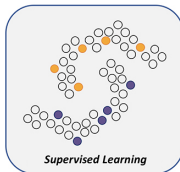


Частичное обучение

Виктор Китов

victorkitov.github.io

Курс поддержан
фондом
'Интеллект'



Победитель
конкурса VK среди
курсов по IT

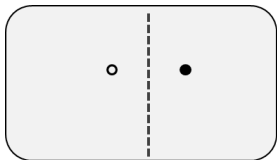


Частичное обучение

- Доступны данные:
$$L = \{(x_1, y_1), \dots (x_N, y_N)\}; U = \{x_{N+1}, \dots x_{N+M}\}.$$
- Использование частичного обучения (semi-supervised learning):
 - N мало, а $M \gg N$ - велико.
- Достаточно типичная ситуация:
 - классификация документов, изображений, речи
 - много неразмеченных объектов в интернете!
- Применимо к трансдуктивному обучению (transductive learning)
 - когда тестовая выборка известна заранее, например kaggle
- Будем рассматривать в основном задачу классификации.
- Частичное обучение есть в [scikit-learn](#).

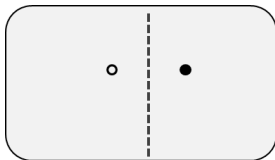
Мотивационный пример

Обучение с учителем:

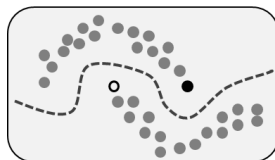


Мотивационный пример

Обучение с учителем:

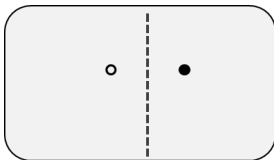


Частичное обучение:

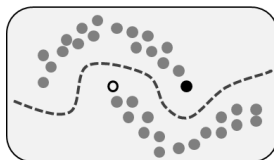


Мотивационный пример

Обучение с учителем:



Частичное обучение:



Предположение частичного обучения

Выход зависит плавно от входа.

- Кластеры/многообразия похожих объектов должны принадлежать одному классу.
- Если предположение не выполнено, то частичное обучение может работать хуже обучения с учителем.

Мотивационный пример¹

Рассмотрим классификацию документов на "астрономию" и "путешествия" или классификацию рукописных цифр.

документ 1	документ 2	документ 3	документ 4	документ 5
разработка	код			
	код	баг		
		баг	тест	
			тест	релиз



not similar



'indirectly' similar

Напрямую объекты несравнимы, но могут быть сравнимы за счет попарной схожести с неразмеченными объектами.

¹Источник иллюстрации.

Методы частичного обучения

- **Типы методов частичного обучения:**
 - препроцессинг на основе большой неразмеченной выборки
 - ↓ размерности, используя PCA
 - word2vec.
 - оценка расстояния Махаланобиса
 - мета-алгоритмы, использующие любой базовый алгоритм
 - самообучение (self-learning)
 - совместное обучение (co-learning)
 - специальные алгоритмы, использующие как размеченные, так и неразмеченные данные
 - кластеризация с метками
 - частичная генеративная классификация
 - трансдуктивный метод опорных векторов (transductive SVM)

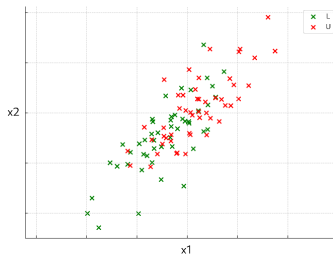
Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации
- 4 Генеративные модели
- 5 Трансдуктивный метод опорных векторов
- 6 Графовые методы

Самообучение

классификатор: $f(x) = \arg \max_c g_c(x)$

уверенность прогноза: $M_f(x) = g_{f(x)}(x) - \max_{c \in C \setminus f(x)} g_c(x)$



Объекты: L (labeled) - размеченные; U (unlabeled) - неразмеченные.

Самообучение

- Метод самообучения (self-training):

$Z=L$ # выборка, по которой учимся

ПОВТОРЯТЬ до условия остановки:

обучить $f(x)$ на Z

применить $f(x)$ к $U \setminus Z$

зададим расширение $\Delta = \{(x_i, f(x_i)) \in U \setminus Z : M_f(x) \geq t\}$

$Z = Z \cup \Delta$

- Выход: обученный $f(x)$ либо разметка тестовой выборки.
- Параметр t может выбираться, чтобы $|\Delta| = 0.05|U|$
- Реализация в scikit-learn: [SelfTrainingClassifier](#).

