

# Стохастическое программирование: Data Mining. Введение

Бояров Андрей Александрович  
Сенов Александр Алексеевич

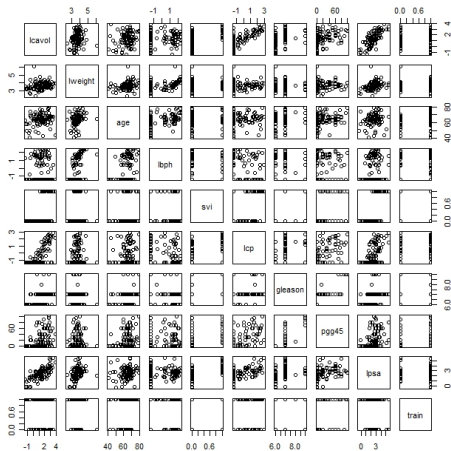
Санкт-Петербургский государственный университет,  
математико-механический факультет

19 февраля 2014

- 1 Постановка проблемы обучения, обучение с учителем, эффект переобучения, обучение без учителя
- 2 Введение в кластеризацию, алгоритм  $k$ -средних, EM алгоритм
- 3 Спектральная кластеризация, алгоритм Ng-Jordan-Weiss (NJW)
- 4 Устойчивость кластеризации, критерии устойчивости, вероятностные метрики
- 5 Ядерные расстояния
- 6 Минимальное остовное дерево, метод ближайших соседей
- 7 Метрики Хоттелинга, устойчивость кластеризации с точки зрения задачи оптимизации
- 8 Заключение, применение алгоритмов кластеризации в различных проектах

- clusterization@mail.ru
- Данные в текстовом формате, описание задания
- Любой язык программирования (Python, R, Matlab (Octave))
- Ответ содержит результат (график, разбиение на кластеры), краткое описание, код
  
- Линейная регрессия, с 19.02 по 12.03
- Тест  $\chi^2$ , с 12.03 по 9.04
- Кластеризация k-means, с 9.04 по 30.04
- Устойчивость кластеризации, с 30.04 по 21.05

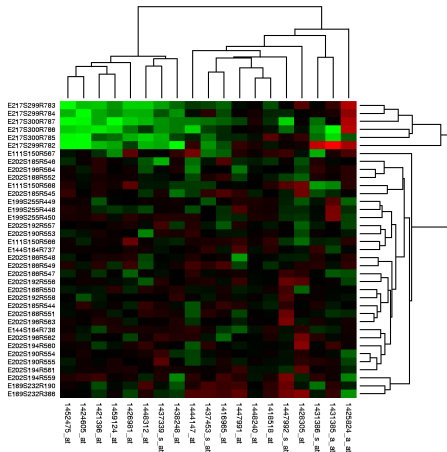
## Данные о раке простаты (Stamey et al. (1989))



## Распознавание рукописных цифр



## Теплокарта уровней экспрессии генов



## Рекомендательные системы

The screenshot shows the Netflix interface for a user's 'Local Favorites' in Boulder, Colorado. The page title is 'Local Favorites' and it lists 10 movies. Each movie has an 'Add' button (or 'Play Add' for some) and a star rating. The text above the list states: 'Members in and around Boulder, Colorado are currently renting these titles much more than other Netflix members.'

Rank	Action	Title	Rating
1.	Add	It Might Get Loud	★★★★★
2.	Add	The Aviator	★★★★★
3.	Add	Vicky Cristina Barcelona	★★★★★
4.	Add	When Harry Met Sally	★★★★★
5.	Add	I Am	★★★★★
6.	Add	West Side Story	★★★★★
7.	Play Add	Toast	★★★★★
8.	Play Add	The First Grader	★★★★★
9.	Play Add	GasLand	★★★★★
10.	Add	The Da Vinci Code	★★★☆☆

On the right side, there is a search box for 'See What's Popular Somewhere Else' with dropdown menus for 'Colorado' and 'Boulder', a text input for 'OR just enter a ZIP Code', and a 'GO' button. Below that is a promotional banner for 'Give FREE rentals!' with a 'Tell a friend' button.

## Машинное зрение, робототехника (Google car)





# Примеры

Обработка естественного языка:

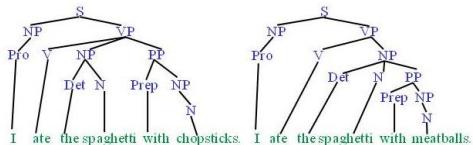


Рис. : Парсинг предложений

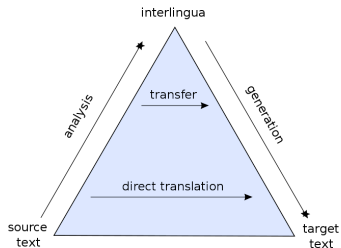
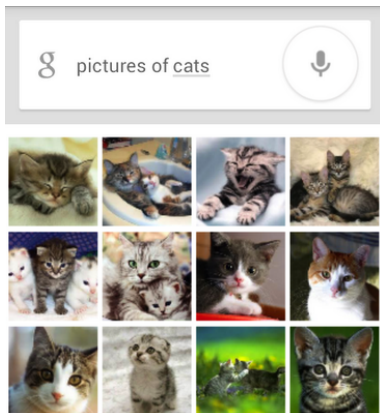


