Ответы на вопросы государственного экзамена

Дарина Дементьева

18 января 2019 г.

Вопросы

- Методы кластеризации: графовые, иерархические, статистические.
- Локализация точки в планарном подразбиении при массовом запросе. Метод полос. Метод Киркпатрика.
- Задача разметки последовательностей. Гибридная модель biLSTM-CRF.

Постановка задачи кластеризации

Дано:

X — пространство объектов;

$X^l = \{x\}_{i=1}^l$ — обучающая выборка;

$\rho : X \times X \to [0, \infty)$ — функция расстояния между объектами.

Найти:

Y — множество кластеров и

$a : X \to Y$ — алгоритм кластеризации, такие, что:

— каждый кластер состоит из близких объектов;

— объекты разных кластеров существенно различны.

Методы кластеризации

1. Графовые методы кластеризации
   - Алгоритм выделения связных компонент
   - Алгоритм КНП
   - Алгоритм ФОРЭЛ
2. Иерархическая кластеризация (таксономия)
   - Агломеративная иерархическая кластеризация
3. Статистические методы кластеризации
   - EM-алгоритм
   - Метод k-средних

Алгоритм выделения связных компонент

Выборка представляется в виде графа:
— вершины графа — объекты x_i
— рёбра — пары объектов с расстоянием $\rho_{ij} = \rho(x_i, x_j)$
Идея:
— Задается входной параметр R и в графе удаляются все ребра, для которых расстояния больше R.
— Надо подобрать такое значение R, лежащее в диапазон всех «расстояний», при котором граф «развалится» на несколько связных компонент.
— Полученные компоненты и есть кластеры.
Недостатки:
— задаётся неудобный параметр R;
— высокая чувствительность к шуму.

Алгоритм КНП — "Кратчайший Незамкнутый Путь"

Алгоритм:

1. Найти пару вершин (i, j) с наименьшим ρ_{ij} и соединить их ребром;
2. пока в выборке остаются изолированные точки
3. найти изолированную точку, ближайшую к некоторой неизолированной
4. соединить эти две точки ребром;
5. удалить $K1$ самых длинных рёбер;

Достоинство:

— задаётся число кластеров K.

Недостаток:

— высокая чувствительность к шуму.

Алгоритм ФОРЭЛ ФОРмальные ЭЛементы

1: $U := X^\ell$ — множество некластеризованных точек;
2: **пока** в выборке есть некластеризованные точки, $U \neq \varnothing$:
3: взять случайную точку $x_0 \in U$;
4: **повторять**
5: образовать кластер с центром в x_0 и радиусом R:
$K_0 := \{x_i \in U \mid \rho(x_i, x_0) \leqslant R\}$;
6: переместить центр x_0 в центр масс кластера:
$x_0 := \frac{1}{|K_0|} \sum_{x_i \in K_0} x_i$;
7: **пока** состав кластера K_0 не стабилизируется;
8: пометить все точки K_0 как кластеризованные:
$U := U \setminus K_0$;
9: применить алгоритм КНП к множеству центров кластеров;
10: каждый $x_i \in X^\ell$ приписать кластеру с ближайшим центром;

Алгоритм ФОРЭЛ ФОРмальные ЭЛементы

Преимущества ФОРЭЛ:

- получаем двухуровневую структуру кластеров;
- кластеры могут быть произвольной формы;
- варьируя R, можно управлять детальностью кластеризации.

Недостаток ФОРЭЛ:

- чувствительность к R и начальному выбору точки x0.

Агломеративная иерархическая кластеризация

Алгоритм Ланса-Уильямса [1967]

1: сначала все кластеры одноэлементные:
$t := 1; \quad C_t = \{\{x_1\}, \ldots, \{x_\ell\}\};$
$R(\{x_i\}, \{x_j\}) := \rho(x_i, x_j);$
2: **для всех** $t = 2, \ldots, \ell$ (t — номер итерации):
3: найти в C_{t-1} два ближайших кластера:
$(U, V) := \arg\min_{U \neq V} R(U, V);$
$R_t := R(U, V);$
4: слить их в один кластер:
$W := U \cup V;$
$C_t := C_{t-1} \cup \{W\} \setminus \{U, V\};$
5: **для всех** $S \in C_t$
6: вычислить $R(W, S)$ по формуле Ланса-Уильямса;

Формула Ланса-Уильямса

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

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

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

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

Гипотезы о данных

Дано:

— выборка X^l - выборка случайных незваисимых наблюдений из смехи $p(x)$

Предполагаем:

— $X = R^n$

— кластеры n-мерные гауссовские

— μ_y - центр кластера y.

