрам:

$$\sum_{i=1}^{\ell} \log P(y_i | \theta) P(x_i | y_i, \theta) \rightarrow \max_{\theta}$$

Если  $X^\ell$  и  $X^u$ :

Неразмеченные данные хотим описать как смесь распределений классов

$$\sum_{i=1}^{\ell} \log P(y_i | \theta) P(x_i | y_i, \theta) + \sum_{i=\ell+1}^n \log \sum_{y=1}^K p(y | \theta) p(x_i | y, \theta) \rightarrow \max_{\theta}$$

Используем ЕМ-алгоритм для поиска  $\theta$

Второе слагаемое может перевесить - нужно добавить  $\lambda$

## 3. Упрощенная версия: Cluster-and-label:

Обучаем алгоритм кластеризации и помечаем в каждом кластере объекты самым популярным классом

## 4. Методы на основе моделей

(a) Логит  $\rightarrow$  Expectation Regularization

(b) Semi-Supervised SVM = S3VM

Безусловная задача оптимизации SVM:

$$\|w\|^2 + C \sum_{i=1}^{\ell} \max(0, 1 - y_i \langle w, x_i \rangle) \rightarrow \min_w$$

Лосс  $> 0$  когда расстояние до гиперплоскости отрицательное

Для unsupervised: чего требовать?

Если объект близко к гиперплоскости - штрафует, если далеко - не штрафует

Задача:

$$\|w\|^2 + C_1 \sum_{i=1}^{\ell} \max(0, 1 - y_i \langle w, x_i \rangle) + C_2 \sum_{i=\ell+1}^n \max(0, 1 - |\langle w, x_i \rangle|) \rightarrow \min_w$$

Может подобрать гиперплоскость, которая просто лежит далеко от данных и не разделяет их

Можно потребовать, чтобы баланс классов был такой же, как и на размеченных данных

$$\frac{1}{n - \ell} \sum_{i=\ell+1}^n a(x_i) = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i$$

Тогда гиперплоскость не может просто не разбивать классы  
Очень сложная задача с точки зрения оптимизации- максимум, модуль, ограничения

### СССР: ConCave Convex Procedure

Метод для оптимизации суммы выпуклой и вогнутой функции

## 5. Графовые методы

$$a(x) = \infty \sum_{i=1}^{\ell} (a(x_i) - y_i)^2 + \sum_{i,j=1}^n w_{ij} (a(x_i) - a(x_j))$$

$$w_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right)$$

Согласованные метки на соседних объектах для неразмеченных

Бесконечно более важно оптимизировать первый элемент

Упрощение задачи (Manifold Regularization):

$$\sum_{i=1}^{\ell} L(y_i, a(x_i)) + \lambda_1 R(a) + \lambda_2 \sum_{i,j=1}^n w_{ij} (a(x_i) - a(x_j))$$

$$\sum_{i,j=1}^n w_{ij} (a(x_i) - a(x_j)) = a^T L a$$

## 9 Метрические методы

Case-based reasoning - не очень зависит от параметров

$\rho : X \times X \rightarrow (0, +\infty)$  - функция расстояния

Обучение: запоминаем  $X$

Применение:

$u$  - новый объект

Строим вариационный ряд:

$$\rho(u, x_1) \leq \rho(u, x_2) \dots$$

Классификация:

$$a(u) = \arg \max_{y \in Y} \sum_{k=1}^K w(K, u, x_k) [y_k = y]$$

Веса учитывают расстояния до точки:

$$w(K, u, x_k) = K \left( \frac{\rho(u, x_k)}{h} \right)$$

Регрессия (Формула Надарая-Ватсона):

$$a(u) = \frac{\sum_{k=1}^K w y_k}{\sum_{k=1}^K w}$$

Зачем нужен kNN?

1. Если легко задать расстояния, но сложно придумать признаки
2. Если задача решается через сходство
3. Мало представителей каждого класса

Оптимальность kNN:

$$Y = \{-1, +1\}$$

$$p(y = +1 | x)$$

$u$  - хотим классифицировать

$x_u$  - ближайший сосед

$p_{bayes}^*$  - вероятность ошибки опт. Байесовского класс

$$p_{1nn} \leq 2p_{bayes}, if l \rightarrow \infty$$

---

### Пример хитрой метрики

Хотим сделать расстояние на текстах

$C(i, j)$  - расстояние между представлениями  $i$  и  $j$  слов

$t_{ij}$  - количество смысла, перетекающего из  $x_i$  в  $z_j$

$x_i$  - число вхождений некоторого слова  $i$  из словаря в текст  $x$

$$\sum_{j=1}^d t_{ij} = x_i$$

$$\sum_{i=1}^d t_{ij} = z_j$$

$$t_{ij} \geq 0$$

Стоимость переноса смысла

$$\sum_{i,j=1}^d t_{ij} C(i, j) = \mu(x, z) \rightarrow \min_{t_{ij}}$$

Для оптимального  $t_{ij}$ :  $\mu(x, z) = \rho(x, z)$

Задача оптимизации:  $\min \text{cost} \max \text{flow}$

---

## 9.1 Быстрый поиск ближайших соседей

Зачем?

1. Задачи retrieval
2. Рекомендации
3. ...

### 9.1.1 Точные методы

1. KD-деревья и другие структуры

При росте  $d$  сложность приближается к линейной

### 9.1.2 Приближенные решения

1. LSH - locality sensitive hashing

(а) Определение:

Семейство функций  $\mathcal{F} = \{f : X \rightarrow H\}$  с распределением  $P(f)$  называется  $(d_1, d_2, p_1, p_2)$  чувствительным, если

- i.  $p(x, z) \leq d_1 \Rightarrow P_{f \in \mathcal{F}}[f(x) = f(z)] \geq p_1$
- ii.  $p(x, z) \geq d_1 \Rightarrow P_{f \in \mathcal{F}}[f(x) = f(z)] \leq p_2$

---

Пример: MinHash

Объекты - множества,  $x \subset U = \{u_1, \dots, u_n\}$

$\pi$  - перестановка на множестве  $U$

$f_\pi(x) = \min\{\pi(i) \mid u_i \in A\}$

Утверждение:

$$P_\pi[f_\pi(A) = f_\pi(B)] = \frac{|A \cap B|}{|A \cup B|}$$

Док-во:

Три категории  $u \in U$ :

- i.  $u \in A, u \in B$
- ii.  $u \in A, u \notin B$
- или
- $u \notin A, u \in B$
- iii.  $u \notin A, u \notin B$

$$\pi = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

Одинаковые хэши для  $A$  и  $B$ ?