Самообучение

- Условия остановки:
 - вся тестовая выборка размечена
 - точность на валидации перестала \uparrow
- Можно составлять Δ по наиболее уверенным предсказаниям, *сохраняя исходное распределение на классах*.
- Метод применим к любому классификатору.
- Предположение: прогнозы, полученные с большой уверенностью, считаются верными.
 - сильно увеличивает переобученность модели.
 - отчасти исправляется совместным обучением (co-training)

Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации
- 4 Генеративные модели
- 5 Трансдуктивный метод опорных векторов
- 6 Графовые методы

Совместное обучение через ансамбль

- Самообучение усиливает переобученность метода.
- Для \downarrow переобучения будем использовать разные методы для разметки.

Идея совместного обучения

Разные методы дообучают друг друга.

- Совместное обучение через ансамбль:
 - применяем самообучение к ансамблю $f_1(x), \dots, f_K(x)$.
 - объекты, на которых большинство прогнозов базовых моделей сходятся, добавляются в выборку
- \downarrow переобучение каждой индивидуальной модели.

Совместное обучение

- Пусть $f_1(\cdot)$ и $f_2(\cdot)$ - одинаковые классификаторы, использующие различные наборы признаков F_1 и F_2 , $F_1 \cap F_2 = \emptyset$.
- Совместное обучение (co-training):

$Z_1 = L$ на признаках F_1

$Z_2 = L$ на признаках F_2

ПОВТОРЯТЬ до условия остановки:

обучить $f_1(x)$ на Z_1

применить $f_1(x)$ к $U \setminus Z_2$

$\Delta_1 = \{(x_i, f_1(x_i)) \in U \setminus Z_2 : M_{f_1}(x_i) \geq t\}$

$Z_2 = Z_2 \cup \Delta_1$

обучить $f_2(x)$ на Z_2

применить $f_2(x)$ к $U \setminus Z_1$

$\Delta_2 = \{(x_i, f_2(x_i)) \in U \setminus Z_1 : M_{f_2}(x_i) \geq t\}$

$Z_1 = Z_1 \cup \Delta_2$

Совместное обучение

- Выход: обученные $f_1(x)$, $f_2(x)$ или разметка тестовой выборки.
- Предположение метода (когда прогнозы одной модели случайны для другой):

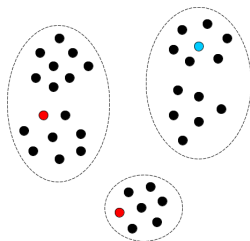
$$p(F_1, F_2|y) = p(F_1|y)p(F_2|y)$$

- Альтернативно $f_1(\cdot)$ и $f_2(\cdot)$ используют одинаковые признаки, но разные модели.
 - в этом случае инициализация $Z_1 = Z_2 = L$.

Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации**
- 4 Генеративные модели
- 5 Трансдуктивный метод опорных векторов
- 6 Графовые методы

Расширение меток на кластер



- Кластеризовать $L \cup U$.
- Расширить метки на кластер.
 - если нет меток - оставить неразмеченными / взять ближайшие
 - если несколько - голосование по большинству
- Простой, но слишком грубый метод обобщения меток

K-средних для частичного обучения

Инициализировать μ_k , $k = 1, 2, \dots K$.

ПОВТОРЯТЬ до сходимости:

для $n = N + 1, 2, \dots N + M$:

определить кластер для x_i :

$$z_n = \arg \min_{k \in \{1, 2, \dots K\}} \|x_n - \mu_k\|_2^2$$

для $k = 1, 2, \dots K$:

пересчитать центры:

$$\mu_k = \frac{1}{|C_k|} \sum_{n \in C_k} x_n$$

- μ_1, μ_2, \dots инициализируются средними для размеченных объектов.