Метод k-средних

1: начальное приближение центров μ_y, $y \in Y$;

2: **повторять**

3: аналог E-шага:

отнести каждый x_i к ближайшему центру:

$y_i := \arg\min_{y \in Y} \rho(x_i, \mu_y)$, $i = 1, \ldots, \ell$;

4: аналог M-шага:

вычислить новые положения центров:

$$\mu_{yj} := \frac{\sum_{i=1}^{\ell} [y_i = y] f_j(x_i)}{\sum_{i=1}^{\ell} [y_i = y]}, \quad y \in Y, \quad j = 1, \ldots, n;$$

5: **пока** y_i не перестанут изменяться;

Недостатки k-means

- Чувствительность к выбору начального приближения.
- Необходимость задавать k.

Способы устранения этих недостатков:

- Несколько случайных кластеризаций; выбор лучшей по функционалу качества.
- Постепенное наращивание числа кластеров k.

Локализация точки в планарном подразбиении

Дано:

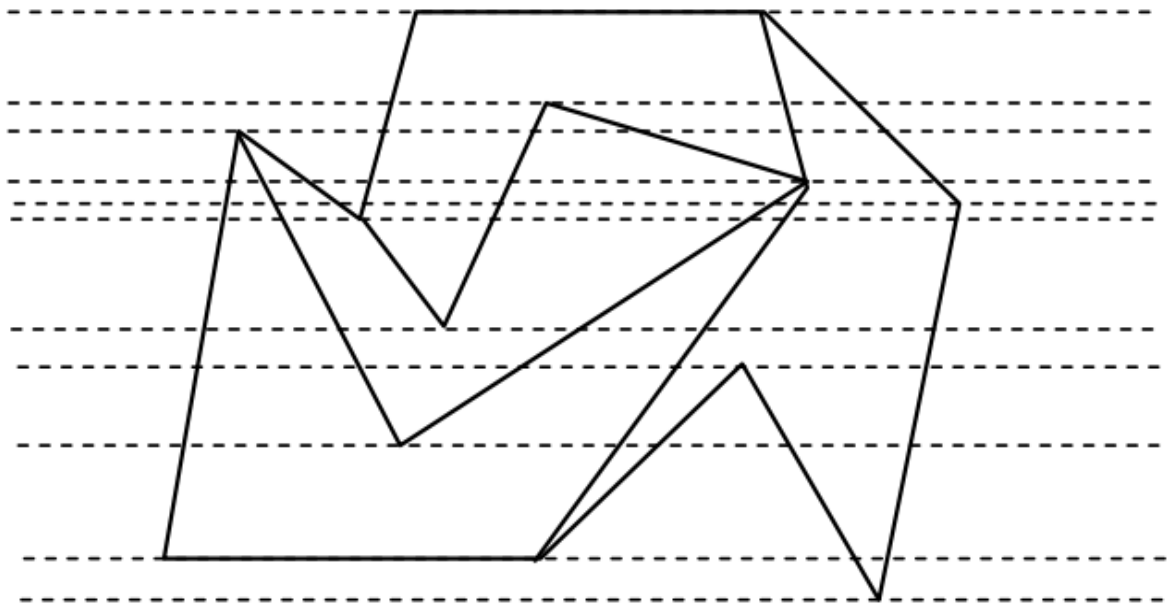
ППЛГ G, имеющий n вершин;

точка x.

