Введение в АД

О курсе

Команда Физтех.Статистики

Проводимые нами учебные курсы

- Введение в анализ данных;
- DS-поток;
- Phystech@DataScience (в т.ч. мат. статистика на ЛФИ);
- Мат. статистика на ФБМФ и ФМХФ (семинары);
- Два курса в магистратуре кафедры X5 Group;
- Машинное обучение в ШАД (частично);

Организационная информация

Телеграм-бот

@miptstats_ds22_bot

Сайт команды mipt-stats.gitlab.io

Почта mipt.stats@yandex.ru

Кафедра дискретной математики

Цели курса

1. Дать представление об анализе данных;
2. Обучить базовым инструментам анализа данных;
3. Рассказать о практическом смысле объектов теории вероятностей;
4. Помочь определиться с кафедрой.

Яндекс

В курсе предполагается участие кафедры анализа данных:

- Гостевые лекции от Яндекса;
- Работа по курсу учитывается при отборе на кафедру.

План занятий первой части (обязательная)

Дата	Тема	Лектор		Дедлайн по ДЗ
5.02	*Введение, инструменты*		Никита Волков	25.02
12.02	*Сбор данных. Модели данных*		Роман Логинов	25.02
19.02	*Самостоятельная работа*			
26.02	*Теория вероятностей на практике*		Екатерина Юшина	11.03
05.03	*Линейная регрессия*		Никита Волков	18.03
12.03	*Гостевая лекция*			

План занятий второй части (факультативная)

Дата	Тема	Лектор		Дедлайн по ДЗ
19.03	*Решающие деревья, случайные леса*		Никита Волков	29.03
26.03	*Основы статистики. Свойства оценок.*		Никита Волков	5.04
2.04	*Где-то тут контрольная на кафедру АД (инф. позже)*			
9.04	*Байесовские классификаторы*		Никита Волков	19.04
16.04	*Введение в нейросети*		Виталия Жихарева	26.04
23.04	*Гостевая лекция*			

Система оценивания первой части

Правила:

1. если Л $>$ 25% и (В $<$ 50% или С $<$ 20%),

   то **О = 3+4*Л**;

2. если Л $>$ 25% и В $\geqslant$ 50% и С $\geqslant$ 20%,

   то **О = max(3+4*Л, 3+В+7*ЛС)**.

3. если Л $\leqslant$ 25%,

   то **О = 1+3*Т**.

Обозначения:

- Л — доля выполнения легких заданий (их немного);
- С — доля выполнения сложных заданий (их много);
- Т — доля выполнения тестов;
- ЛС — доля выполнения всех заданий (кроме тестов);
- В — доля правильных ответов на вопросы в боте на занятии;
- О — итоговая оценка, округляется вверх.

Система оценивания первой части

Правила:

1. если Л $>$ 25% и (В $<$ 50% или С $<$ 20%),

   то **О = 3+4*Л**;

2. если Л $>$ 25% и В $\geqslant$ 50% и С $\geqslant$ 20%,

   то **О = max(3+4*Л, 3+В+7*ЛС)**.

3. если Л $\leqslant$ 25%,

   то **О = 1+3*Т**.

Следствия: для получения оценки

- уд(3) достаточно выполнить 33.4% тестов.
- хор(5) достаточно выполнить 25.1% легких заданий.
- отл(8) достаточно выполнить 50.1% всех заданий и ответить на 50% вопросов.

Если списать: все участники списывания сдают устный зачет.

DS-поток

DS-поток

Семестр	DS-поток	Основной поток ПМИ
5	**Математическая статистика**	**Математическая статистика**
	Машинное обучение	**Машинное обучение**
	Практика	*Практика по мат. статистике*
	Основы прикладной статистики	Курс по выбору x 2
	Курс по выбору	
6	**Дискр. случ. процессы и временные ряды**	**Случайные процессы**
	Прикладная статистика и анализ данных	Курс по выбору x 3
	Практика	
	Курс по выбору	
Кафедра АД	Курс ШАД	Методы прикладной статистики
7	**Байесовский подход в анализе данных**	**Курс по выбору x 2**
	Практика	
8	**Прикладные задачи машинного обучения**	**Курс по выбору x 2**
	Практика	

Примечания.

1. По одним и тем же курсам лекторы разные.
2. По факту практика не является отдельным курсов.
3. Полужирным выделены курсы, по которым экзамен.
4. В процессе обучения перейти в DS-поток невозможно.

DS-поток

Чему мы учим

1. В меру глубокое математическое понимание статистики и машинного обучения.
2. Применение математических моделей на реальных данных, в том числе на реальных задачах.
3. Умение составлять полноценные выводы.

Реальная практика на DS-потоке

1. Реальные примеры из практики;
2. Соревнования на Kaggle;
3. Гостевые лекторы, применяющие анализ данных на практике;
4. Разбор статей на тему анализа данных;
5. Бонусы за участие в хакатонах, соревнованиях по анализу данных и прочую активность;

Отбор на DS-поток

Что будет учитываться?

1. **Необходимое условие:**
   оценка не менее отл(8) по первой части курса
   "Введение в анализ данных" и не менее хор(7) по второй части.
2. Работа в семестре по курсу, грамотное оформление ДЗ.
3. Оценка по теории вер., в меньшей степень — другие предметы.

Что нужно делать для отбора?

1. Трудиться в течение семестра.
2. В июне подать заявку.
3. Ждать. Результаты летом.

DS-поток адаптирован для ПМИ. Студенты КТ и ИВТ могут попасть на DS-поток, но возможна доп. нагрузка и проблемы с расписанием.

Куда пойти?

2 курс

Введение в анализ данных

3-4 курсы

DS-поток

Кафедра анализа данных

Школа анализа данных (ШАД)

1. Если анализ данных интересен, то хорошее решение: DS-поток + кафедра анализа данных.
2. Если выбираете кафедру анализа данных, то кафедра рекомендует пойти на DS-поток.

Что такое анализ данных?

Кому нужен анализ данных

1. "*Я математик, практическое применение не интересует*".

   Скорее всего АД не нужен,
   но часто математики им начинают интересоваться.

2. "*Я математик, но хочу применять свои знания на практике*".

   АД для вас, ждем вас на DS-потоке

3. "*Я программист, и хочу писать только код*".

   Скорее всего АД в подробностях не нужен,
   но стоит понимать, чем занимаются коллеги-аналитики.

4. "*Я программист, но хочу глубоко разбираться в тонкостях математических методов*".

   АД для вас, ждем вас на DS-потоке

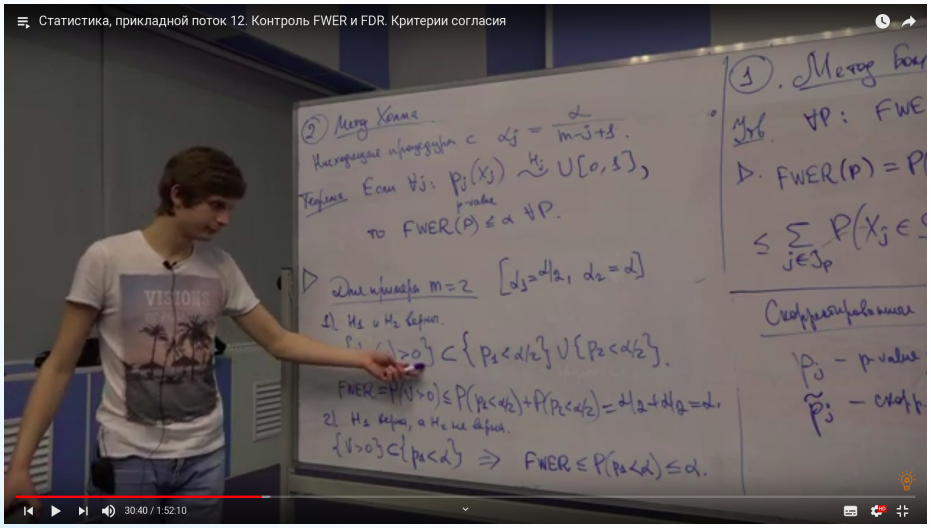
Так что, анализ данных

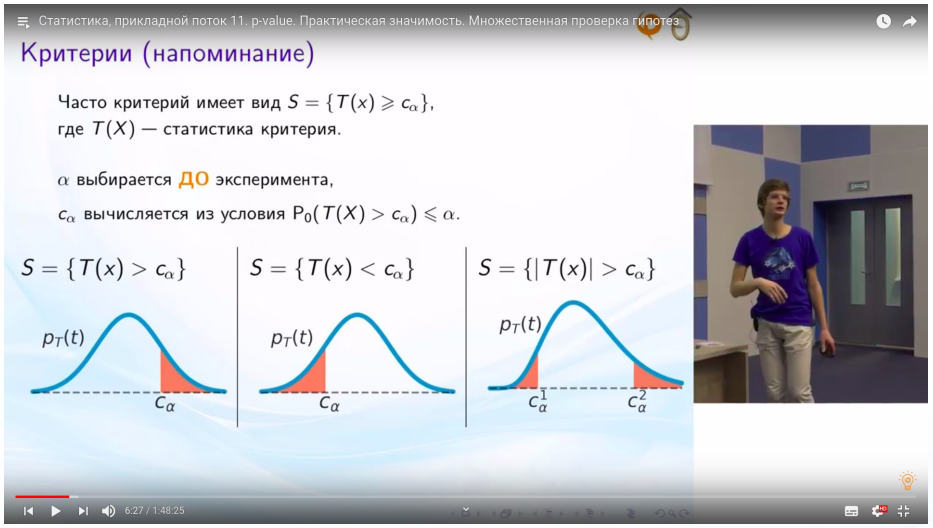
это математика

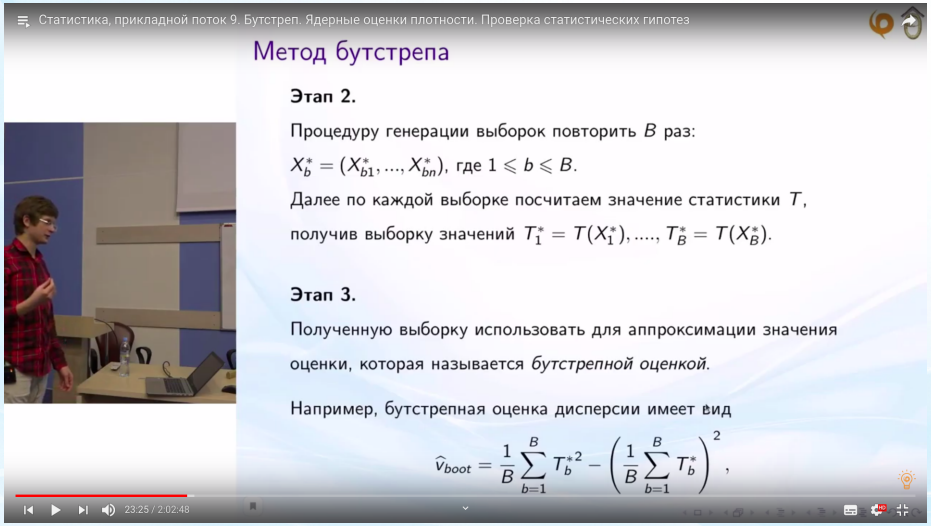
или программирование?

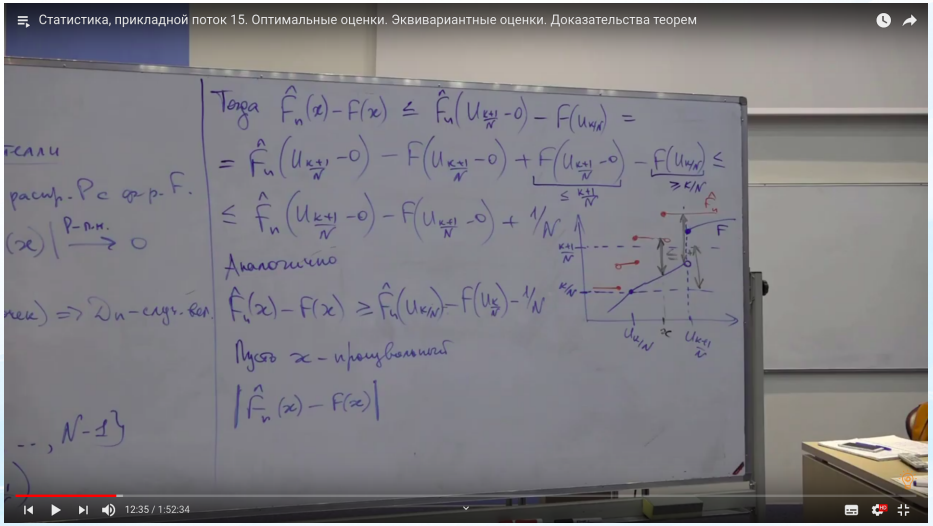
Давайте разбираться...