$u$  из первой группы должны иметь меньший хэш

Какова доля перестановок, где хотя бы 1 элемент первого типа идет раньше всех второго типа:

$$\frac{p}{p+q} = \frac{|A \cap B|}{|A \cup B|}$$

$$\rho(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

$$\rho(A, B) \leq d_1 \Rightarrow P_{f \in \mathcal{F}}[f(x) = f(z)] = \frac{|A \cap B|}{|A \cup B|} = 1 - \rho(A, B) \geq 1 - d_1$$

$$\rho(A, B) \geq d_2 \Rightarrow P \leq 1 - d_1$$

MinHash -  $(d_1, d_2, 1 - d_1, 1 - d_2)$  чувствителен

---

(b) Для косинусного расстояния:

$$\rho(x, y) = \arccos \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

$$\mathcal{F} = \{f_w(x) = \text{sign} \langle w, x \rangle \mid w \in R^d\}$$

Геометрически: Накидываем случайные гиперплоскости  $w$  и смотрим с какой стороны от них находятся точки

Если между точками угла маленький - вероятность их рассеяния плоскостью мала

(с) Для евклидова расстояния:

$$\mathcal{F} = \{f_{w,b}(x) = \frac{\leq w, x \geq + b}{r} \mid w \in R^d, b \in [0, r)\}$$

Геометрически: проводим случайную прямую, разбиваем на отрезки длины  $r$ . Значение хэш функции - номер отрезка

$$w \sim \mathcal{N}(0, I) \quad b \sim U[0, r)$$

Если варьировать распределения, то можно получить другие метрики:

Метрики Минковского -  $\rho \in (0, 2]$

Манхэттенская метрика ( $\rho = 1$ ):  $w \sim Cauchy$

(d) Проблема этих методов

- i. Не очень устойчиво из-за вероятностного подхода
- ii. При большом расстоянии все равно есть вероятность совпадения хэшей
- iii. Хотим форму сигмоиды, а не линейно убывающую вероятность равенства хэшей

(e) Композиция хэш-функций:

$$g(x) = (f_1(x), \dots, f_m(x))$$

Алгоритм:  $x \rightarrow g(x) \rightarrow \{x_i \in X \mid g(x_i) = g(x)\} \rightarrow$  Ищем ближайших соседей среди  $N(x)$

$$d = \rho(x, z)$$

$$P(f_1(x) = f_1(z)) = p$$

$$P(g(x) = g(z)) = p^m$$

Степенная функция это не то, что нам нужно

Модифицируем:

$$g_1(x) = (f_{11}(x), \dots, f_{1m}(x))$$

$\vdots$

$$g_L(x) = (f_{L1}(x), \dots, f_{Lm}(x))$$

$$P[[g_1(x) = g_1(z)] \parallel \dots \parallel [g_L(x) = g_L(z)]] = 1 - (1 - p^m)^L$$

(f) Теория:

Алгоритм решает задачу поиска  $s$ -ближайшего соседа, если для нового объекта  $u$  с вероятностью  $1 - \epsilon$ , алгоритм возвращает объект выборки  $z \in X : \rho(u, z) \leq C\rho(u, x_*)$ , где  $x_*$  - ближайший сосед  $u$

Для LSH:

$\exists L, m : O(d\ell^n \log \ell)$  - время поиска в LSH

## 2. NSW (Navigable Small World)

- (a) Small world graph - если сгенерировать случайный граф - среднее расстояние между двумя парами вершин очень близко
- (b) Представляем выборку в виде графа, где у каждой вершины небольшая степень, но выполняется свойство малого мира
- (c)  $G = (X, E)$   
Задаем алгоритм жадного поиска:
  - i.  $u$  - новый объект
  - ii. Берем случайную вершину  $v$  в  $G$
  - iii. В цикле: Среди всех соседей  $v$  ищем вершину  $v' : \rho(v', u) < \rho(v, u)$
  - iv. Если такой сосед нашелся - переходим в него
  - v. Используем мультистарт  
Находим множество результатов  $C_u$ , можно расширить это множество окрестностями  $C_u$  - выбираем ближайшие
- (d) Добавление вершины  $u$ :
  - i. Мультистарт -  $C(u)$
  - ii.  $D(u) = C(u) \cup \text{окрестности вершины}$
  - iii. Соединяем  $u$  с  $k$  ближайшими соседями из  $D(u)$
- (e) Особенность метода:
  - i. В графе есть области плотности и связующие цепочки
  - ii. Свойство малого мира достигается за счет связующих цепочек - вершин с очень высокой степенью

## 3. HNSW (hierarchical NSW):

- (a) На следующий уровень пропускаем только  $\log$  от числа вершин с некоторой вероятностью
- (b) Начинаем с самого верхнего уровня графа: находим ближайшего соседа
- (c) Начинаем из этой точки на более низком уровне, ищем новую точку
- (d) ...

## 10 Задача ранжирования

$$X = \{x_1, \dots, x_l\} \subset X$$

$$(i, j) \in R \subset \{1, \dots, \ell\}^2 \Rightarrow a(x_i) > a(x_j)$$

$\{1, \dots, \ell\}^2$  - множество всех пар

Обычно так:

$x = (q, d)$  -  $q$  - запрос, документ

в  $R$  - пары  $x_i = (q_i, d_i), x_j = (q_j, d_j)$ , где  $q_i = q_j$

Метрики качества ранжирования

1. Наивный подход:

(a)  $R \rightarrow y_1, \dots, y_\ell : (i, j) \in R \Rightarrow y_i > y_j$

(b) Обучаем  $a(x_i) \approx y_i$ , метрика - точность прогнозов

(c) Модель может правильно ранжировать, но при этом выдавать лейблы далекие от исходных и качество будет плохим при хорошем ранжировании

2. Дефектные пары:

$$\frac{1}{|R|} \sum_{(i,j) \in R} [a(x_i) \leq a(x_j)]$$

3. precision@k:

Работает, когда таргет является разметкой релевантности документов под запрос  $y \in \{0, 1\}$

$$\sum_{i=1}^K [y(i) = 1]$$

Проблема:

Не учитывает порядок выдачи в топ k документов

4. Average Precision@k(q)

$$AP@k = \sum_{i=1}^K \frac{y_i}{\sum_{j=1}^K y_j} \times precision@i$$

5. MAP@k

$Q$  - множество запросов

$$MAP@k = \frac{1}{|Q|} \sum_{q \in Q} AP@k(q)$$

6. DCG@k (для небинарных меток)

$$DCG@k(q) = \sum_{i=1}^K g(y_i) d(i)$$

$$g(y) 2^y - 1$$

$$d(i) = \frac{1}{\log(i+1)}$$



7. nDCG@k(q)

$$nDCG@k(q) = \frac{DCG@k(q)}{\max DCG@k(q)}$$

8. Каскадные метрики

pFound:

Пытается промоделировать поведение пользователей

$y \in [0, 1]$  - вероятность найти ответ в документе

$p_i$  - вероятность, что пользователь дойдет до  $i$ -ой позиции в выдаче

$$p_i = 1$$

$$p_{i+1} = p_i(1 - y_i)(1 - p_{out})$$

$p_{out}$  - вероятность, что пользователь уйдет

$$pFound@k(q) = \sum_{i=1}^K p_i \times y_i$$

Признаки для моделей ранжирования

1. Запросные признаки

- (a) Эмбединг запроса
- (b) Популярность
- (c) Категория
- (d) Персонализация
- (e) Признаки про пользователя

2. Статические - только про документ

- (a) Эмбединг документа
- (b) Категория документа
- (c) PageRank

$$PR(d) = \frac{1-\delta}{|D|} + \delta \sum_{c \in D_d^{in}} \frac{PR(c)}{|D_c^{out}|}$$

$D_d^{in}$  - мн-во документов, ссылающихся на  $d$

$D_c^{out}$  - мн-во док., на которые ссылается  $c$

3. Динамические признаки - про пару/запрос документ

- (a) Косинусное расстояние между эмбедингами документа и запроса

(b) BM25

$q = q_1, q_2, \dots, q_n$  - слова

$$BM25(q, d) = \sum_{i=1}^n IDF(q_i) \times \frac{TF(q_i, d) \times (K_1 + 1)}{TF(q_i, d) + K_1(1 + b \frac{|D|}{n_d})}$$

## Методы ранжирования

1. Pointwise - поточечные методы

$$q : (d_1, y_1), \dots, (d_{n_q}, y_{n_q})$$

$$\sum_{q \in Q} \sum_{i=1}^{n_q} L(y_i, a(q, d_i)) \rightarrow \min_a$$

2. Парные методы

Главное, чтобы пары были правильно расположены относительно друг друга

$$(i, j) \in R \Rightarrow a(x_i) > a(x_j)$$

$$\frac{1}{|R|} \sum_{(i, j) \in R} [a(x_i) < a(x_j)] \rightarrow \min_a$$

$$\frac{1}{|R|} \sum [a(x_i) - a(x_j) < 0] \leq$$

$$\leq \frac{1}{|R|} \sum_{(i, j) \in R} \tilde{L}(a(x_i) - a(x_j)) \rightarrow \min_a$$

$$\frac{1}{|R|} \sum \log(1 + e^{a(x_j) - a(x_i)}) \rightarrow \min_a$$

Частный случай (RankNet):

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

$$\tilde{L}(z) = \log(1 + e^{-\sigma z})$$

$$w^t = w^{t-1} + \eta \frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)} (x_j - x_i)$$

Используем метрику

$\delta F_{ij}$  - как изменится метрика качества, если поменять  $x_i x_j$  местами

LambdaRank:

$$w^t = w^{t-1} + \eta \frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)} |\delta F_{ij}| (x_j - x_i)$$

3. Списочные методы

Напрямую оптимизируем метрики качества:

ListNet

$$\begin{aligned}
& q_1, \dots, q_m \\
& q : d_1, \dots, d_{n_q} \\
& a(q, d_1) = z_1, \dots, a(q, d_{n_q}) = z_{n_q} \\
& \frac{1}{m} \sum_{i=1}^m nDCG@k(q_i) \rightarrow \max \\
& nDCG@k(q) = \sum_{i=1}^K \frac{2^y(i)}{\log(i+1)}
\end{aligned}$$

Параметры модели защиты в порядке ранжирования

$$\begin{aligned}
nDCG@k(q, \pi(a)) &= \sum_{i=1}^K \frac{2^{y(\pi(i))}}{\log(i+1)} \downarrow \\
E_\pi nDCG@k(q, \pi) &= \sum_{\pi \in \text{Sym}(q_1, \dots, q_{n_q})} p(\pi) nDCG@k(q, \pi)
\end{aligned}$$

$\phi$  - неубывающая строго + функция

$$p_z(\pi) = \prod_{j=1}^{n_q} \frac{\phi(z_{\pi(j)})}{\sum_{k=j}^{n_q} \phi(z_{\pi(k)})}$$

Свойства этого распределения:

- (a) Распределение на мн-ве всех перестановок  $n_q$  документов
- (b)  $\pi_i : d_i$  выше  $d_j, z_i > z_j$   
 $\pi' : \text{как } \pi_i \text{ только } d_i \text{ ниже } d_j$   
 $p_z(\pi) > p_z(\pi')$
- (c) Максимальную вероятность имеет перестановка с отсортированной по модели выборке

Можно посчитать  $\frac{\partial p_z(\pi)}{\partial a}$

Задача:  $E_\pi nDCG@k(q, \pi) \rightarrow \max_a$

$$\sum_{\pi \in \text{Sym}(q_1, \dots, q_{n_q})} p(\pi) nDCG@k(q, \pi) \rightarrow \max_a$$

Очень много ( $n_q$ ) слагаемых

В ListNet перестановки делаются иначе:

Вероятность, что  $j$  документ попадет на первое место в перестановке:

$$\begin{aligned}
p_z(j) &= \frac{\phi(z_j)}{\sum_{i=1}^n \phi(x_i)} \\
Q(y, z) &= - \sum_{j=1}^{n_q} p_y(j) \log p_z(j) \rightarrow \min_a
\end{aligned}$$

## 11 Рекомендательные системы

Задача, где любые две сущности надо сопоставлять друг с другом

Определения:

Множество пользователей:

$$U = \{u_1, \dots, u_n\}$$

Множество айтемов:

$$I = \{i_1, \dots, i_m\}$$

Для некоторых  $(u, i) \exists r_{ui}$

$$R = \{(u, i) \mid \exists r_{ui}\}$$

Нужно найти  $a(u, i)$

$U \xrightarrow{\text{Запрос}}$  Отбор кандидатов  $\rightarrow$  Ранжирование  $\rightarrow$  Переранжирование с учетом бизнес-требований  $\rightarrow$  выдача

### 11.1 Методы

1. Коллаборационная фильтрация:

Используем только информацию о взаимодействиях

$$R = \begin{pmatrix} r_{u_1 i_1} & \dots & r_{u_1 i_m} \\ r_{u_2 i_1} & \dots & \dots \end{pmatrix}$$

- (a) Memory-based методы

Просто используем эвристики - находим похожих пользователей и рекомендуем то, что понравилось им

- (b) Модели со скрытыми переменными

$p_u \in R^d$  - вектор пользователя

$q_i \in R^d$  - вектор айтема

Обучаем так, чтобы:

$$r_{ui} \approx \langle p_u, q_i \rangle$$

$$(\star) \sum_{u,i \in R} (r_{ui} - w_u - w_i - \langle p_u, q_i \rangle) \rightarrow \min_{w_u, w_i} p_u, q_i$$

