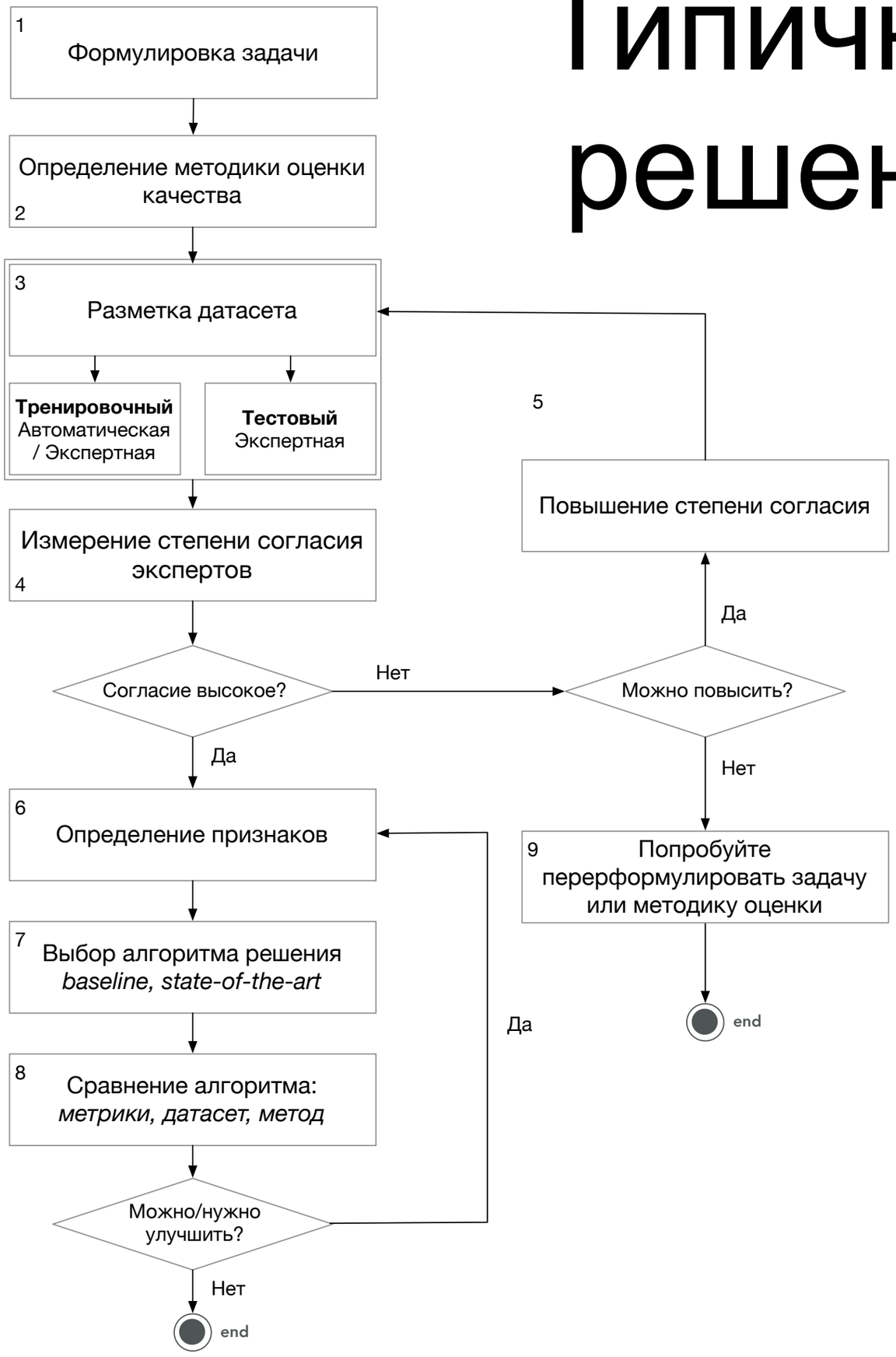
Основы обработки текстов

Лекция 2

Методы классификации и кластеризации

Типичная схема решения задач



- 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

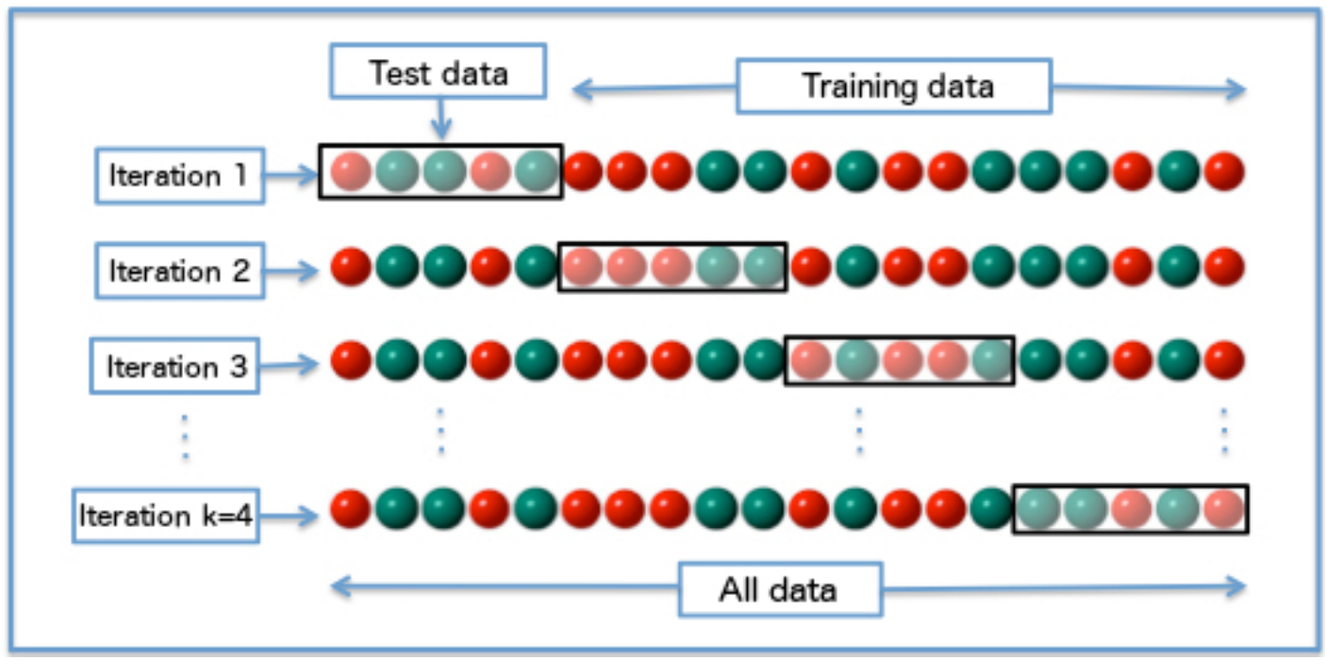
научная дисциплина НД НД

Оценка качества

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

Наборы 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)



https://en.wikipedia.org/wiki/Cross-validation_(statistics)