Посмотрим на лекции по статистике









Посмотрим на научные статьи

McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426, 2018

The choice of non-expansive maps in Definition 6 is due to Spiv
note that it closely mirrors the work of Carlsson and Memoli in [1
logical methods for clustering as applied to finite metric spaces. Th
significant since pure isometries are too strict and do not provide la
Hom-sets.

In [43] Spivak constructs a pair of adjoint functors, Real and Si
the categories **sFuzz** and **EPMet**. These functors are the natural e
the classical realization and singular set functors from algebraic top
functor Real is defined in terms of standard fuzzy simplices $\Delta^n_{<a}$ as

$$\mathsf{Real}(\Delta^n_{<a}) \triangleq \left\{ (t_0, \ldots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} t_i = -\log(a), t_i \geq \right.$$

similarly to the classical realization functor $|\cdot|$. The metric on R
simply inherited from $\mathbb{R}^{n+1}$. A morphism $\Delta^n_{<a} \to \Delta^m_{<b}$ exists only if
is determined by a $\mathbf{\Delta}$ morphism $\sigma : [n] \to [m]$. The action of Rea
morphism is given by the map

$$(x_0, x_1, \ldots, x_n) \mapsto \frac{\log(b)}{\log(a)} \left(\sum_{i_0 \in \sigma^{-1}(0)} x_{i_0}, \sum_{i_0 \in \sigma^{-1}(1)} x_{i_0}, \ldots, \sum_{i_0 \in \sigma^{-1}(m)} x_{i_0} \right).$$

Such a map is clearly non-expansive since $0 \leq a \leq b \leq 1$ implies that $\log(b)/\log$
1.

We then extend this to a general simplicial set X via colimits, defining

$$\mathsf{Real}(X) \triangleq \operatorname*{colim}_{\Delta^n_{<a} \to X} \mathsf{Real}(\Delta^n_{<a}).$$

Since the functor Real preserves colimits, it follows that there exists a righ
adjoint functor. Again, analogously to the classical case, we find the right adjoint
denoted Sing, is defined for an extended pseudo metric space Y in terms of its
action on the category $\mathbf{\Delta} \times I$:

$$\mathsf{Sing}(Y) : ([n], [0, a)) \mapsto \hom_{\mathbf{EPMet}}(\mathsf{Real}(\Delta^n_{<a}), Y).$$

For our case we are only interested in finite metric spaces. To correspond with
this we consider the subcategory of bounded fuzzy simplicial sets **Fin-sFuzz**. We
therefore use the analogous adjoint pair FinReal and FinSing. Formally we define
the finite fuzzy realization functor as follows:

Algorithm 2 Constructing a local fuzzy simplicial set

```
function LocalFuzzySimplicialSet(X, x, n)
    knn, knn-dists ← ApproxNearestNeighbors(X, x, n)
    ρ ← knn-dists[1]                          ▷ Distance to nearest neighbor
    σ ← SmoothKNNDist(knn-dists, n, ρ)        ▷ Smooth approximator to knn-distance
    fs-set_0 ← X
    fs-set_1 ← {([x, y], 0) | y ∈ X}
    for all y ∈ knn do
        d_{x,y} ← max{0, dist(x, y) − ρ}/σ
        fs-set_1 ← fs-set_1 ∪ ([x, y], exp(−d_{x,y}))
    return fs-set
```

Algorithm 3 Compute the normalizing factor for distances σ

```
function SmoothKNNDist(knn-dists, n, ρ)
    Binary search for σ such that Σ_{i=1}^{n} exp(−(knn-dists_i − ρ)/σ) = log_2(n)
    return σ
```

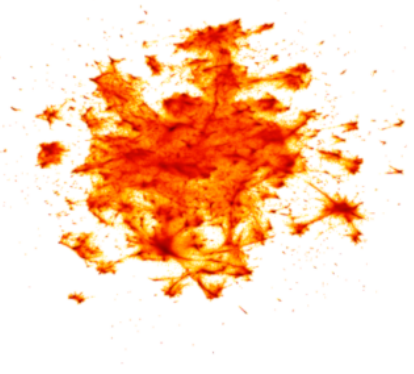


Figure 8: Visualization of the full 3 million word vectors from the GoogleNews dataset as embedded by UMAP.

is contained in U, then g is constant in B and hence $\sqrt{\det(g)}$ is constant
can be brought outside the integral. Thus, the volume of B is

$$\sqrt{det(g)} \int_B dx^1 \wedge \ldots \wedge dx^n = \sqrt{det(g)} \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)},$$

re r is the radius of the ball in the ambient $\mathbb{R}^n$. If we fix the volume of the
to be $\frac{\pi^{n/2}}{\Gamma(n/2+1)}$ we arrive at the requirement that

$$\det(g) = \frac{1}{r^{2n}}.$$

, since g is assumed to be diagonal with constant entries we can solve for g
f as

$$g_{ij} = \begin{cases} \frac{1}{r^2} & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

geodesic distance on $\mathcal{M}$ under g from p to q (where $p, q \in B$) is defined as

$$\inf_{c \in C} \int_a^b \sqrt{g(\dot{c}(t), \dot{c}(t))} dt,$$

where C is the class of smooth curves c on $\mathcal{M}$ such that $c(a) = p$ and $c(b) = q$,
and $\dot{c}$ denotes the first derivative of c on $\mathcal{M}$. Given that g is as defined in (2) we
see that this can be simplified to

$$\begin{aligned} &\frac{1}{r} \inf_{c \in C} \int_a^b \langle \sqrt{\dot{c}(t), \dot{c}(t)} \rangle dt \\ =&\frac{1}{r} \inf_{c \in C} \int_a^b \langle \|\dot{c}(t), \dot{c}(t)\| dt \\ =&\frac{1}{r} d_{\mathbb{R}^n}(p, q). \end{aligned} \tag{3}$$

□

B Proof that FinReal and FinSing are adjoint

Theorem 2. *The functors* FinReal : **Fin-sFuzz** → **FinEPMet** *and* FinSing : **FinEPMet** → **Fin-sFuzz** *form an adjunction with* FinReal *the left adjoint and* FinSing *the right adjoint.*

Посмотрим на научные статьи

Diederik P Kingma, Max Welling: *Auto-Encoding Variational Bayes*,

ArXiv 1312.6114, 2014

2.2 The variational bound

The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$, which can each be rewritten as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \quad (1)$$

The first RHS term is the KL divergence of the approximate from the true posterior. Since this KL-divergence is non-negative, the second RHS term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ is called the (variational) *lower bound* on the marginal likelihood of datapoint i, and can be written as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})] \quad (2)$$

which can also be written as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}}$$

We want to differentiate and optimize the lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}$ parameters $\boldsymbol{\phi}$ and generative parameters $\boldsymbol{\theta}$. However, the gradien is a bit problematic. The usual (naïve) Monte Carlo gradient estir is: $\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}[f(\mathbf{z})] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z})}[f(\mathbf{z}) \nabla_{q_{\boldsymbol{\phi}}(\mathbf{z})} \log q_{\boldsymbol{\phi}}(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^{L} f$ $\mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$. This gradient estimator exhibits exhibits very hi and is impractical for our purposes.

2.3 The SGVB estimator and AEVB algorithm

In this section we introduce a practical estimator of the lower bour parameters. We assume an approximate posterior in the form $q_{\boldsymbol{\phi}}($ technique can be applied to the case $q_{\boldsymbol{\phi}}(\mathbf{z})$, i.e. where we do not con variational Bayesian method for inferring a posterior over the parame

Under certain mild conditions outlined in section 2.4 for a chosen app can reparameterize the random variable $\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ using a differen of an (auxiliary) noise variable $\boldsymbol{\epsilon}$:

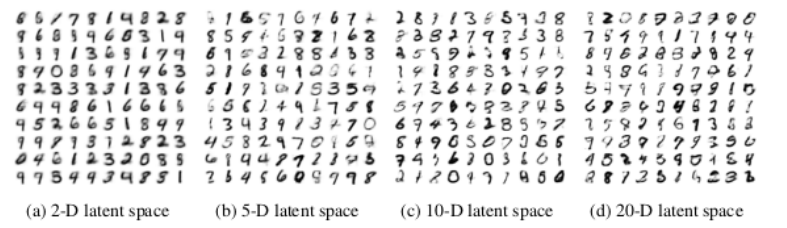
$$\tilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

See section 2.4 for general strategies for chosing such an approriate $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$. We can now form Monte Carlo estimates of expectations $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ as follows:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[f(\mathbf{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))\right] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i}$$

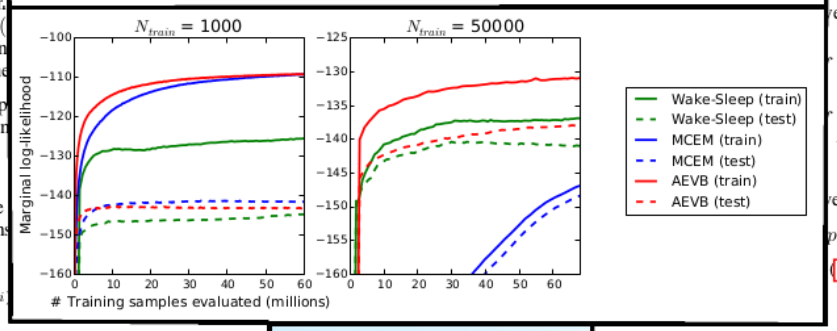
We apply this technique to the variational lower bound (eq. (2)), yielding our generic Stochastic Gradient Variational Bayes (SGVB) estimator $\tilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$:

$$\tilde{\mathcal{L}}^{A}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \frac{1}{L} \sum^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)})$$



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.



$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x})$ is the variational lower bound of the marginal likelihood of datapoint i:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z} \quad (16)$$

The expectations on the RHS of eqs (14) and (16) can obviously be written as a sum of three separate expectations, of which the second and third component can sometimes be analytically solved, e.g. when both $p_{\boldsymbol{\theta}}(\mathbf{x})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are Gaussian. For generality we will here assume that each of these expectations is intractable.

Under certain mild conditions outlined in section (see paper) for chosen approximate posteriors parameterize conditional samples $\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ as

$$\tilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \quad (17)$$

) and a function $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$ such that the following holds:

$$z|x) \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z}$$

$$\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right)\Big|_{\mathbf{z}=g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x}^{(i)})} d\boldsymbol{\epsilon} \quad (18)$$

approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$:

$$\tilde{\boldsymbol{\theta}} = h_{\boldsymbol{\phi}}(\boldsymbol{\zeta}) \quad \text{with} \quad \boldsymbol{\zeta} \sim p(\boldsymbol{\zeta}) \quad (19)$$

e, choose a prior $p(\boldsymbol{\zeta})$ and a function $h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})$ such that the following

$$q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right) d\boldsymbol{\theta}$$

$$p(\boldsymbol{\zeta}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right)\Big|_{\boldsymbol{\theta}=h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})} d\boldsymbol{\zeta} \quad (20)$$

e introduce a shorthand notation $f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}, \boldsymbol{\theta})$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \quad (21)$$

(18), the Monte Carlo estimate of the variational lower bound, given

$$\mathcal{L}(\boldsymbol{\phi}; \mathbf{X}) \simeq \frac{1}{L} \sum_{l=1}^{L} f_{\boldsymbol{\phi}}(\mathbf{x}^{(l)}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(l)}), h_{\boldsymbol{\phi}}(\zeta^{(l)})) \quad (22)$$

where $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$ and $\zeta^{(l)} \sim p(\boldsymbol{\zeta})$. The estimator only depends on samples from $p(\boldsymbol{\epsilon})$ and $p(\boldsymbol{\zeta})$ which are obviously not influenced by $\boldsymbol{\phi}$, therefore the estimator can be differentiated w.r.t. $\boldsymbol{\phi}$.