Наблюдение 1:

$$R \in R^{n \times m}$$

$$P = (p_1 \mid \dots \mid p_n) \in R^{d \times n}$$

$$Q = (q_1 \mid \dots \mid q_m) \in R^{d \times m}$$

$$(P^T Q)_{ui} = \langle p_u, q_i \rangle$$

$$\|R - P^T Q\|_F \rightarrow \min_{P, Q}$$

Если матрица R известна, то это задача низкорангового приближения

R - SVD

PureSVD: Заполняем все пропуски нулями и применяем обычный SVD

Наблюдение 2:

Если знаем все  $q_i$

Рассмотрим конкретного пользователя:

$$\sum_{i: \exists r_{ui}} (r_{ui} - w_u - w_i - \langle p_u, q_i \rangle)^2 \rightarrow \min_{p_u \in R^d}$$

Задача сводится к линейной регрессии

Где это может пригодиться?

- i. Обучили (\*), получили  $P, Q$  - матрицу  $Q$  можно хранить на серверах  
Когда приходит и решаем задачу линейной регрессии, обучая  $p_u$   
Не нужно хранить  $p$   
Можно учесть самые свежие действия пользователя
- ii. Можно фиксировать все  $q_i$  на основе контента айтема  $i$

Обучение LFM:

- i. SGD  
Задача невыпуклая - можно попасть в локальный минимум
- ii. ALS (alternating least squares)  
Фиксируем  $P$ , находим  $Q \rightarrow$  Фиксируем  $Q$ , находим  $P \rightarrow$   
...

Наблюдение 3:

Если нашли идеальное решение  $R = P^T Q$

$\|r_i\|$  - насколько всем пользователям нравится этот айтем

Даже если много единиц, то норма все равно будет большой  
 $\rightarrow$  популярность айтема  $i$

$$r_i = P^T q_i$$

$$\sigma_{\min}(P) \times \|q_i\| \geq \|r_i\| \leq \sigma_{\max}(P) \times \|q_i\|$$

---

### Сингулярное разложение

Пусть дана матрица  $F_{n \times m}$ . Тогда  $F$  можно представить в следующем виде:

$$F_{n \times m} = U_{n \times n} \Sigma_{n \times m} V_{m \times m}^T$$

Основные свойства сингулярного разложения:

- i.  $n \times n$ -матрица  $U = (u_1, \dots, u_n)$  ортогональна,  $U^T U = I_n$ , столбцы  $u_j$  — собственные векторы матрицы  $FF^T$
- ii.  $m \times m$ -матрица  $V = (v_1, \dots, v_m)$  ортогональна,  $V^T V = I_m$ , столбцы  $v_j$  — собственные векторы матрицы  $F^T F$
- iii.  $n \times m$ -матрица  $\Sigma_{n \times m}$  диагональная,  $\Sigma_{n \times m} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$ ,  $\lambda_j \geq 0$  — собственные значения матриц  $F^T F$  и  $FF^T$   $\sqrt{\lambda_i}$  — сингулярные числа

Пусть  $i$  популярен -  $\|q_i\| \gg 0 \rightarrow$  для многих пользователей  $\langle p_u, q_i \rangle \gg 0$

Хак: Заменяем  $a(x) = \langle p_u, q_i \rangle$  на  $a(x) = \frac{\langle p_u, q_i \rangle}{\|p_u\| \|q_i\|}$

Наблюдение 3':  $i$ : 5 показов, 4 успешных  $\rightarrow$  высокий CTR  $\rightarrow$  норма становится высокой, но большой CTR может быть случайностью на низком количестве показов

Надо регуляризовать с учетом норм  $q_i, p_u$

Наблюдение 4:

LFM:  $(|U| + |I|)(d + 1)$  параметров

Увеличиваем количество параметров с помощью Neural CF:

- i. Берем эмбединги  $p_u, q_i$
- ii. Конкатенируем эмбединги в один вектор
- iii. Наворачиваем сверху полносвязные слои
- iv. В конце они должны предсказывать  $r_{ui}$

Для задачи лучше использовать не полносвязные слои, чтобы не застревать в локальных минимумах

Factorization Machines - обобщение LFM

## 11.2 Работа с неявным фидбэком

Явный фидбэк: пользователь непосредственно поставил оценку

Неявный фидбэк: факт покупки, факт просмотра, ...

**iALS**

$$S_{ui} = \begin{cases} 1, \exists r_{ui} \\ 0, \nexists r_{ui} \end{cases}$$

$$C_{ui} = 1 + [\exists r_{ui}] \alpha r_{ui}$$

$\alpha$  может принимать различные значения в зависимости от градации позитивности неявного фидбэка

Модель:

$$\sum_{\substack{u \in U \\ i \in I}} C_{ui} (S_{ui} - w_u - w_i - \langle p_u, q_i \rangle)^2 + \lambda \dots$$

### 11.3 Контентные рекомендации

Подходы

1.  $q_i = f(i)$  - считаем  $q_i$  по контенту  
Обучаем  $(a_i)$
2. DSSM, Siamese Nets  
 $i \rightarrow$  контент  $\rightarrow$  превращаем в вектор  $q_i$   
 $u \rightarrow$  история  $(i_1, i_2, \dots) \rightarrow$  превращаем в вектор  $p_u$   
Требуем, чтобы  $\text{corr}(\rho(p_u, q_i), r_{ui}) \uparrow$   
Для обучения используем триплетную функцию потерь

### 11.4 Как это все используется

1. Отбор кандидатов
  - (a) Эвристики
  - (b) Легкая модель
  - (c) Поиск ближайших соседей
  - (d) Графовые методы
    - i. Двудольный граф с пользователями и айтемами
    - ii. Запускаем случайное блуждание
2. Ранжирование  $(u, i) \rightarrow$  признаки

## Part II

# Семинары

## 12 Семинар: Задачи условной оптимизации

Учебник: *Boyd, Convex Optimization*

$$\begin{cases} f_0(x) \rightarrow \min_{x \in R^d} \\ f_i(x) \leq 0, i = 1, \dots, m \\ h_i(x) = 0, i = 1, \dots, p \end{cases}$$

$$G(x) = f_0(x) + \sum_{i=1}^m I_-(f_i(x)) + \sum_{i=1}^p I_0(h_i(x)) \rightarrow \min$$

Штрафы за нарушение ограничений:

$$I_-(z) = \begin{cases} 0, z \leq 0 \\ +\infty, z > 0 \end{cases}$$

$$I_0 = \begin{cases} 0, z = 0 \\ +\infty, z \neq 0 \end{cases}$$

$G(x) \rightarrow \infty$  в точках где не выполняется условие

Проблема: Недифференцируема

Заменяем функции на их аппроксимации ( $\hat{I}_- = ax$ )

