

# УПРАВЛЕНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

## искусственный интеллект, машинное обучение

А.В. Макаренко

`avm@rdcn.ru`

Научно-исследовательская группа «Конструктивная Кибернетика»  
Москва, Россия, [www.rdcn.ru](http://www.rdcn.ru)

Институт проблем управления РАН  
Москва, Россия

Учебный курс – Лекция 3

15 марта 2018 г.

ИПУ РАН, Москва, Россия

- ① Искусственный интеллект
  - Общие положения
  
- ② Машинное обучение
  - Общие положения
  - О данных
  - Основные классы задач
  - Основные методы обучения
  - Прикладные аспекты
  
- ③ Заключение

# Outline section

- ① Искусственный интеллект
  - Общие положения
  
- ② Машинное обучение
  - Общие положения
  - О данных
  - Основные классы задач
  - Основные методы обучения
  - Прикладные аспекты
  
- ③ Заключение

## Определение понятия

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ** – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

## Определение понятия

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ** – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

**ИСКУССТВЕННАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА** – это аппаратно-программный комплекс, способный решать творческие задачи, традиционно считающиеся прерогативой человека.

## Определение понятия

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ** – это самостоятельное направление информатики, специализирующееся на разработке и исследовании искусственных интеллектуальных систем.

**ИСКУССТВЕННАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА** – это аппаратно-программный комплекс, способный решать творческие задачи, традиционно считающиеся прерогативой человека.

**ТВОРЧЕСТВО** – процесс деятельности, создающий качественно новые материальные и духовные ценности или итог создания объективно нового. Основным критерий, отличающий творчество от изготовления (производства) – уникальность его результата. Результат творчества невозможно прямо вывести из начальных условий. Именно этот факт придаёт продуктам творчества дополнительную ценность в сравнении с продуктами производства.

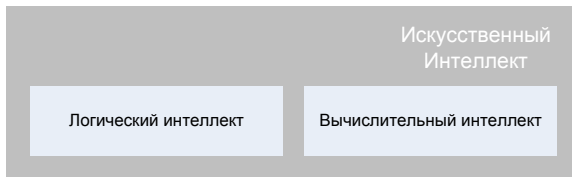
## Классификация по возможностям

ИИ	узкий	широкий	
слабый	Распознавание образов, Сложные логические игры	Управление сложными системами	Адаптивное поведение
сильный	Формулирование и доказательство новых матем. теорем	42?	Разум
	конкретная предметная область	пересечение предметных областей	

**АДАПТИВНОЕ ПОВЕДЕНИЕ** – (в кибернетике) – способность системы к целенаправленному приспособляющемуся поведению в сложных средах при изменении как внутренних, так и внешних условий. **Особенность:** не требуется понимание смысла оперируемой информации.

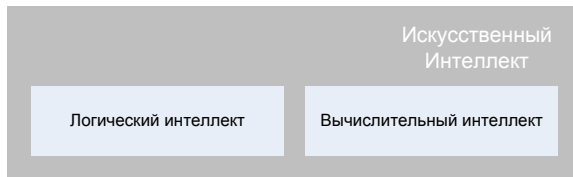
**РАЗУМ** – высший тип мыслительной (познавательной) деятельности, способность мыслить всеобще, способность анализа, абстрагирования и обобщения. **Особенность:** прохождение теста Тьюринга.

# Классификация по методам I



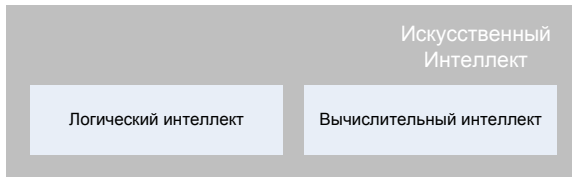


# Классификация по методам I



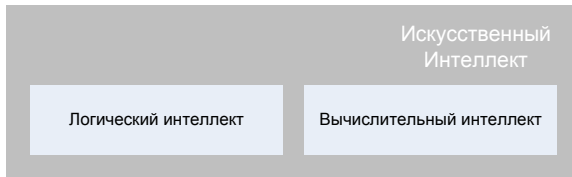
**ЛОГИЧЕСКИЙ ИНТЕЛЛЕКТ** – основан на строгом логическом выводе, в качестве основного математического инструментария применяются описательные логики.

# Классификация по методам I



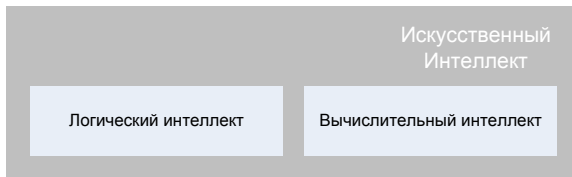
**ОПИСАТЕЛЬНЫЕ ЛОГИКИ** – семейство языков представления знаний, позволяющих описывать понятия предметной области в недвусмысленном, формализованном виде. Они сочетают в себе, с одной стороны, богатые выразительные возможности, а с другой – хорошие вычислительные свойства, такие как разрешимость и относительно невысокая вычислительная сложность основных логических проблем, что делает возможным их применение на практике. Являются основой для построения онтологий.

# Классификация по методам I

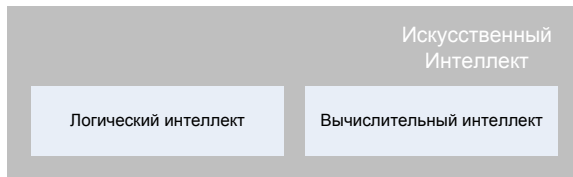


**ОНТОЛОГИЯ** – (в информатике) – это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры данных, содержащей все релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой предметной области.

# Классификация по методам II

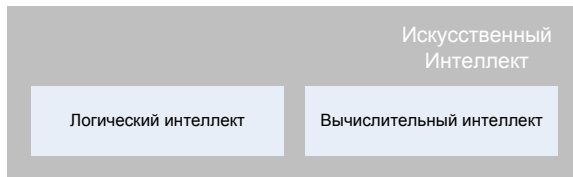


## Классификация по методам II



**ВЫЧИСЛИТЕЛЬНЫЙ ИНТЕЛЛЕКТ** – опирается на эвристические алгоритмы, в качестве основного математического инструментария применяется машинное обучение.

# Классификация по методам II



**ВЫЧИСЛИТЕЛЬНЫЙ ИНТЕЛЛЕКТ** – опирается на эвристические алгоритмы, в качестве основного математического инструментария применяется машинное обучение.

Тесно связан с концепцией мягких вычислений и кибернетикой.

Дополнительно: [IEEE Computational Intelligence Society](#).

# Проблематика



Джон Маккарти  
(04.09.1927 – 24.10.2011)

«Проблема состоит в том, что пока мы не можем в целом определить, какие вычислительные процедуры мы хотим называть интеллектуальными. Мы понимаем некоторые механизмы интеллекта и не понимаем остальные. Поэтому под интеллектом в пределах этой науки понимается только вычислительная составляющая способности достигать целей в мире.»

