

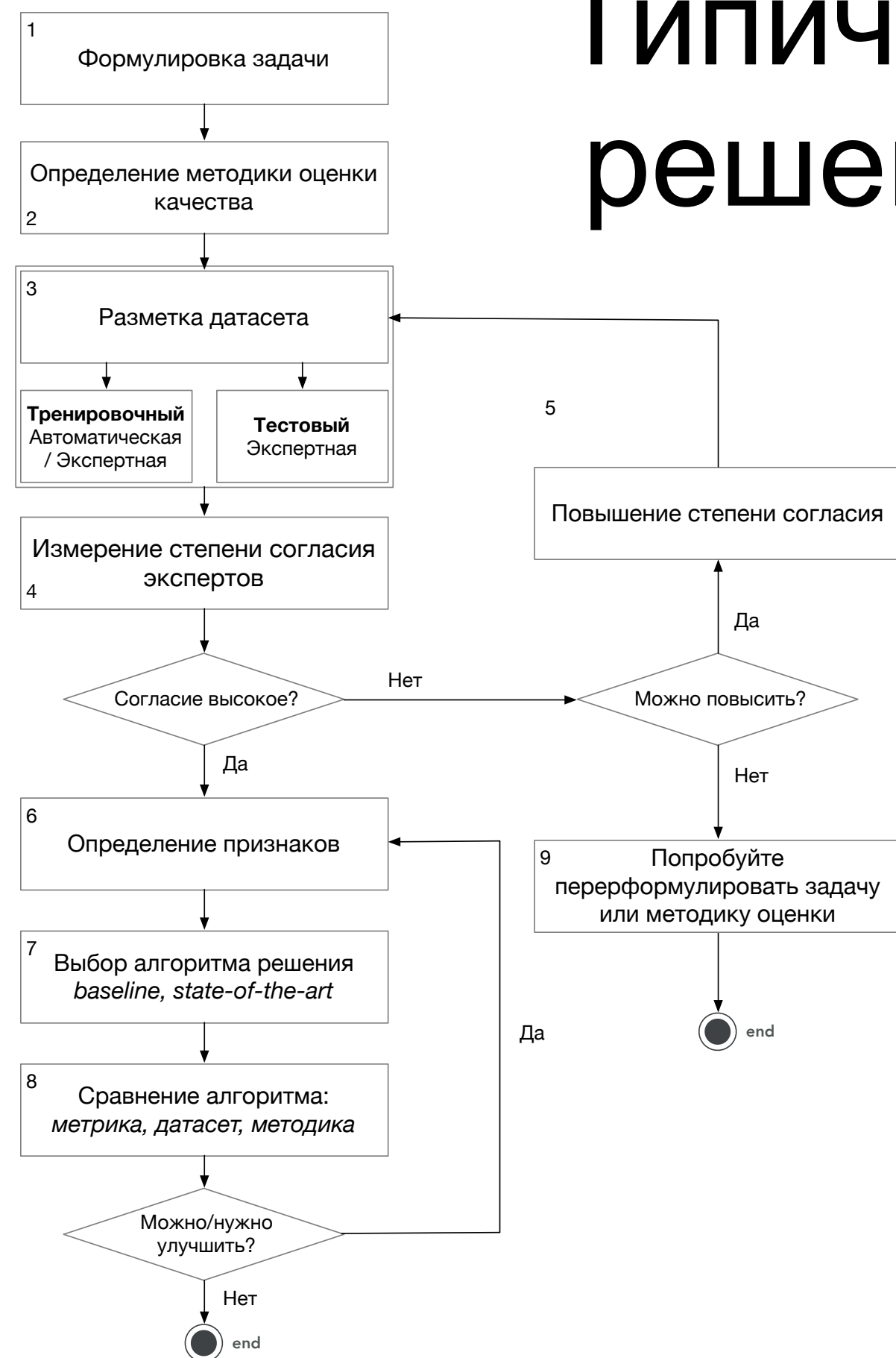
Введение в обработку ТЕКСТОВ

Лекция 2

Методы классификации и кластеризации

Типичная схема решения задач

- На каждом шаге производится *исследовательский поиск*
- Способ оценки качества часто влияет на постановку задачи
- Тестировать алгоритм надо на данных, которые он никогда “не видел”
- Согласие экспертов показывает разрешимость задачи людьми и определяет *верхнюю границу* качества
- Для честного сравнения алгоритмов должны быть зафиксированы датасет и методика
- В реальных задачах шаги 1-5 занимают до 90% всего времени



План

- Задача определения и классификации именованных сущностей (NERC)
- Основные понятия машинного обучения
- Метод опорных векторов (SVM)
- Линейная регрессия, Логистическая регрессия
- Скрытая марковская модель (HMM). Алгоритм Витерби
- Марковская модель максимальной энтропии (MEMM)
- Условные случайные поля (CRF)

Распознавание и классификация именованных сущностей

- Named Entity Recognition and Classification
- На входе: текст, разбитый на предложения и токены
- На выходе: множество сущностей (начало, конец, тип)

Александр Пушкин родился в Москве, столице России
личность город страна

Microsoft — один из крупнейших производителей ПО в мире
компания

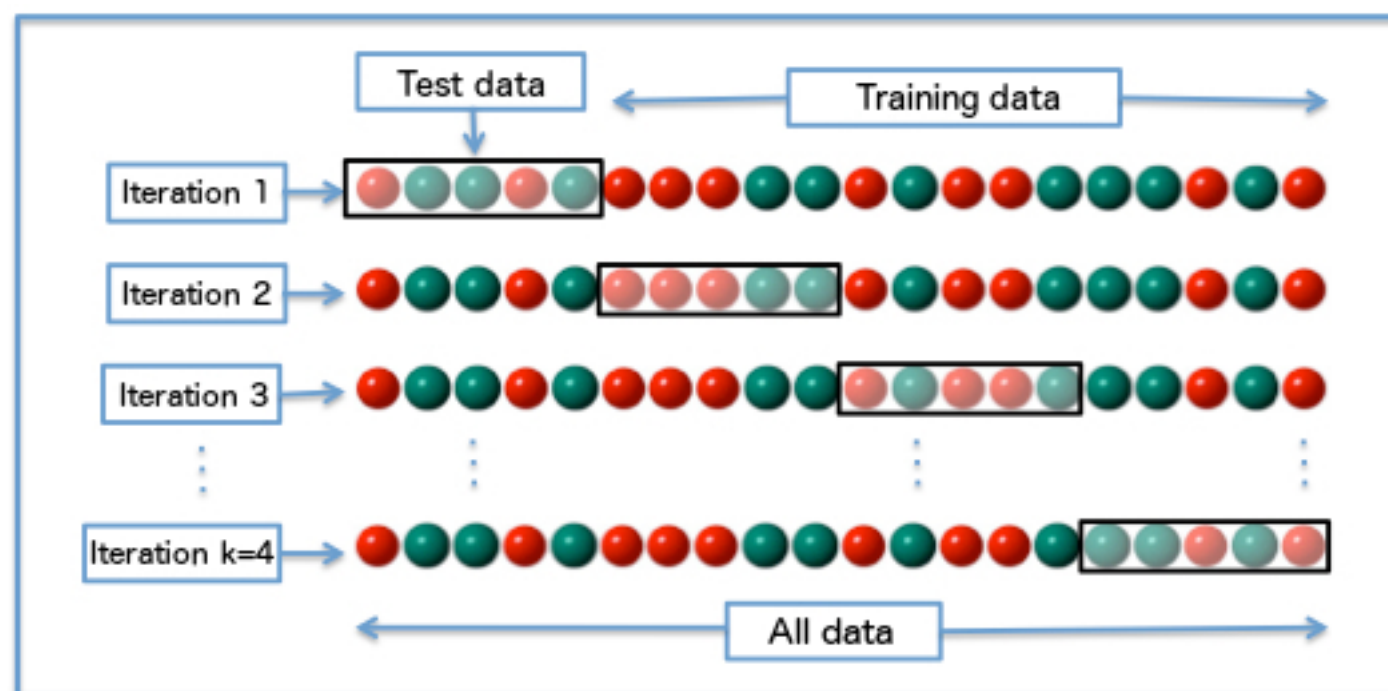
Обработка текстов — область на стыке ИИ и лингвистики
научная дисциплина НД НД