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**

* есть классы, где можно перечислить все варианты: например, страны.
Но нужно понимать ограничения предметной области: словарь со странами может не сработать при анализе фантастической литературы

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**

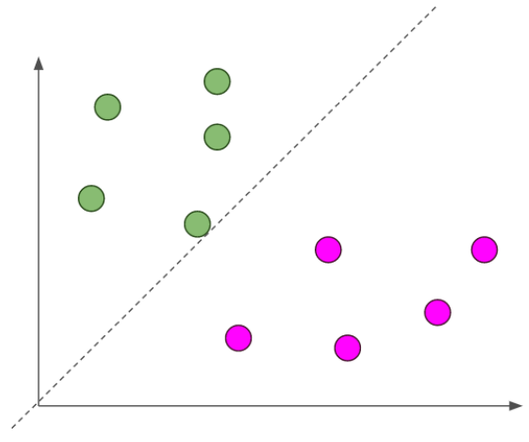
Машинное обучение с учителем

- 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 классам
- Факторы, на основе которых делается предсказание класса, называются **признаками** (feature)

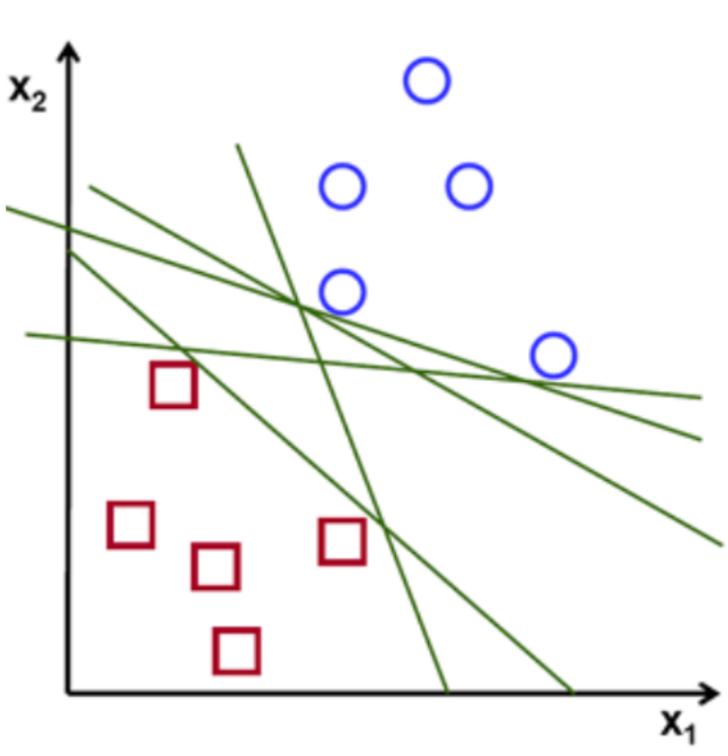
Линейные классификаторы

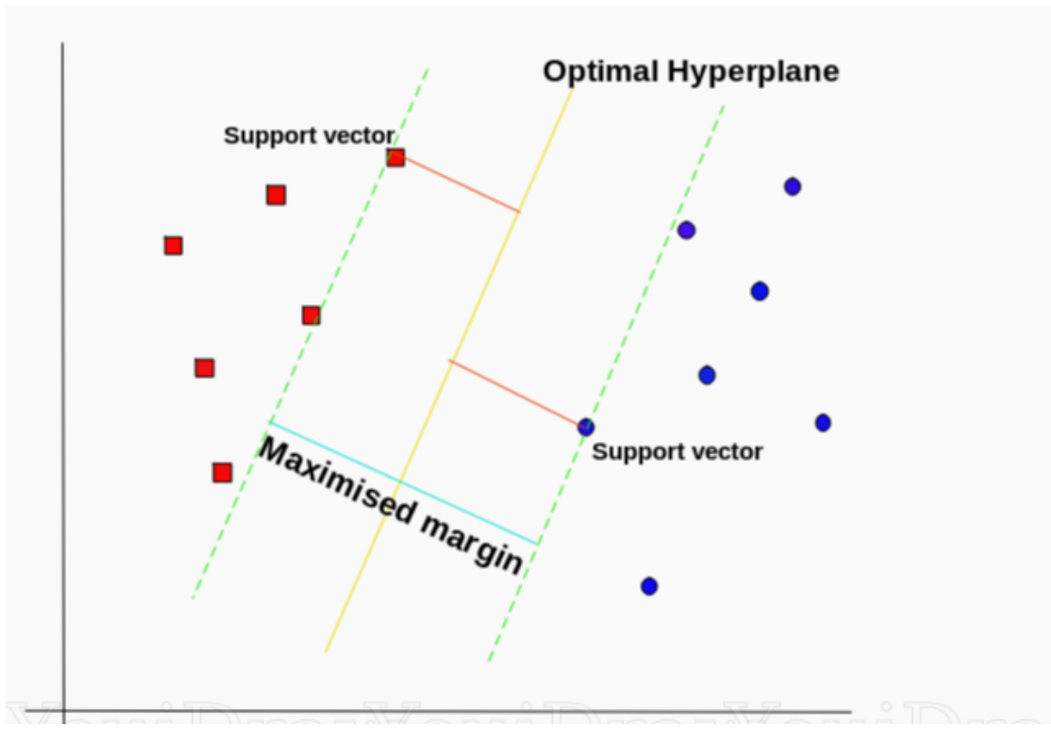


- Классификация на 2 класса «+» и «-»: слово в контексте — имя человека? текст — спам? будет ли сегодня дождь?
- Каждый классифицируемый объект (слово в контексте, текст, изображение неба) представляется точкой в «признаковом» N-мерном пространстве
- По обучающей выборке строится гиперплоскость, разделяющая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