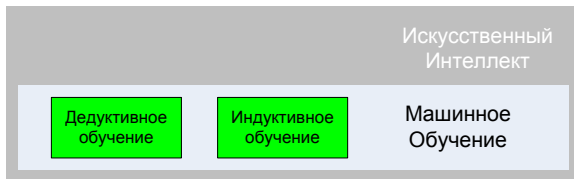
John McCarthy, WHAT IS ARTIFICIAL INTELLIGENCE? [Читать...](#)

# Outline section

- ① Искусственный интеллект
  - Общие положения
- ② Машинное обучение
  - Общие положения
  - О данных
  - Основные классы задач
  - Основные методы обучения
  - Прикладные аспекты
- ③ Заключение

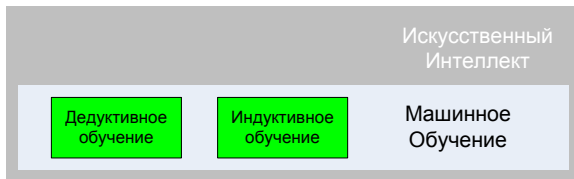


# Определение понятия



**МАШИННОЕ ОБУЧЕНИЕ** – обширный (центральный) подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

# Определение понятия



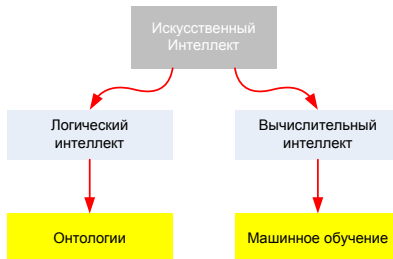
**МАШИННОЕ ОБУЧЕНИЕ** – обширный (центральный) подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Различают два типа обучения машин:

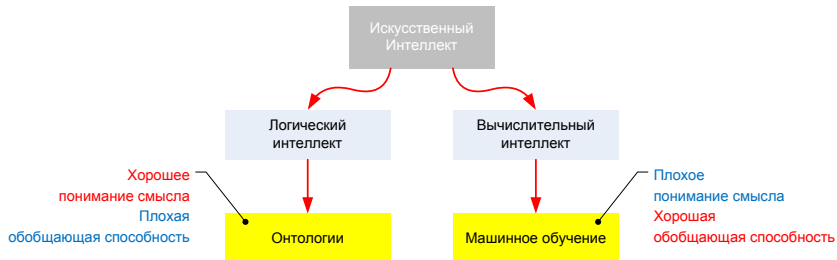
- Дедуктивное обучение – предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний (область экспертных систем).
- Индуктивное обучение – (обучение по прецедентам) – основано на выявлении общих закономерностей по частным эмпирическим (экспериментальным) данным.

Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Наука, 1974.

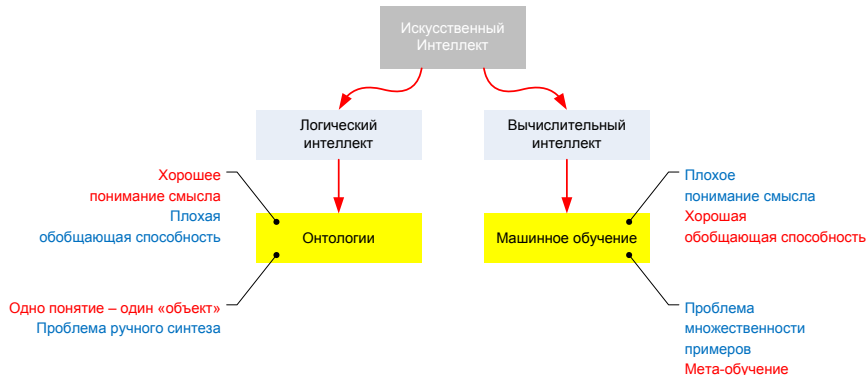
# Проблематика ИИ 2-го порядка



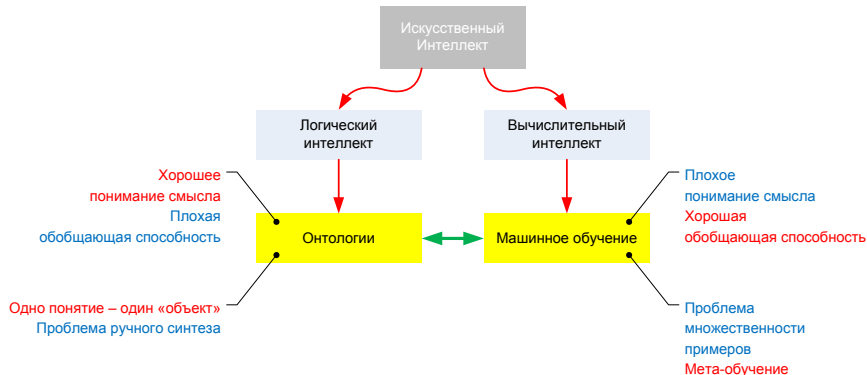
# Проблематика ИИ 2-го порядка



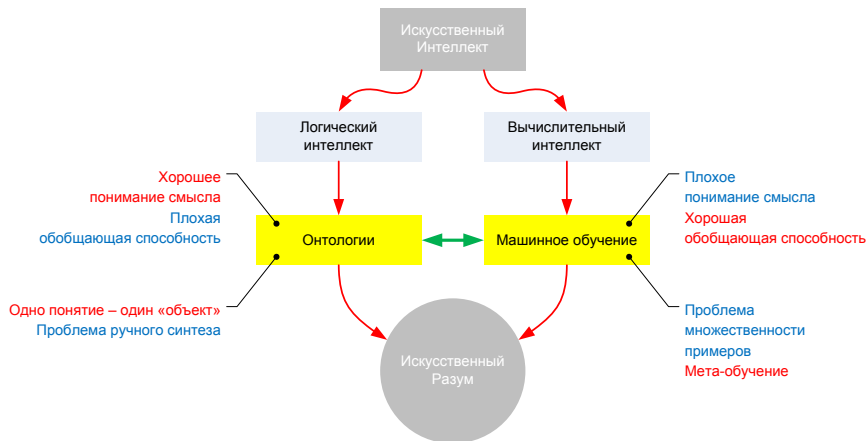
# Проблематика ИИ 2-го порядка



# Проблематика ИИ 2-го порядка

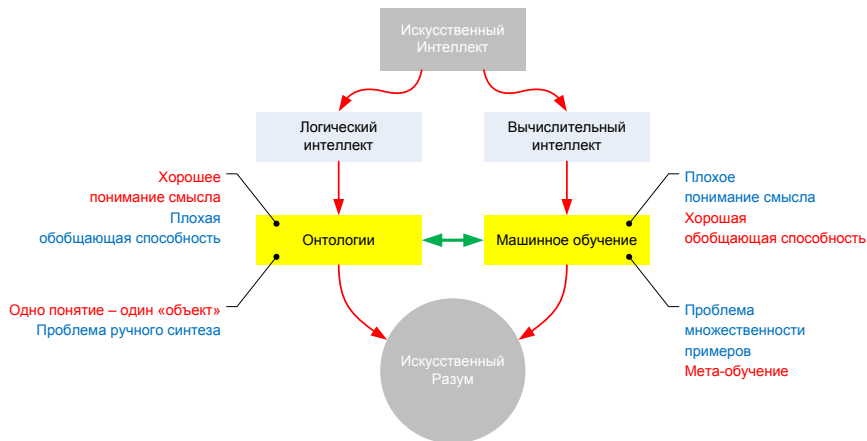


# Проблематика ИИ 2-го порядка



По теме: Besold T.R. et al. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. ArXiv: [1711.03902](https://arxiv.org/abs/1711.03902).

# Проблематика ИИ 2-го порядка



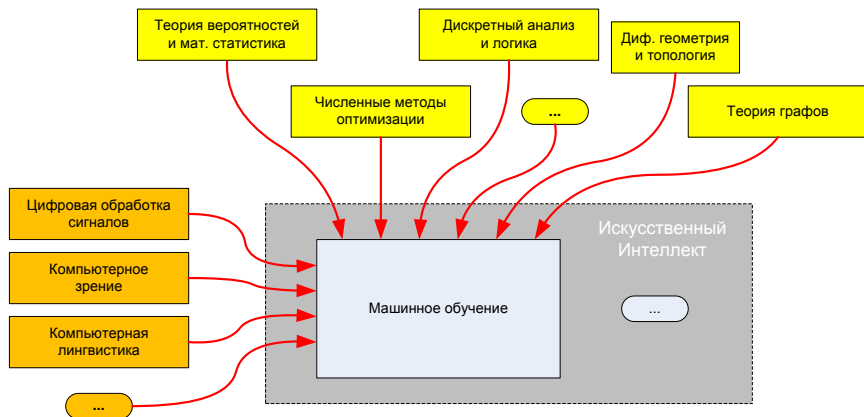
По теме: Besold T.R. et al. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. ArXiv: [1711.03902](https://arxiv.org/abs/1711.03902).