Оценка качества

- Все ли распознанные сущности распознаны верно?
 - Точность (precision)
- Все ли имеющиеся сущности распознаны?
 - Полнота (recall)
- Сбалансированная мера
 - $F_1 = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$

Коллекции данных

- <https://github.com/juand-r/entity-recognition-datasets>
- Зависят от классов (определяют классы) NERC
- Делятся на несколько частей
 - Тренировочная
 - Тестовая
 - Валидационная
- Перекрестная проверка (cross-validation)



Решение через словари

Александр Пушкин родился в Москве

Министерство путей сообщения было упразднено в 2004 году

- Можно составить словари для актуальных типов сущностей и сопоставлять словосочетания в тексте с ними
- Разреженность сущностей:
 - Словари всегда будут неполными
 - Во многих языках слова склоняются => лемматизация
 - Имена и фамилии могут встречаться в произвольном порядке
 - Имеются явные шаблоны «министерство ...» => правила

Решение через правила

Юрий Левитан родился во Владимире

Всеволод Юрьевич «Большое Гнездо» умер во Владимире

Игорь Тальков убит в Санкт-Петербурге

Инцидент произошел в Туле

- Многозначность сущностей:

- «Владимир» - это «личность» или «город»? => анализ контекста

- Разреженность контекста:

- «(родился|умер|убит|...) в(|о) <город>» => (квази-)синонимия
- «<глагол> в(|о) <город>» => части речи

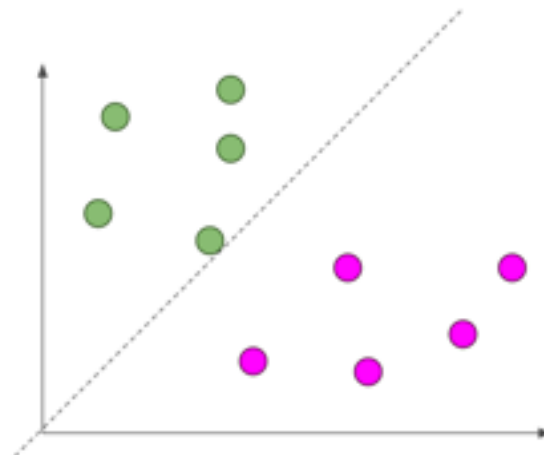
Машинное обучение с учителем

- Составление правил — крайне трудоемкий процесс: в случае задачи NER могут потребоваться тысячи правил
- Разработанные правила скорее всего будут специфичны для какого-то набора типов сущностей, их будет сложно адаптировать к другому набору типов
- Вместо правил мы можем составить выборку текстов, в которой будут вручную размечены примеры сущностей
- Нам понадобится алгоритм, который, «просмотрев» выборку примеров, будет выделять сущности на произвольном тексте

Задача классификации

- Есть множество классов и множество объектов, которые могут относиться к одному или более классам.
- Задача состоит в отнесении объектов с неизвестным классом к одному или более классов
- Факторы, на основе которых делается предсказание класса, называются **признаками** (feature)
- Пример, классификация людей по расам на основе цвета кожи и формы глаз.

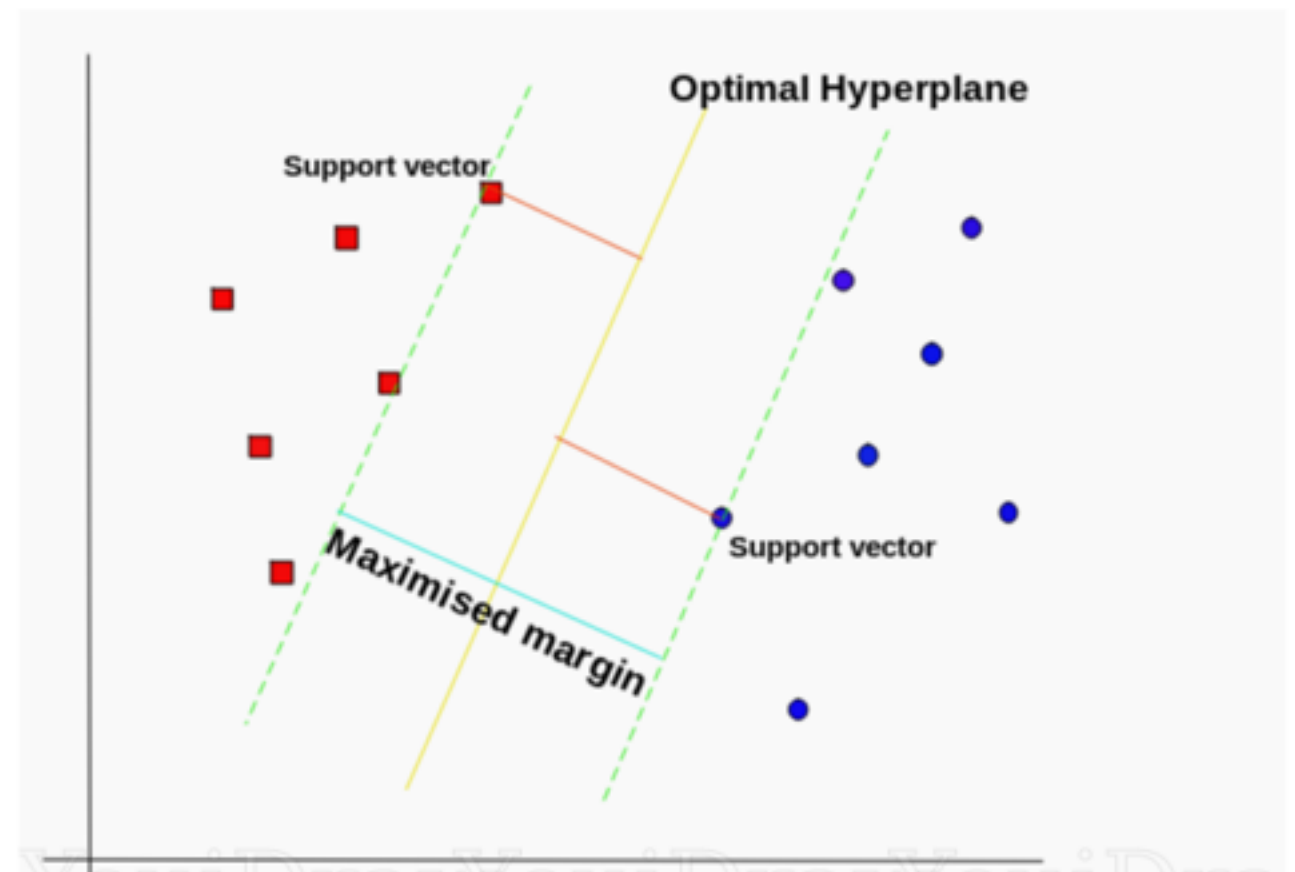
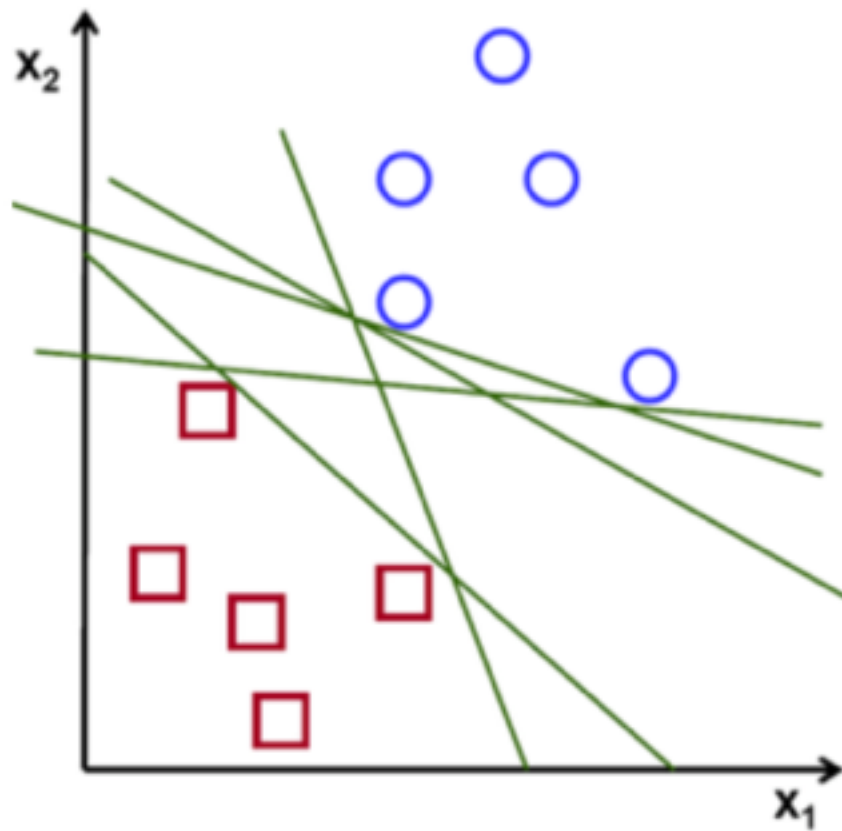
Линейные классификаторы