Метод опорных векторов

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





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

Обучение

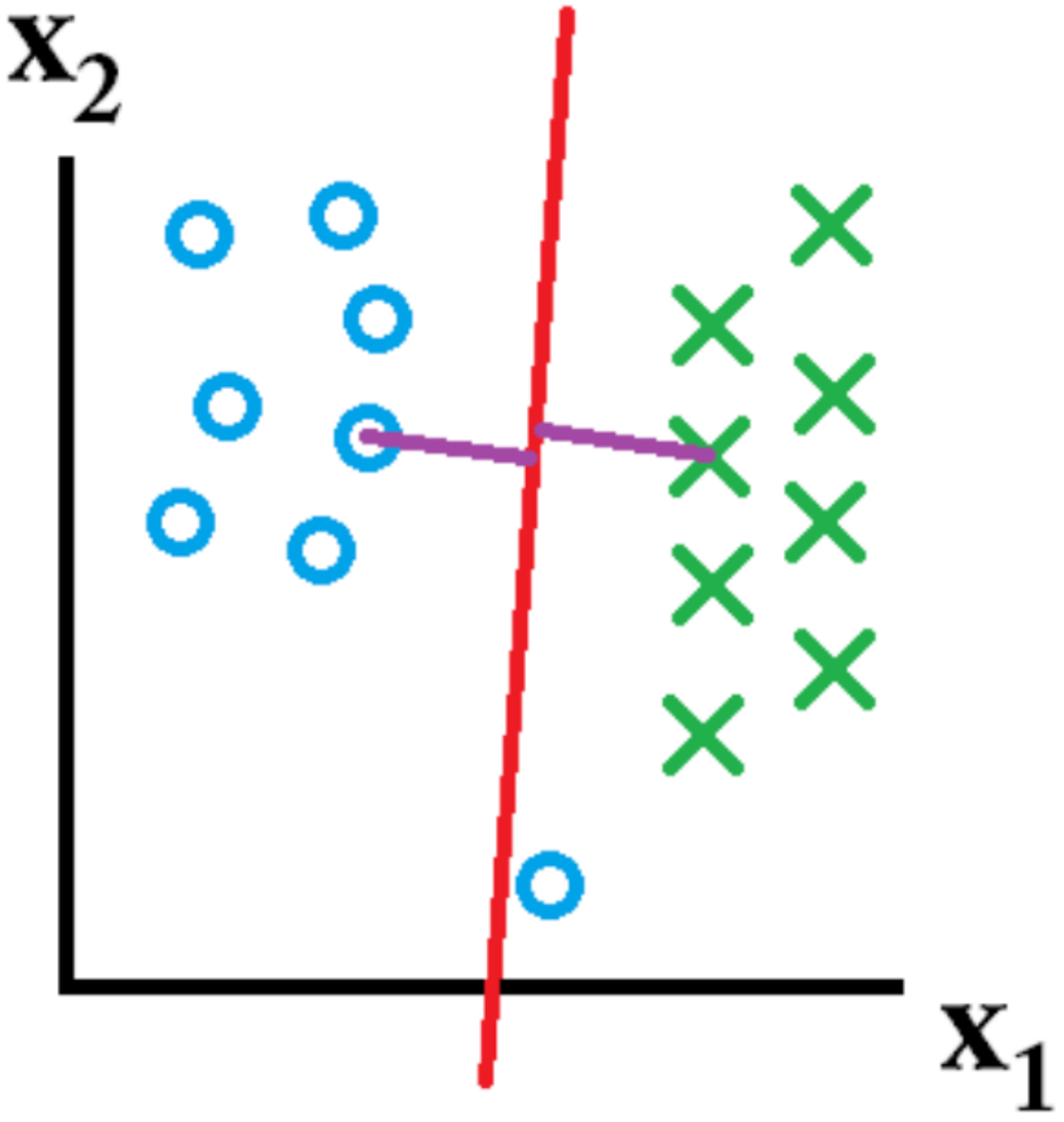
- Поиск весов сводится к минимизации функции потерь

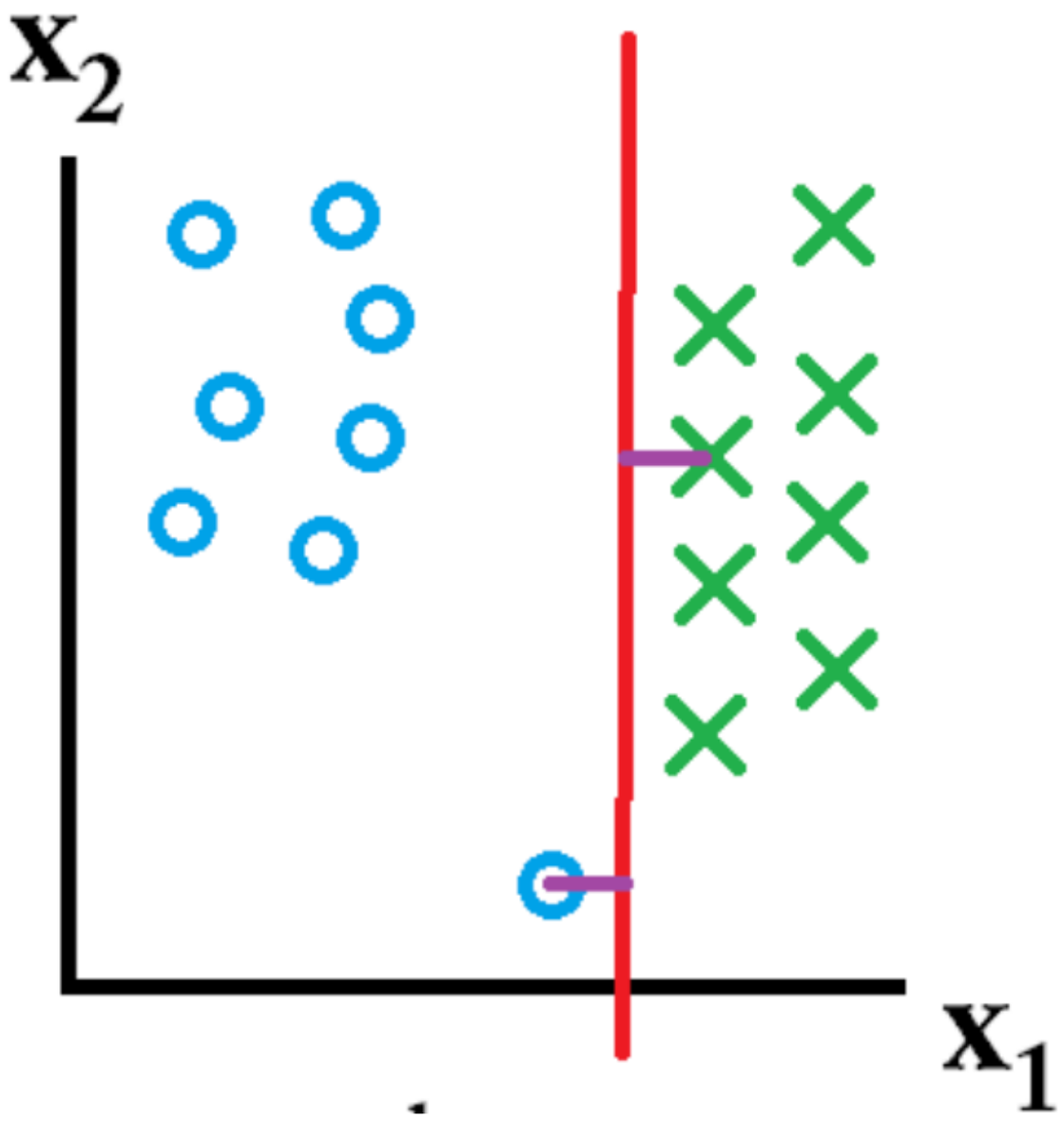
$$\min_w \lambda\|w\|^2 + \sum_i \max\{0, 1 - y_i w^T x_i\}$$