Посмотрим на примеры использования библиотек

Variational Autoencoders

https://github.com/pyro-ppl/pyro/blob/dev/examples/vae/vae.py

```python
# Copyright (c) 2017-2019 Uber Technologies, Inc.
# SPDX-License-Identifier: Apache-2.0

import argparse

import numpy as np
import torch
import torch.nn as nn
import visdom

import pyro
import pyro.distributions as dist
from pyro.infer import SVI, JitTrace_ELBO, Trace_ELBO
from pyro.optim import Adam
from utils.mnist_cached import MNISTCached as MNIST
from utils.mnist_cached import setup_data_loaders
from utils.vae_plots import mnist_test_tsne, plot_llk, plot_vae_samp


# define the PyTorch module that parameterizes the
# diagonal gaussian distribution q(z|x)
class Encoder(nn.Module):
    def __init__(self, z_dim, hidden_dim):
        super().__init__()
        # setup the three linear transformations used
        self.fc1 = nn.Linear(784, hidden_dim)
        self.fc21 = nn.Linear(hidden_dim, z_dim)
        self.fc22 = nn.Linear(hidden_dim, z_dim)
        # setup the non-linearities
        self.softplus = nn.Softplus()

    def forward(self, x):
        # define the forward computation on the image x
        # first shape the mini-batch to have pixels in the rightmost
        x = x.reshape(-1, 784)
        # then compute the hidden units
        hidden = self.softplus(self.fc1(x))
        # then return a mean vector and a (positive) square root cov
        # each of size batch_size x z_dim
        z_loc = self.fc21(hidden)
        z_scale = torch.exp(self.fc22(hidden))
        return z_loc, z_scale


# define the PyTorch module that parameterizes the
# observation likelihood p(x|z)
class Decoder(nn.Module):
```

```python
# define the PyTorch module that parameterizes the
# observation likelihood p(x|z)
class Decoder(nn.Module):
    def __init__(self, z_dim, hidden_dim):
        super().__init__()
        # setup the two linear transformations used
        self.fc1 = nn.Linear(z_dim, hidden_dim)
        self.fc21 = nn.Linear(hidden_dim, 784)
        # setup the non-linearities
        self.softplus = nn.Softplus()

    def forward(self, z):
        # define the forward computation on the latent z
        # first compute the hidden units
        hidden = self.softplus(self.fc1(z))
        # return the parameter for the output Bernoulli
        # each is of size batch_size x 784
        loc_img = torch.sigmoid(self.fc21(hidden))
        return loc_img


# define a PyTorch module for the VAE
class VAE(nn.Module):
    # by default our latent space is 50-dimensional
    # and we use 400 hidden units
    def __init__(self, z_dim=50, hidden_dim=400, use_cuda=False):
        super().__init__()
        # create the encoder and decoder networks
        self.encoder = Encoder(z_dim, hidden_dim)
        self.decoder = Decoder(z_dim, hidden_dim)

        if use_cuda:
            # calling cuda() here will put all the parameters of
            # the encoder and decoder networks into gpu memory
            self.cuda()
        self.use_cuda = use_cuda
        self.z_dim = z_dim

    # define the model p(x|z)p(z)
    def model(self, x):
        # register PyTorch module `decoder` with Pyro
        pyro.module("decoder", self.decoder)
        with pyro.plate("data", x.shape[0]):
            # setup hyperparameters for prior p(z)
            z_loc = torch.zeros(x.shape[0], self.z_dim, dtype=x.dtype, dev
            z_scale = torch.ones(x.shape[0], self.z_dim, dtype=x.dtype, de
            # sample from prior (value will be sampled by guide when compu
```

```python
            # decode the latent code z
            loc_img = self.decoder.forward(z)
            # score against actual images
            pyro.sample("obs", dist.Bernoulli(loc_img).to_event(1), obs=x
            # return the loc so we can visualize it later
            return loc_img

    # define the guide (i.e. variational distribution) q(z|x)
    def guide(self, x):
        # register PyTorch module `encoder` with Pyro
        pyro.module("encoder", self.encoder)
        with pyro.plate("data", x.shape[0]):
            # use the encoder to get the parameters used to define q(z|x)
            z_loc, z_scale = self.encoder.forward(x)
            # sample the latent code z
            pyro.sample("latent", dist.Normal(z_loc, z_scale).to_event(1)

    # define a helper function for reconstructing images
    def reconstruct_img(self, x):
        # encode image x
        z_loc, z_scale = self.encoder(x)
        # sample in latent space
        z = dist.Normal(z_loc, z_scale).sample()
        # decode the image (note we don't sample in image space)
        loc_img = self.decoder(z)
        return loc_img


def main(args):
    # clear param store
    pyro.clear_param_store()

    # setup MNIST data loaders
    # train_loader, test_loader
    train_loader, test_loader = setup_data_loaders(MNIST, use_cuda=args.c

    # setup the VAE
    vae = VAE(use_cuda=args.cuda)

    # setup the optimizer
    adam_args = {"lr": args.learning_rate}
    optimizer = Adam(adam_args)

    # setup the inference algorithm
    elbo = JitTrace_ELBO() if args.jit else Trace_ELBO()
    svi = SVI(vae.model, vae.guide, optimizer, loss=elbo)
```

```python
    # setup visdom for visualizati
    if args.visdom_flag:
        vis = visdom.Visdom()

    train_elbo = []
    test_elbo = []
    # training loop
    for epoch in range(args.num_ep
        # initialize loss accumula
        epoch_loss = 0.
        # do a training epoch over
        # by the data loader
        for x, _ in train_loader:
            # if on GPU put mini-b
            if args.cuda:
                x = x.cuda()
            # do ELBO gradient and
            epoch_loss += svi.step

        # report training diagnost
        normalizer_train = len(tra
        total_epoch_loss_train = e
        train_elbo.append(total_ep
        print("[epoch %03d]  avera

        if epoch % args.test_frequ
            # initialize loss accu
            test_loss = 0.
            # compute the loss ove
            for i, (x, _) in enume
                # if on GPU put mi
                if args.cuda:
                    x = x.cuda()
                # compute ELBO est
                test_loss += svi.e

                # pick three rando
                # visualize how we
                if i == 0:
                    if args.visdom
                        plot_vae_s
                        reco_indic
                        for index
                            test_i
                            reco_i
                            vis.im

                            vis.im
```

Какие вообще инструменты могут потребоваться

Вывод:

и математика

и программирование

Анализ данных это

процесс поиска закономерностей

в данных при помощи

- средств визуализации данных,
- математических методов,
- программных алгоритмов.

Искусственный интеллект

Отличительная особенность:

нет четко зафиксированного ответа на каждый входящий объект.

Что можно почитать:

Анализ данных — основы и терминология

https://habr.com/ru/post/352812/

Всё, что вам нужно знать об ИИ — за несколько минут

https://habr.com/ru/post/416889/

Сравним задачи

Алгоритмы и структуры данных

Задача: дан массив x, нужно его отсортировать.

Ровно один правильный ответ, можно получить с помощью четких алгоритмов.

Комбинаторика

Задача: Сколько имеется способов раздать 11 разных цветков, трём девушкам: какой-то – 5, а остальным – по 3 цветка? [ОКТЧ 2019]

Ровно один правильный ответ.

Анализ данных

Задача: Имеются данные $(x_1, y_1), \ldots, (x_n, y_n)$. Восстановите по ним функцию $f : x \mapsto y$.

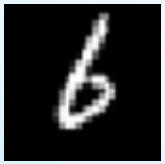
Особенности: нет четкого ответа, требуется только приближение, но есть критерии качества.

Пример — распознавание рукописных цифр

Вход: 

Ожидается на выходе: **5**

Но как четко алгоритмически определить границу между 6 и 8?

 — 2 или 9?

 — 4 или 7?

Актуальность в научной среде

Число статей по запросам в Google Scholar с 2016:

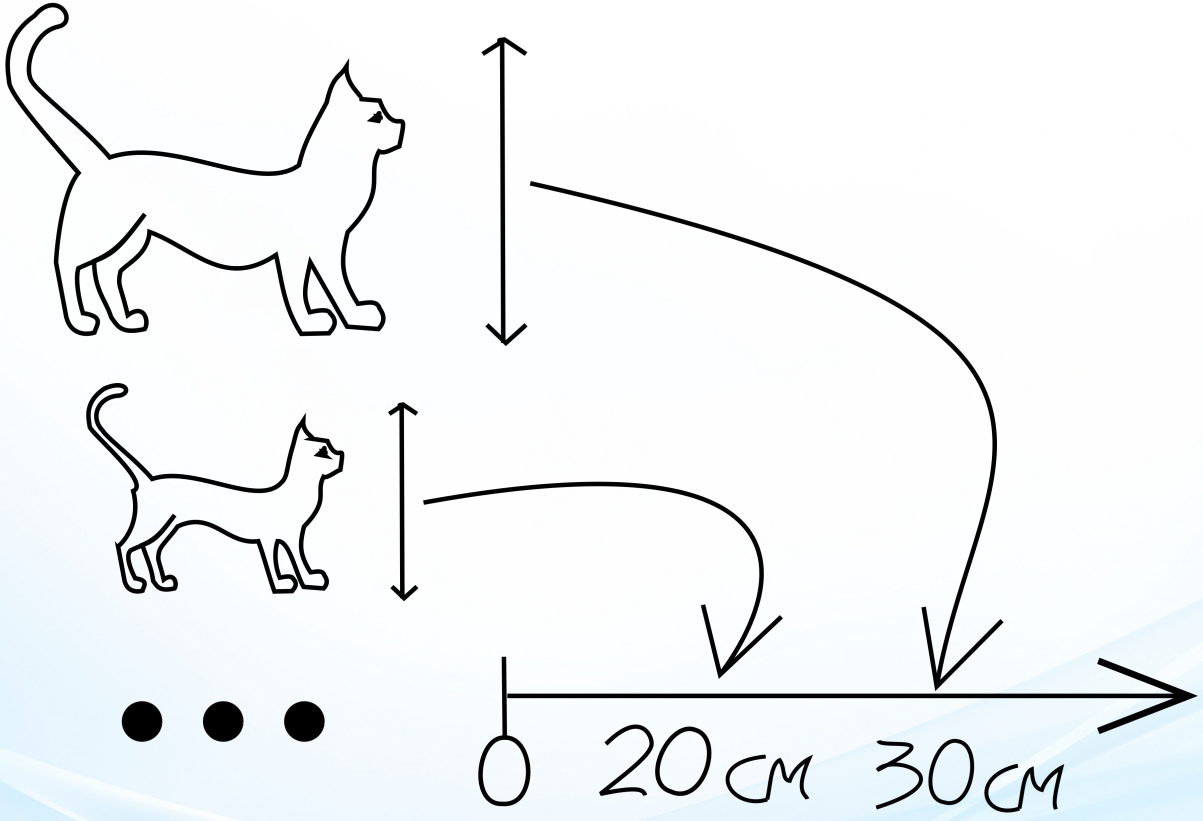
- *statistics* $\approx$ 2 010 000 статей
- *machine learning* $\approx$ 1 360 000 статей
- *artificial intelligence* $\approx$ 595 000 статей
- *neural network* $\approx$ 970 000 статей
- *computer vision* $\approx$ 854 000 статей

- *статистика* $\approx$ 60 200 статей
- *машинное обучение* $\approx$ 15 200 статей
- *искусственный интеллект* $\approx$ 16 600 статей
- *нейронные сети* $\approx$ 16 700 статей
- *компьютерное зрение* $\approx$ 12 300 статей