Лагранжиан:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$
$$\lambda_i \geq 0$$

$x$  - прямые (primal) переменные

$\lambda, \nu$  - двойственные переменные

**Двойственная функция**

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$$

- Двойственная функция всегда вогнутая



- Дает нижнюю оценку на минимум функции в прямой задаче  
 $x'$  - допустимая точка (все условия выполнены)

$$L(x', \lambda, \nu) = f_0(x') + \sum_{i=1}^m \lambda_i f_i(x') + \sum_{i=1}^p \nu_i h_i(x')$$

$$f_i(x) \leq 0, h_i(x) = 0 \rightarrow L(x', \lambda, \nu) \leq f_0(x')$$

$$\inf_x L(x, \lambda, \nu) \leq \inf_{x'} L(x', \lambda, \nu) \leq \inf_{x'} f_0(x')$$

$\uparrow$  - это и есть решение исходной задачи

$$g(\lambda, \nu) \leq f_0(x_*)$$

$$g(\lambda, \nu) \rightarrow \max_{\lambda, \nu}, \lambda_i \geq 0$$

$\lambda^*, \nu^*$  - решение двойственной задачи

$g(\lambda^*, \nu^*) \leq f_0(x_*)$  - слабая двойственность

$g(\lambda^*, \nu^*) = f_0(x_*)$  - сильная двойственность

Достаточное условие сильной двойственности (Условие Слейтера)

– Задача выпуклая:

$f_0, f_1, \dots, f_m$  - выпуклые

$h_1, \dots, h_p$  - линейные

–  $\exists x'$ , что все ограничения выполнены строго

Пусть имеет место сильная двойственность:

$$g(\lambda^*, \nu^*) = f_0(x_*)$$

$$g(\lambda^*, \nu^*) = \inf_x (f_0(x) + \sum \lambda^* f_i(x) + \sum \nu^* h_i(x)) \leq f_0(x_*) + \sum \lambda^* f_i(x_*) + \sum \nu^* h_i(x_*) \leq f_0(x_*)$$

Все неравенства являются равенствами:

- Если решить безусловную задачу при подставлении  $\lambda^*, \nu^*$ , то получим решение прямой задачи
- $\lambda_i^* f_i(x^*) = 0$  - условие дополняющей нежесткости

## Теорема Куна-Такера

Необходимые условия для

$$\begin{cases} \nabla_x L(x_*, \lambda^*, \nu^*) = 0 \\ f_i(x) \leq 0 \\ h_i(x) = 0 \\ \lambda_i \geq 0 \\ \lambda_i f_i(x_*) = 0 \\ \text{Сильная двойственность} \end{cases} \leftrightarrow x_*, \lambda^*, \nu^* \text{ решения}$$

## 13 Семинар 3: ЕМ алгоритм

На М-шаге:

$$\Theta = \arg \max_{\Theta} E_q \log p(X, Z \mid \Theta)$$

$$\log p(X \mid \Theta_{i+1}) > \log p(X \mid \Theta_i)$$

Задача: **Шумная разметка изображений 100 экспертами**

i - изображение, j - эксперт:  $l_{ij} \in \{0, 1\}$

Истинный класс для картинки  $Z_i \in \{0, 1\}$

Дополнительные параметры:

$$\beta_i \in (0, +\infty), \alpha_j \in \mathcal{R}$$

$\beta$  - сложность изображения,  $\alpha$  - уровень эксперта

$$p(l_{ij} = Z_i \mid Z_i, \alpha_j, \beta_i) = \sigma(\alpha_j \beta_i) = \frac{1}{1 + e^{-\alpha_j \beta_i}}$$

$$p(Z_i, l_i \mid \alpha, \beta) = p(Z_i) \prod_j p(l_{ij} \mid Z_i, \alpha_j, \beta_i)$$

$p(Z_i)$ ?

1. Задать как 1/2, т.к. имеем два класса
2. Задать как баланс классов
3. Найти как параметр  $p(1) = \pi$

$$p(Z, l \mid \alpha, \beta) = \prod_i Z_i \prod_j p(l_{ij} \mid Z_i, \alpha_j, \beta_i)$$

Необходимо свести вероятность  $l_{ij} = Z_i$  к вероятности  $l_{ij}$

$$p(l_{ij} = Z_i \mid \dots) = \sigma(\alpha\beta)$$

$$p(l_{ij} \neq Z_i \mid \dots) = 1 - \sigma(\alpha\beta)$$

Бернулли:

$$p(l \mid \dots) = p(l = Z \mid \dots)^{[l=Z]} \times p(l \neq Z \mid \dots)^{[l \neq Z]} = \sigma(\alpha\beta)^{[l=Z]} \sigma(-\alpha\beta)^{[l \neq Z]}$$

$$p(Z_i, l_{ij} \mid \dots) = p(Z_i) \prod_j \sigma(\alpha\beta)^{[l=Z]} \sigma(-\alpha\beta)^{[l \neq Z]}$$

**Е-шаг:**

$$q^*(Z_i) = p(Z_i \mid l_{ij}, \alpha_j, \beta_i) \xrightarrow{\text{Теорема Байеса}} \frac{p(Z_i, l_{ij} \mid \alpha, \beta)}{p(l_{ij} \mid \alpha, \beta)} = \frac{p(Z_i, l_{ij} \mid \alpha, \beta)}{\sum_t p(t, l_{ij} \mid \alpha, \beta)}$$

$$q^*(Z) = \frac{p(Z_i) \prod_j \sigma(\alpha\beta)^{[l=Z]} \sigma(-\alpha\beta)^{[l \neq Z]}}{\sum_{t \in \{0,1\}} p(t_i) \prod_j \sigma(\alpha\beta)^{[l=t]} \sigma(-\alpha\beta)^{[l \neq t]}} = \frac{\gamma_i^{Z_i}}{\gamma_i^0 + \gamma_i^1} = \frac{e^{\log \gamma_i^{Z_i}}}{e^{\log \gamma_i^0 + \gamma_i^1}}$$

**М-шаг:**

$$E_{q^*} \log p(Z, l \mid \alpha, \beta) \rightarrow \max_{\alpha, \beta}$$

$$\begin{aligned} E_{q^*} \log \prod_i p(Z, l \mid \alpha, \beta) &= \sum_i E_{q_i^*} \log p(Z, l \mid \alpha, \beta) = \\ &= \sum_i E_{q_i^*} [\log p(Z_i) + \sum_j [l_{ij} = Z_i] \log \sigma(\alpha\beta) + [l_{ij} \neq Z_i] \log \sigma(-\alpha\beta)] \rightarrow \max_{\alpha, \beta} \\ &\quad \sum_i \sum_j \sum_{t \in \{0,1\}} q_i^*(t) [[l_{ij} = t] \log \sigma(\alpha\beta) + [l_{ij} \neq t] \log \sigma(-\alpha\beta)] \end{aligned}$$