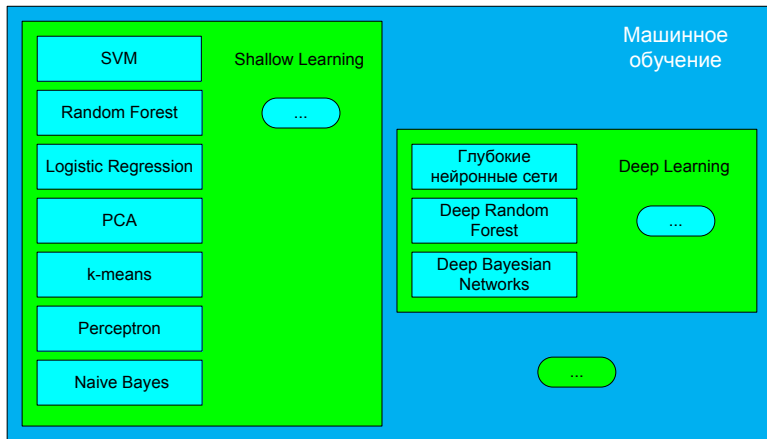
## Замечание

Далее, если специально не будет оговорено иное, под искусственным интеллектом мы будем понимать вычислительный интеллект в его слабой форме, и машинное обучение в форме обучения по прецедентам.

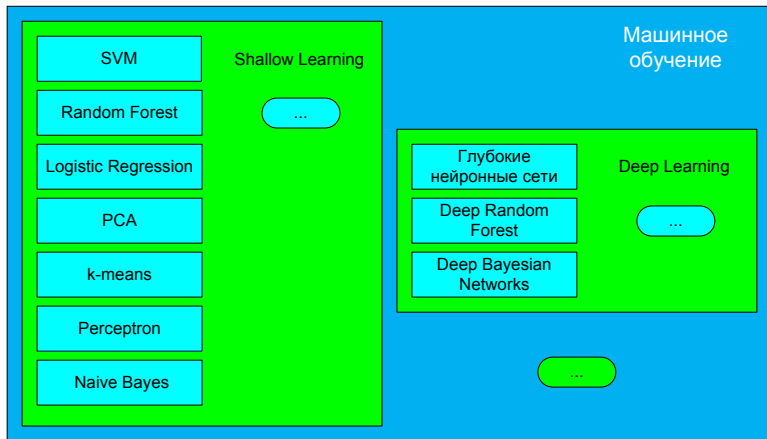
# Внешние составляющие



# Внутренняя структура



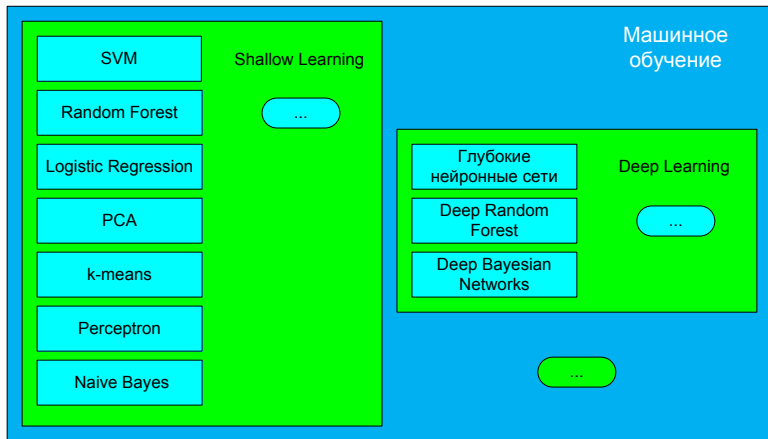
# Внутренняя структура



Основная особенность **Shallow Learning** алгоритмов:

- Для их обучения требуются (как правило) вручную синтезируемые высокоуровневые признаки.

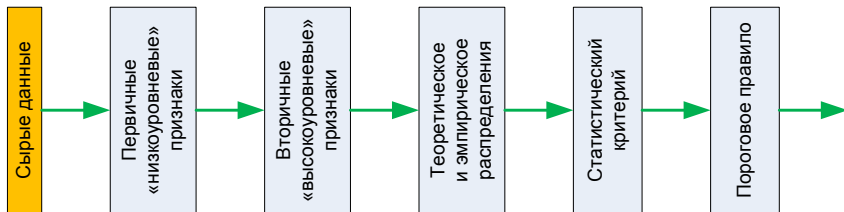
# Внутренняя структура



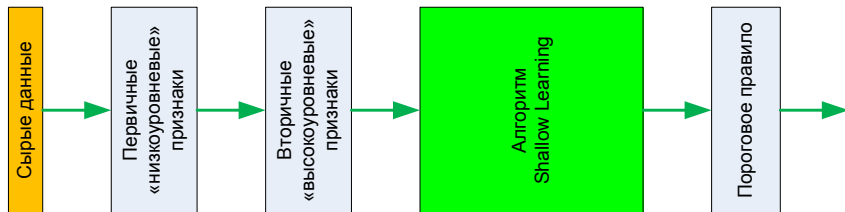
Основная особенность **Deep Learning** алгоритмов:

- Они работают с исходными данными (низкоуровневыми признаками) и самостоятельно извлекают (формируют) признаковое описание объектов.

## Объёмы «ручного труда»



## Объёмы «ручного труда»

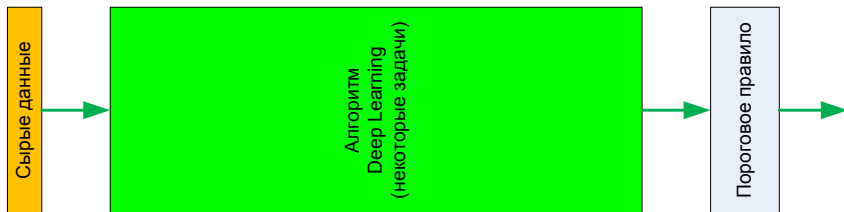


## Объёмы «ручного труда»





## Объёмы «ручного труда»



# Шкалы измерений I

**ШКАЛА ИЗМЕРЕНИЙ** – (согласно РМГ 83-2007)<sup>1</sup> – отображение множества различных проявлений количественного или качественного свойства [наблюдаемого объекта] на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений).

---

<sup>1</sup>Государственная система обеспечения единства измерений. Шкалы измерений. Термины и определения.

# Шкалы измерений I

**ШКАЛА ИЗМЕРЕНИЙ** – (согласно РМГ 83-2007)<sup>1</sup> – отображение множества различных проявлений количественного или качественного свойства [наблюдаемого объекта] на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений).

**ИЗМЕРЕНИЕ** – сравнение конкретного проявления измеряемого свойства (величины) [наблюдаемого объекта] со шкалой измерений этого свойства (величины) в целях получения результата измерений (оценки свойства или значения величины).

---

<sup>1</sup>Государственная система обеспечения единства измерений. Шкалы измерений. Термины и определения.

# Шкалы измерений I

**ШКАЛА ИЗМЕРЕНИЙ** – (согласно РМГ 83-2007)<sup>1</sup> – отображение множества различных проявлений количественного или качественного свойства [наблюдаемого объекта] на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений).