Рис. : Машинный перевод

Распознавание человеческой речи: Google (Android), Siri



Методы машинного обучения применяются, когда

- Существует некоторая закономерность
- Невозможно определить её математически
- Для её аппроксимации используются данные

# Общий алгоритм обучения



# Виды обучения

$\mathbf{w} = (w_1, \dots, w_d)$  — входной сигнал,  $\mathbf{x}$  — параметры,  $y$  — выходное значение

- С учителем:  $(\mathbf{w}, y)$  (регрессия, классификация)
- Без учителя:  $\mathbf{w}$  (кластеризация)
- Обучение с подкреплением, коллаборативная фильтрация

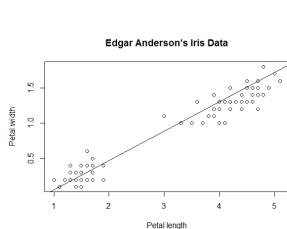


Рис. : Регрессия

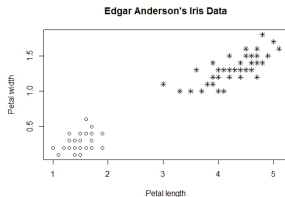


Рис. : Классификация

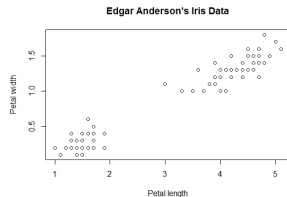
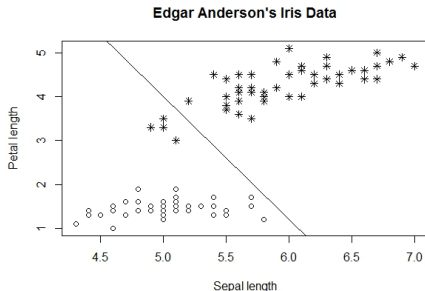


Рис. : Кластеризация

- На вход подается сигнал  $\mathbf{w}$
- Выходное значение:  $y$
- Искомая функция:  $f_t : \mathbb{W} \rightarrow \mathbb{Y}$
- Тренировочная выборка:  $(\mathbf{w}_1, y_1), (\mathbf{w}_2, y_2), \dots, (\mathbf{w}_N, y_N)$
- Результат работы алгоритма: гипотеза  $g : \mathbb{W} \rightarrow \mathbb{Y}$

# Перцептрон

- 1  $s(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^d x_i w_i = \mathbf{x}^T \mathbf{w}$ ,  $w_0 = 1$  — порог
- 2  $\mathbb{X}_1(\mathbf{x}) = \{\mathbf{w} : s(\mathbf{x}, \mathbf{w}) > 0\}$ ,  $\mathbb{X}_2(\mathbf{x}) = \{\mathbf{w} : s(\mathbf{x}, \mathbf{w}) < 0\}$
- 3  $h(\mathbf{w}) = \text{sign}(\mathbf{x}^T \mathbf{w})$
- 4 На каждой итерации алгоритма выбираем ошибочно классифицированную точку данных  $(\mathbf{w}_k, y_k)$ :  $\text{sign}(\mathbf{x}^T \mathbf{w}_k) \neq y_k$ .  
Для неё пересчитываем параметры:  $\mathbf{x} = \mathbf{x} + y_k \mathbf{w}_k$



- Данные:  $(\mathbf{w}_1, y_1), (\mathbf{w}_2, y_2), \dots, (\mathbf{w}_N, y_N)$ ,  $y_k \in \mathbb{R}$ ,  $k \in 1, \dots, N$
- $h(\mathbf{w}) = \mathbf{x}^T \mathbf{w}$ ,  $F(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N (h(\mathbf{w}_i) - y_i)^2 \rightarrow \min_{\mathbf{w}}$
- Градиентный спуск («пакетный», «batch»):  $t = 0, 1, 2, \dots$   
 $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t - \alpha \nabla F$
- Матрица входных данных и вектор выходных данных:

$$\mathbf{W} = \begin{bmatrix} -\mathbf{w}_1^T & - \\ -\mathbf{w}_2^T & - \\ \dots & \\ -\mathbf{w}_N^T & - \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

- $F(\mathbf{x}) = \frac{1}{N} \|\mathbf{W}\mathbf{x} - \mathbf{y}\|^2$ ,  $\mathbf{W}^\dagger = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T$ ,  $\mathbf{x} = \mathbf{W}^\dagger \mathbf{y}$
- Градиентный спуск: итерации,  $\alpha$ , хорошо работает при большом N
- Псевдообратная матрица: много вычислений при большом N