Оптимизируем:

$$\frac{\partial}{\partial x} \log \sigma(x) = \sigma(-x)$$

$$\frac{\partial}{\partial \alpha} \log \sigma(\alpha \beta) = \beta \sigma(-\alpha \beta)$$

$$\frac{\partial}{\partial \alpha} \log \sigma(-\alpha \beta) = -\beta \sigma(\alpha \beta)$$

$$\frac{\partial}{\partial \alpha} E_{q^*} \log p(Z, l \mid \alpha, \beta) = \sum_i \sum_t q_i^* \beta ([l = t] \sigma(-\alpha \beta)) - [l \neq t] \sigma(\alpha \beta))$$

По  $\beta$  аналогично

## 14 Семинар 4: Основы байсовских методов

Существует распределение  $p(x, y)$

Интересует распределение:  $p(y \mid x)$

*Формула Байеса*

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

$p(x \mid y)$  - правдоподобие  $X$ , распределение объектов для некоторого класса

$p(y)$  - априорное распределение, доли классов в обучающей выборке

$p(x)$  - нормировочная константа

*Функционал среднего риска*

$$R(a) = \int_Y \int_X L(y, a(x)) p(x, y) dx dy$$

$$E_{y,x} L(y, a(x))$$

*Как использовать оптимальное распределение, когда оно найдено?*

$$L(y, a) = [y \neq a]$$

Функционал среднего риска:

$$R(a) = \int_Y \int_X [y \neq a(x)] p(x, y) dx dy = \sum_{y=1}^K \int_X [y \neq a(x)] p(x, y) dx dy =$$

$$= \int_X \sum_{y \neq a(x)} p(x, y) dx dy = \int_X (1 - \sum_{y=a(x)} p(x, y)) dx dy =$$

$$1 - \int_X p(x, a(x)) dx dy \rightarrow \min \Rightarrow a_*(x) = \arg \max_{y \in Y} P(y | x)$$

Для регрессии:

$$L(y, a) = (y - a)^2$$

$$a_*(x) = E(y | x)$$

Как найти  $p(y | x)$

В классификации:

$$a_*(x) = \arg \max_{y \in Y} p(y | x) = \arg \max_{y \in Y} \frac{p(x | y)p(y)}{p(x)} =$$

$$= \arg \max_{y \in Y} p(x | y)p(y)$$

$p(y)$  задается исходя из распределения  $y$

$p(x | y, \theta)$  находим  $\theta$  ММП

Пример:

$$p(y | x, w) = \mathcal{N}(< w, x >, \sigma^2)$$

Правдоподобие:

$$\prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - < w, x_i >)^2}{2\sigma^2}\right) \rightarrow \max_w$$

$$\log L = -\ell \log \sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2} \sum_{i=1}^{\ell} (y_i - < w, x_i >)^2 \rightarrow \max_w \Rightarrow$$

$$\Rightarrow \sum_{i=1}^{\ell} (y_i - < w, x_i >)^2 \rightarrow \min_w$$

Классификация:

Нужно найти  $p(x | y, \theta)$  для всех классов

$$p(x | y, \theta) = \mathcal{N}(\mu_y, \Sigma_y)$$

Можем найти  $\mu_y, \Sigma_y$  по ММП

Если параметры распределены нормально - Нормальный дискриминантный анализ

Если  $\Sigma_y = \Sigma$ , метод называется линейный дискриминант Фишера

Разделяющая поверхность:

$$p(y = +1 | x, \theta) = p(y = -1 | x, \theta)$$

$\Sigma_{-1} \neq \Sigma_{+1} \Rightarrow$  квадратичная поверхность

$\Sigma_{-1} = \Sigma_{+1} \Rightarrow$  Линейная поверхность

---

**Больше распределений:**

$$p(w | y, x) = \frac{p(y | x, w)p(w)}{p(x, y)}$$

$p(w) \sim \mathcal{N}(0, \sigma^2 I)$  - запрещаем модели большие веса

$$\log P(w | y, x) = -\frac{1}{2\sigma^2} \sum_{i=1}^{\ell} (y_i - \langle w, x_i \rangle)^2 - \frac{\ell}{2\alpha^2} \sum_{j=1}^{\alpha} w_j^2 \rightarrow \max_w$$

Фактически: MSE с регуляризацией  $L^2$

$$\lambda = \frac{\ell \sigma^2}{\alpha^2}$$

Что если  $w_j \sim \mathcal{N}(0, \alpha_j^2)$ ?

Отдельный коэффициент регуляризации для каждого параметра - такое не особо выводится в классическом машинном обучении  $\rightarrow$  RVM

---

**Наивный Байесовский алгоритм**

Исходя из предположения о независимости признаков:

$$p(x | y) = \prod_{j=1}^d p(x_j | y)$$

$$a(x) = \arg \max_{y \in Y} p(y | x) = \arg \max_y (\ln P(y) + \sum_{j=1}^d \ln P(x_j | y))$$

## 15 Семинар 5: Спектральная кластеризация

Алгоритм:

1.  $L = D - W, D = \text{diag}(d_1, \dots, d_\ell), d_i = \sum_{j=1}^{\ell} w_{ij}$
2.  $u_1, \dots, u_m$  - собственные векторы, соотв. минимальным собственным значениям  $L$
3.  $U = (u_1 \mid \dots \mid u_m) \in R^{l \times m}$
4. K-means над  $U$

Почему не делать кластеризацию t-SNE или Umap?