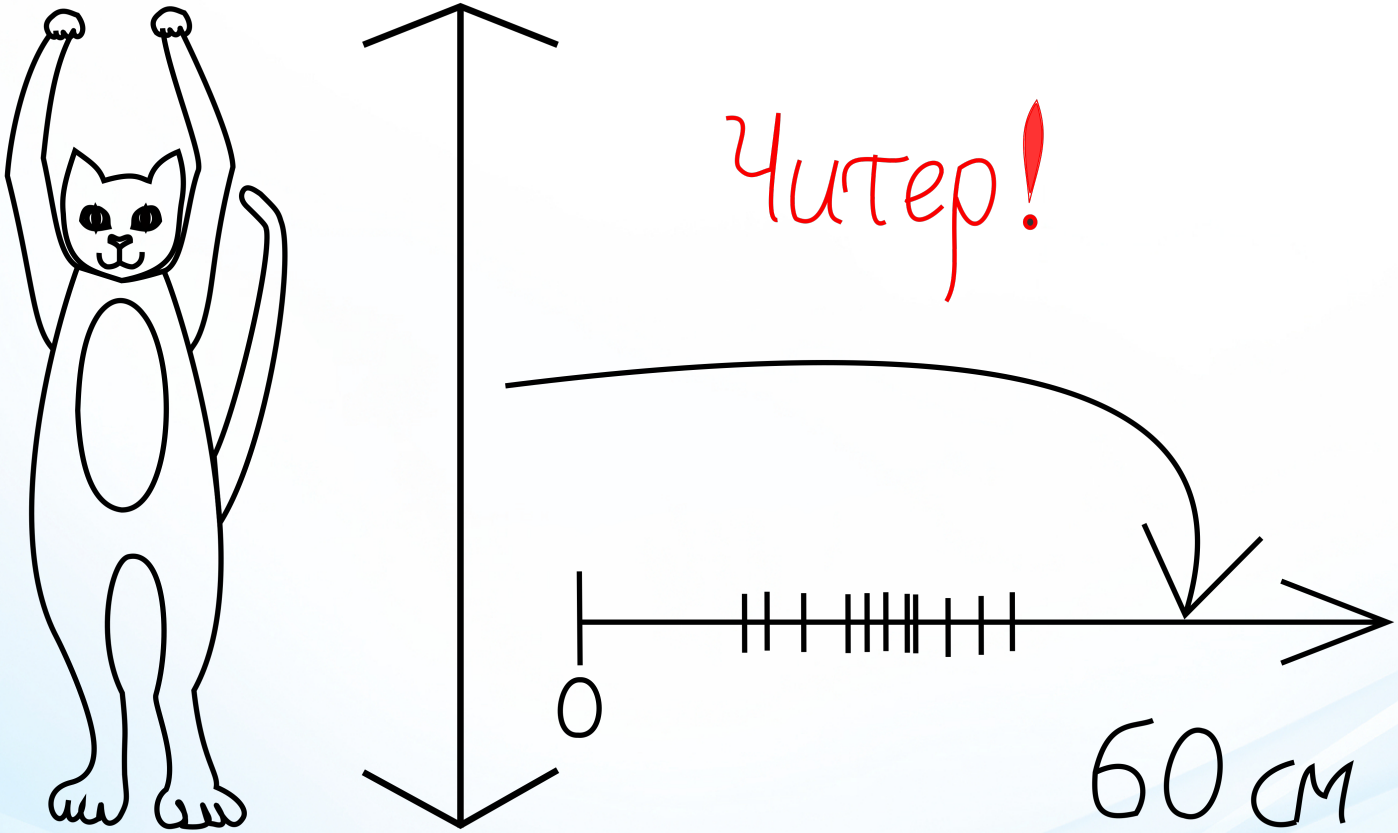
Обзор задач анализа данных

Мурмурландия

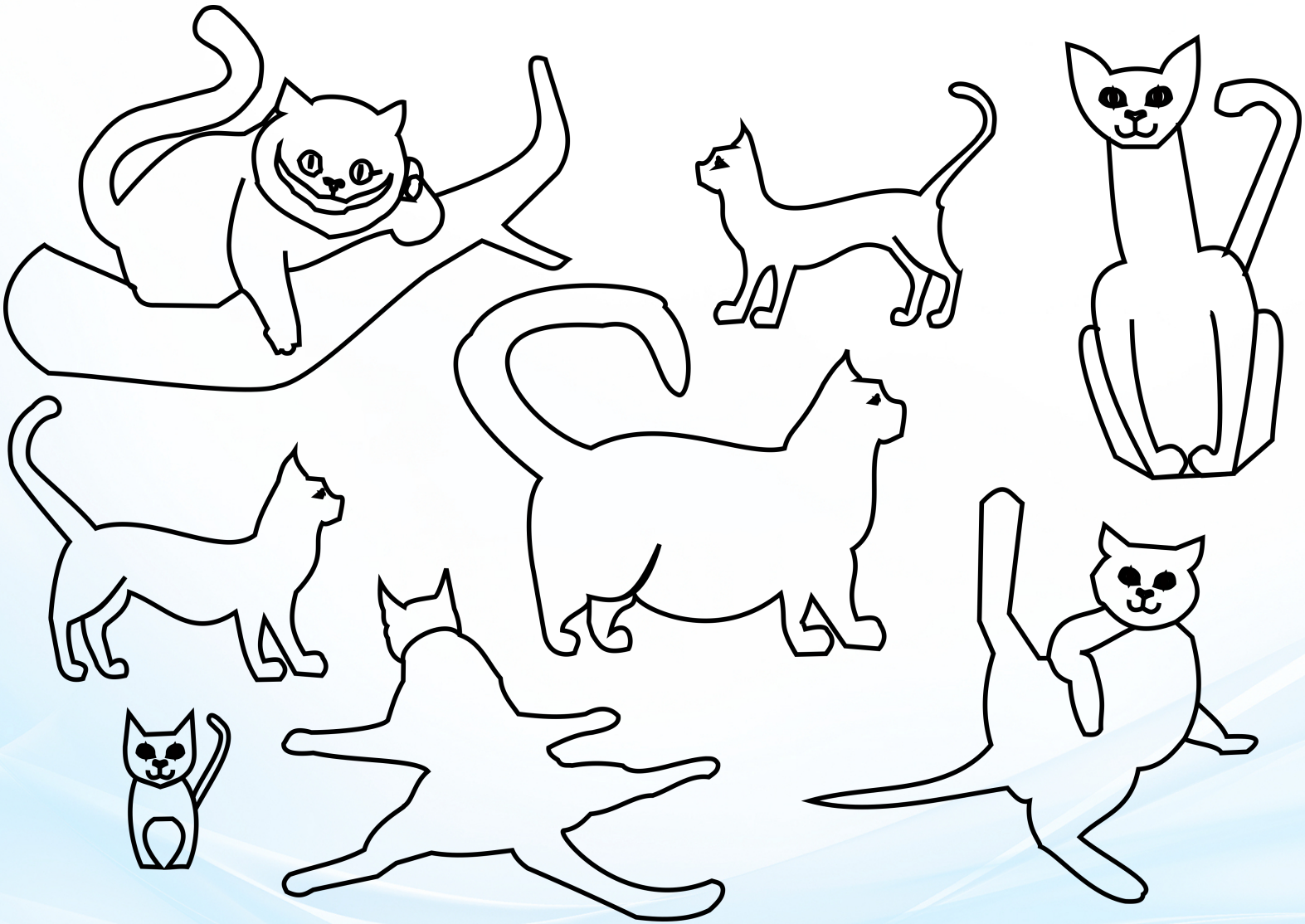
Каков средний рост котиков?