- Классификация на 2 класса «+» и «-»: слово в контексте — имя человека? текст — спам? будет ли сегодня дождь?
- Каждый классифицируемый объект (слово в контексте, текст, изображение неба) представляется точкой в «признаковом» N-мерном пространстве
- По обучающей выборке строится гиперплоскость, разделяющая точки из противоположных классов

Метод опорных векторов

- Максимизируем расстояние до гиперплоскости
- $w_1x_1 + w_2x_2 + \dots + w_nx_n + b = (w, x) + b = 0$



- liblinear (C, Java), scikit-learn (Python), weka (Java)

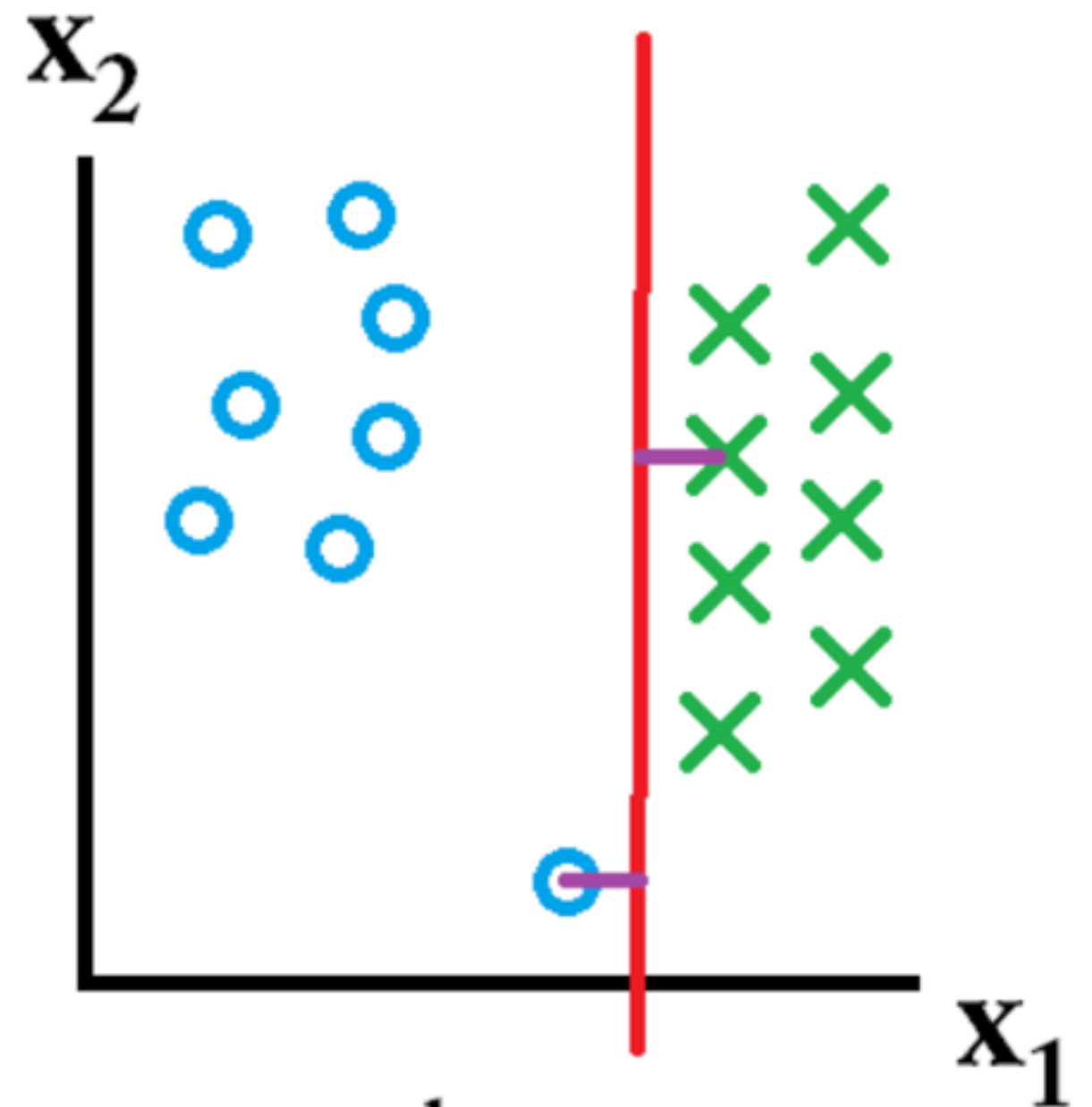
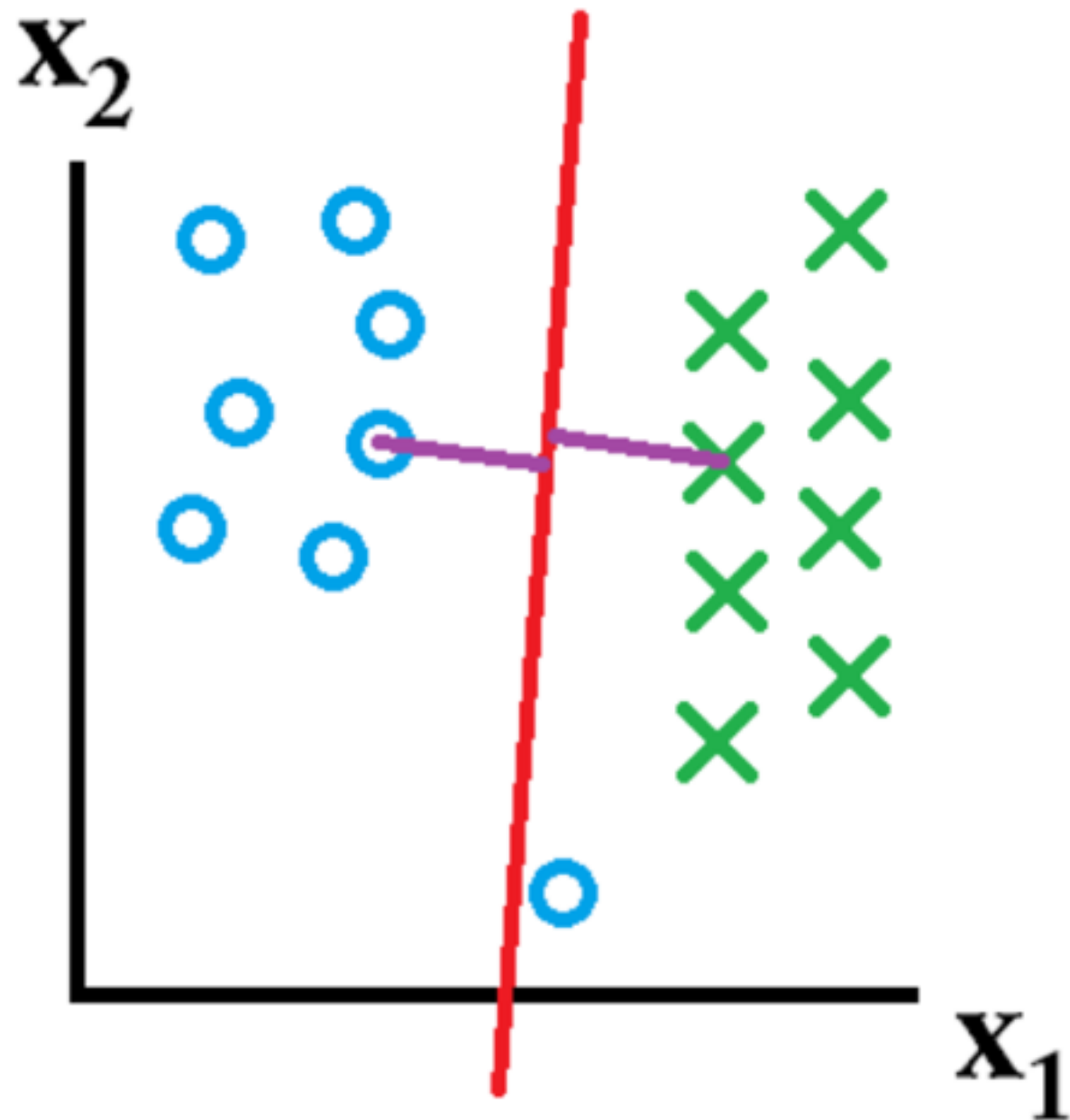
Обучение