1. Оптимизирует положение точек, а не расположение кластеров
2. В t-SNE ошибка может быть неограничено большой  $\rightarrow$  зашумленные представления объектов
3. Есть шанс, что PSA будет лучше, чем t-SNE

Задача кластеризации:

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

$$A, B \subset X$$

$$A \cap B = \emptyset$$

$$\bar{A} = X \setminus A$$

$$\text{Ratio Cut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^K \frac{W(A_i, \bar{A}_i)}{|A_i|} \rightarrow \min_{A_1, \dots, A_K}$$

K = 2:

$$\text{Ratio Cut}(A, \bar{A}) \rightarrow \min_{A \subset X}$$

Задача поиска минимального разреза

$$X = A \cup \bar{A}$$

$$f : f_i = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}}, x_i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}}, x_i \in \bar{A} \end{cases}$$

Квадратичная форма:

$$\begin{aligned}
f^T L f &= \frac{1}{2} \sum_{i,j=1}^{\ell} w_{ij} (f_i - f_j)^2 = \\
&= \frac{1}{2} \left( \sum_{x_i \in A, x_j \in \bar{A}} w_{ij} \sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 + \frac{1}{2} \left( \sum_{x_i \in \bar{A}, x_j \in A} w_{ij} \sqrt{-\frac{|A|}{|\bar{A}|}} + \sqrt{-\frac{|\bar{A}|}{|A|}} \right)^2 \\
&= \frac{1}{2} \left( \sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 (W(A, \bar{A}) + W(\bar{A}, A)) \Rightarrow \\
&\Rightarrow \left( \sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 = \left( \frac{|\bar{A}| + |A|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|} \right) = \ell \left( \frac{1}{|A|} + \frac{1}{|\bar{A}|} \right); \\
&\quad (W(A, \bar{A}) + W(\bar{A}, A)) = 2W(A, \bar{A}) \Rightarrow \\
f^T L f &= \ell \left( \frac{1}{|A|} + \frac{1}{|\bar{A}|} \right) W(A, \bar{A}) = 2\ell \text{Ratio Cut}(A, \bar{A}) \propto \text{Ratio Cut}(A, \bar{A}) \\
\sum_{i=1}^{\ell} f_i &= |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = \sqrt{|A| |\bar{A}|} - \sqrt{|A| |\bar{A}|} = 0 \\
\sum_{i=1}^{\ell} f_i^2 &= |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = \ell
\end{aligned}$$

Перепишем оптимизационную задачу:

$$\begin{cases} f^T L f \rightarrow \min_{f_i \in \{\dots\}} \\ \langle f, \vec{1} \rangle = 0 \\ \|f\|^2 = \ell \end{cases}$$

Релаксация:  $f_i \in R$

$$\begin{cases} f^T L f \rightarrow \min_{f \in R^\ell} \\ \langle f, \vec{1} \rangle = 0 \\ \|f\|^2 = \ell \end{cases}$$



Лагранжиан:

$$\mathcal{L} = f^T L f + \lambda_1 \langle f, \vec{1} \rangle + \lambda_2 (\|f\|_2 - \sqrt{\ell})$$

$$\nabla_f \mathcal{L} = 2L_f + \lambda_1 \vec{1} + \lambda_2 \frac{1}{\|f\|} f = 0 \quad | \times \vec{1}^T$$

$$2\vec{1}^T L f + \lambda_1 \ell + \lambda_2 \frac{1}{\|f\|} \langle \vec{1}, f \rangle = 0 \Rightarrow$$

$$L \vec{1} = 0 \Rightarrow f^T L \vec{1} = 0 \Rightarrow \lambda_1 \ell = 0$$

$$\lambda_1 = 0$$

$$2L f + \lambda_2 \frac{1}{\|f\|} f = 0$$

$$L f = -\frac{\lambda_2}{2\|f\|} f \Rightarrow$$

$f$  - собственный вектор  $L$ , соотв. с.з.  $\mu$

$$f^T L f = \mu f^T f = \|f\|_2 \mu = \ell \mu$$

$$\text{Новая задача: } \begin{cases} \mu \rightarrow \min_{\mu - \text{с.з. } L, f - \text{с.в.}} \\ \langle f, \vec{1} \rangle = 0 \\ \|f\| = \sqrt{\ell} \end{cases}$$

Если  $G$ -связный, то с.в. соотв. 0 с.з. не подходит из-за невыполнения первого ограничения, в неполном графе может подходить

Решение - это с.в., соотв. второму собств. знач.

Находим  $f \rightarrow$  запускаем K-means

## 16 Семинар 6: Отбор признаков

### 16.1 Deep Clustering

1. Прогоняем объекты через нейросеть
2. По векторным описаниям строим псевдоразметку с помощью KMeans
3. Обучаем на псевдоразметке
4. Повторяем каждую эпоху

Даже необученная сеть не супер плохо размечает

Проблемы:

1. Несбалансированные выборки - использование взвешенных потерь

2. Наличие пустых кластеров - берем случайный центр другого кластера и добавляем шум
3. Делаем PCA перед кластеризацией, L2 нормализацию
4. Сбрасываем линейный слой на каждой эпохе

Есть возможности для улучшений:

1. Использовать не PCA, а MLP
2. Инициализируем матрицу классификатора в виде центроидов

## 16.2 Positional Encoding

Более простая задача:

1. Подаем сети координату точки
2. Она восстанавливает цвет пикселя с координатами
3. Можно воссоздавать 3D сцены (NERF)

## 16.3 Спектральный анализ

Берем картинку и парсим ее на частоты с помощью преобразования Фурье

При высоких частотах - быстрые изменения цвета

Обычный персептрон не умеет передавать высокие частоты

## 16.4 Positional encoding

Подаем не только саму картинку, но и гармоники - сеть сможет извлекать высокие частоты и обучаться на них

## 16.5 Feature extraction

Методы:

1. Filter
  - (a) Relevancy - удаляем близкие фичи
  - (b) Redundancy - используем Mutual Info classifier
  - (c) MRMR classifier
2. Wrapper - переучиваем модель на подмножествах фичей

## 17 Работа с признаками

1. Придумывание признаков
2. Feature selection
3. Понижение размерности

### 17.1 Отбор признаков

1. Методы фильтрации
  - (a) Корреляция  $x_j$  с  $y$  - не учитывает нелинейность и попарную корреляцию
  - (b) Для корреляции: t-score

$$R_j = \frac{|\mu_{-1} - \mu_{+1}|}{\sqrt{\frac{\sigma_{-1}^2}{n_{-1}} + \frac{\sigma_{+1}^2}{n_{+1}}}}$$

- (c) Для многоклассовой f-score
2. Методы обертки
    - (a) Ищем подмножество признаков, при котором ошибка модели на валидации поменьше
    - (b) Жадное удаление / добавление
    - (c) Генетические алгоритмы
      - $\beta \in \{0, 1\}^\alpha$  - вхождение признака в подмножество признаковИтерация:
      - i. Популяция:  $B = \{\beta_1, \dots\}$
      - ii. Скрещивание:  $\beta_j = \beta' \times \beta''$ 
$$\beta_j = \begin{cases} \beta', p = \frac{1}{2} \\ \beta'', p = \frac{1}{2} \end{cases}$$
      - iii. Мутация:  $\sim \beta' \rightarrow \beta_j = \begin{cases} \beta'_j, p \\ 1 - \beta'_j, 1 - p \end{cases}$
      - iv. Новая популяция:  $B' = \{\sim \beta' \times \beta''\}$  для какого-то числа пар из B
      - v. Делаем селекцию: Оставляем n лучших организмов
    - (d) Отбор признаков на основе моделей: Лассо, Out of bag

## 17.2 Понижение размерности

Метод главных компонент (РСА)

$$X \in R^{1 \times D}$$

$u_1, \dots, u_D \in R^D$  - главные компоненты, если

$$(1) : \langle u_i, u_j \rangle = 0, \forall i \neq j$$

$$(2) : \|u_j\|^2 = 1$$

$$(3) : \text{При проецировании выборки } X \text{ на } u_1, \dots, u_d : var \rightarrow \max$$

Поиск первой компоненты:

$$\begin{cases} u_1^T X^T X u_1 \rightarrow \max \\ \|u_1\|^2 = 1 \end{cases}$$

Лагранжиан:

$$2X^T X u_1 + 2\lambda u_1 = 0 \Rightarrow \lambda \rightarrow \max$$

$u_1$  - собств. вектор  $X^T X$  соотв. наибольшему с.з.