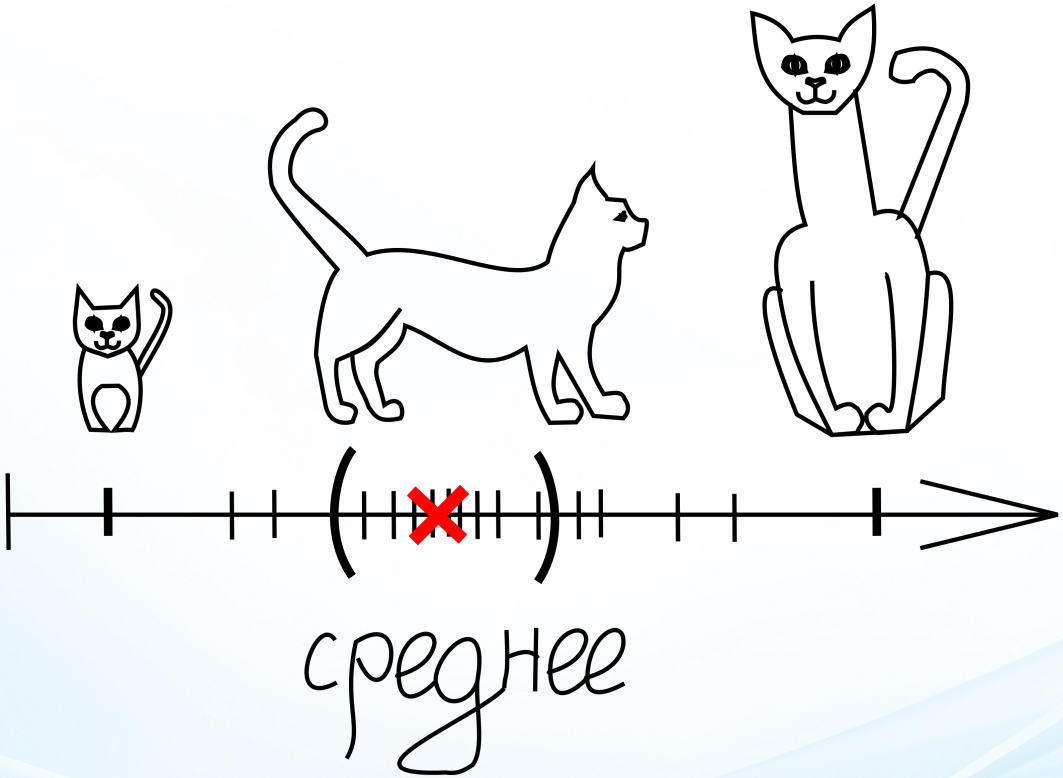
Точечное оценивание



Выбросы

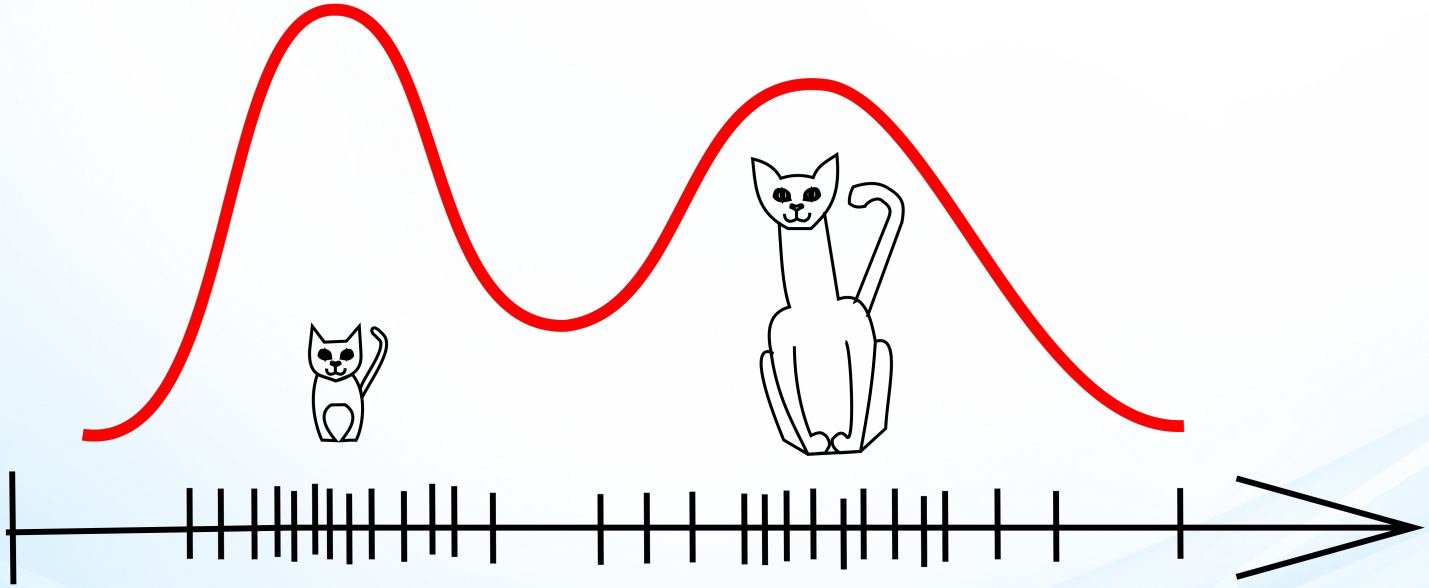


Среднее определяется неточно

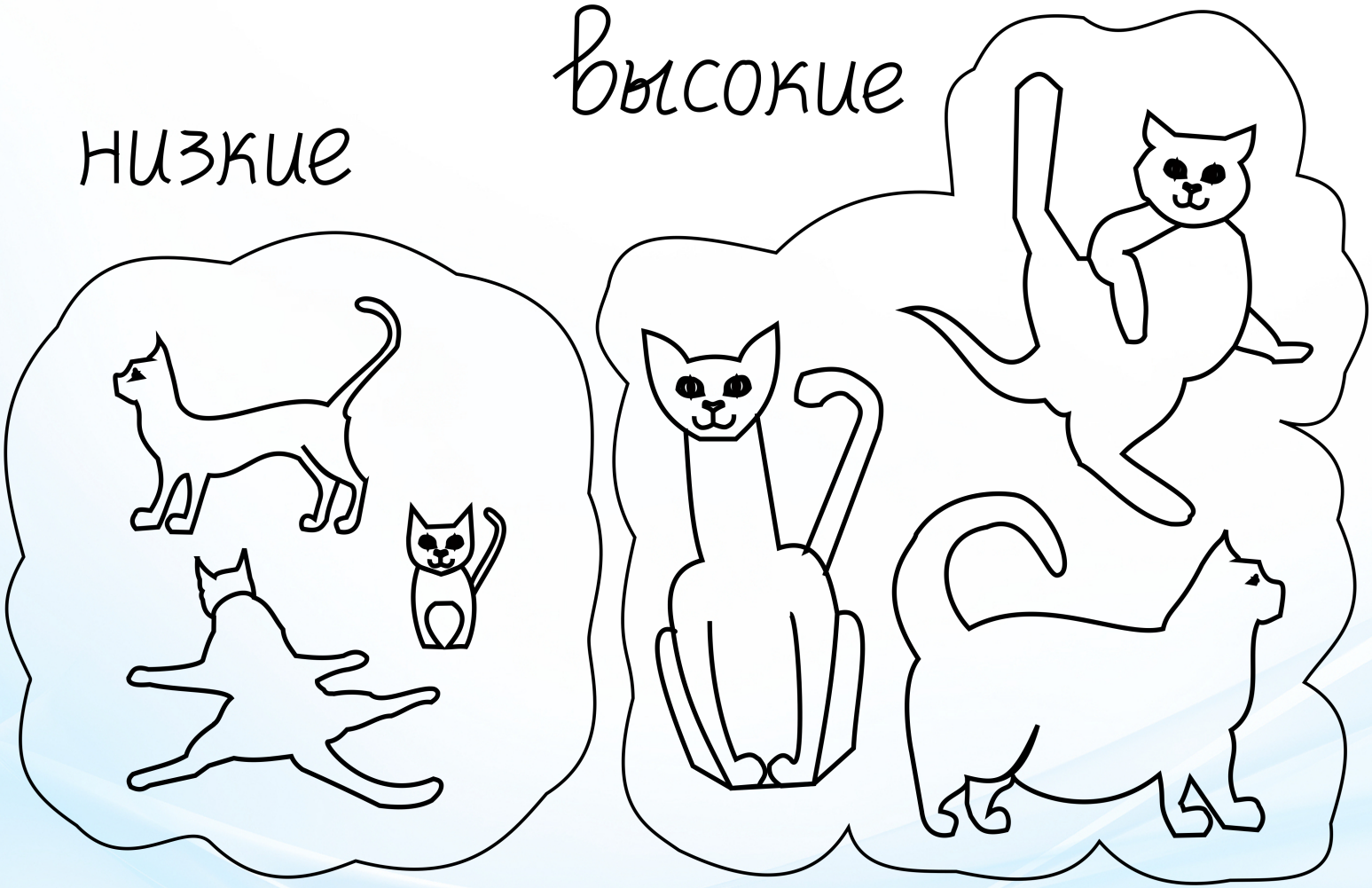


Интервальное оценивание

Характер распределения

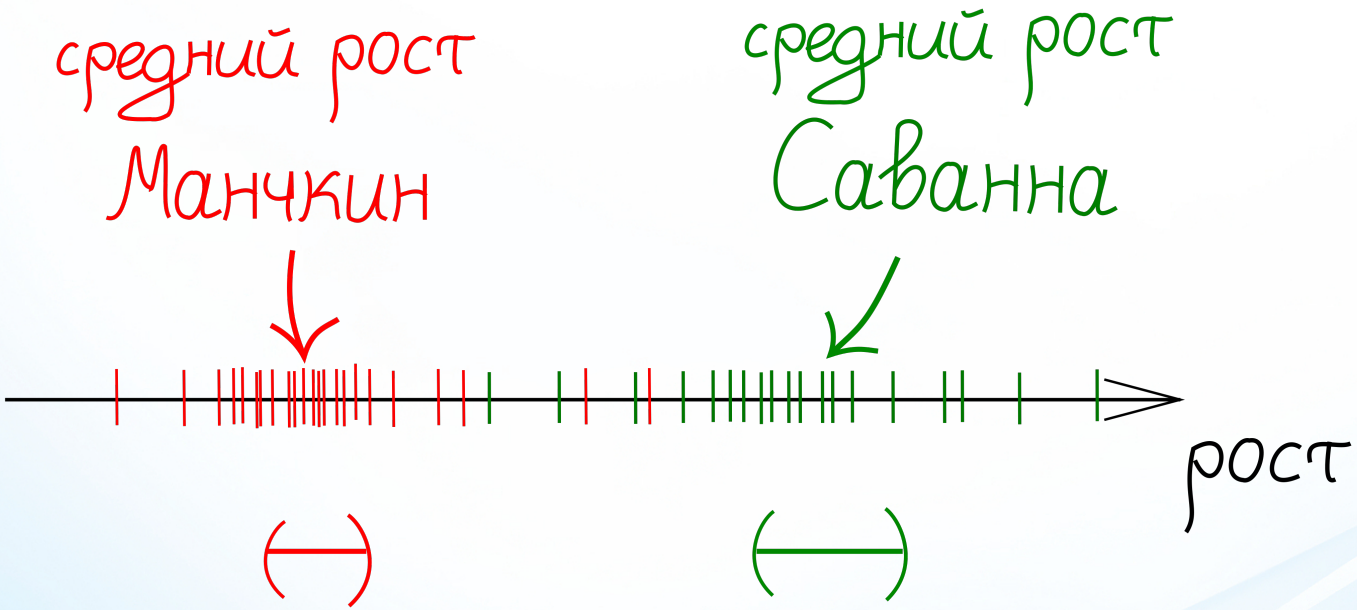


Непараметрическое оценивание


низкие
высокие

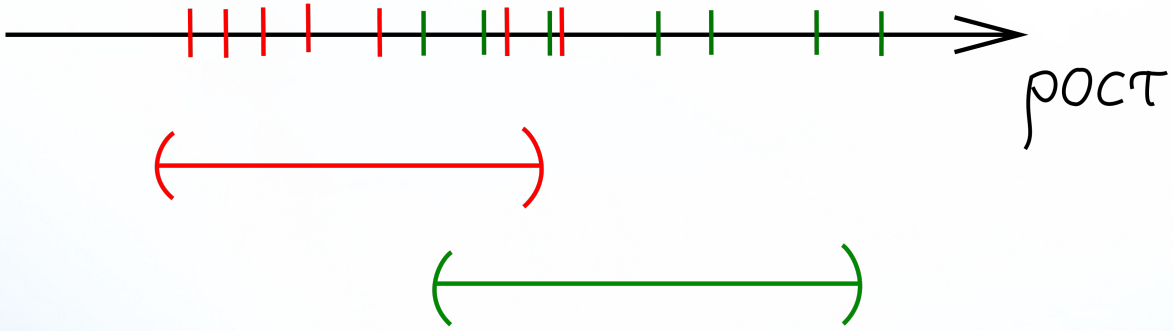
Отличается ли их средний рост?