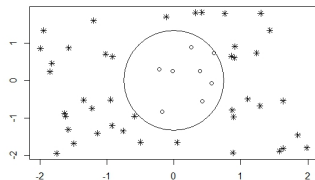
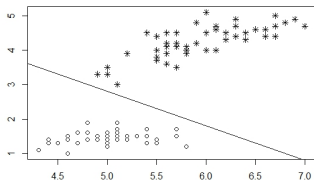
- $s(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^d x_i w_i$ ,  $h(\mathbf{w}) = \theta(s) = \frac{e^s}{1+e^s}$  — сигмоида
- $h(\mathbf{w}) = \theta(s)$  интерпретируется, как вероятность для данных  $(\mathbf{w}, y)$

$$P(y|\mathbf{w}) = \begin{cases} h(\mathbf{w}), & \text{для } y = +1; \\ 1 - h(\mathbf{w}), & \text{для } y = -1. \end{cases}$$

- $\theta(-s) = 1 - \theta(s)$ , поэтому  $P(y|\mathbf{w}) = \theta(y\mathbf{x}^T \mathbf{w})$
- Построим оценку максимального правдоподобия для данных  $(\mathbf{w}_1, y_1), (\mathbf{w}_2, y_2), \dots, (\mathbf{w}_N, y_N)$ :  $\prod_{i=1}^N P(y_i|\mathbf{w}_i) = \prod_{i=1}^N \theta(y_i \mathbf{x}^T \mathbf{w}_i)$
- $F(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \log(1 + e^{-y_i \mathbf{x}^T \mathbf{w}_i})$ ,  $F(\mathbf{x}) \rightarrow \min_{\mathbf{x}}$

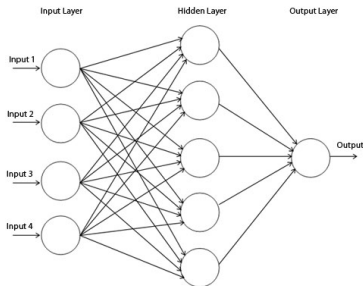
# Преобразование координат

- $s(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^d x_i w_i$  — граница принятия решения



- $\Phi : \mathbf{w} = (w_1, \dots, w_d) \rightarrow \mathbf{z} = (z_1, \dots, z_p)$ ,  $z_i = \phi_i(\mathbf{w})$  для всех  $i = 1, \dots, p$
- Например:  $\mathbf{w} = (w_1, w_2)$ , тогда  $\mathbf{z} = (1, w_1, w_2, w_1 w_2, w_1^2, w_2^2)$
- Находим решение в новом пространстве:  $\tilde{\mathbf{x}}$ , тогда  $g(\mathbf{w}) = \text{sign}(\tilde{\mathbf{x}}^T \Phi(\mathbf{w}))$

- Аналогии с биологической сетью
- Комбинирование перцептронов для решения задач, которые не могут быть решены обычным перцептроном
- Искусственная неронная сеть:



# Нейронные сети (модель)

- $\mathbf{w}$  — на вход,  $1 \leq l < L$  — скрытые уровни,  $l = L$  — выходной уровень
- Параметры:

$$x_{ij}^{(l)} = \begin{cases} 1 \leq l \leq L & \text{уровни} \\ 0 \leq i \leq d^{(l-1)} & \text{входные данные} \\ 1 \leq j \leq d^{(l)}, & \text{выходные данные} \end{cases}$$

- $\theta(s) = \text{th}(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$
- $w_j^{(l)} = \theta(s_j^{(l)}) = \theta(\sum_{i=0}^{d^{(l-1)}} x_{ij}^{(l)} w_i^{(l-1)})$
- Входные данные:  $\mathbf{w} = (w_1^{(0)}, \dots, w_{d^{(0)}}^{(0)}) \rightarrow w_1^{(L)} = h(\mathbf{w})$

# Нейронные сети (обучение)

- Стохастический градиентный спуск: на каждом шаге выбирается один экземпляр выборки  $(\mathbf{w}_k, y_k)$
- Ошибка на этом экземпляре:  $f(\mathbf{x})$ . Необходимо вычислить  $\nabla f(\mathbf{x})$ :  $\frac{\partial f}{\partial x_{ij}^{(l)}}$  для всех  $i, j, l$
- $\frac{\partial f}{\partial x_{ij}^{(l)}} = \frac{\partial f}{\partial s_j^{(l)}} \frac{\partial s_j^{(l)}}{\partial x_{ij}^{(l)}} = w_i^{(l-1)} \delta_j^{(l)}$ , где  $\delta_j^{(l)} = \frac{\partial f}{\partial s_j^{(l)}}$
- $\delta_i^{(l-1)} = (1 - (w_i^{(l-1)})^2) \sum_{j=1}^{d^{(l)}} x_{ij}^{(l)} \delta_j^{(l)}$

Алгоритм обратного распространения ошибки:

- 1 Инициализируем  $\hat{x}_{ij}^{(l)}$  случайным образом
- 2 На каждой итерации:
  - Выбираем экземпляр  $(\mathbf{w}_k, y_k)$
  - Вычисляем все  $w_j^{(l)}$ ; Вычисляем все  $\delta_j^{(l)}$
  - $\hat{x}_{ij}^{(l)} = \hat{x}_{ij}^{(l)} - \alpha w_i^{(l-1)} \delta_j^{(l)}$
- 3 Возвращаем веса  $\hat{x}_{ij}^{(l)}$