Постановка 2:

$$X \in R^{\ell \times D}$$

$$Z \in R^{\ell \times d}$$

$$U \in R^{d \times D}$$

Задача:

$$\|X - ZU^T\|_F^2 \rightarrow \min$$

Решается с помощью сингулярного разложения

## 18 Метод k ближайших соседей

$$X = \{(x_i, y_i)\}_{i=1}^{\ell}$$

$$\rho : X \times X \rightarrow (0, +\infty)$$

$$U : \rho(u, x_1) \leq \dots \leq \rho(u, x_{\ell})$$

$$a(u) = \arg \max_{y \in Y} \sum_{i=1}^K w_i [y_i = y]$$

Особенности метода:

### 1. Шумовые признаки

Очень чувствителен к шумовым признакам, т.к. использует все признаки для подсчета расстояния

### 2. Проклятие размерности

Все объекты находятся по краям гиперкуба - трудно быстро искать близких соседей

### 3. Функции расстояния

(а) Метрика Минковского:  $\rho_p(x, z) = (\sum |x_j - z_j|^p)^{\frac{1}{p}}$

$$\rho_\infty(x, z) = \max |x_j - z_j|$$

$$\rho_0(x, z) = \sum_{j=1}^d [x_j \neq z_j]$$

Можно добавить веса для отдельных признаков:

Веса можно подбирать покоординатным спуском

(b) Расстояние Махаланобиса

$$\rho(x, z) = \sqrt{(x - z)^T S^{-1} (x - z)}$$

$S$  - симметричная, положительно определенная матрица

(c) Косинусная мера

$$\rho_{cos}(x, z) = \arccos\left(\frac{\langle x, z \rangle}{\|x\| \|z\|}\right)$$

## 19 Метрические методы 2

### 19.1 Расстояния на категориальных признаках

Один категориальный признак

Как измерить расстояния:

1.  $\rho(x, z) = [x \neq z]$

2.  $\rho(x, z) = \alpha[x \neq z] + \beta[x = z]$

3. Сделать  $\alpha, \beta$  зависимыми от признака

4.  $\rho(x, z) = [x \neq z] \log(f(x) + 1) \log(f(z) + 1)$

$f(x)$  - сколько раз в обучающей выборке встречается категория  $x$

5.  $\rho(x, z) = [x \neq z] + [x = z] \times \sum_{q: p(q) \leq p(x)} p_j^2(q)$

$p(x)$  - частота

$p_j^2(x)$  - вероятность, что у пары объектов категория  $x$

## 19.2 Обучение метрик

Зачем:

1. Подобрать метрику для улучшения kNN
2. Когда необходимо разносить разные объекты по дальности

Самая параметризованная метрика: Метрика Махаланобиса

$$\rho(x, z) = \|Ax - Az\|^2 = (x - z)^T A^T A (x - z)$$

Выучиваем матрицу  $A \in R^{n \times d}$

Методы обучения:

1. NCA - neighborhood component analysis

$x_i \rightarrow$  выбираем  $x_j$  из некоторого распределения  $\rightarrow$  относим  $x_i$  к  $y_j$

$$p_{ij} = \begin{cases} \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{i \neq j} \exp(-\|Ax_i - Ax_j\|^2)}, & i \neq j \\ 0, & i = j \end{cases}$$

Вероятность отнесения к правильному классу:

$$C_i = \{j \mid y_j = y_i\}$$

$$p_i = \sum_{j \in C_i} p_{ij}$$

$$Q(A) = \sum_{i=1}^{\ell} p_i \rightarrow \max_A$$

2. LMNN - Large Margin NN

Используем триплетный лосс:

Берем для  $x_i$  положительные объекты и отрицательные объекты

$\eta_{ij} \in \{0, 1\}$  - является ли  $x_j$  целевым объектом для  $x_i$  (входит в  $k$  соседей)

Целевые объекты должны быть близки:

$$\begin{aligned} & \sum_{i \neq j} \xi_{ij} \|Ax_i - Ax_j\|^2 + C \sum_{i=1}^{\ell} \sum_{j \neq i} \sum_{\substack{m \neq i \\ m \neq j}} \xi_{ij} [y_m \neq y_i] \times \\ & \times \max(0, \alpha + \|Ax_i - Ax_j\|^2 - \|Ax_i - Ax_m\|^2) \rightarrow \min_A \end{aligned}$$

### 3. ITML

$$p(x | A) = \frac{1}{z} \exp(-\frac{1}{2} \|Ax - A\mu\|^2)$$

$z$  - нормировочная константа

$\mu$  - центр распределения

$S$  - мн-во пар, которые похожи

$D$  - мн-во пар, которые не похожи

$A_0$  - априорная матрица для расстояния Махаланобиса:

Можно ввести на основе выборочной ков. матрицы

Можно ввести как диагональную

$$\begin{cases} KL(p(x | A_0) || p(x | A)) \rightarrow \min_A \\ \rho_A(x_i, x_j) \leq u, (i, j) \in S \\ \rho_A(x_i, x_j) \geq L, (i, j) \in D \end{cases}$$

### 4. MCML (Maximally collapsing metric learning)

$$p_A(j | i) = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}$$

$$p_0(j | i) \propto \begin{cases} 1, y_i = y_j \\ 0, y_i \neq y_j \end{cases}$$

$$\sum_{i=1}^{\ell} KL(p_0(|i) || p_A(|i)) \rightarrow \min_A$$

### 5. Ядровой переход

$$K, \phi : X \rightarrow H$$

$$L : H \rightarrow R^n$$

$$\rho(x, z) = \|L\phi(x) - L\phi(z)\|^2$$

Ищем  $L$ :

Из функционального анализа:

$$L(h) = (< h, w_1 >, \dots, < h, w_n >), w_1, \dots, w_n \in H$$

$$w_i = \sum_{j=1}^{\ell} \alpha_{ij} \phi(x_j)$$