- Поиск весов сводится к минимизации функции потерь

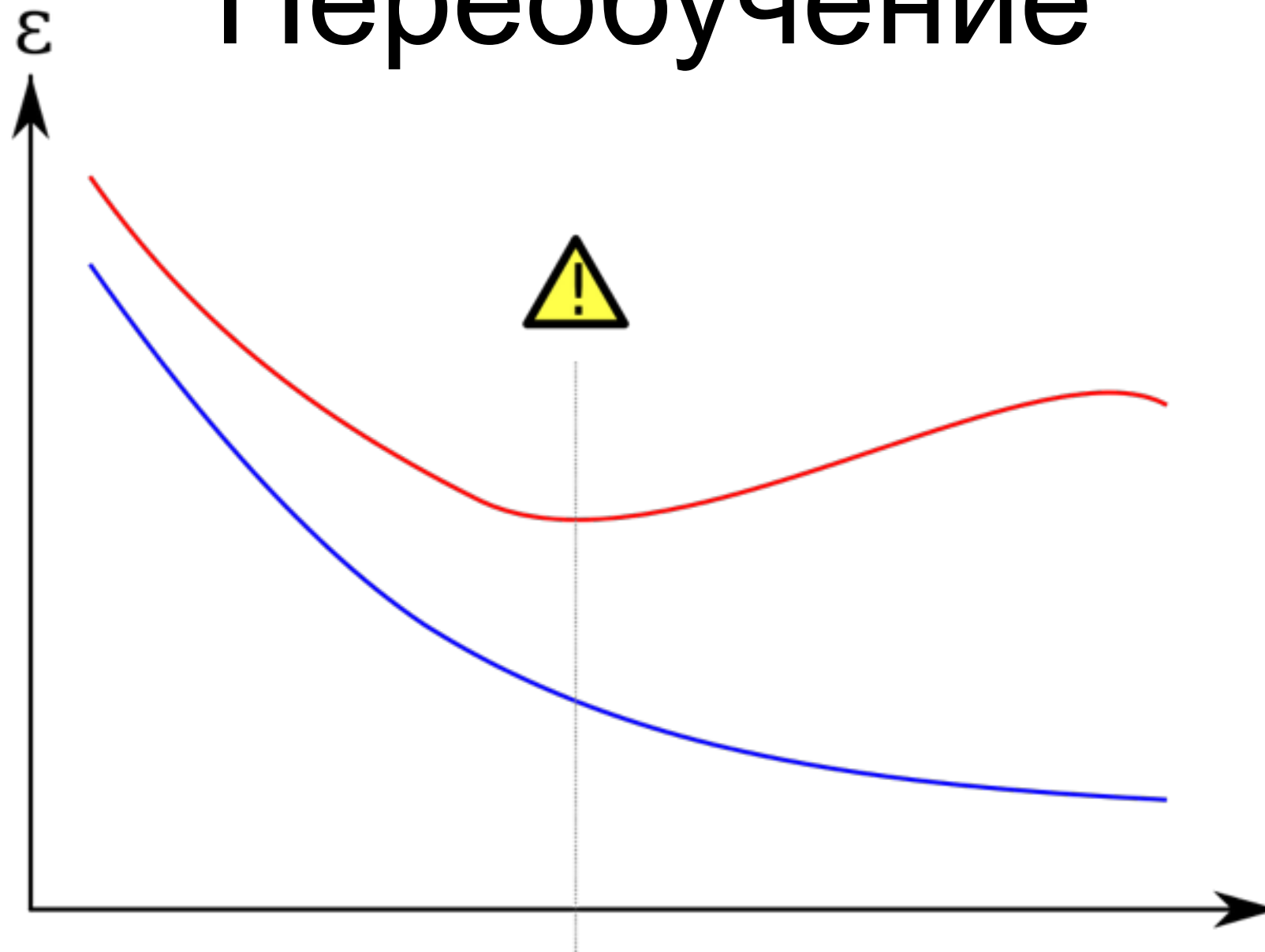
$$\min_w \lambda \|w\|^2 + \sum_i \max\{0, 1 - y_i w^T x_i\}$$

- Задача выпуклой оптимизации
- Например, можно использовать метод градиентного спуска

Переобучение



Переобучение



- (Синий) Ошибка на тренировочных данных
- (Красный) Ошибка на валидационных данных

Пример

```
from sklearn import svm
X = [[0, 0], [1, 1]]
y = [0, 1]

clf = svm.LinearSVC()
clf.fit(X, y)

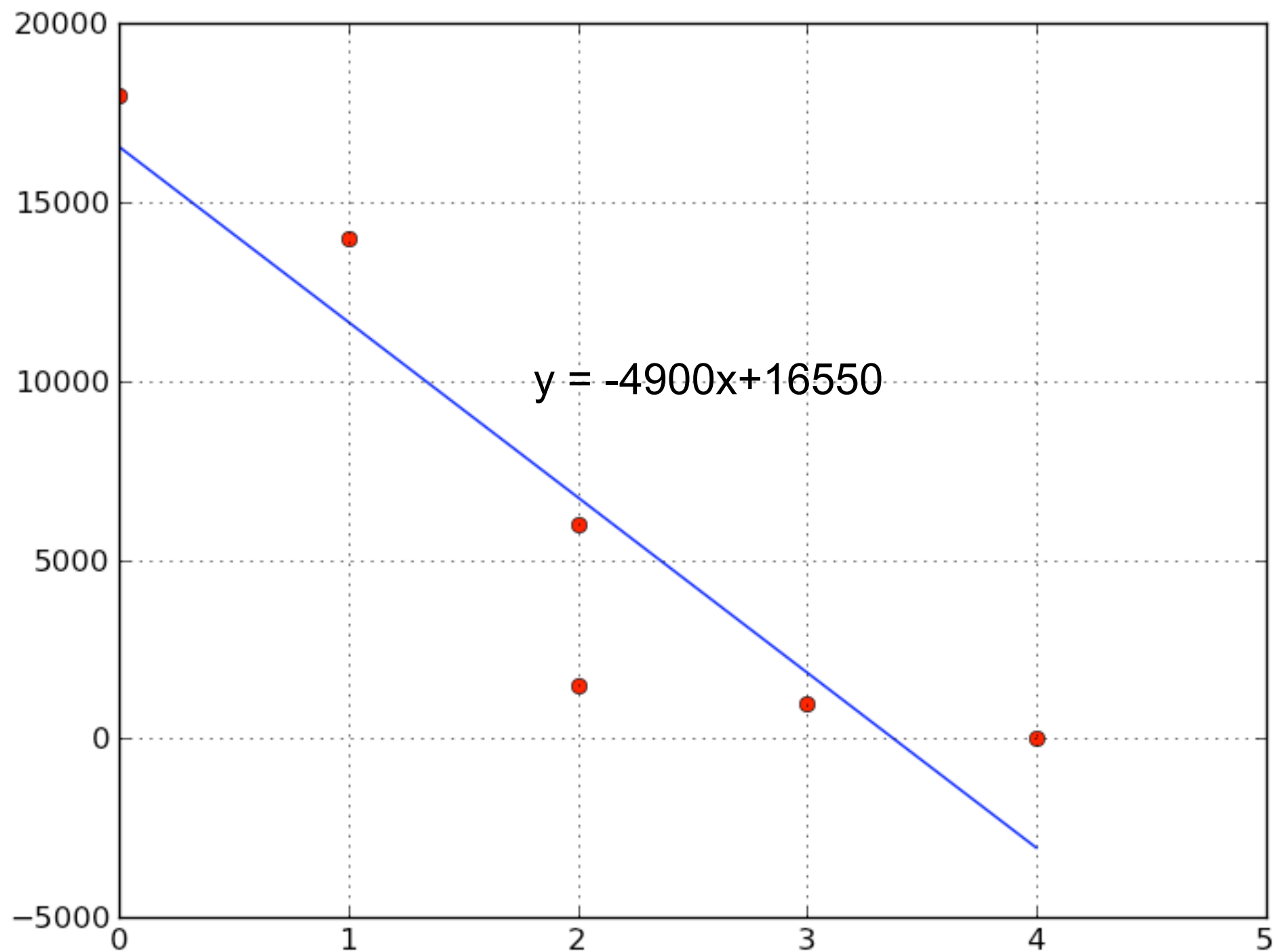
clf.predict([[2., 2.]])
> array([1])
```