- Задача выпуклой оптимизации
- Например, можно использовать метод градиентного спуска
  - На самом деле решается задача квадратичного программирования, для которой есть более эффективные методы

Подробнее про SVM в лекции К.В. Воронцова: http://www.ccas.ru/voron/download/SVM.pdf

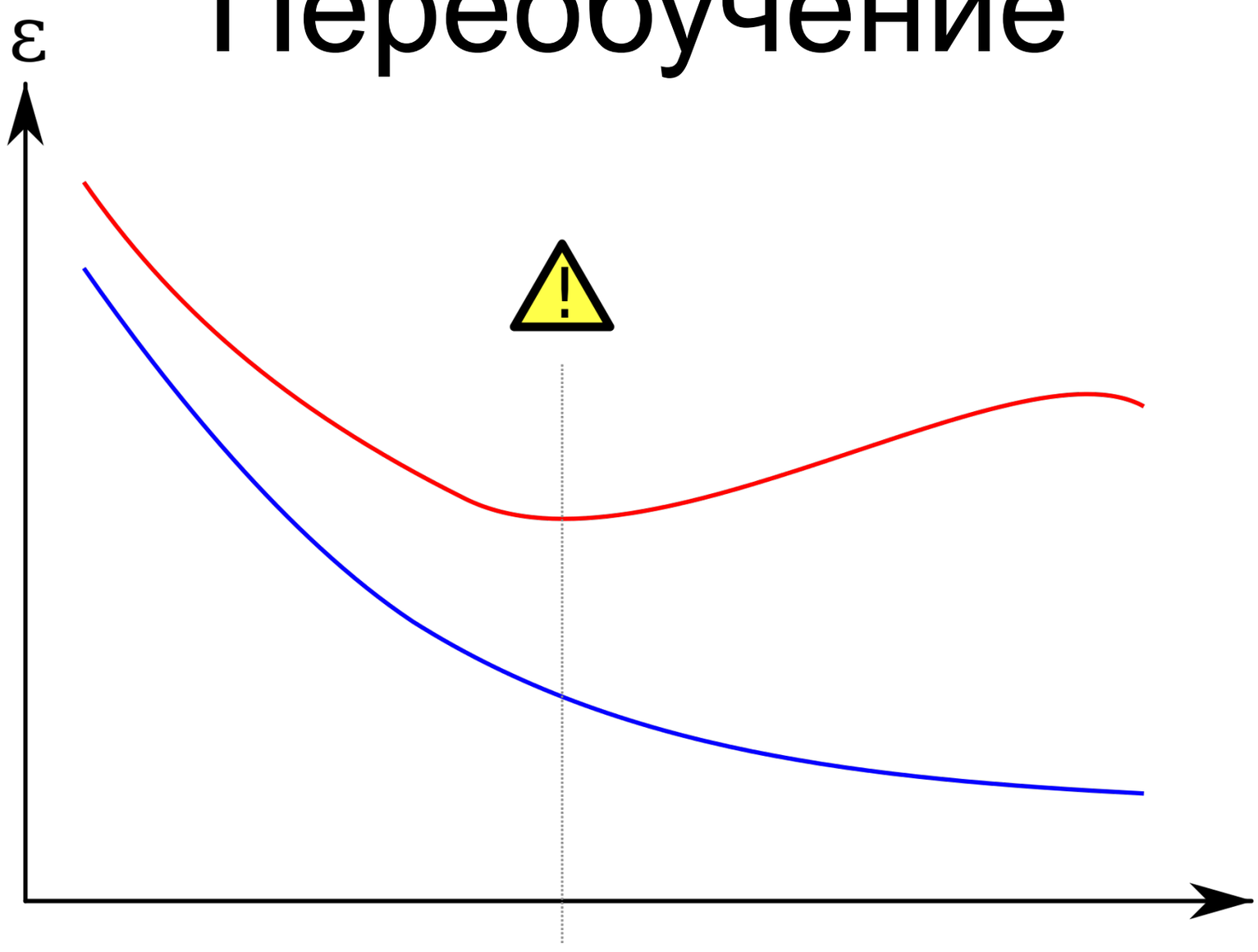
Переобучение





https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

Переобучение



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

https://en.wikipedia.org/wiki/Overfitting

Пример

```
from sklearn import svm
X = [[0, 0], [1, 1]]
y = [0, 1]
clf = svm.LinearSVC()
clf.fit(X, y)

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

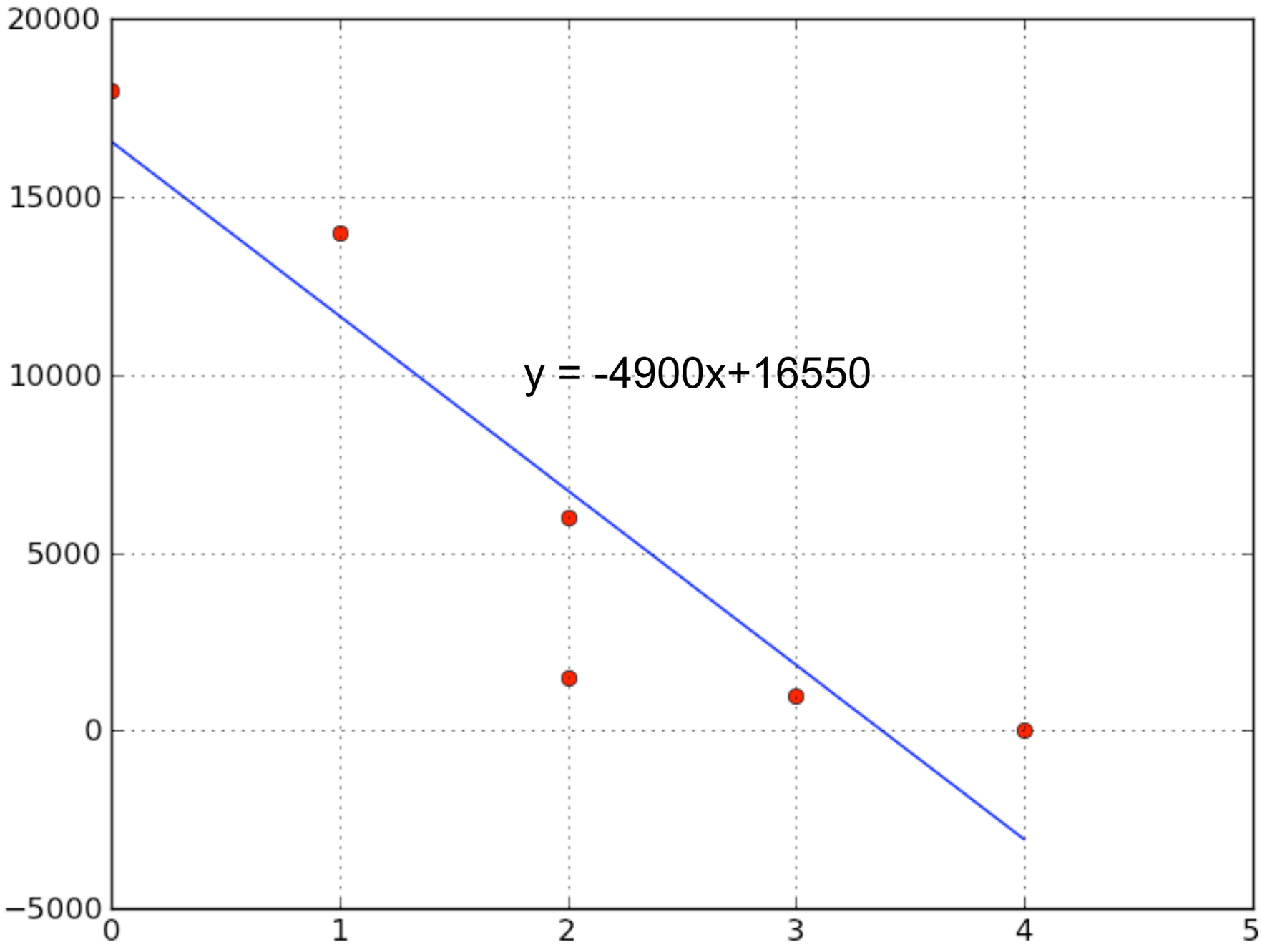
> array([1])

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 $f_0 = 1$

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

Вычисление коэффициентов

- Минимизировать квадратичную погрешность

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

- Вычисляется по формуле

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