Аггломеративная кластеризация - алгоритм

инициализировать матрицу попарных расстояний $M \in \mathbb{R}^{N \times N}$ между кластерами из отдельных объектов $\{x_1\}, \dots, \{x_N\}$

ПОВТОРЯТЬ:

- 1) выбрать ближайшие кластеры i и j
- 2) объединить $i, j \rightarrow \{i + j\}$, **если нет разных меток**
- 3) удалить строки/столбцы i, j из матрицы расстояний
- 4) добавить строку/столбец для нового $\{i + j\}$ в матрицу

ПОКА не выполнено условие остановки

ВЕРНУТЬ иерархическую кластеризацию

Объединяем самые близкие $\{i\}$ и $\{j\}$,
в которых нет меток разных классов.

Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации
- 4 Генеративные модели**
- 5 Трансдуктивный метод опорных векторов
- 6 Графовые методы

Частичное обучение в генеративных моделях

- Генеративная модель оценивает $p(x, y|\theta)$, поэтому можем оценить $p(x|\theta) = \sum_y p(x, y|\theta)$ для U .

Частичное обучение в генеративных моделях

- Генеративная модель оценивает $p(x, y|\theta)$, поэтому можем оценить $p(x|\theta) = \sum_y p(x, y|\theta)$ для U .

$$\begin{aligned}
 \ln p(X, Y|\theta) &= \sum_{n=1}^N \ln p(x_n, y_n|\theta) + \lambda \sum_{i=N+1}^{N+M} \ln p(x_i|\theta) \\
 &= \sum_{n=1}^N \ln p(x_n, y_n|\theta) + \lambda \sum_{n=N+1}^{N+M} \ln \left[\sum_{y=1}^C p(x_n, y|\theta) \right] \\
 &= \sum_{n=1}^N \ln [p(y_n|\theta)p(x_n|y_n, \theta)] + \lambda \sum_{n=N+1}^{N+M} \ln \left[\sum_{y=1}^C p(y|\theta)p(x_n|y, \theta) \right]
 \end{aligned}$$

Частичное обучение в генеративных моделях

- Генеративная модель оценивает $p(x, y|\theta)$, поэтому можем оценить $p(x|\theta) = \sum_y p(x, y|\theta)$ для U .

$$\begin{aligned}
 \ln p(X, Y|\theta) &= \sum_{n=1}^N \ln p(x_n, y_n|\theta) + \lambda \sum_{i=N+1}^{N+M} \ln p(x_i|\theta) \\
 &= \sum_{n=1}^N \ln p(x_n, y_n|\theta) + \lambda \sum_{n=N+1}^{N+M} \ln \left[\sum_{y=1}^C p(x_n, y|\theta) \right] \\
 &= \sum_{n=1}^N \ln [p(y_n|\theta)p(x_n|y_n, \theta)] + \lambda \sum_{n=N+1}^{N+M} \ln \left[\sum_{y=1}^C p(y|\theta)p(x_n|y, \theta) \right]
 \end{aligned}$$

- $\lambda \in [0, 1]$ - значимость неразмеченной части.
- Важна адекватность генеративной модели $p(x|y)$.
- $\ln(\sum \dots)$ - нет численного решения, используем ЭМ-алгоритм (латентные переменные y_{N+1}, \dots, y_{N+M}).

ЕМ алгоритм

ЕМ алгоритм: повторять до сходимости

- для $n = N + 1, \dots N + M, c = 1, \dots C$:
 - найти $p_{ny} = p(y_n = y | x_n, \hat{\theta})$
 - уточнить $\hat{\theta}$, решив:

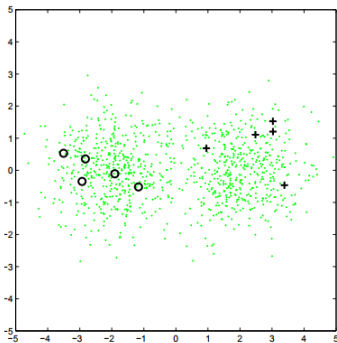
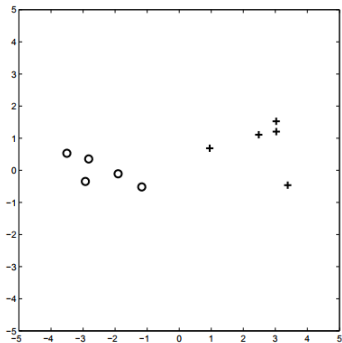
$$\sum_{n=1}^N \ln [p(y_n | \theta) p(x_n | y_n, \theta)] + \lambda \sum_{n=N+1}^{N+M} \sum_{y=1}^C p_{ny} \ln [p(y | \theta) p(x_n | y, \theta)]$$