Линейная регрессия

Кол-во неопределенных прилагательных	Прибыль сверх запрашиваемой
4	0
3	\$1000
2	\$1500
2	\$6000
1	\$14000
0	\$18000

$$price = w_0 + w_1 * Num_Adjectives$$

Линейная регрессия



Линейная регрессия

$$price = w_0 + w_1 * Num_Adjectives + w_2 * Mortgage_Rate + w_3 * Num_Unsold_Houses$$

- В терминах признаков

$$price = w_0 + \sum_{i=1}^N w_i \times f_i$$

- введем дополнительный признак $f_0 = 1$

$$y = \sum_{i=0}^N w_i \times f_i \quad \text{или} \quad y = w \cdot f$$

Вычисление коэффициентов

- Минимизировать квадратичную погрешность

$$cost(W) = \sum_{j=0}^M (y_{pred}^j - y_{obs}^j)^2$$

- Вычисляется по формуле

$$W = (X^T X)^{-1} X^T \vec{y}$$

Логистическая регрессия

- Перейдем к задаче классификации
- Определить вероятность, с которой наблюдение относится к классу
- Попробуем определить вероятность через линейную модель

$$P(y = true|x) = \sum_{i=0}^N w_i \times f_i = w \cdot f$$

Логистическая регрессия

- Попробуем определить отношение вероятности принадлежать классу к вероятности не принадлежать классу

$$\frac{P(y = true|x)}{1 - P(y = true|x)} = w \cdot f$$

Логистическая регрессия

- Проблема с несоответствием области значений решается введением натурального логарифма

$$\ln \left(\frac{P(y = true|x)}{1 - P(y = true|x)} \right) = w \cdot f$$

- Логит-преобразование

$$\text{logit}(P(x)) = \ln \left(\frac{P(x)}{1 - P(x)} \right)$$

- Определим вероятность ...

Логистическая регрессия

$$P(y = true|x) = \frac{e^{w \cdot f}}{1 + e^{w \cdot f}} \quad P(y = false|x) = \frac{1}{1 + e^{w \cdot f}}$$

- Или

$$P(y = true|x) = \frac{1}{1 + e^{-w \cdot f}} \quad P(y = false|x) = \frac{e^{-w \cdot f}}{1 + e^{-w \cdot f}}$$

- Логистическая функция

$$\frac{1}{1 + e^{-x}}$$

Логистическая регрессия

$$P(y = true|x) > P(y = false|x)$$

$$\frac{P(y = true|x)}{1 - P(y = true|x)} > 1$$

$$e^{w \cdot f} > 1$$

$$w \cdot f > 0$$

$$\sum_{i=0}^N w_i f_i > 0 \quad \text{разделяющая гиперплоскость}$$

Обучение

- Оптимизация функции потерь

$$\min_w \lambda \|w\|^2 + \sum_i \log(1 + \exp(1 - y_i w^T x_i))$$

Мультиномиальная логистическая регрессия

- Классификация на множество классов

$$p(c|x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i\right)$$

$$p(c|x) = \frac{\exp\left(\sum_{i=0}^N w_{ci} f_i\right)}{\sum_{c' \in C} \exp\left(\sum_{i=0}^N w_{c'i} f_i\right)}$$

Пример

```
from sklearn.linear_model import LogisticRegression
```

```
X = [[0, 0], [1, 1]]
```

```
y = [0, 1]
```

```
clf = LogisticRegression()  
clf.fit(X, y)
```

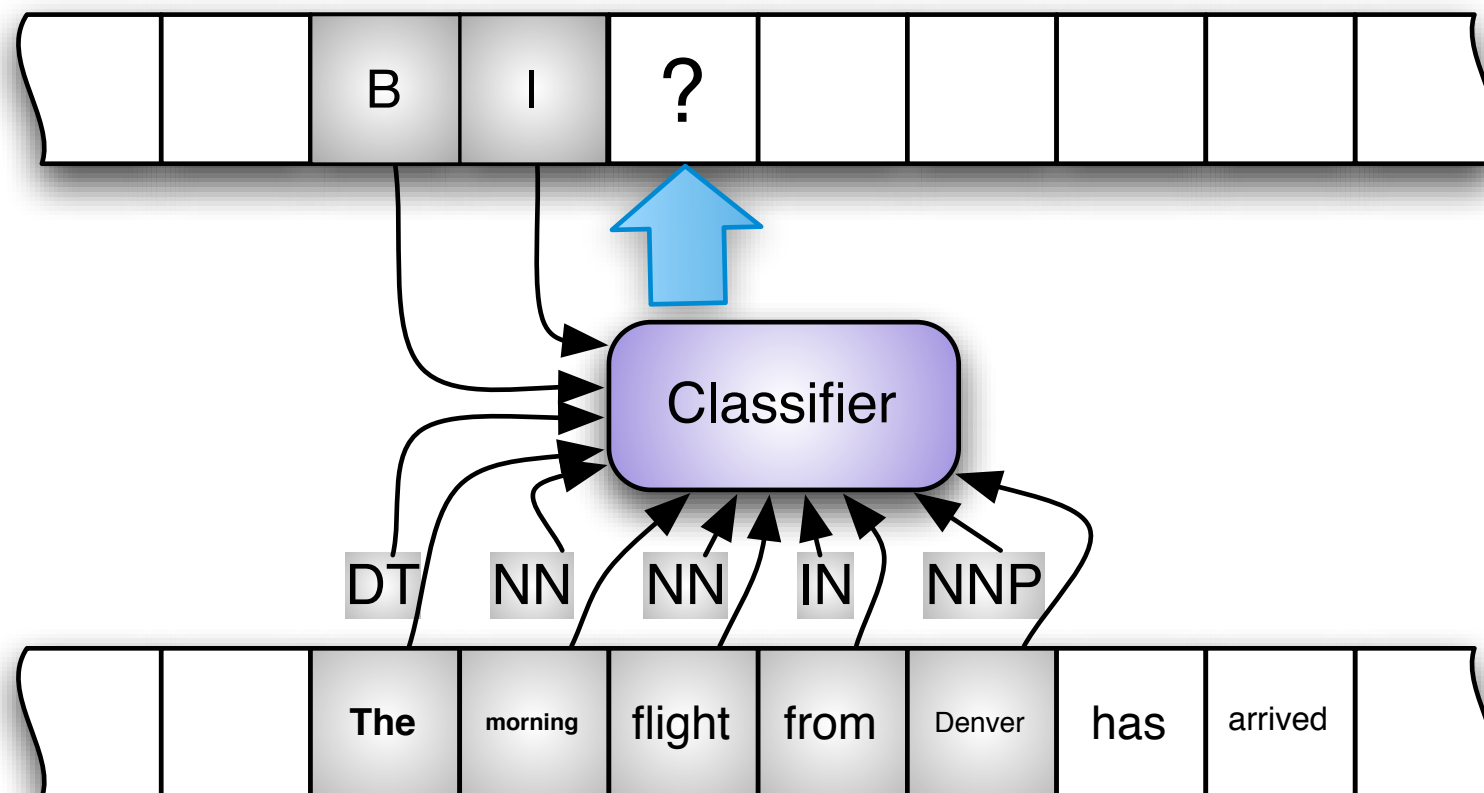
```
clf.predict([[2., 2.]])
```

```
> array([1])
```

NERC и классификация

• Классы + метки

- IO (inside, outside)
- BIO (begin, inside, outside) - стандарт
- BMEWO (begin, middle, end, whole, outside)



Скрытая мартовская модель (НММ)