Собираем данные

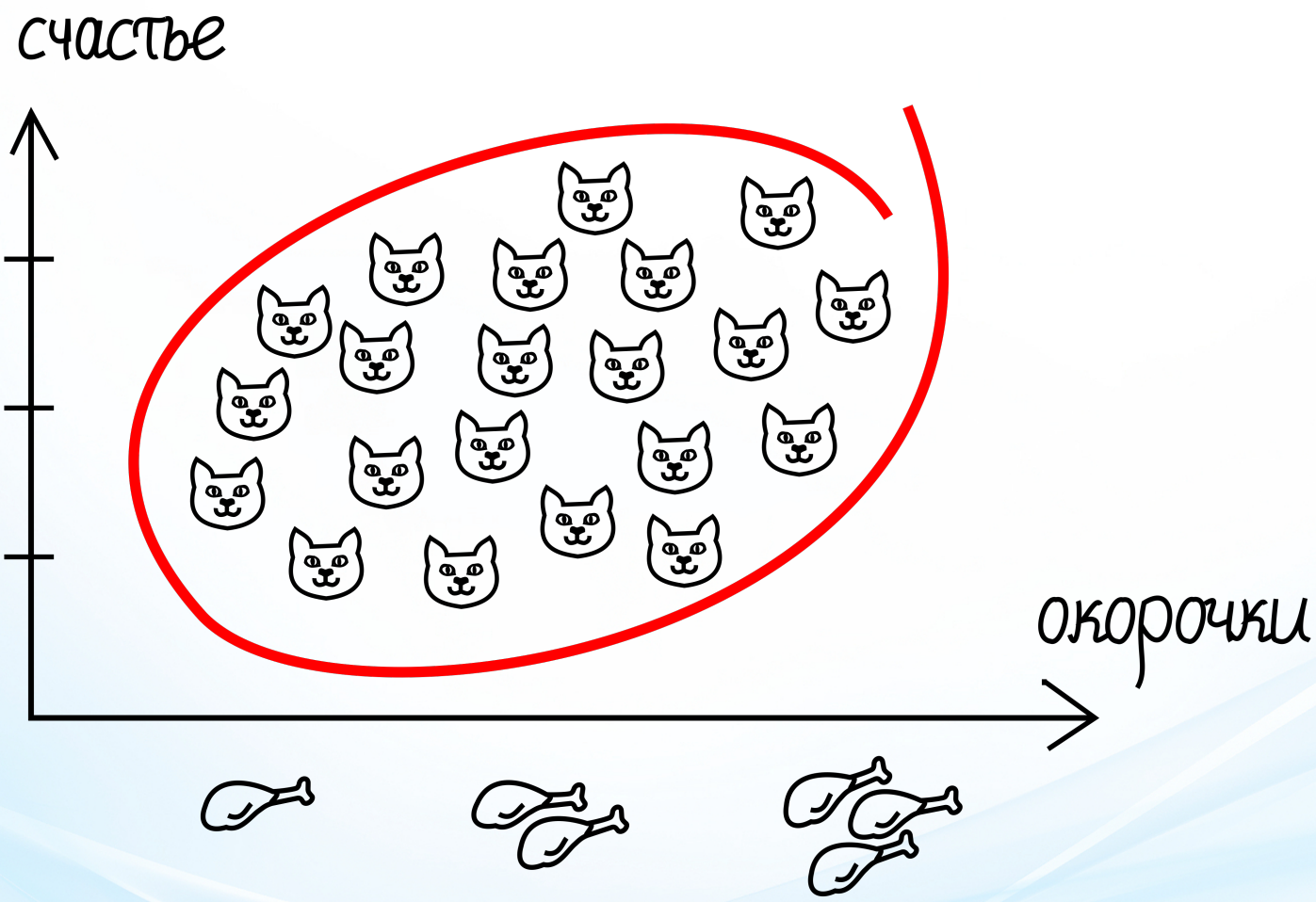


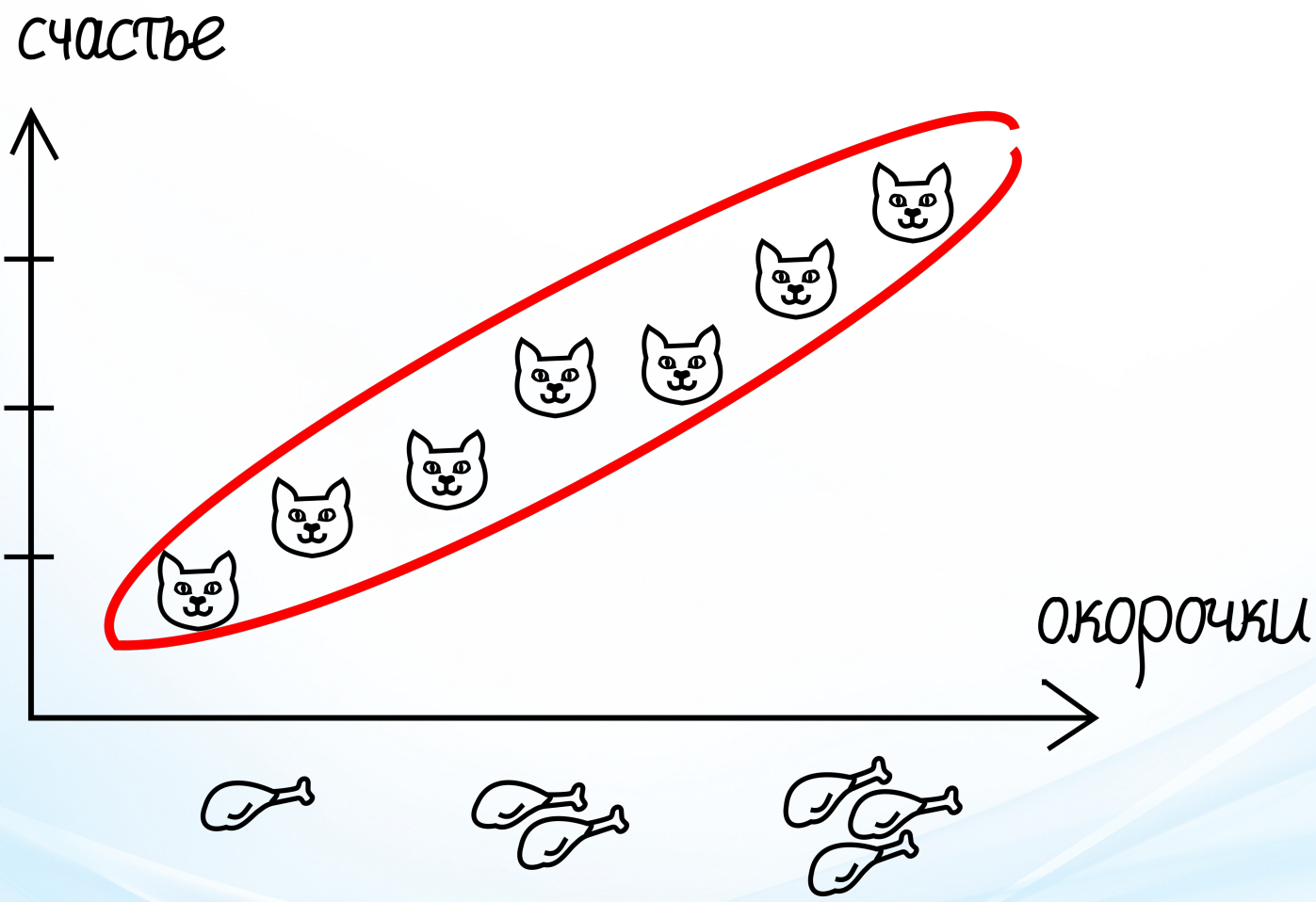
отличается

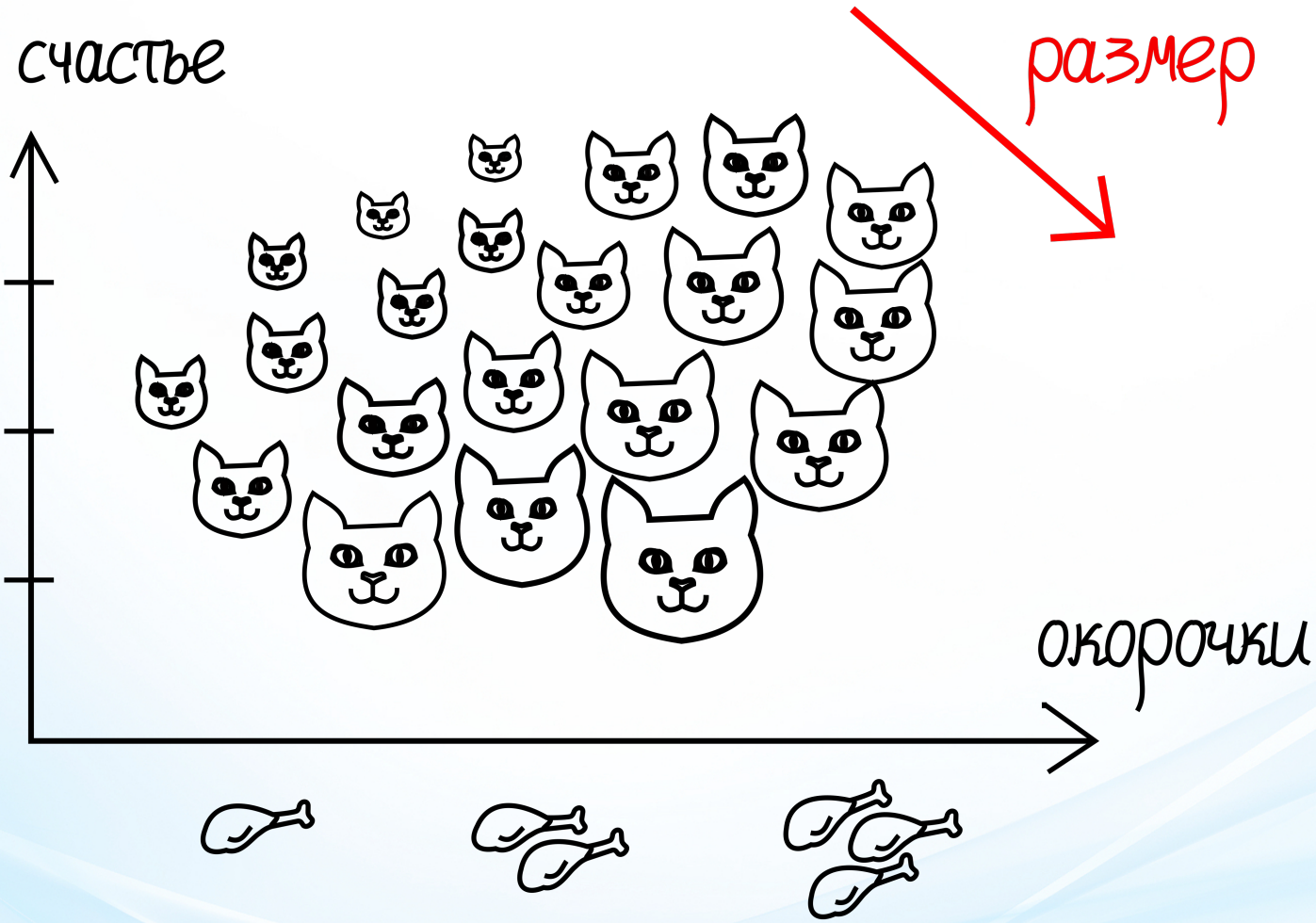
Если данных мало



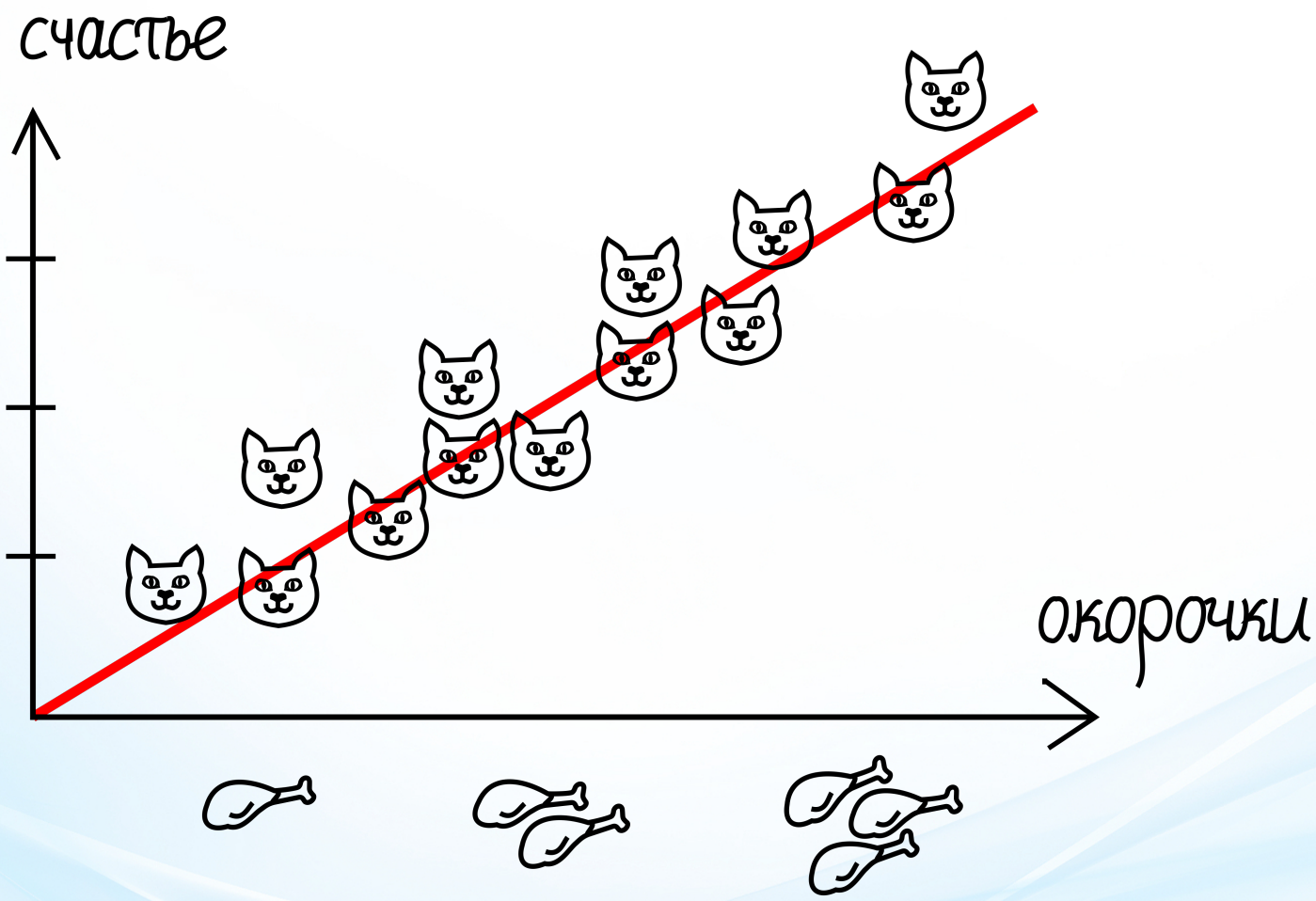
непонятно


счастье
окорочки

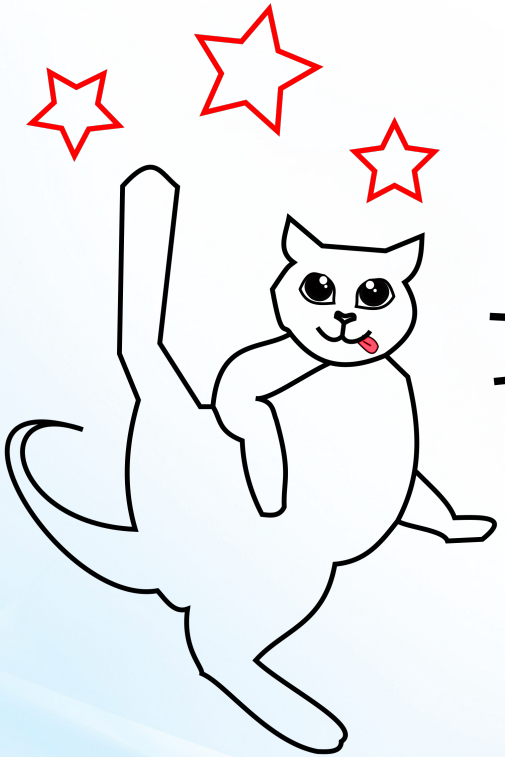

счастье
окорочки



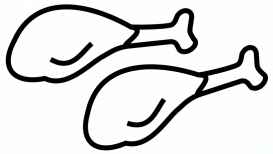
Корреляционный анализ


счастье
окорочки

Формула счастья

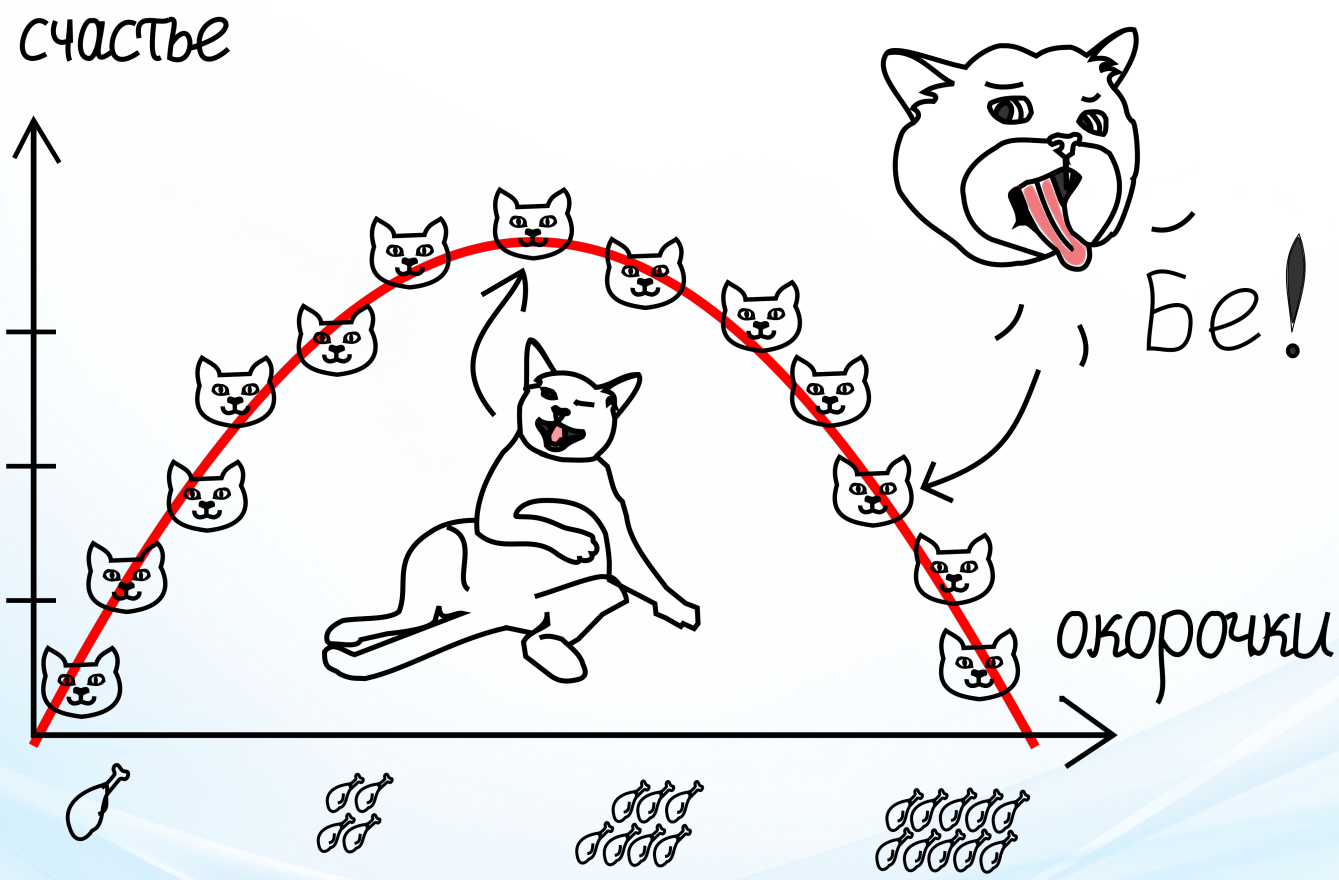


$$= \theta_0 + \theta_1 \times \text{кол-во}$$

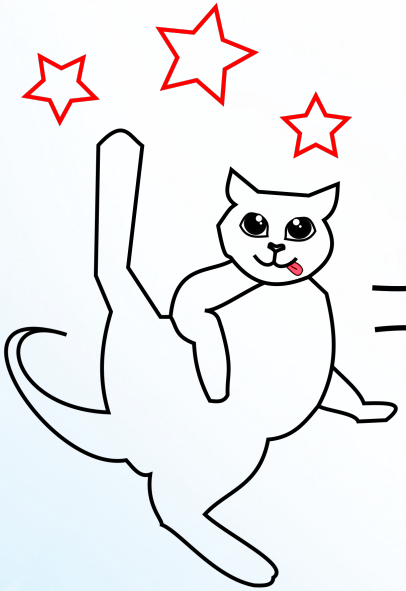


$$+ \text{ погрешности}$$

Больше окорочков