$$\rightarrow \max_{\theta}$$

Пример использования

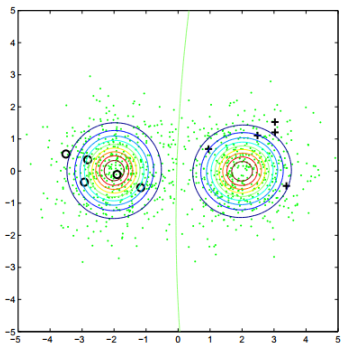
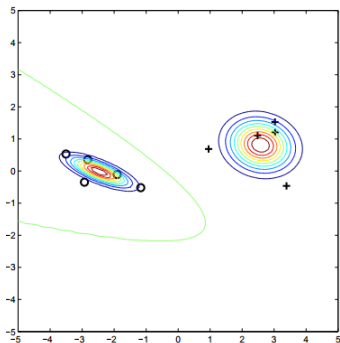
Пусть $y \in \{+1, -1\}$, $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Размеченные и неразмеченные данные:



Пример использования

Решение без/с использованием неразмеченных данных:



Мультиномиальная модель

- w_1, w_2, \dots, w_D - уникальные токены языка
- Решающее правило:

$$\hat{y}(x) = \arg \max_y p(y)p(x|y)$$

- $x \in \mathbb{R}^D$, $x^i = [\text{сколько раз } w_i \text{ встретилось в документе}]$,
 $i = \overline{1, D}$
- $\theta_i^y = p(w_i \text{ на словопозиции } i | y)$ - не зависит от i и др. слов документа
- Генерация документа класса y :
 - для каждой словопозиции $i = 1, 2, \dots, n_{\text{document}}$:
 - сгенерировать слово $z_i \sim \text{Categorical}(\theta_1^y, \theta_2^y, \dots, \theta_D^y)$

Мультиномиальная модель

- $(\sum_i x^i)!$ - # перестановок всех слов документа
- $\prod_i (x^i)!$ - # перестановок в рамках встречи каждого слова
- $\frac{(\sum_i x^i)!}{\prod_i (x^i)!}$ - # документов где w_1, w_2, \dots встретились x^1, x^2, \dots раз.
- Вероятность:

$$p(x|y) = \frac{(\sum_i x^i)!}{\prod_i (x^i)!} \prod_{i=1}^D (\theta_i^y)^{x^i}$$

Оценка параметров

$$p(y) = \frac{\sum_{d=1}^N \mathbb{I}[y_d = y] + \lambda \sum_{d=N+1}^{N+M} p_{dy}}{N + \lambda M}$$

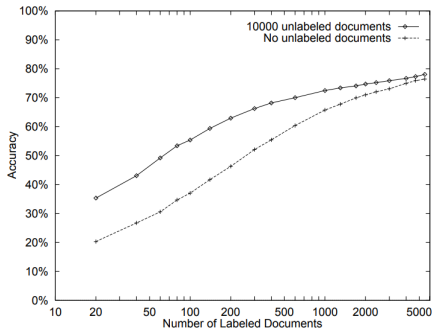
$$\theta_i^y = \frac{\sum_{d=1}^N n_{di} \mathbb{I}[y_d = y] + \alpha + \lambda \sum_{d=N+1}^{N+M} p_{dy} n_{di}}{\sum_{d=1}^N \sum_{i=1}^D n_{di} \mathbb{I}[y_d = y] + \alpha D + \lambda \sum_{d=N+1}^{N+M} \sum_{i=1}^D p_{dy} n_{di}}$$

- $n_{di} = \#$ раз w_i встретилось в документе d
- $D = \#$ документов
- $\alpha \geq 0$ - сглаживание Лапласа
- $\lambda \in [0, 1]$ - важность частичного обучения

$$p_{dy} = p(y|d) = \frac{p(y, d)}{p(d)} = \frac{p(y) p(d|y)}{\sum_y p(y) p(d|y)}$$