**ИЗМЕРЕНИЕ** – сравнение конкретного проявления измеряемого свойства (величины) [наблюдаемого объекта] со шкалой измерений этого свойства (величины) в целях получения результата измерений (оценки свойства или значения величины).

**СРАВНЕНИЕ** – познавательная операция, заключающаяся в нахождении сходства и различия между предметами, явлениями, событиями и лежащая в основе суждений о сходстве или различии объектов. Один из главных способов познания окружающего мира. При сравнении устанавливают закономерности, присущие объектам, системам объектов и их характеристикам.

---

<sup>1</sup>Государственная система обеспечения единства измерений. Шкалы измерений. Термины и определения.

# Шкалы измерений I

**ШКАЛА ИЗМЕРЕНИЙ** – (согласно РМГ 83-2007)<sup>1</sup> – отображение множества различных проявлений количественного или качественного свойства [наблюдаемого объекта] на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений).

**ИЗМЕРЕНИЕ** – сравнение конкретного проявления измеряемого свойства (величины) [наблюдаемого объекта] со шкалой измерений этого свойства (величины) в целях получения результата измерений (оценки свойства или значения величины).

**СРАВНЕНИЕ** – познавательная операция, заключающаяся в нахождении сходства и различия между предметами, явлениями, событиями и лежащая в основе суждений о сходстве или различии объектов. Один из главных способов познания окружающего мира. При сравнении устанавливают закономерности, присущие объектам, системам объектов и их характеристикам.

**Два замечания:**

- Различают пять основных типов шкал.
- Различают одномерные и многомерные шкалы измерений.

---

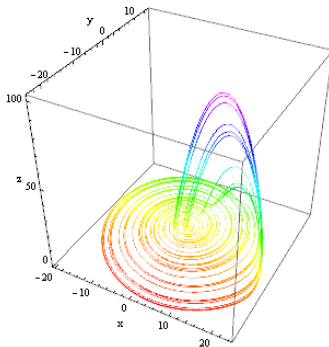
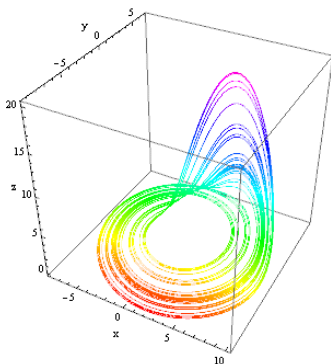
<sup>1</sup>Государственная система обеспечения единства измерений. Шкалы измерений. Термины и определения.

## Шкалы измерений II

Признак типа шкалы измерений	Тип шкалы измерений				
	Наименований	Порядка	Интервалов	Отношений	Абсолютные
Эквивалентность $A = B, A \neq B$	+	+	+	+	+
Порядок $A < B < C$		+	+	+	+
Метрика $d(A, B)$			+	+	+
Логарифмирование $\log A, \log B$			+	+	+
Единица измерения (масштаб)	Не применимо	Не применимо $\neg \exists A - B$	По соглашению $\neg \exists A / B$	По соглашению $\exists A / B$	Естественный $\exists A / B$
Наличие нуля	Не применимо	Необязательно	По соглашению	Естественный	Естественный
Допустимые преобразования $A' = f(A), B' = f(B)$	Изоморфное отображение	Монотонные преобразования	Линейные преобразования	Умножение (1, 2 р.) Сумма (2 р.)	Тождественные преобразования
Пример	Москва, Рубашка, Планета, ...	..., Холодно, Тепло, Жарко, ...	Шкала Времени Шкала Тем-ры, °C	Шкала Массы Шкала Тем-ры, К	Кол-во квантов Коеф. усиления

## Формальное представление

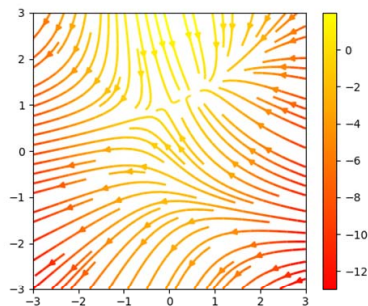
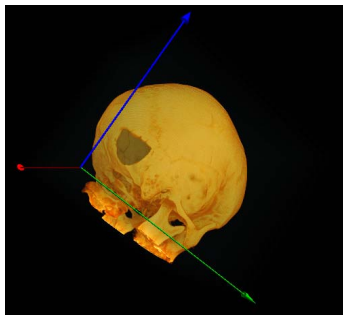
Временной ряд:



$$\text{TS} = \left\{ \{s_k\}_{k=1}^K, \{t_k\}_{k=1}^K \right\}, \quad s \in S \subset \mathbb{R}^N, \quad t \in T \subset \mathbb{R}, \quad t_k < t_{k+1}.$$

# Формальное представление

Дискретное скалярное (векторное) поле:



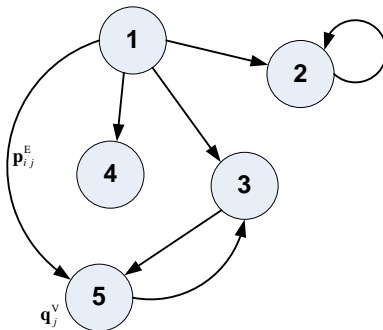
$$\mathbf{s}(\mathbf{r}_m, t_k) = \mathbf{S}\left(r_m^{(0)}, \dots, r_m^{(M)}, t_k\right), \quad \mathbf{s} \in \mathbf{S} \subset \mathbb{R}^N, \quad \mathbf{r} \in \mathbf{R} \subset \mathbb{R}^M,$$

$$t \in \mathbf{T} \subset \mathbb{R}, \quad t_k \prec t_{k+1}.$$



## Формальное представление

Граф:



$\Gamma|t_k = \langle V, E \rangle | t_k$ ,  $V$  – множество вершин,  $E$  – множество рёбер,  
 $t \in T \subset \mathbb{R}$ ,  $t_k \prec t_{k+1}$ .

# Формальное представление

Мультимножество:



$$M|t_k = \{A_1, A_2, A_1, A_3, \dots\} | t_k \quad A_i - \text{множество},$$

$$t \in T \subset \mathbb{R}, \quad t_k \prec t_{k+1}.$$

# Большие данные

**БОЛЬШИЕ ДАННЫЕ** – это набор стратегий, методов и технологий связанных со сбором, хранением и обработкой наборов данных, отвечающих следующим условиям:

- Большой объём данных, превосходящий возможности их сохранения «на одном сервере».
- Высокая скорость поступления данных (режим реального времени).
- Существенная неструктурированность и гетерогенность поступающих данных.

# Большие данные

**БОЛЬШИЕ ДАННЫЕ** – это набор стратегий, методов и технологий связанных со сбором, хранением и обработкой наборов данных, отвечающих следующим условиям:

- Большой объём данных, превосходящий возможности их сохранения «на одном сервере».
- Высокая скорость поступления данных (режим реального времени).
- Существенная неструктурированность и гетерогенность поступающих данных.

Три важных аспекта обработки Больших Данных:

- глубина анализа данных, позволяющая обнаружить, осмыслить и понять суть явления, а также спрогнозировать его последствия.
- проявление (статистически значимых только на больших объёмах выборок) тонких, порой контринтуитивных, эффектов.
- особые требования к алгоритмам и программному обеспечению:  
(i) – производительность (массовая параллельность, алгоритмы типа  $\log N$ ,  $N$ ,  $N \log N$ ); (ii) – устойчивость (отсутствие рекурсивных вызовов, вспомогательная память не более  $N$ ); (iii) – организация (асинхронная среда с распределённой памятью, минимизация дисковых и коммуникационных операций).