- *Из окна сильно дуло*

$$\hat{t}_1^n = \arg \max_{t_1^n} P(t_1^n | w_1^n)$$

- Правило Байеса $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$

- В нашем случае

$$\hat{t}_1^n = \arg \max_{t_1^n} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$

Оценка параметров

$$\hat{t}_1^n = \arg \max_{t_1^n} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \arg \max_{t_1^n} P(w_1^n | t_1^n) P(t_1^n)$$

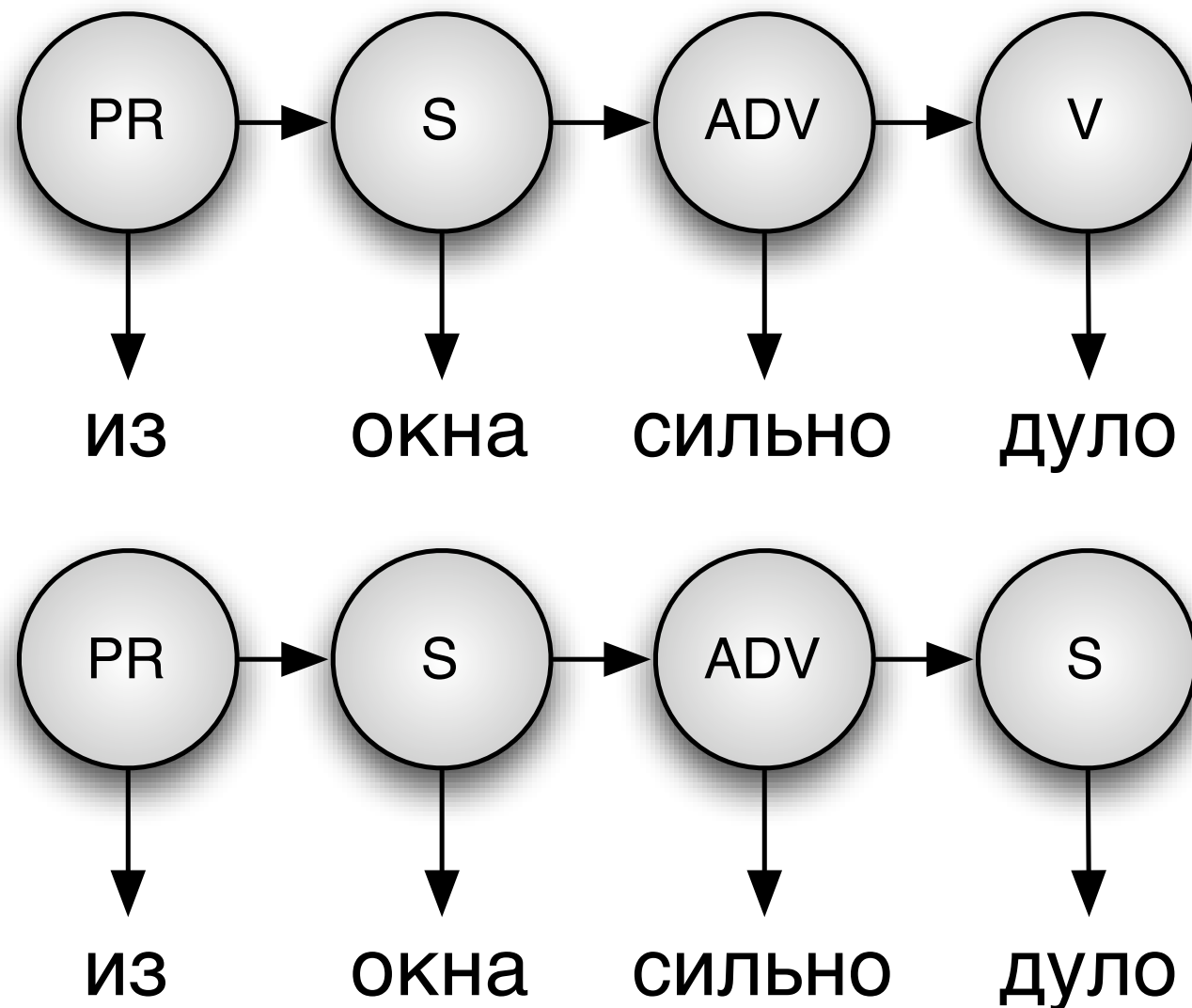
- Предположение 1

$$P(w_1^n | t_1^n) = \prod_{i=1}^n P(w_i | t_i)$$

- Предположение 2

$$P(t_1^n) = \prod_{i=1}^n P(t_i | t_{i-1})$$

Автомат



- Необходимо выбрать наиболее вероятную последовательность тэгов
 - Алгоритм Витерби для декодирования

Алгоритм Витерби

- Алгоритм динамического программирования
- Находит наиболее вероятную последовательность скрытых состояний (тэгов) за линейное (от длины входа) время
- Идея: Для подсчета наиболее вероятной последовательности длины $k+1$ нужно знать:
 - вероятность перехода между тэгами
 - вероятность слова при условии тэга
 - наиболее вероятные последовательности тэгов для последовательностей длины k

Алгоритм Витерби

```

1 comment: Given: a sentence of length  $n$ 
2 comment: Initialization
3  $\delta_1(\text{PERIOD}) = 1.0$ 
4  $\delta_1(t) = 0.0$  for  $t \neq \text{PERIOD}$ 
5 comment: Induction
6 for  $i := 1$  to  $n$  step 1 do
7   for all tags  $t^j$  do
8      $\delta_{i+1}(t^j) := \max_{1 \leq k \leq T} [\delta_i(t^k) \times P(w_{i+1}|t^j) \times P(t^j|t^k)]$ 
9      $\psi_{i+1}(t^j) := \arg \max_{1 \leq k \leq T} [\delta_i(t^k) \times P(w_{i+1}|t^j) \times P(t^j|t^k)]$ 
10  end
11 end
12 comment: Termination and path-readout
13  $X_{n+1} = \arg \max_{1 \leq j \leq T} \delta_{n+1}(j)$ 
14 for  $j := n$  to 1 step  $-1$  do
15    $X_j = \psi_{j+1}(X_{j+1})$ 
16 end
17  $P(X_1, \dots, X_n) = \max_{1 \leq j \leq T} \delta_{n+1}(t^j)$ 

```

Пример

The bear is on the move

First tag	Second tag					
	AT	BEZ	IN	NN	VB	PERIOD
AT	0	0	0	48636	0	19
BEZ	1973	0	426	187	0	38
IN	43322	0	1325	17314	0	185
NN	1067	3720	42470	11773	614	21392
VB	6072	42	4758	1476	129	1522
PERIOD	8016	75	4656	1329	954	0

	AT	BEZ	IN	NN	VB	PERIOD
<i>bear</i>	0	0	10	0	43	0
<i>is</i>	0	10065	0	0	0	0
<i>move</i>	0	0	0	36	133	0
<i>on</i>	0	0	5484	0	0	0
<i>president</i>	0	0	0	382	0	0
<i>progress</i>	0	0	0	108	4	0
<i>the</i>	69016	0	0	0	0	0
<i>.</i>	0	0	0	0	0	48809

+ добавим сглаживание Лапласа

Пример

Считаем вероятности

	AT	BEZ	IN	NN	VB	PERIOD
AT	2.05478e-05	2.05478e-05	2.05478e-05	0.999384	2.05478e-05	0.000410956
BEZ	0.748862	0.000379363	0.161988	0.0713202	0.000379363	0.0147951
IN	0.69687	1.60854e-05	0.0213293	0.278519	0.00017694	0.00299189
NN	0.0131774	0.0459111	0.524023	0.145272	0.0075881	0.263955
VB	0.433445	0.00306902	0.339662	0.105417	0.00927842	0.1087
PERIOD	0.532974	0.00505252	0.3096	0.0884191	0.0634889	6.64805e-05

	AT	BEZ	IN	NN	VB	PERIOD
bear	1.44877e-05	9.92753e-05	0.00199927	0.00187266	0.234043	2.04847e-05
is	1.44877e-05	0.999305	0.000181752	0.00187266	0.00531915	2.04847e-05
move	1.44877e-05	9.92753e-05	0.000181752	0.0692884	0.712766	2.04847e-05
on	1.44877e-05	9.92753e-05	0.99691	0.00187266	0.00531915	2.04847e-05
president	1.44877e-05	9.92753e-05	0.000181752	0.717228	0.00531915	2.04847e-05
progress	1.44877e-05	9.92753e-05	0.000181752	0.20412	0.0265957	2.04847e-05
the	0.999899	9.92753e-05	0.000181752	0.00187266	0.00531915	2.04847e-05
.	1.44877e-05	9.92753e-05	0.000181752	0.00187266	0.00531915	0.999857