Задача:

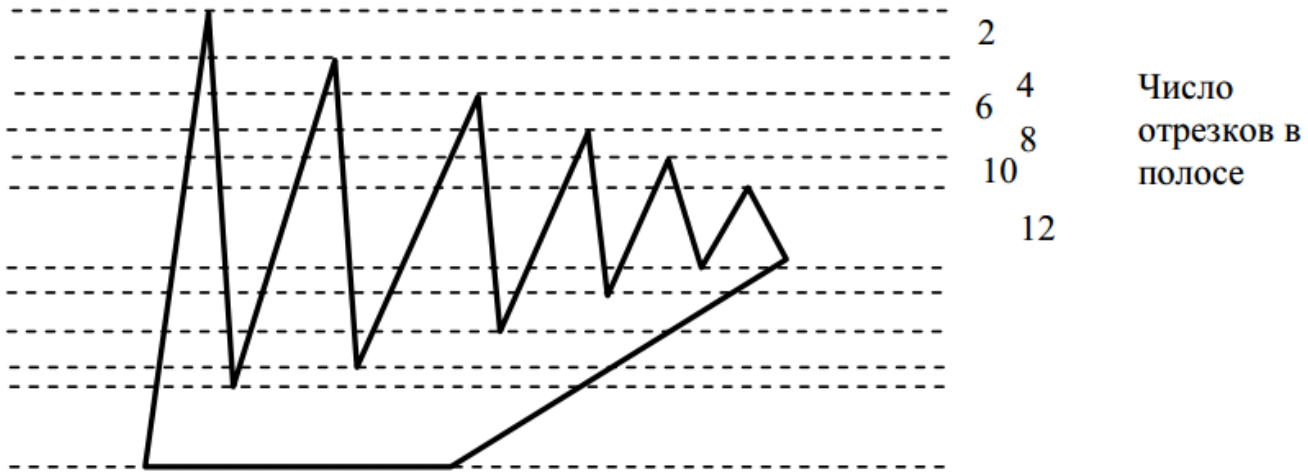
поиск многоугольника, содержащего заданную точку.

Метод полос



Если провести сортировку полос по координате y на этапе предобработки, то появится возможность найти ту полосу, в которой лежит пробная точка, за время $O(log\ n)$.

Метод полос

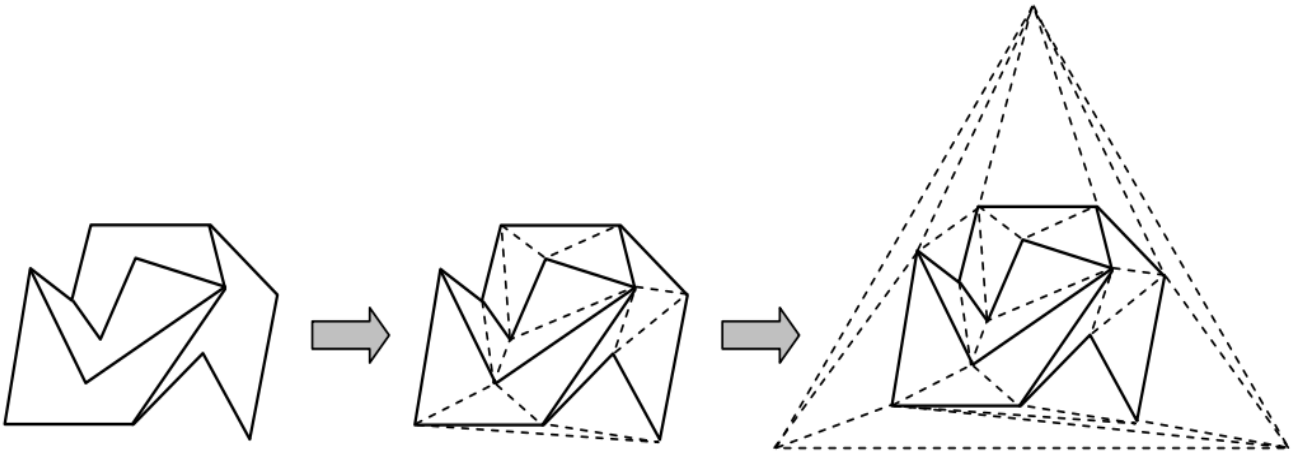


Простая реализация идеи метода полос состоит в сортировке трапеций и треугольников в каждой полосе. Поскольку количество полос $O(n)$, а в каждой полосе число трапеций составляет также $O(n)$, получаем, что общее время составит $O(n^2 log\ n)$.