# Основные обозначения

Даны мультимножества:

$X$  – описаний объектов (характеристики, признаки, features);

$R$  – решений алгоритма (ответы, метки, patterns, labels).

Существует, но неизвестна, целевая функция (target function):  $G' : X \rightarrow R$ .

Логическая пара:  $d_n = (x_n, r_n)$  – составляет  $n$ -й прецедент.

Необходимо найти алгоритм (решающую функцию, decision function):  
 $G : X \rightarrow R$ , которая восстанавливает оценку  $G'$ .

Подмножества прецедентов, выборки:

$D^{Tr} = \{(x_n, r_n)\}_{n=1}^{N_{Tr}}$  – обучающая (train set),

$D^{Ts} = \{(x_n, r_n)\}_{n=1}^{N_{Ts}}$  – тестовая (test set).

Исключение протечек данных (leaked data):

$X^{Tr} \cap X^{Ts} = \emptyset$  – необходимое условие.

Множество  $W$  – допустимые параметры алгоритма  $G$ .

Функционал  $Q$  – оценивание качества функционирования алгоритма  $G$ .

## Признаки

Введём отображение:  $U' : X \rightarrow V$ , где  $V^{(m)}$  – множество допустимых значений  $m$ -го признака,  $m = \overline{1, M}$ .

В зависимости от структуры множества  $V^{(m)}$ , признаки делятся на следующие типы:

- $V^{(m)} = \{0, 1\}$  – **бинарный**.
- $V^{(m)} \subset \mathbb{Z}$ ,  $|V| < \infty$  – **именованный** (номинальный).
- $V^{(m)} \subset \mathbb{Z}$ ,  $|V| < \infty$ ,  $v_i \prec v_{i+1}$  – **порядковый**.
- $V^{(m)} \subset (\mathbb{Z}, \mathbb{R}, \mathbb{C})$ ,  $v_i < v_j$ ,  $d(v_i, v_j)$  – **количественный**.

Вектор  $\mathbf{x}_j = [x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(M)}]$  называют **признаковым описанием**  $j$ -го наблюдаемого объекта.

Если тип всех признаков  $m = \overline{1, M}$  одинаков, то признаковое описание называю **однородным**, иначе **гетерогенным**.

Наиболее распространённым способом представления множества  $X$  в прикладных задачах является **матрица объектов–признаков**:

$$\begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(M)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(M)} \\ \dots & \dots & \dots & \dots \\ x_N^{(1)} & x_N^{(2)} & \dots & x_N^{(M)} \end{bmatrix}.$$

# Задача Классификации

Множество допустимых значений  $R$  – выражается в шкале Наименований (или Порядка):

$$U' : R \rightarrow V, \quad V^{(m)} \subset \mathbb{Z}, \quad |V| < \infty, \quad (v_i^{(m)} \prec v_{i+1}^{(m)}).$$

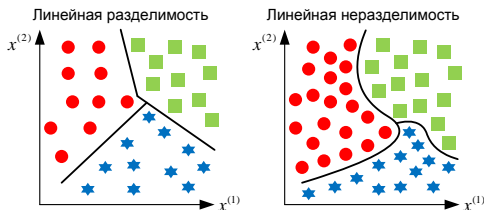
Дано обучающее множество

$$D^{\text{Tr}} \neq \emptyset.$$

Структура  $V$  – известна, её тип определяет тип классификатора:

- $V^{(m)} = \{0, 1\}$  – **бинарный**.
- $V^{(m)} = \{1, \dots, M\}$  – **многоклассовый** на  $M$  непересекающихся классов.
- $V^{(m)} = \{0, 1\}^M$  – **многоклассовый** на  $M$  пересекающихся классов.

Два фундаментальных типа классифицируемости  $X$ :



# Задача Кластеризации

Множество допустимых значений  $R$  – выражается в шкале Наименований:

$$U' : R \rightarrow V, \quad V \subset \mathbb{N}, \quad |V| < \infty.$$

Под  $v_i \in V$  понимается номер кластера.

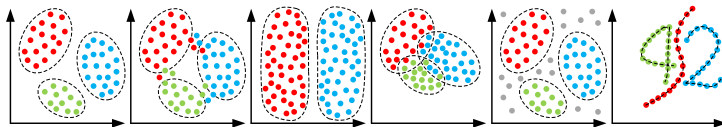
Обучающее множество – отсутствует:

$$D^{\text{Tr}} \equiv \emptyset.$$

Структура  $V$  – в общем случае неизвестна. Априорные предположения:

- Возможно указать число кластеров  $\hat{M}$ ,  $m = \overline{1, \hat{M}}$ .
- Возможно указать координаты центров кластеров  $\hat{\mathbf{x}}_c^{(m)}$ .

Типы кластерных многообразий:





# Задача Регрессии

Множество допустимых значений  $R$  – выражается в шкале Интервалов, Отношений или Абсолютной:

$$U' : R \rightarrow V, \quad V^{(m)} \subset (\mathbb{R}, \mathbb{C})^N, \quad v_i < v_j, \quad d(v_i, v_j).$$

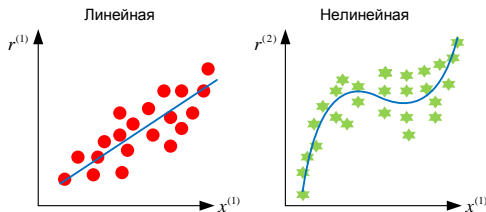
Дано обучающее множество

$$D^{\text{Tr}} \neq \emptyset.$$

Структура  $V$  – известна (это, как правило, результаты случайных измерений).

$X \xrightarrow{G} R$  – регрессионная модель (функция регрессии).

Два фундаментальных типа регрессионных моделей:



## Задача Редукции Размерности

$\mathbf{x}_{\text{src}}$ ,  $\mathbf{x}_{\text{red}}$  – исходное и редуцированное признаковое описание объектов.

Обучающее множество – отсутствует:

$$D^{\text{Tr}} \equiv \emptyset.$$

Множество  $R$  – не требуется.

Суть задачи  $X_{\text{src}} \xrightarrow{G} X_{\text{red}}$  (manifold learning):

$$\dim \mathbf{x}_{\text{src}} > \dim \mathbf{x}_{\text{red}},$$

при том, что:

$$Q^*[R_{\text{red}}|X_{\text{red}}] \geq Q^*[R_{\text{src}}|X_{\text{src}}] - \epsilon, \quad \epsilon \rightarrow +0,$$

где  $Q^*$  – качество решения сопряжённой задачи (классификация, кластеризация, регрессия, и т.п.).

Назначение редуцирующей модели  $G$ :

- Снижение вычислительной нагрузки при функционировании моделей машинного обучения.
- Повышение качества функционирования моделей.
- Когнитивная визуализация, исследование локальной и глобальной структур данных.

## Общие положения

Рассмотрим алгоритм  $G : X \times W \rightarrow R$ , где  $W$  – множество допустимых значений вектора параметра  $\mathbf{w}$ . Множество  $W$  также называют **пространством параметров** или **пространством поиска**.

Тогда **моделью алгоритмов** будет параметрическое семейство функций:

$$A = \{G(\mathbf{x}, \mathbf{w}) | \mathbf{w} \in W\}.$$

**Метод обучения** (learning algorithm) – это отображение вида:

$$\mu : (X \times R)^{N_{ds}} \rightarrow A,$$

которое произвольной конечной выборке  $D^{ds} = \{(\mathbf{x}_n, \mathbf{r}_n)\}_{n=1}^{N_{ds}}$  ставит в соответствие некоторый алгоритм  $a \in A$ , фактически находит оптимальное значение вектора параметров  $\tilde{\mathbf{w}} : a \equiv G(\mathbf{x}, \tilde{\mathbf{w}})$ .