Пример

Чтобы не работать произведением вероятностей будем суммировать логарифмы вероятностей

	AT	BEZ	IN	NN	VB	PERIOD
AT	-10.7928	-10.7928	-10.7928	-0.00061662	-10.7928	-7.79702
BEZ	-0.289201	-7.87702	-1.82023	-2.64058	-7.87702	0.0147951
IN	-0.361157	-11.0376	-3.84767	-1.27827	-8.6397	-5.81185
NN	-4.32925	-3.08105	-0.64622	-1.92915	-4.88117	-1.33198
VB	-0.83599	-5.7864	-1.07981	-2.24983	-4.68006	-2.21916
PERIOD	-0.629282	-5.28787	-1.17247	-2.42567	-2.75689	-9.6186

	AT	BEZ	IN	NN	VB	PERIOD
bear	-11.1422	-9.21761	-6.21497	-6.2804	-1.45225	-10.7958
is	-11.1422	-0.000695169	-8.61287	-6.2804	-5.23644	-10.7958
move	-11.1422	-9.21761	-8.61287	-2.66948	-0.338602	-10.7958
on	-11.1422	-9.21761	-0.00309457	-6.2804	-5.23644	-10.7958
president	-11.1422	-9.21761	-8.61287	-0.332361	-5.23644	-10.7958
progress	-11.1422	-9.21761	-8.61287	-1.58905	-3627	-10.7958
the	-0.000101419	-9.21761	-8.61287	-6.2804	-5.23644	-10.7958
.	-11.1422	-9.21761	-8.61287	-6.2804	-5.23644	-0.000143403

		The	bear	is	on	the	move
AT	-1.79						
BEZ	-1.79						
IN	-1.79						
NN	-1.79						
VB	-1.79						
(.)	-1.79						

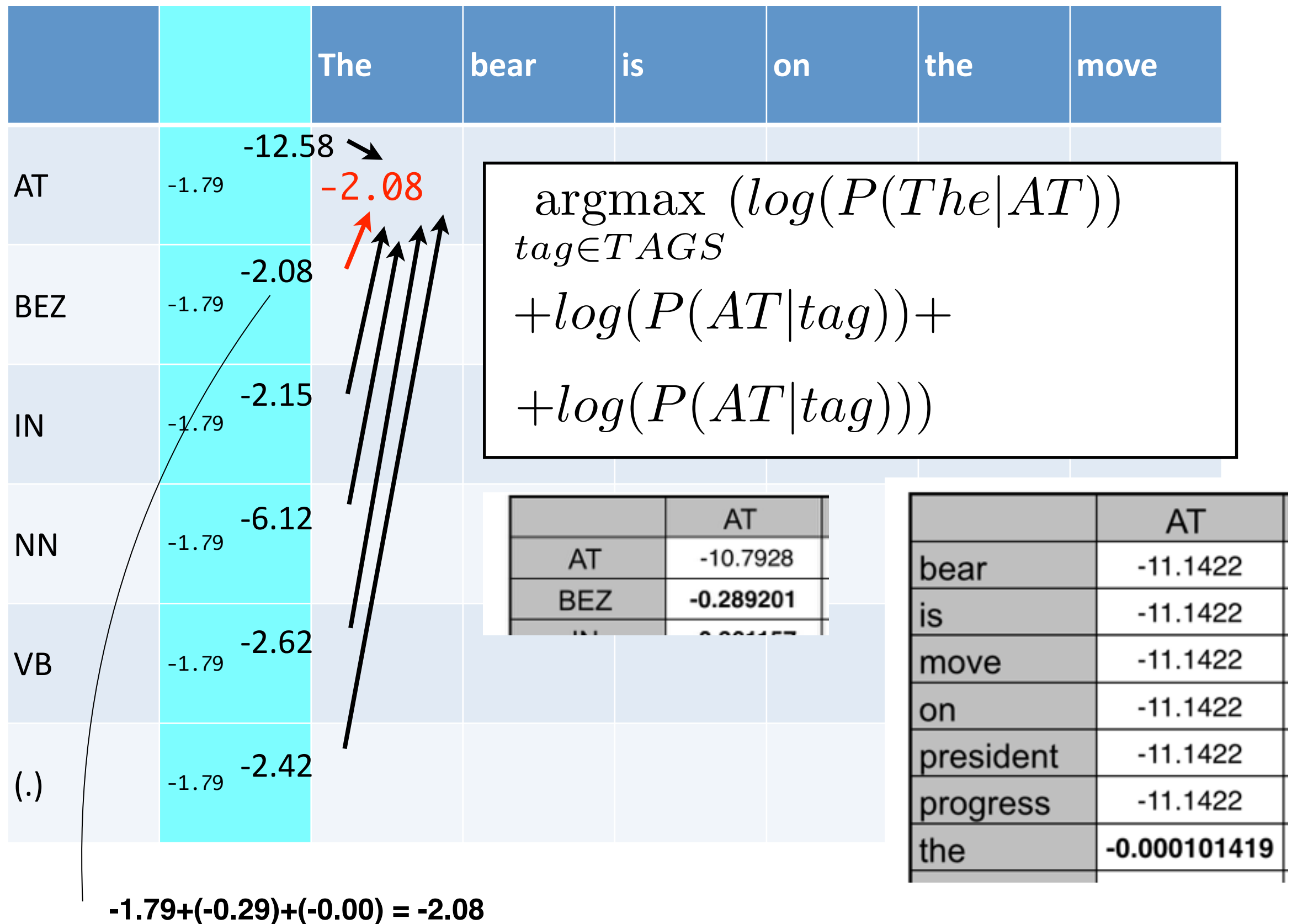
		The	bear	is	on	the	move
AT	-12.58	-1.79					
BEZ	-2.08	-1.79					
IN	-2.15	-1.79					
NN	-6.12	-1.79					
VB	-2.62	-1.79					
(.)	-2.42	-1.79					

$$\operatorname{argmax}_{tag \in TAGS} (\log(P(The|AT)) + \log(P(AT|tag)) + \log(P(AT|tag)))$$

	AT
AT	-10.7928
BEZ	-0.289201
IN	-0.001457

	AT
bear	-11.1422
is	-11.1422
move	-11.1422
on	-11.1422
president	-11.1422
progress	-11.1422
the	-0.000101419

$-1.79 + (-0.29) + (-0.00) = -2.08$



		The	bear	is	on	the	move
AT	-1.79	-2.08					
BEZ	-1.79	-14.09					
IN	-1.79	-11.05					
NN	-1.79	-8.07					
VB	-1.79	-9.78					
(.)	-1.79	13.91					

The diagram illustrates a transformation or mapping from the second column (labeled with values -1.79) to the third column (labeled with various numerical values). Arrows point from the second column to the third column for each row, indicating a relationship or transformation between the two columns.