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 вводом натурального логарифма

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

- Логит-преобразование

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

- Определим вероятность ...

Логистическая регрессия

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

- Или

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

- Логистическая функция

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

Логистическая регрессия

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

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

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

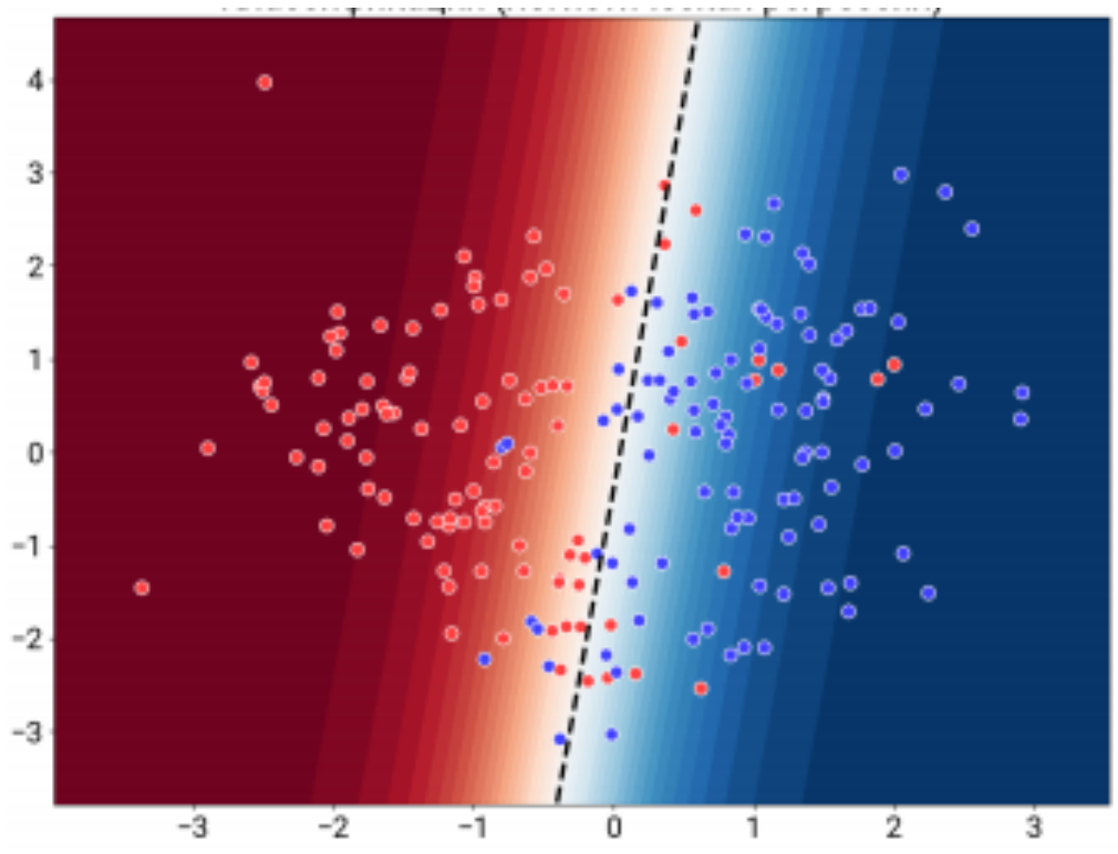
$$w \cdot f > 0$$

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

Обучение

- Оптимизация функции потерь (классы -1,1)

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



Мультиномиальная логистическая регрессия

- Классификация на множество классов

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

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

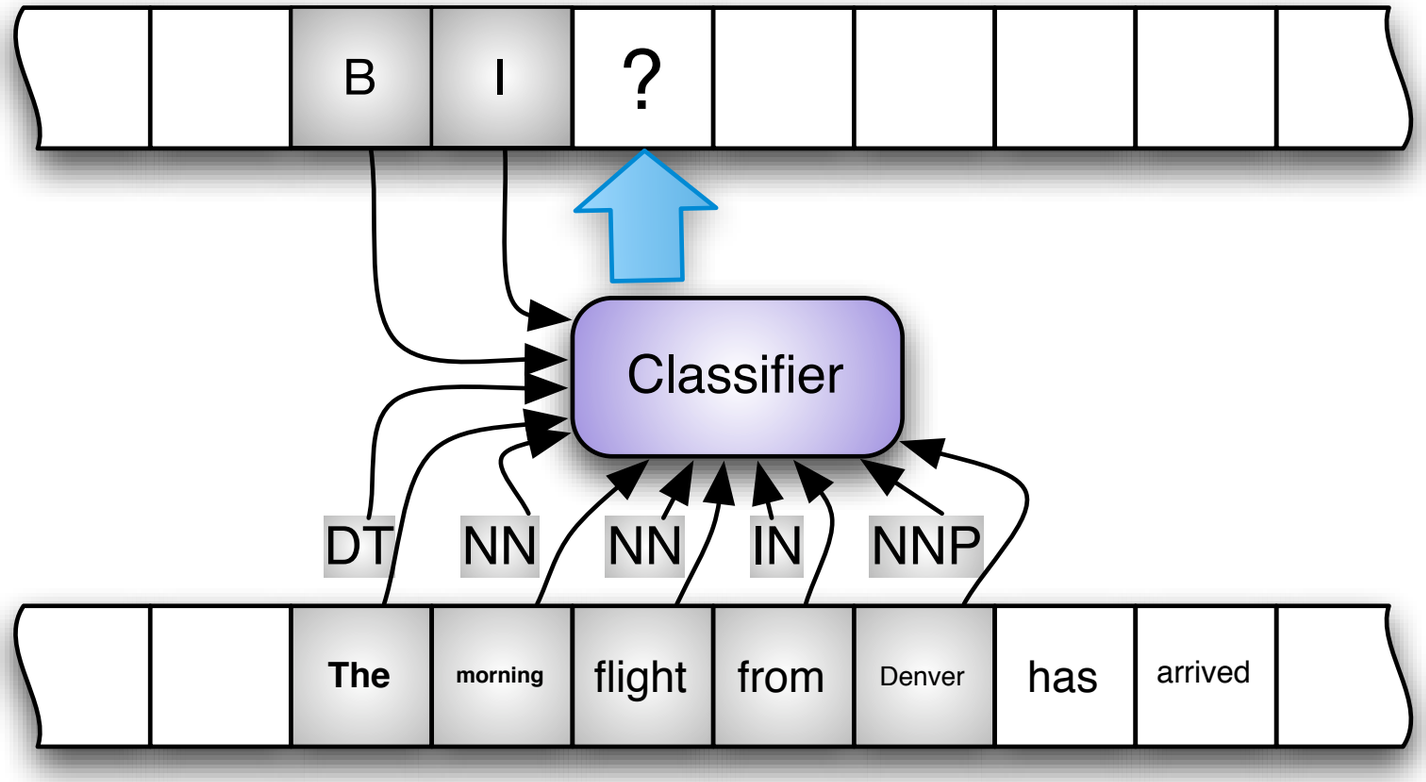
Пример