В машинном обучении по прецедентам выделяют два основных этапа:

- Собственно **обучение** (training) – построение алгоритма  $G(\mathbf{x}, \tilde{\mathbf{w}})$  по выборке  $D^{ds}$ .
- **Применение** (testing) – эксплуатация алгоритма  $G(\mathbf{x}, \tilde{\mathbf{w}})$ .

# Обучение с учителем

Основные области применения:

- Классификация.
- Регрессия.

Основной момент: доступность обучающего множества  $D^{\text{Tr}} \neq \emptyset$ .

# Обучение без учителя

Основные области применения:

- Кластеризация.
- Manifold Learning.

Основной момент: отсутствие обучающего множества  $D^{\text{Tr}} \equiv \emptyset$ .

## Обучение с подкреплением

**ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ** (reinforcement learning) – один из способов машинного обучения, в ходе которого испытуемая система (агент) обучается, взаимодействуя с некоторой средой. Откликом среды (а не специальной системы управления подкреплением, как это происходит в обучении с учителем) на принятые решения являются сигналы подкрепления, поэтому такое обучение является частным случаем обучения с учителем, но учителем является среда или её модель.

## Обучение с подкреплением

**Q-ОБУЧЕНИЕ** (Q-learning) – метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучение с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии даёт ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения в том, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений.

## Обучение с подкреплением

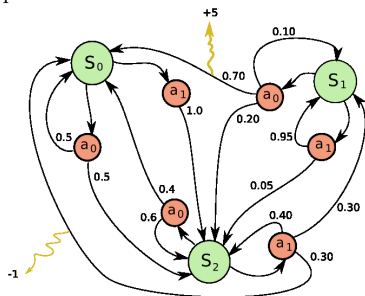
**Q-ОБУЧЕНИЕ** (Q-learning) – метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучение с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии даёт ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения в том, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений.

**МАРКОВСКИЙ ПРОЦЕСС ПРИНЯТИЯ РЕШЕНИЙ** (Markov decision process, MDP) – спецификация задачи последовательного принятия решений для полностью наблюдаемой среды с марковской моделью перехода и дополнительными вознаграждениями. Служит математической основой для того, чтобы смоделировать принятие решения в ситуациях, где результаты частично случайны и частично под контролем лица, принимающего решения.



# Обучение с подкреплением

**Q-ОБУЧЕНИЕ** (Q-learning) – метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучение с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии даёт ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения в том, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений.



Автор: MistWiz

## Функционал качества

**ФУНКЦИЯ ПОТЕРЬ** (loss function) – это неотрицательная функция  $L(\tilde{\mathbf{r}}, \mathbf{r})$ , характеризующая величину ошибки алгоритма  $\tilde{\mathbf{r}}_n = a(\mathbf{x}_n)$  на  $n$ -м прецеденте  $\mathbf{d}_n$ . Если  $L(\tilde{\mathbf{r}}, \mathbf{r}) = 0$ , то ответ  $a(\mathbf{x}_n)$  называется корректным.

Напоминание:  $a \equiv G(\mathbf{x}, \tilde{\mathbf{w}})$ ,  $\mathbf{d}_n = (\mathbf{x}_n, \mathbf{r}_n)$ .

## Функционал качества

**ФУНКЦИЯ ПОТЕРЬ** (loss function) – это неотрицательная функция  $L(\tilde{\mathbf{r}}, \mathbf{r})$ , характеризующая величину ошибки алгоритма  $\tilde{\mathbf{r}}_n = a(\mathbf{x}_n)$  на  $n$ -м прецеденте  $\mathbf{d}_n$ . Если  $L(\tilde{\mathbf{r}}, \mathbf{r}) = 0$ , то ответ  $a(\mathbf{x}_n)$  называется корректным.

Напоминание:  $a \equiv G(\mathbf{x}, \tilde{\mathbf{w}})$ ,  $\mathbf{d}_n = (\mathbf{x}_n, \mathbf{r}_n)$ .

**Функционал качества алгоритма**  $a$  на произвольной конечной выборке  $D^{\text{ds}}$

$$Q(a, D^{\text{ds}}) = \frac{1}{N^{\text{ds}}} \sum_{n=1}^{N^{\text{ds}}} L(a(\mathbf{x}_n), \mathbf{r}_n),$$

называют также функционалом средних потерь или эмпирическим риском, так как он вычисляется по эмпирическим данным  $D^{\text{ds}}$ .

## Функционал качества

**ФУНКЦИЯ ПОТЕРЬ** (loss function) – это неотрицательная функция  $L(\tilde{\mathbf{r}}, \mathbf{r})$ , характеризующая величину ошибки алгоритма  $\tilde{\mathbf{r}}_n = a(\mathbf{x}_n)$  на  $n$ -м прецеденте  $\mathbf{d}_n$ . Если  $L(\tilde{\mathbf{r}}, \mathbf{r}) = 0$ , то ответ  $a(\mathbf{x}_n)$  называется корректным.

Напоминание:  $a \equiv G(\mathbf{x}, \tilde{\mathbf{w}})$ ,  $\mathbf{d}_n = (\mathbf{x}_n, \mathbf{r}_n)$ .

**Функционал качества алгоритма**  $a$  на произвольной конечной выборке  $D^{\text{ds}}$

$$Q(a, D^{\text{ds}}) = \frac{1}{N^{\text{ds}}} \sum_{n=1}^{N^{\text{ds}}} L(a(\mathbf{x}_n), \mathbf{r}_n),$$

называют также функционалом средних потерь или эмпирическим риском, так как он вычисляется по эмпирическим данным  $D^{\text{ds}}$ .

**МЕТОД МИНИМИЗАЦИИ ЭМПИРИЧЕСКОГО РИСКА** (empirical risk minimization, ERM) – один из наиболее распространённых подходов к обучению алгоритмов по прецедентам. Он заключается в том, чтобы в заданной модели алгоритмов  $A = \{G(\mathbf{x}, \mathbf{w}) | \mathbf{w} \in W\}$  найти алгоритм  $a$ , минимизирующий среднюю ошибку на обучающей выборке:

$$a = \arg \min_{a \in A} Q(a, D^{\text{Tr}})$$

Следует отметить, что ERM сводит задачу обучения к оптимизации и задача может быть решена численными методами оптимизации.

## Переобучение алгоритма

Чем плохо  $Q(a, D^{\text{Tr}}) = 0$ ?

## Переобучение алгоритма

Чем плохо  $Q(a, D^{\text{Tr}}) = 0$ ?

Как минимум тем, что ничего не известно о величине  $Q(a, D^{\text{Ts}})$ .

## Переобучение алгоритма

**ПЕРЕОБУЧЕНИЕ** (переподгонка, overfitting) – явление в машинном обучении и статистике, при котором построенная модель  $a$  хорошо объясняет примеры из обучающей выборки  $D^{Tr}$ , но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки  $D^{Ts}$ ). Как правило, переобучение возникает при использовании избыточно сложных моделей.

## Переобучение алгоритма

**ПЕРЕОБУЧЕНИЕ** (переподгонка, overfitting) – явление в машинном обучении и статистике, при котором построенная модель  $a$  хорошо объясняет примеры из обучающей выборки  $D^{Tr}$ , но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки  $D^{Ts}$ ). Как правило, переобучение возникает при использовании избыточно сложных моделей.

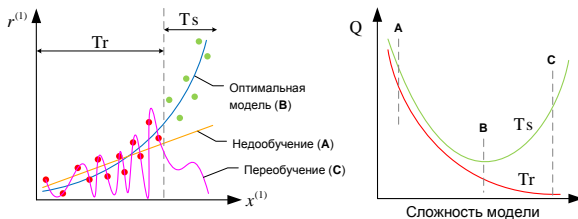
**НЕДООБУЧЕНИЕ** – явление в машинном обучении по прецедентам, когда алгоритм обучения  $\mu$  не обеспечивает достаточно малой величины средней ошибки  $Q$  на обучающей выборке  $D^{Tr}$ . Как правило, недообучение возникает при использовании недостаточно сложных моделей.