Формула счастья



$$= \Theta_0 + \Theta_1 \times \overset{\text{кол-во}}{\text{🐟}} - \Theta_2 \times \left(\overset{\text{кол-во}}{\text{🐟}}\right)^2$$

$$+ \text{ погрешности}$$

Другие факторы

$$= \theta_0 + \theta_1 \times \text{🍗} - \theta_2 \times (\text{🍗})^2$$

$$+ \theta_3 \times \text{🧶}$$

$$+ \theta_4 \times \text{🛋}$$

$$+ \text{погрешности}$$

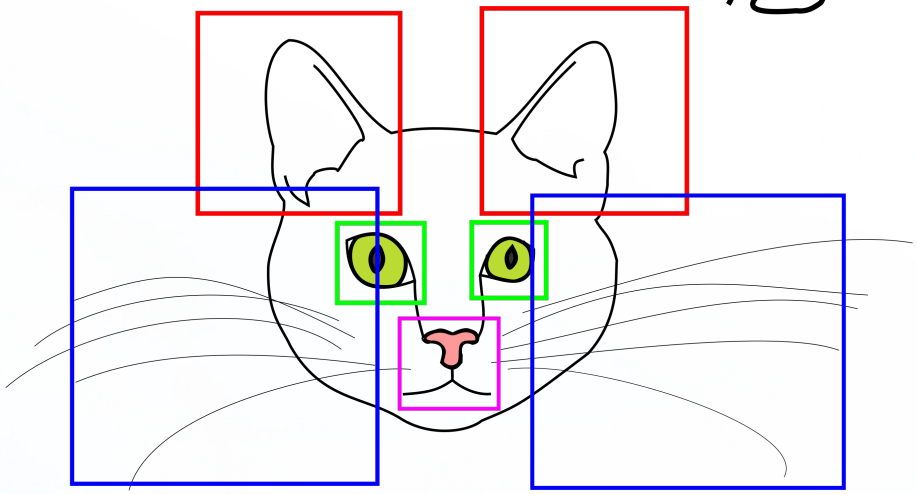
Классификация котиков

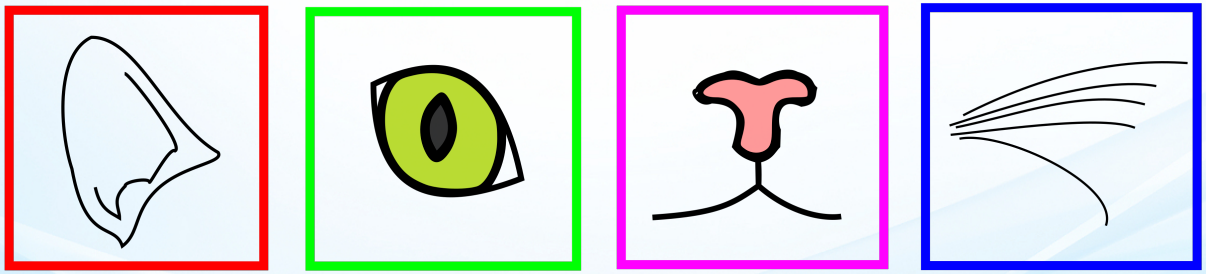
Классификация

Собираем данные

котик	порода	рост	шерсть
	Саванна	50 см	да
	Сфинкс	30 см	нет
	Манчкин	15 см	да
	Саванна	40 см	да