Эксперимент

- Классификация новостей (20NewsGroups).
- 20 - 5000 размеченных документов, 10000 неразмеченных.
- Частичное обучение работает лучше:



Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации
- 4 Генеративные модели
- 5 Трансдуктивный метод опорных векторов**
- 6 Графовые методы

Обычный метод опорных векторов

- Метод опорных векторов (SVM) - линейный классификатор:

$$f(x) = \text{sign}(w^T x + w_0), \quad w, x \in \mathbb{R}^D, w_0 \in \mathbb{R}$$

- Отступ объекта x_n :

$$M(x_n, y_n) = (w^T x_n + w_0) y_n$$

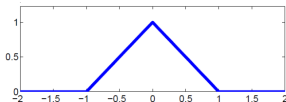
- Оптимизационная задача:

$$\frac{1}{2C} \|w\|^2 + \sum_{n=1}^N [1 - M(x_n, y_n)]_+ \rightarrow \min_{w, w_0}$$

- $\mathcal{L}(M) = [1 - M]_+$ штрафует за $M \leq 1$.

Трансдуктивный метод опорных векторов

$$\tilde{\mathcal{L}}(M) = [1 - |M|]_+ = [1 - |w^T x_n + w_0|]_+$$

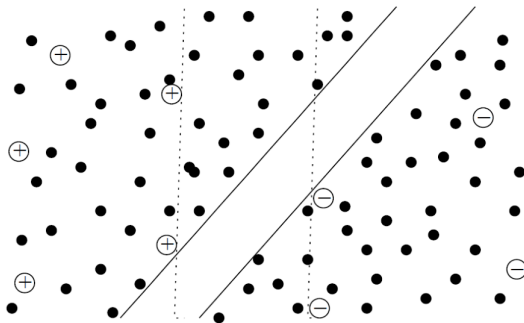


- не зависит от y_n
- штрафует объекты за близость к разделяющей гиперплоскости

Трансдуктивный метод опорных векторов (transductive SVM, TSVM):

$$\frac{1}{2C} \|w\|^2 + \sum_{n=1}^N [1 - M(x_n, y_n)]_+ + \lambda \sum_{n=N+1}^{N+M} [1 - |M(x_n, y_n)|]_+ \rightarrow \min_{w, w_0}$$

Иллюстрация



- В кругах - размеченные объекты.
- Пунктиром - разделяющая граница SVM
- Сплошные линии - разделяющая граница TSVM

Идея метода - разделение областей низкой плотности.

Обсуждение

Преимущества:

- может быть обобщено ядрами
- существуют эффективные реализации

Недостатки:

- задача перестаёт быть выпуклой:
 - много локальных минимумов, нужно искать наилучший
- поощряет тривиальное решение, когда гиперплоскость далека от всех объектов
 - т.е. прогноз одним классом
 - рекомендуется оптимизировать при доп. ограничении (balancing constraint²):

$$\frac{1}{M} \sum_{n=N+1}^{N+M} (w^T x + w_0) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y_n = +1]$$

²Large Scale Transductive SVMs.

Содержание

- 1 Самообучение
- 2 Совместное обучение
- 3 Использование кластеризации
- 4 Генеративные модели
- 5 Трансдуктивный метод опорных векторов
- 6 Графовые методы**

Алгоритм распространения меток³

❶ строим граф связей похожих объектов:

- узлы - $x \in L \cup U$ и связи между близкими x_i, x_j :

$$w_{ij} = e^{\|x_i - x_j\|^2 / (2\sigma^2)} = w_{ji}; \quad w_{ij} = \begin{cases} 1, & x_j \in \text{KNN}(i) \\ 0, & x_j \notin \text{KNN}(i) \end{cases}$$

❷ вычисляем матрицу переходов $P \in \mathbb{R}^{(N+M) \times (N+M)}$

$$p_{ij} = P(x_i \rightarrow x_j) = \frac{w_{ij}}{\sum_{k=1}^{N+M} w_{ik}}$$

❸ Инициализируем ответы на объектах $f \in \mathbb{R}^{N+M}$.

- $f_n := y_n$ для $n = 1, 2, \dots, N$.
- $f_n := 0$ (не принципиально) для $n = N + 1, \dots, N + M$.

³Детали метода.