Основы обработки текстов

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76	-23.83	-22.87	-13.62	-35.56
BEZ	-1.79	-14.09	-20.37	-11.44	-28.53	-32.66	-33.12
IN	-1.79	-11.05	-14.93	-17.62	-13.26	-25.72	-30.08
NN	-1.79	-8.07	-8.36	-16.57	-20.36	-20.82	-16.29
VB	-1.79	-9.78	-14.32	-18.47	-24.55	-27.14	-24.75
(.)	-1.79	13.91	-20.20	-20.48	-26.45	-29.87	-32.22

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76	-23.83	-22.87	-13.62	-35.56
BEZ	-1.79	-14.09	-20.37	-11.44	-28.53	-32.66	-33.12
IN	-1.79	-11.05	-14.93	-17.62	-13.26	-25.72	-30.08
NN	-1.79	-8.07	-8.36	-16.57	-20.36	-20.82	-16.29
VB	-1.79	-9.78	-14.32	-18.47	-24.55	-27.14	-24.75
(.)	-1.79	13.91	-20.20	-20.48	-26.45	-29.87	-32.22

the/AT bear/NN is/BEZ on/IN the/AT move/NN

Вероятность: 8.34932985587e-08

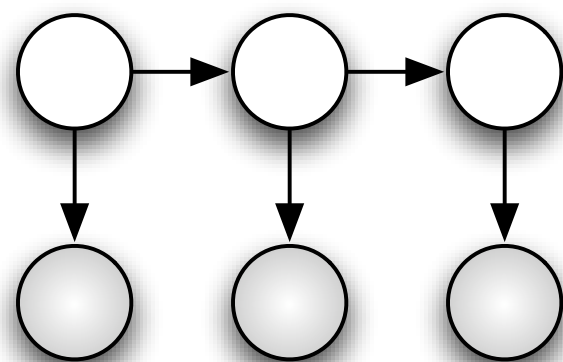
Марковская модель максимальной энтропии

- Позволяет смоделировать сложные признаки (например для определения части речи)

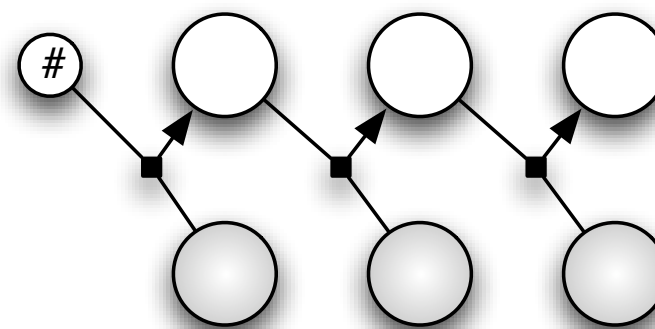
$$\hat{T} = \arg \max_t P(T|W) = \arg \max_T \prod_i P(tag_i | word_i, tag_{i-1})$$

- Сравнить с марковской моделью

$$\hat{T} = \arg \max_t P(T|W) = \arg \max_T \prod_i P(word_i | tag_i) P(tag_i, tag_{i-1})$$

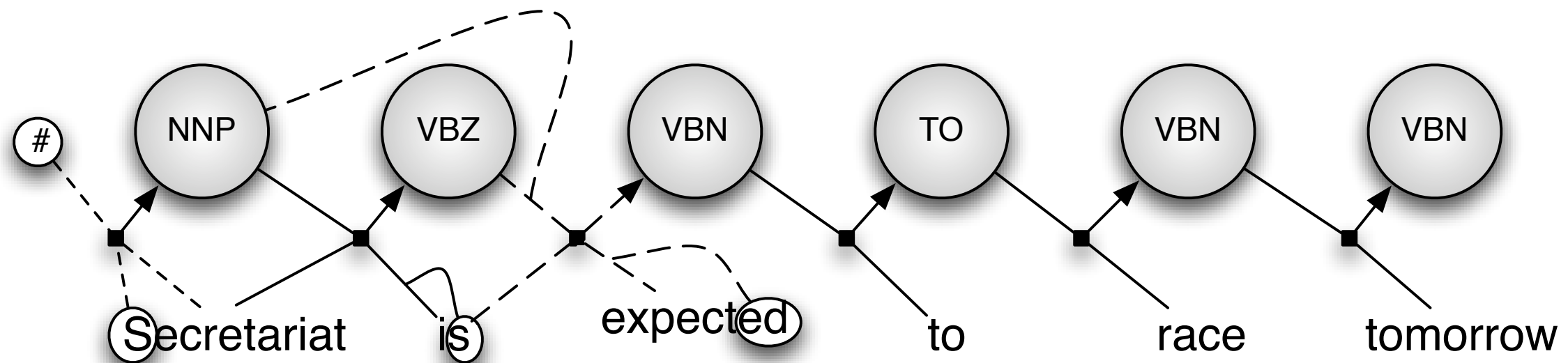


Скрытые марковские
модели



Скрытые марковские
модели максимальной
энтропии

Признаки в MEMM



$$P(q|q', o) = \frac{1}{Z(o, q')} \exp \left(\sum_i w_i f_i(o, q) \right)$$

Декодирование и обучение

- Декодирование - алгоритм Витерби, где на каждом шаге вычисляется

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) P(s_j | s_i, o_t), 1 \leq j \leq N, 1 < t \leq T$$

- Обучение аналогично логистической регрессии

Условные случайные поля (CRF)

$$\bar{s} = s_1, s_2, \dots, s_n$$

$$\bar{o} = o_1, o_2, \dots, o_n$$

HMM

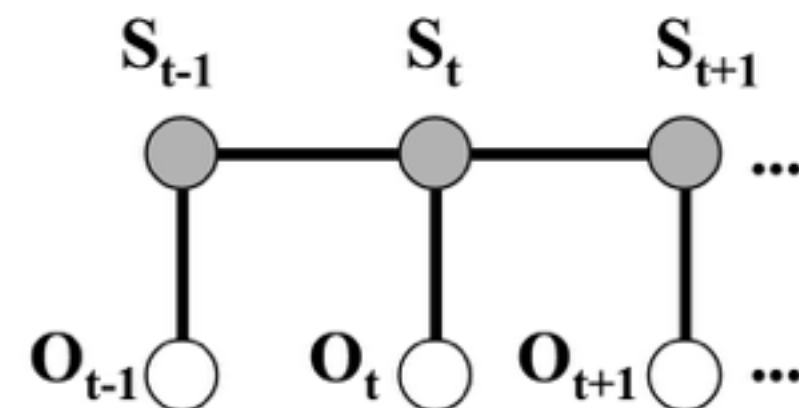
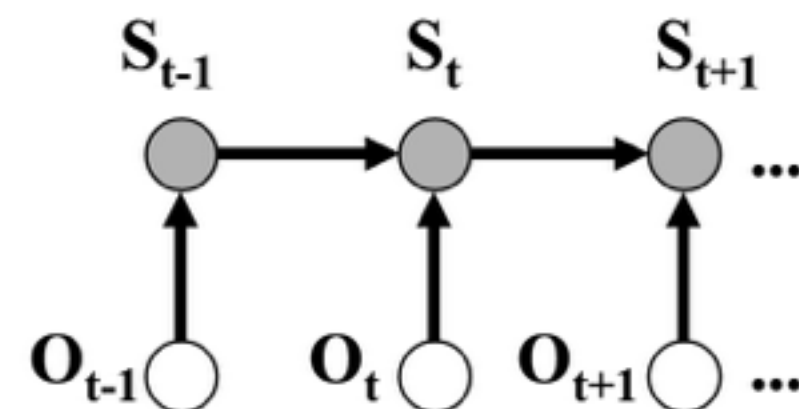
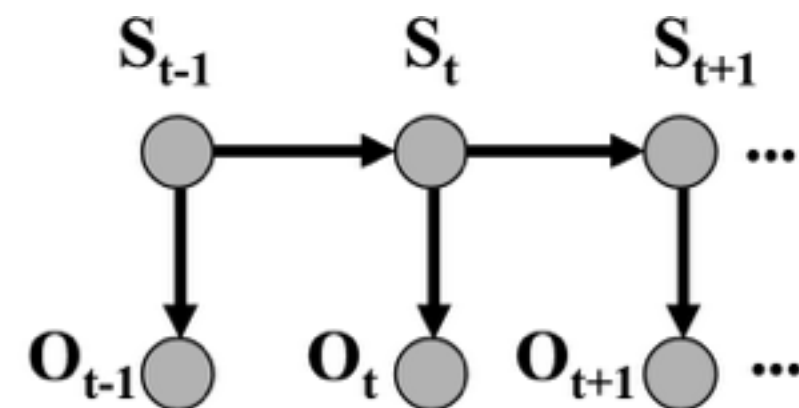
$$P(\bar{s}, \bar{o}) \propto \prod_{t=1}^{|\bar{o}|} P(s_t | s_{t-1}) P(o_t | s_t)$$

MEMM

$$P(\bar{s} | \bar{o}) \propto \prod_{t=1}^{|\bar{o}|} P(s_t | s_{t-1}, o_t)$$

CRF

$$P(\bar{s} | \bar{o}) \propto \frac{1}{Z_{\bar{o}}} \prod_{t=1}^{|\bar{o}|} \exp \left(\sum_j \lambda_j f_j(s_t, s_{t-1}) + \sum_k \mu_k g_k(s_t, o_t) \right)$$



Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

Следующая лекция

- Методы классификации текстов на основе искусственных нейронных сетей