## Переобучение алгоритма

**ПЕРЕОБУЧЕНИЕ** (переподгонка, overfitting) – явление в машинном обучении и статистике, при котором построенная модель *a* хорошо объясняет примеры из обучающей выборки  $D^{Tr}$ , но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки  $D^{Ts}$ ). Как правило, переобучение возникает при использовании избыточно сложных моделей.

**НЕДООБУЧЕНИЕ** – явление в машинном обучении по прецедентам, когда алгоритм обучения  $\mu$  не обеспечивает достаточно малой величины средней ошибки  $Q$  на обучающей выборке  $D^{Tr}$ . Как правило, недообучение возникает при использовании недостаточно сложных моделей.



Успешное обучение требует не только умения **запоминать** (memorization), но и способности **обобщать** (generalization).

## Регуляризация

Один из эффективных способов борьбы с переобучением модели – это регуляризация.

## Регуляризация

**РЕГУЛЯРИЗАЦИЯ** – (в машинном обучении, статистике, теории обратных задач) – метод добавления некоторой дополнительной информации к условию с целью решить некорректно поставленную задачу или предотвратить переобучение. Эта информация часто имеет вид штрафа за сложность модели. Например, это могут быть ограничения гладкости результирующей функции или ограничения по норме векторного пространства.

## Регуляризация

**РЕГУЛЯРИЗАЦИЯ** – (в машинном обучении, статистике, теории обратных задач) – метод добавления некоторой дополнительной информации к условию с целью решить некорректно поставленную задачу или предотвратить переобучение. Эта информация часто имеет вид штрафа за сложность модели. Например, это могут быть ограничения гладкости результирующей функции или ограничения по норме векторного пространства.

**Наиболее употребимые виды регуляризации** ( $\lambda \geq 0$  – параметр регуляризации):

$L_1$  – lasso regression (разряженные модели, отбор признаков):

$$L_1 = \lambda \sum_{i=1}^D |w_i|.$$

$L_2$  – ridge regression (Регуляризация Тихонова):

$$L_2 = \lambda \sum_{i=1}^D w_i^2.$$

Elastic Net – эластичная сеть:

$$L_x | \beta = \beta L_1 + (1 - \beta) L_2.$$

$$L'(\tilde{\mathbf{r}}, \mathbf{r}) = L(\tilde{\mathbf{r}}, \mathbf{r}) + L_o$$

## Варианты проведения анализа данных

Полу-ручная обработка экспертом малых выборок данных:

Ограничения:

- Низкая производительность труда и повторяемость результатов.
- Высокий уровень субъективизма в результатах (низкие уровни стат. значимости).
- Низкая обнаружительная способность – пропуск значимых эффектов.

## Варианты проведения анализа данных

### Полу-ручная обработка экспертом малых выборок данных:

#### Ограничения:

- Низкая производительность труда и повторяемость результатов.
- Высокий уровень субъективизма в результатах (низкие уровни стат. значимости).
- Низкая обнаружительная способность – пропуск значимых эффектов.

### Автоматизированная обработка средних и больших выборок данных:

#### Ограничения:

- Квалификация специалиста в области параллельного программирования.
- Существенные затраты на сбор релевантных данных.
- Существенные затраты на разработку аналитических инструментов.

## Варианты проведения анализа данных

### Полу-ручная обработка экспертом малых выборок данных:

#### Ограничения:

- Низкая производительность труда и повторяемость результатов.
- Высокий уровень субъективизма в результатах (низкие уровни стат. значимости).
- Низкая обнаружительная способность – пропуск значимых эффектов.

### Автоматизированная обработка средних и больших выборок данных:

#### Ограничения:

- Квалификация специалиста в области параллельного программирования.
- Существенные затраты на сбор релевантных данных.
- Существенные затраты на разработку аналитических инструментов.

### Тотальный автоматический скрининг всех доступных наборов данных:

#### Ограничения:

- Высокие затраты на интегрирование разнородных данных.
- Высокие затраты на разработку и поддержание аналитической системы.
- Высокий риск получения результата: «гора родила мышь».

## Проблематика построения аналитических систем

Основные направления задач требующих решения:

- Высоко-производительные вычислительные системы.
- Специализированные математические методы.
- Специализированное алгоритмическое и программное обеспечение.



# Проблематика построения аналитических систем

## Основные направления задач требующих решения:

- Высоко-производительные вычислительные системы.
- Специализированные математические методы.
- Специализированное алгоритмическое и программное обеспечение.

## Основные классы задач требующих решения:

- Сбор, структурирование и хранение первичных разнородных данных.
- Выявление паттернов, идентификация структурных и динамических свойств изучаемых систем и процессов.
- Понижение размерности данных и когнитивная визуализация.
- Модели предметной области для возможностей изучения систем, явлений и процессов, прогнозирования и управления.

## Граф обработки (базовые положения)

**ГРАФ ОБРАБОТКИ** – набор логически связанных вычислительных процедур, которые должны быть применены к каждому объекту (набору) данных с целью решения задач анализа и достижения поставленных целей исследования.

$$\Gamma = \langle V E \rangle, \quad V - \text{процедуры}, \quad E - \text{данные}.$$

## Граф обработки (базовые положения)

**ГРАФ ОБРАБОТКИ** – набор логически связанных вычислительных процедур, которые должны быть применены к каждому объекту (набору) данных с целью решения задач анализа и достижения поставленных целей исследования.

$$\Gamma = \langle V E \rangle, \quad V - \text{процедуры}, \quad E - \text{данные}.$$

Два полюса вычислительных процедур:

## Граф обработки (базовые положения)

**ГРАФ ОБРАБОТКИ** – набор логически связанных вычислительных процедур, которые должны быть применены к каждому объекту (набору) данных с целью решения задач анализа и достижения поставленных целей исследования.

$$\Gamma = \langle V E \rangle, \quad V - \text{процедуры}, \quad E - \text{данные}.$$

Два полюса вычислительных процедур:

**Изолированная по данным** – для выполнения не требуется доступ к данным смежных объектов (размер документа, частота слов в документе, фурье-преобразование сигнала с элемента ФАР, и т.п.) – хорошо реализуется через модель MapReduce, удобна для распараллеливания и конвейеризации.

## Граф обработки (базовые положения)

**ГРАФ ОБРАБОТКИ** – набор логически связанных вычислительных процедур, которые должны быть применены к каждому объекту (набору) данных с целью решения задач анализа и достижения поставленных целей исследования.

$$\Gamma = \langle V E \rangle, \quad V - \text{процедуры}, \quad E - \text{данные}.$$

Два полюса вычислительных процедур:

**Изолированная по данным** – для выполнения не требуется доступ к данным смежных объектов (размер документа, частота слов в документе, фурье-преобразование сигнала с элемента ФАР, и т.п.) – хорошо реализуется через модель MapReduce, удобна для распараллеливания и конвейеризации.

**Полносвязная по данным** – для выполнения требуется доступ к данным всех объектов выборки (поиск цитирований, построение компонент связности графа, пространственно-временная фильтрация сигнала, и т.п.) – требует моделей матрично-графовых вычислений, сложно распараллеливается, является высоконагруженной вычислительной задачей.