Алгоритм распространения меток⁵

Алгоритм распространения меток (label propagation)

- повторять до сходимости f :

- 1 усреднить ответы по исходам из каждой вершины

$$f := Pf \quad (\text{покомпонентно } f(x_i) := \sum_{j:i \rightarrow j} p_{ij} f(x_j))$$

либо по входам в каждую вершину⁴

$$f^T := f^T P \quad (\text{покомпонентно } f(x_j) := \sum_{i:i \rightarrow j} f(x_i) p_{ij})$$

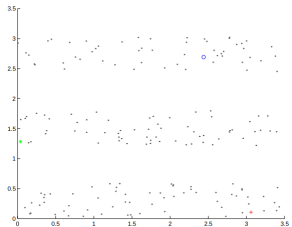
- 2 перезадать известные метки: $f_L := Y_L$

⁴Эквивалентно для ненаправленных графов ($P = P^T$), т.к.

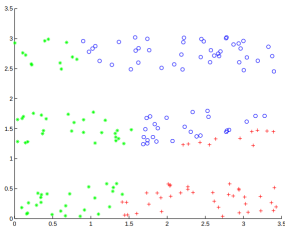
$$f = (f^T)^T = (f^T P)^T = P^T f = Pf.$$

⁵Реализация: [LabelPropagation](#), [LabelSpreading](#)

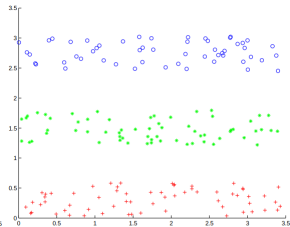
Визуализация работы



(a) The data



(b) 1NN



(c) Label Propagation

Комментарии

- Идейно это self-learning для KNN.
- $F^0 = \begin{pmatrix} Y_L \\ 0 \end{pmatrix}$
- повторять до сходимости:
 - $F^{t+1} := PF^t$
 - $F_{:|Y_L|}^{t+1} := Y_L$
- Вариант: вершина с большим числом соседей слабее отдаёт сигнал.
- повторять до сходимости:
 - $F^{t+1} := D^{-1}PF^t$
 - $F_{:|Y_L|}^{t+1} := Y_L$

Регуляризация энергии графа

- Воспользуемся графом из алгоритма распространения меток.
- Энергия графа измеряет согласованность меток для соседних узлов

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(x_i) - f(x_j))^2 = f^T \Delta f$$

- Найдем $f(x)$ из задачи

$$\sum_{n=1}^N \mathcal{L}(f(x_n), y_n) + \lambda_1 R(f) + \lambda_2 E(f) \rightarrow \min_f$$

- Варианты оптимизации:
 - по значениям $f \in \mathbb{R}^{N+M}$
 - по параметрам $f_w(x)$

Лапласиан графа

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(x_i) - f(x_j))^2 = f^T \Delta f$$

где $\Delta = D - W$ - Лапласиан графа, $D, W \in \mathbb{R}^{(N+M) \times (N+M)}$

$$D = \text{diag} \left(\sum_{j=1}^{N+M} w_{1j}, \sum_{j=1}^{N+M} w_{2j}, \dots, \sum_{j=1}^{N+M} w_{(N+M)j} \right)$$

$$W = \{w_{ij}\}_{i,j=1,\dots,N+M}.$$

Разобьём Лапласиан на блоки: $\Delta = \begin{bmatrix} \Delta_{LL} & \Delta_{LU} \\ \Delta_{UL} & \Delta_{UU} \end{bmatrix}$

$$\sum_{n=1}^N (f_n - y_n)^2 + f^T \Delta f \rightarrow \min_{f \in \mathbb{R}^{N+M}} \Rightarrow f_U = -\Delta_{UU}^{-1} \Delta_{UL} Y_L$$

Заключение

- Частичное обучение - использование неразмеченных объектов для уточнения прогнозов.
- Наиболее эффективно, когда N мало, $M \gg N$.
- Подходы:
 - мета-алгоритмы, строящиеся на базе других
 - самообучение, совместное обучение
 - кластеризация с учётом меток
 - генеративные модели, учитывающие $p(x)$ для $x \in U$.
 - трансдуктивный метод опорных векторов
 - распространение меток на графе, минимизация энергии