```
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 марковская модель (HMM)

- *Из окна сильно дуло*

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

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

- 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

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

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

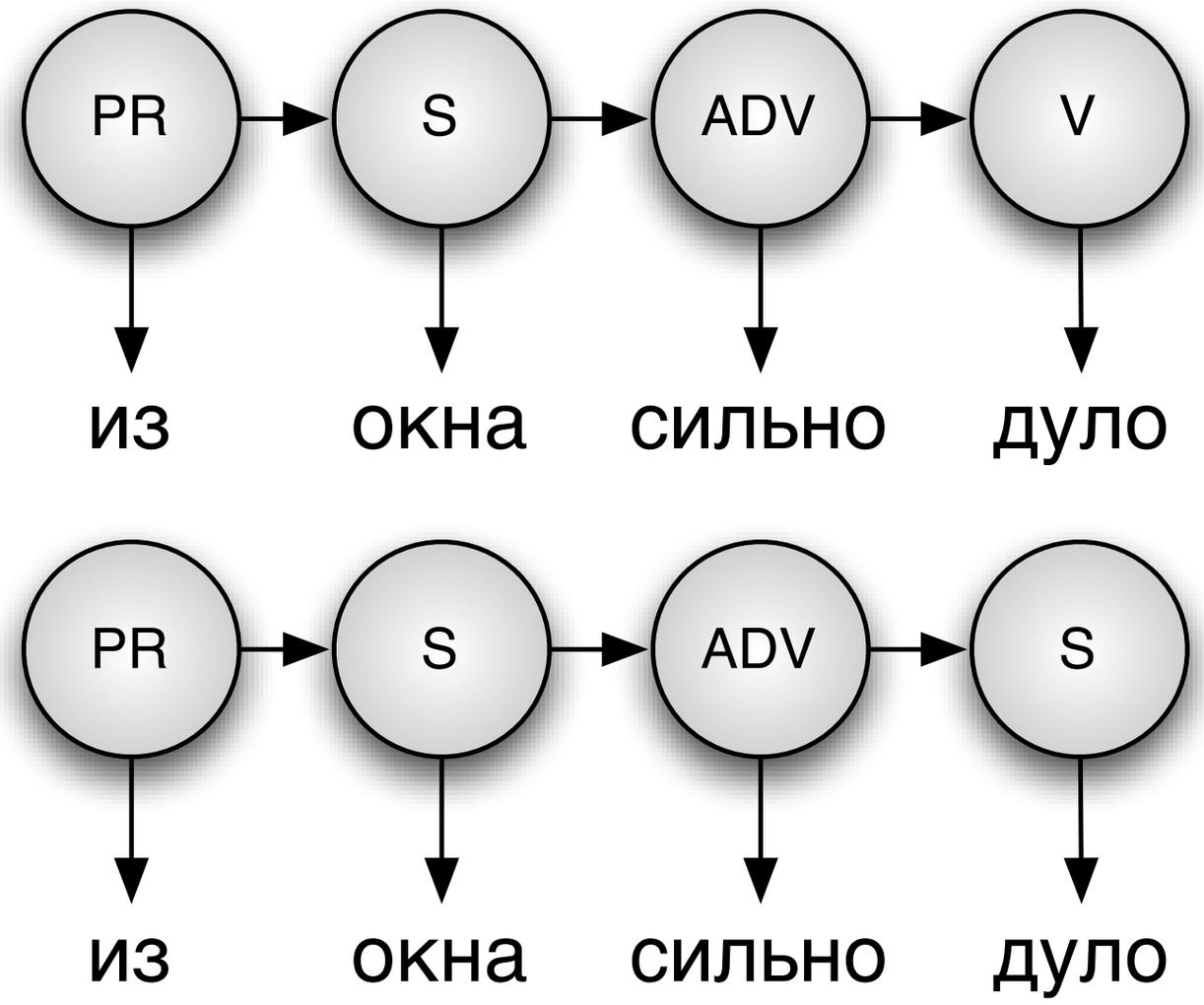
- Предположение 1

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

- Предположение 2

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

HMM для определения частей речи



- 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 $t \neq$ 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 $P(X_1, \ldots, X_n) = \max_{1 \leq j \leq T} \delta_{n+1}(t^j)$

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
bear	0	0	10	0	43	0
is	0	10065	0	0	0	0
move	0	0	0	36	133	0
on	0	0	5484	0	0	0
president	0	0	0	382	0	0
progress	0	0	0	108	4	0
the	69016	0	0	0	0	0
.	0	0	0	0	0	48809

+ добавим везде 1 (сглаживание Лапласа)

Пример

Считаем вероятности

Вероятности тегов — $P(tag_2 \mid tag_1)$:

$tag_1 \setminus tag_2$	AT	BEZ	IN	NN	VB	PERIOD
AT	0.000021	0.000021	0.000021	0.999507	0.000021	0.000411
BEZ	0.75057	0.00038	0.162357	0.071483	0.00038	0.014829
IN	0.697049	0.000016	0.021335	0.278591	0.000016	0.002993
NN	0.013178	0.045914	0.524062	0.145283	0.007589	0.263974
VB	0.433631	0.00307	0.339807	0.105462	0.009282	0.108747
PERIOD	0.533187	0.005055	0.309723	0.088454	0.063514	0.000067

Вероятности слов — P(word | tag):

word \ tag	AT	BEZ	IN	NN	VB	PERIOD
bear	0.000014	0.000099	0.001999	0.001873	0.234043	0.00002
is	0.000014	0.999305	0.000182	0.001873	0.005319	0.00002
move	0.000014	0.000099	0.000182	0.069288	0.712766	0.00002
on	0.000014	0.000099	0.99691	0.001873	0.005319	0.00002
president	0.000014	0.000099	0.000182	0.717228	0.005319	0.00002
progress	0.000014	0.000099	0.000182	0.20412	0.026596	0.00002
the	0.999899	0.000099	0.000182	0.001873	0.005319	0.00002
.	0.000014	0.000099	0.000182	0.001873	0.005319	0.999857

Пример

Чтобы не работать произведением вероятностей будем суммировать логарифмы вероятностей

Логарифмы вероятности тегов — ln $P(tag_2 | tag_1)$:

$tag_1 \setminus tag_2$	AT	BEZ	IN	NN	VB	PERIOD
AT	-10.7926	-10.7926	-10.7926	-0.0005	-10.7926	-7.7969
BEZ	-0.2869	-7.8747	-1.818	-2.6383	-7.8747	-4.2112
IN	-0.3609	-11.0373	-3.8474	-1.278	-11.0373	-5.8116
NN	-4.3292	-3.081	-0.6461	-1.9291	-4.8811	-1.3319
VB	-0.8356	-5.786	-1.0794	-2.2494	-4.6796	-2.2187
PERIOD	-0.6289	-5.2875	-1.1721	-2.4253	-2.7565	-9.6182

Логарифмы вероятности слов — ln P(word | tag):

word \ tag	AT	BEZ	IN	NN	VB	PERIOD
bear	-11.1422	-9.2176	-6.215	-6.2804	-1.4523	-10.7958
is	-11.1422	-0.0007	-8.6129	-6.2804	-5.2364	-10.7958
move	-11.1422	-9.2176	-8.6129	-2.6695	-0.3386	-10.7958
on	-11.1422	-9.2176	-0.0031	-6.2804	-5.2364	-10.7958
president	-11.1422	-9.2176	-8.6129	-0.3324	-5.2364	-10.7958
progress	-11.1422	-9.2176	-8.6129	-1.589	-3.627	-10.7958
the	-0.0001	-9.2176	-8.6129	-6.2804	-5.2364	-10.7958
.	-11.1422	-9.2176	-8.6129	-6.2804	-5.2364	-0.0001

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76				
BEZ	-1.79	-14.09	-20.37				
IN	-1.79	-11.05	-14.93				
NN	-1.79	-8.07	?				
VB	-1.79	-9.78					
(.)	-1.79	-13.92					

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76				
BEZ	-1.79	-14.09	-20.37				
IN	-1.79	-11.05	-14.93				
NN	-1.79	-8.07	?				
VB	-1.79	-9.78					
(.)	-1.79	-13.92					

-8.36
-23.01
-18.61
-16.28
-18.31
-22.36

$tag_1 \setminus tag_2$	NN
AT	-0.0005
BEZ	-2.6383
IN	-1.278
NN	-1.9291
VB	-2.2494
PERIOD	-2.4253

word \ tag	AT	BEZ	IN	NN
bear	-11.1422	-9.2176	-6.215	-6.2804

$\delta_{1,2} = \delta_{1,1} + \ln P(bear \mid NN) + \ln P(NN \mid AT) = -2.08 - 6.28 + 0 = -8.36$
$\delta_{2,2} = \delta_{2,1} + \ln P(bear \mid NN) + \ln P(NN \mid BEZ) = -14.09 - 6.28 - 2.64 = -23.01$
$\delta_{3,2} = \delta_{3,1} + \ln P(bear \mid NN) + \ln P(NN \mid IN) = -11.05 - 6.28 - 1.28 = -18.61$
$\delta_{4,2} = \delta_{4,1} + \ln P(bear \mid NN) + \ln P(NN \mid NN) = -8.07 - 6.28 - 1.93 = -16.28$
$\delta_{5,2} = \delta_{5,1} + \ln P(bear \mid NN) + \ln P(NN \mid VB) = -9.78 - 6.28 - 2.25 = -18.31$
$\delta_{6,2} = \delta_{6,1} + \ln P(bear \mid NN) + \ln P(NN \mid PERIOD) = -13.92 - 6.28 - 2.43 = -22.63$
$v_{4,2} = \max(-8.36, -23.01, -18.61, -16.28, -18.31, -22.63) = -8.36$

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76				
BEZ	-1.79	-14.09	-20.37				
IN	-1.79	-11.05	-14.93				
NN	-1.79	-8.07	-8.36				
VB	-1.79	-9.78	-18.31				
(.)	-1.79	-13.92	-22.36				

-8.36
-23.01
-18.61
-16.28
-18.31
-22.36

$tag_1 \setminus tag_2$	NN
AT	-0.0005
BEZ	-2.6383
IN	-1.278
NN	-1.9291
VB	-2.2494
PERIOD	-2.4253

word \ tag	AT	BEZ	IN	NN
bear	-11.1422	-9.2176	-6.215	-6.2804

$\delta_{1,2} = \delta_{1,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{AT}) = -2.08 - 6.28 + 0 = -8.36$
$\delta_{2,2} = \delta_{2,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{BEZ}) = -14.09 - 6.28 - 2.64 = -23.01$
$\delta_{3,2} = \delta_{3,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{IN}) = -11.05 - 6.28 - 1.28 = -18.61$
$\delta_{4,2} = \delta_{4,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{NN}) = -8.07 - 6.28 - 1.93 = -16.28$
$\delta_{5,2} = \delta_{5,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{VB}) = -9.78 - 6.28 - 2.25 = -18.31$
$\delta_{6,2} = \delta_{6,1} + \ln P(\text{bear} \mid \text{NN}) + \ln P(\text{NN} \mid \text{PERIOD}) = -13.92 - 6.28 - 2.43 = -22.63$
$v_{4,2} = \max(-8.36, -23.01, -18.61, -16.28, -18.31, -22.63) = -8.36$

		The	bear	is	on	the	move
AT	-1.79	-2.08	-21.76				
BEZ	-1.79	-14.09	-20.37				
IN	-1.79	-11.05	-14.93				
NN	-1.79	-8.07	-8.36				
VB	-1.79	-9.78					
(.)	-1.79	13.91					