## Граф обработки (базовые положения)

**ГРАФ ОБРАБОТКИ** – набор логически связанных вычислительных процедур, которые должны быть применены к каждому объекту (набору) данных с целью решения задач анализа и достижения поставленных целей исследования.

$$\Gamma = \langle V E \rangle, \quad V - \text{процедуры}, \quad E - \text{данные}.$$

**Закон Амдала** – иллюстрирует ограничение роста производительности вычислительной системы с увеличением количества вычислителей: «В случае, когда задача разделяется на несколько частей, суммарное время её выполнения на параллельной системе не может быть меньше времени выполнения самого длинного фрагмента».

$$V_{\text{prl}} = \frac{1}{\alpha + \frac{1 - \alpha}{p}},$$

$V_{\text{prl}}$  – ускорение,  $p$  – число узлов (workers),  $\alpha$  – доля последовательных вычислений от всего объёма вычислений.

## Типы и назначения моделей

**ОПИСАНИЕ** – формализованное представление изучаемого явления.

## Типы и назначения моделей

**ОПИСАНИЕ** – формализованное представление изучаемого явления.

**ОБЪЯСНЕНИЕ** – понимание изучаемого явления.



## Типы и назначения моделей

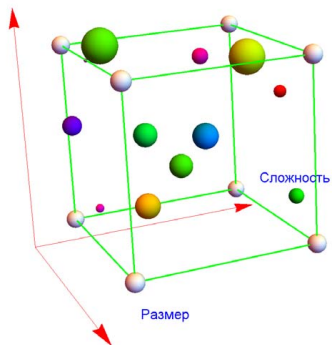
**ОПИСАНИЕ** – формализованное представление изучаемого явления.

**ОБЪЯСНЕНИЕ** – понимание изучаемого явления.

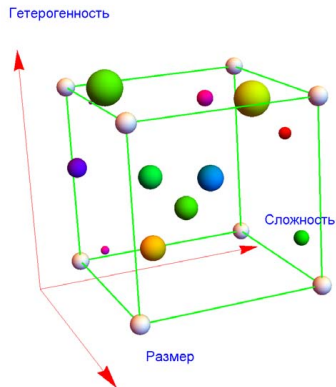
**ПРЕДСКАЗАНИЕ** – выход нового знания.

# Классификация задач Data Science

Гетерогенность

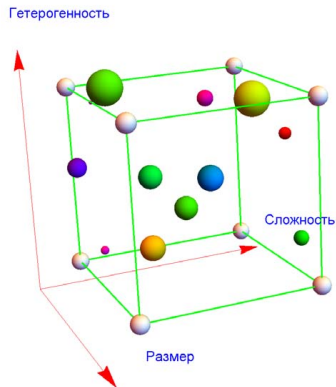


# Классификация задач Data Science



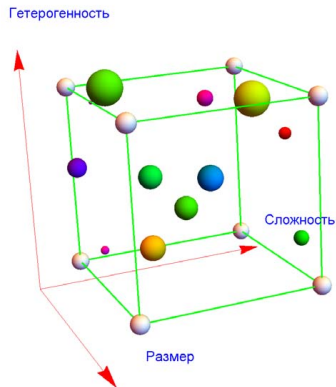
**РАЗМЕР ЗАДАЧИ** – определяется количеством операций и/или объёмом данных (на уровне размерности вектора и количества объектов), требуемых для решения задачи.

# Классификация задач Data Science



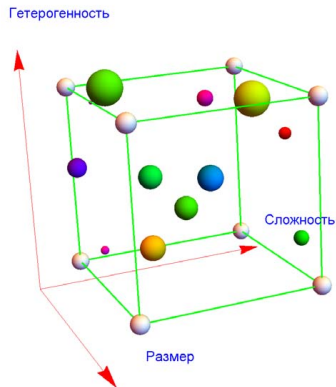
**СЛОЖНОСТЬ ЗАДАЧИ** – зависит от существования «вычислимого» алгоритма решения задачи и его переносимости на компьютерные платформы.

# Классификация задач Data Science



**ГЕТЕРОГЕННОСТЬ ЗАДАЧИ** – определяется количеством разделов математики требующихся для решения задачи и разнородностью и/или неструктурированностью входных данных.

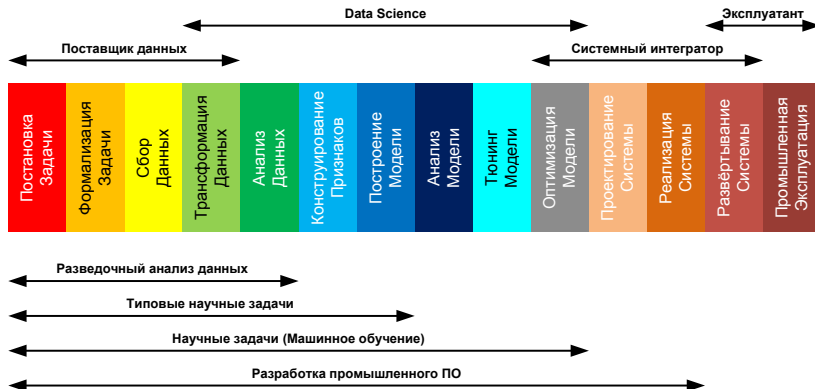
# Классификация задач Data Science



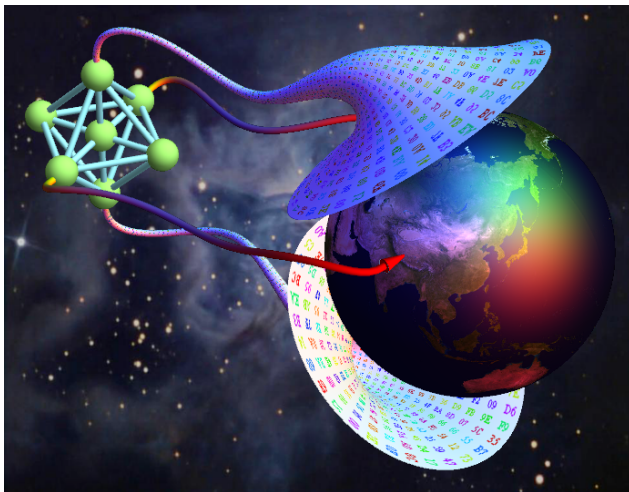
Важный контекст задач:

- *Медианные* – типовые потребности большинства Аналитиков.
- *Экстремальные* – ~~за гранью добра и зла~~ – редкие и специфичные задачи.

## Конвейер Data Science



## Смена технологических парадигм



Андрей Макаренко: От Network-Centric к Big Data – смена моды  
или закономерное развитие технологий?



# Outline section

- ❶ Искусственный интеллект
  - Общие положения
  
- ❷ Машинное обучение
  - Общие положения
  - О данных
  - Основные классы задач
  - Основные методы обучения
  - Прикладные аспекты
  
- ❸ Заключение

# Контрольная работа

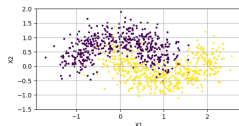
## Задание для слушателей:

- 1 Изучить понятие **confusion matrix** (матрица ошибок), а также смысл мер: **Precision**, **Recall**,  $F_1$ ; для случая бинарного классификатора.

- 2 Дескриптивная статистика:

Изучить внутренний датасет:

```
from sklearn.datasets import make_moons  
X, y = make_moons(1000, noise = 0.275)
```



- 3 Набор  $(X, y)$  разбить на обучающую и тестовую выборки в соотношении 75/25. Изучить на нём функционирование следующих классификаторов (оформить Jupyter Notebook):

- `from sklearn.linear_model import LogisticRegression`
- `from sklearn.neighbors import KNeighborsClassifier`
- `from sklearn.neighbors import KNeighborsClassifier`
- `from sklearn.tree import DecisionTreeClassifier`
- `from sklearn.ensemble import RandomForestClassifier`
- `from sklearn.svm import SVC`