Метод Киркпатрика

Алгоритм:

1. Сведение ППЛГ к триангуляции (триангуляция граней). (Время сведения $O(nlog\ n)$, память $O(n)$.)
2. Построение охватывающего треугольника.



Построение охватывающего треугольника

Строится последовательность триангуляций $S_1, .S_2, ..., S_{h(n)}$, в которой $S_1 = G$, а S_i получается из S_{i-1} по следующим правилам:

1. Удалим некоторое множество независимых (т.е. несмежных) неграничных вершин триангуляции S_{i-1} и инцидентные к ним рёбра.
2. Вновь триангулируем многоугольники, образовавшиеся в результате удаления вершин и рёбер.

Процесс выполняется до момента, когда в $S_{h(n)}$ не остаётся внутренних рёбер.

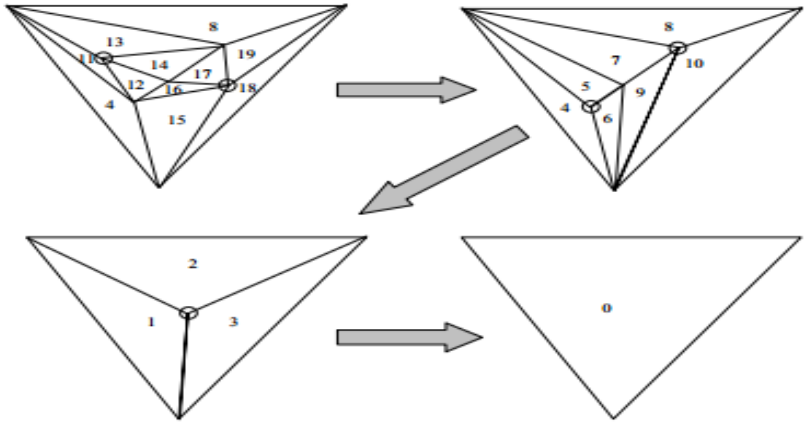
Структура данных

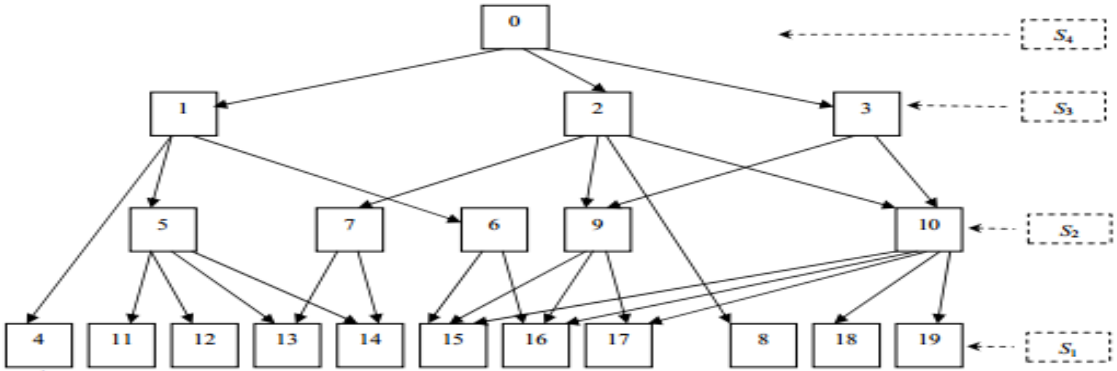
Обозначим треугольник R_j. Введём отношение принадлежности треугольников триангуляциям: $R_j \in S_i$, если R_j создан на шаге 2 при построении S_i.

Структура данных T для поиска – это дерево, узлы которого соответствуют треугольникам. T – ациклический ориентированный граф. От узла R_k к R_j проводится дуга, если при построении S_i S_{i-1} выполнены условия:

1. R_j удаляется из S_{i-1} на шаге (1)
2. R_k создаётся в S_i на шаге (2)
3. $R_j \cap R_k = \varnothing$

Структура данных





Алгоритм локализации

Дано:

— z – точка-запрос,
— (v) – список потомков узла v,
— треугольник(v) – это треугольник, отнесённый к узлу v.

```
PROCEDURE Локализация точки(z);
BEGIN
        IF (z∉треугольник(корень)) THEN
                Result(«z лежит в бесконечной области»)
        ELSE
                V:=корень;
                WHILE (Γ(v)≠∅) DO
                        FOR каждый u∈Γ(v) DO
                                IF (z∈треугольник(u)) THEN
                                        V:=u;
                                END IF;
                        END FOR;
                END WHILE;
                Result(«z лежит в треугольнике(v)»)
        END ELSE;
END Локализация точки.
```

Задача разметки последовательностей

Дано:

последовательность слов (токенов)

Найти:

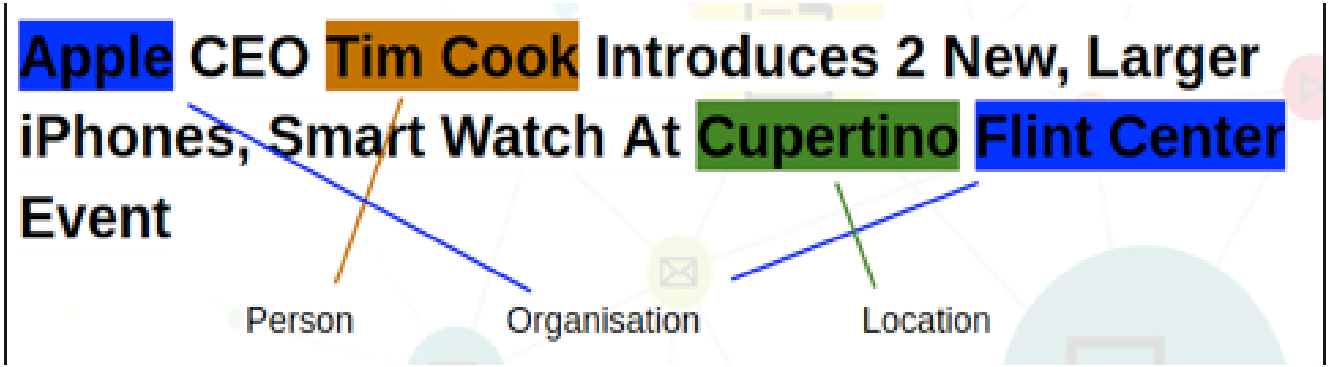
последовательность меток (тэгов)

Примеры задач:

- распознавание частей речи (part of speech tagging, POS)
- распознавание именованных сущностей (named entity recognition, NER)
- выделение семантических ролей (semantic role labeling)
- снятие омонимии слов (word sense disambiguation, WSD)

Распознавание именованных сущностей

Что может быть именованной сущностью:
люди, организации, места, дата, время, количество, ...
Пример определения семантических ролей:



Распознавание именованных сущностей

Задача:
Для каждого слова $(w_1, w_2, ..., w_n), w_i \in W$ в тексте определить, является ли оно частью некоторой именованной сущности
Варианты ответов:

- Входит в именованную сущность/не входит
- BIO-notation: начало сущности (B)/внутри сущности(I)/не сущность(O)
- Тип сущности (персона, место, организация и т.д.)

Задача сводится к классификации каждого слова в последовательности.

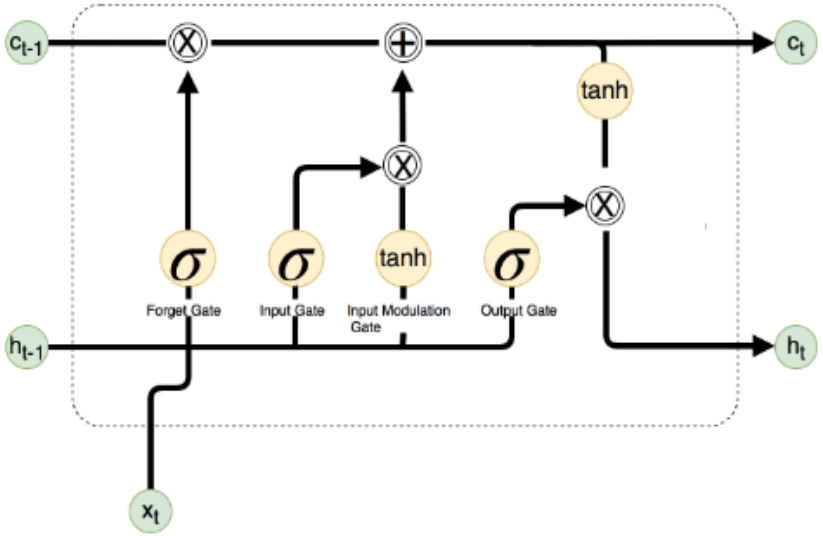
biLSTM + CRF: результаты

Table 2: Comparison of tagging performance on POS, chunking and NER tasks for various models.