		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
```

https://programforyou.ru/tests/viterbi-algorithm

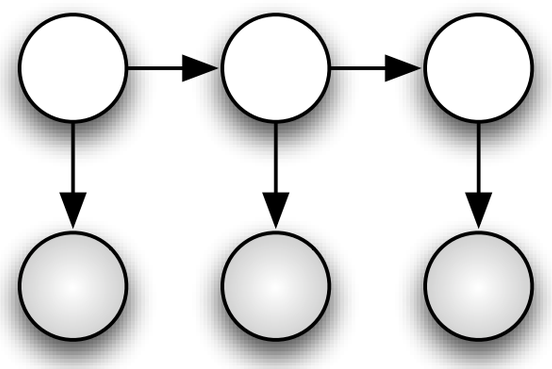
Марковская модель максимальной энтропии

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

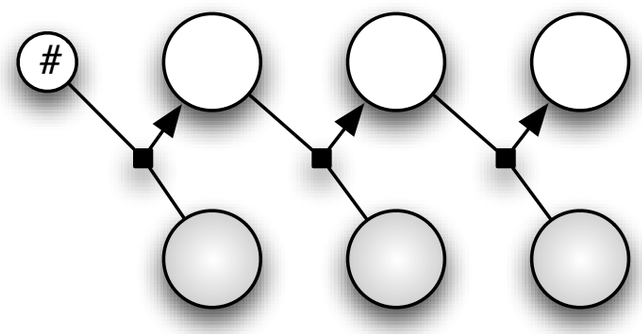
$$\hat{T} = \arg\max_t P(T|W) = \arg\max_T \prod_i P(tag_i|word_i, tag_{i-1})$$

- Сравнить с марковской моделью

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

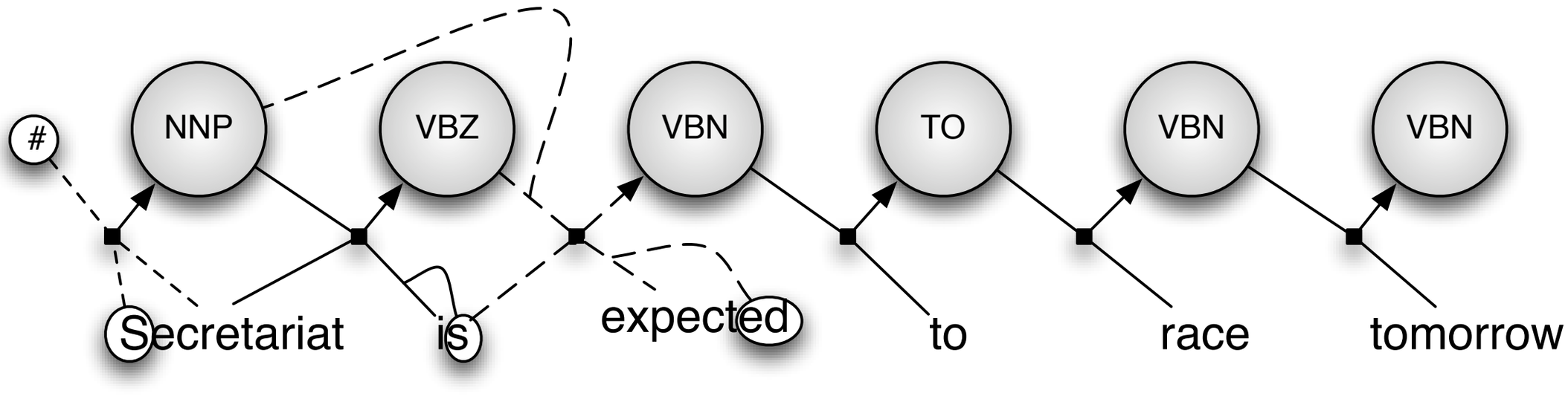


Скрытые марковские модели



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

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

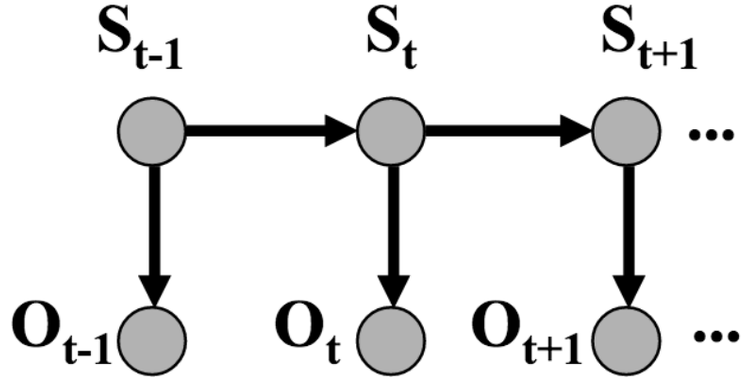
- Обучение аналогично логистической регрессии

Условные случайные поля (CRF)

$\vec{s} = s_1, s_2, \ldots s_n \qquad \vec{o} = o_1, o_2, \ldots o_n$

HMM

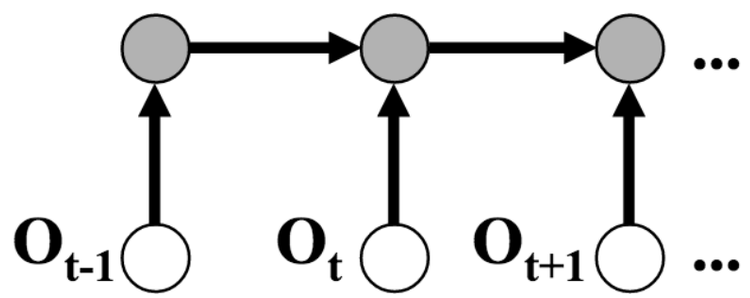
$$P(\vec{s}, \vec{o}) \propto \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$



MEMM

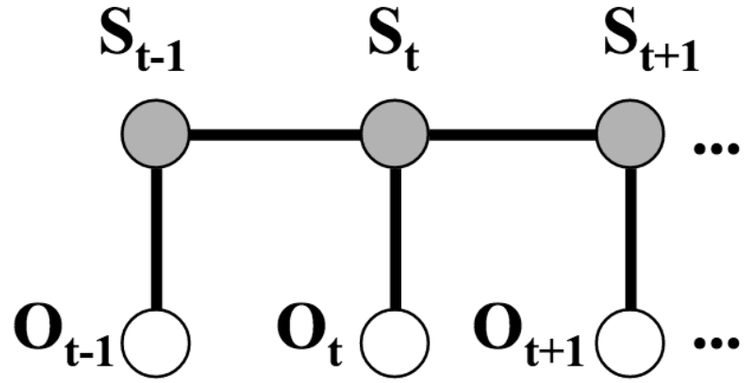
$$P(\vec{s} \mid \vec{o}) \propto \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}, o_t)$$

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



CRF

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



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

Следующая лекция

- Искусственные нейронные сети для обработки текстов