		POS	CoNLL2000	CoNLL2003
Random	Conv-CRF (Collobert et al., 2011)	96.37	90.33	81.47
	LSTM	97.10	92.88	79.82
	BI-LSTM	97.30	93.64	81.11
	CRF	97.30	93.69	83.02
	LSTM-CRF	**97.45**	93.80	84.10
	BI-LSTM-CRF	97.43	**94.13**	**84.26**
Senna	Conv-CRF (Collobert et al., 2011)	97.29	94.32	88.67 (89.59)
	LSTM	97.29	92.99	83.74
	BI-LSTM	97.40	93.92	85.17
	CRF	97.45	93.83	86.13
	LSTM-CRF	97.54	94.27	88.36
	BI-LSTM-CRF	**97.55**	**94.46**	**88.83** (**90.10**)

LSTM ячейка

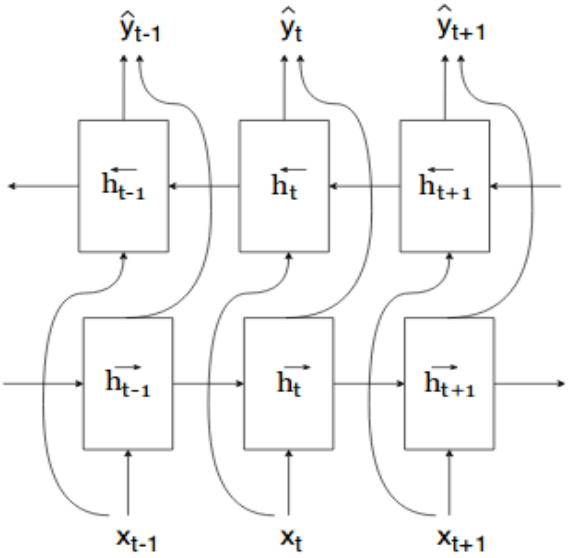
$$z_t = [h_{t-1}, x_t]$$
$$f_t = \sigma(W_f \cdot z_t + b_f)$$
$$i_t = \sigma(W_i \cdot z_t + b_i)$$
$$\hat{C}_t = \text{th}(W_c \cdot z_t + b_c)$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t$$
$$o_t = \sigma(W_o \cdot z_t + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$



bi-LSTM

Конкатенация выходов двух сетей, одна идёт слева направо, другая справа налево:

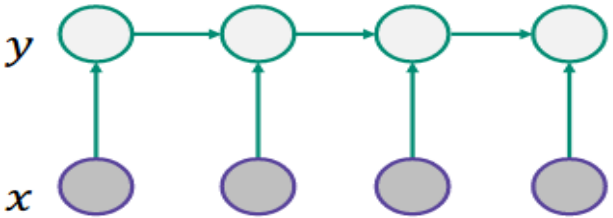
$$\overrightarrow{h_t}, \overrightarrow{C_t} = \overrightarrow{LSTM}(\overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}, x_t)$$
$$\overleftarrow{h_t}, \overleftarrow{C_t} = \overleftarrow{LSTM}(\overleftarrow{h_{t-1}}, \overleftarrow{C_{t-1}}, x_t)$$
$$y_t = g(U[\overrightarrow{h_t}, \overleftarrow{h_t}] + \hat{b})$$



На практике часто работают лучше чем однонаправленные!

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|y_{t-1}, x_t)$$

Дискриминативная модель



$$p(y_t|y_{t-1}, x_t) = \frac{1}{Z_t(y_{t-1}, x_t)} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right)$$

Нормализация

вес

признак

Conditional Random Field (линейный случай)

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right)$$

bi-LSTM + CRF

(Zhiheng Huang, Wei Xu, Kai Yu, Bidirectional LSTM-CRF Models for Sequence Tagging. 2015)