Распознавание мордочек



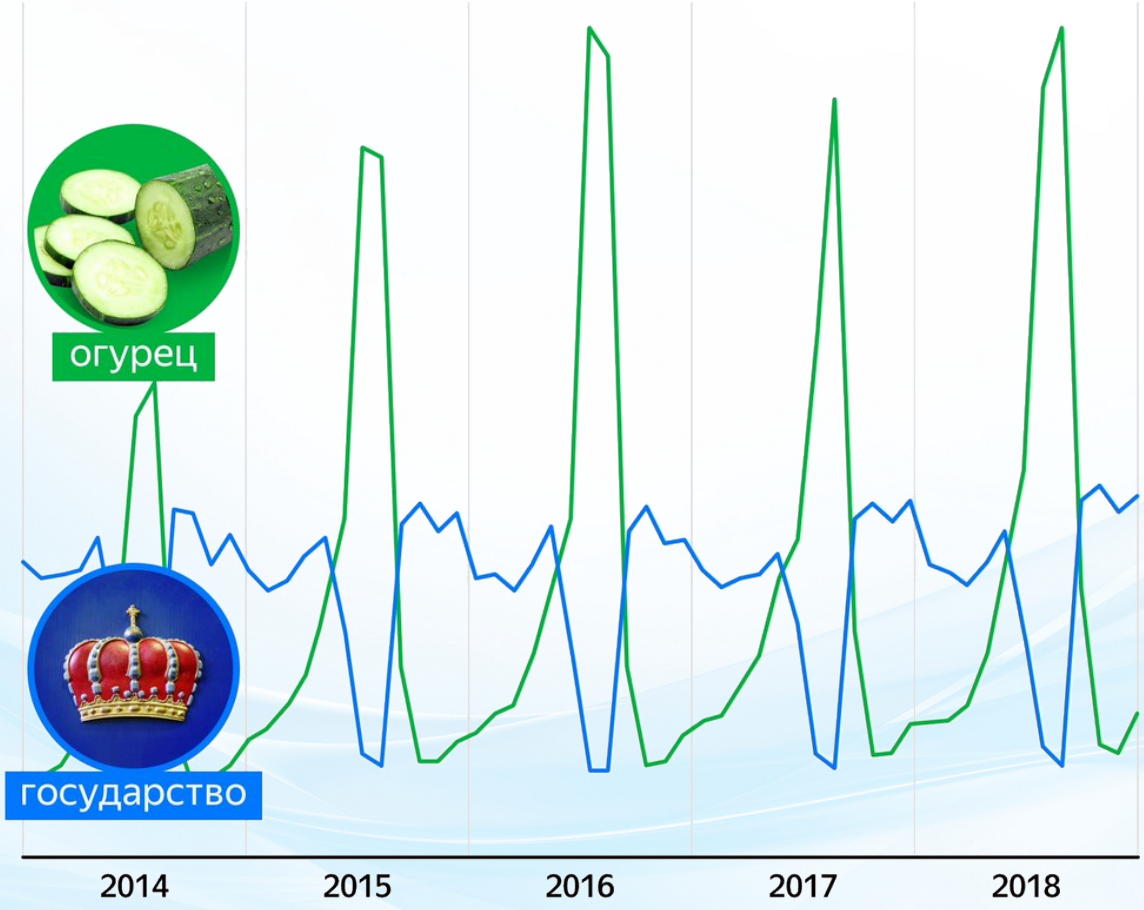


Нейронные сети

Художник курса

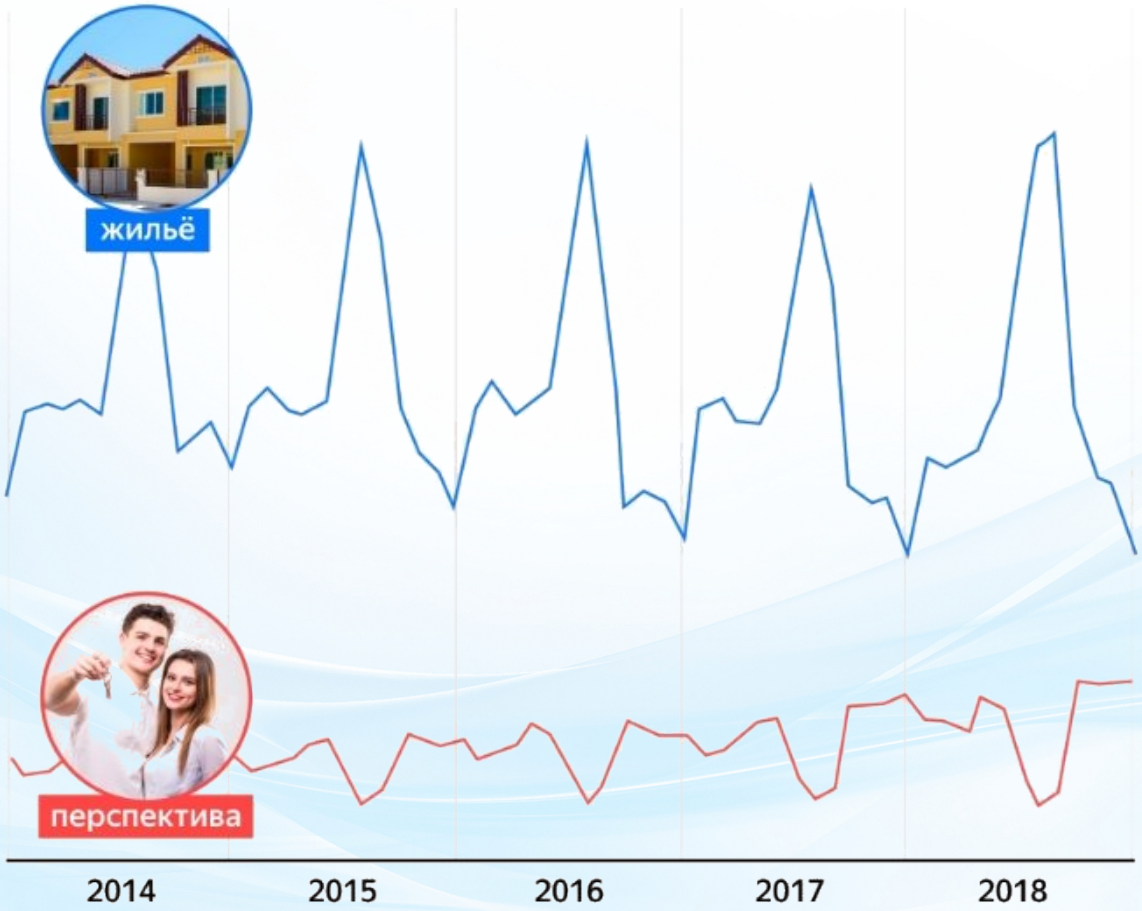
Книга с похожим содержанием

Когда в Поиске растёт интерес к огурцам, снижается доля запросов со словом государство



Когда в Поиске растёт интерес к тату, снижается доля запросов со словом смысл

Когда в Поиске растёт интерес к жилью, снижается доля запросов со словом перспектива



Президентские выборы в США в 1948 г.

Гарри Труман (демократы) vs. Томас Дьюи (республиканцы)

В ночь на оглашение результатов газета Chicago Tribune опубликовала заголовок: **DEWEY DEFEATS TRUMAN**

Chicago Daily Tribune

THE WORLD'S GREATEST NEWSPAPER

HOME

WEDNESDAY, NOVEMBER 3, 1948

DEWEY DEFEATS TRUMAN

G.O.P. Sweep Indicated in State; Boyle Leads in City

REPUBLICAN TICKET AHEAD OF 1944 VOTE

Town Balloting Gives Trend

Tops Coghlan in Hot Race for Attorney

Probable Winners

RECORD CITY VOTE SEEN IN LATE TALLIES

Suburban Ballot Near 575,000

BULLETINS ON ELECTIONS

Early Count Gives G.O.P. Senate Edge

PUTS G.O.P. BACK IN THE WHITE HOUSE

Sizable Electoral Margin Seen

После закрытия участков газета провела опрос, обзвонив большое число избирателей, все предвещало оглушительную победу Дьюи.

Президентские выборы в США в 1948 г.

Смеющийся Труман, победитель выборов 1948 года.

Что же пошло не так?

В 1948 году телефон был доступен только людям определенного достатка и редко встречался у людей с небольшим заработком.
Выборка не учитывала достаточно широкий пласт избирателей Трумана, т.к. как правило демократы имеют большую долю голосов среди бедного населения, которым телефон в свою очередь был недоступен.

Примеры прикладных задач

Рекомендательная система

Продуктовая аналитика

Синтез речи

Где еще?

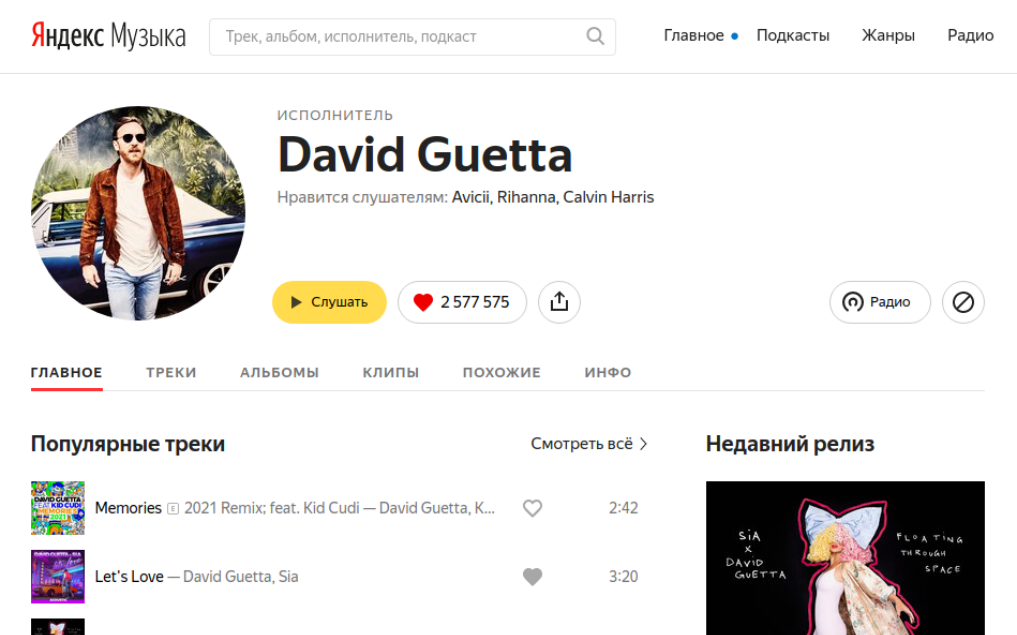
Рекомендательная система

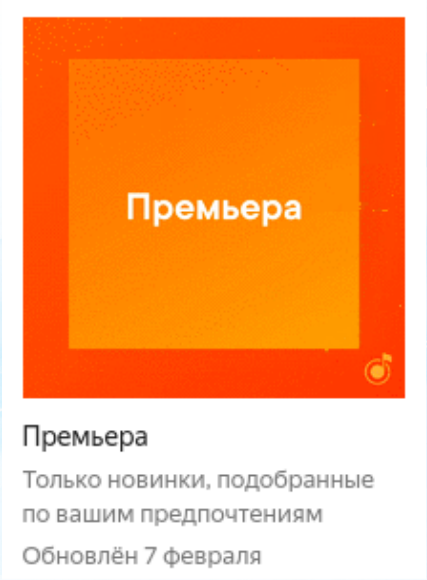
Постановка задачи:

Спроектировать рекомендательную систему для муз. сайта. Система должна рекомендовать музыку каждому пользователю в соответствии с его предпочтениями.

Данные:

- оценки пользователей различным трекам;
- прослушивание треков пользователями.





Рекомендательная система

Пусть U — множество треков, I — множество пользователей.

$R = (r_{ui})_{u \in U, i \in I}$ — матрица лайков/прослуш. пользователями треков.

Идея: T — небольшое множество интересов.

$P = (p_{tu})_{t \in T, u \in U}$ — матрица интересов пользователей.

$Q = (q_{ti})_{t \in T, i \in I}$ — матрица соответствия треков интересам.

Если величины p_{tu} и q_{ti} неотрицательны, то их можно нормировать по темам, получив вероятности:

- $p_{tu} \big/ \sum_{t \in T} p_{tu} = \mathrm{P}(\text{пользователю } u \text{ интересно } t)$,
- $q_{ti} \big/ \sum_{t \in T} q_{ti} = \mathrm{P}(\text{трек } i \text{ соответствует } t)$.

Рекомендательная система

Неотрицательные матричные разложения

Решаем поочередно для каждой t.

$$\begin{cases} \left\| R_t - p_t^T q_t \right\|^2 \longrightarrow \min\limits_{p_t} \\ p_t \geqslant 0 \end{cases} \quad \Longrightarrow \quad p_t = \left(\frac{q_t R_t^T}{q_t q_t^T} \right)^+$$

$$\begin{cases} \left\| R_t - p_t^T q_t \right\|^2 \longrightarrow \min\limits_{q_t} \\ q_t \geqslant 0 \end{cases} \quad \Longrightarrow \quad q_t = \left(\frac{p_t R_t^T}{p_t p_t^T} \right)^+$$

Что нужно помнить при реализации

- Пользователей и треков миллионы;
- Матрица R сильно разрежена — лайков очень малая доля.

Примеры прикладных задач

Рекомендательная система

Продуктовая аналитика

Синтез речи

Где еще?

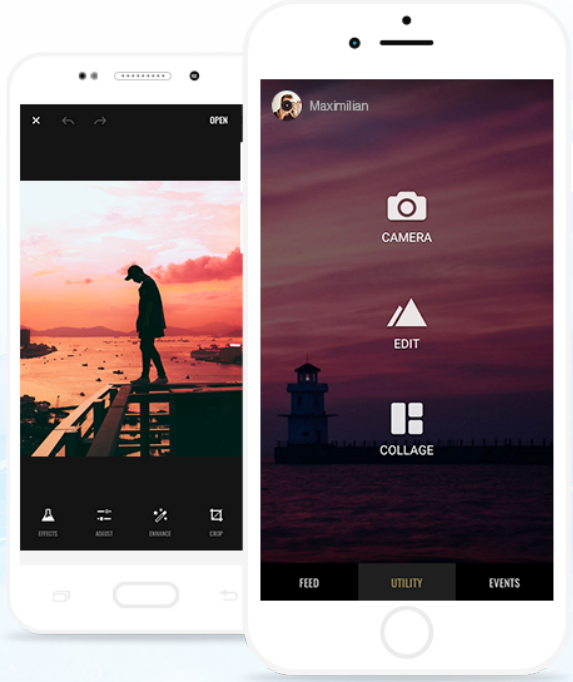
Продуктовая аналитика

Представьте, что вы продуктовый аналитик в фоторедакторе.

Вы всегда хотите, чтобы продукт приносил больше денег.

Реальный вопрос:

пользователи какой страны платят больше?



Продуктовая аналитика

Постановка задачи:

Оказывает ли страна пользователя влияние на его LifeTime Value?

LifeTime Value — доход, который принес пользователь

за все время жизни в продукте.

Данные:

- страна пользователя — фактор;
- LifeTime Value каждого пользователя.

Продуктовая аналитика

Решение задачи:

Однофакторный дисперсионный анализ и Post-Hoc анализ.

Данные методы помогают понять какие значения фактора влияют на исследуемую величину и как именно.

Т.е., можно узнать, пользователи каких стран платят больше.

Как же выглядят эти методы?

Однофакторный дисперсионный анализ

Независимые выборки

1	2	...	k
X_{11}	X_{12}	...	X_{1d}
X_{21}	X_{22}	...	X_{2d}
...	...	...	...
$X_{n_1 1}$	$X_{n_2 2}$	...	$X_{n_k k}$

$$X_{ij} = \overbrace{\mu + \beta_j}^{\mu_j} + \varepsilon_{ij},$$

$i = 1, ..., n_j$ — номер наблюдения в выборке

$j = 1, ..., k$ — номер выборки

μ — неизвестное общее среднее

β_j — неизвестный эффект воздействия фактора для j-й выборки

ε_{ij} — случайная ошибка

Предположение:

ε_{ij} независимы и имеют одинаковое непрерывное распределение.

$H_0: \mu_1 = ... = \mu_k$ *vs.* $H_1: \exists j_1, j_2$ т.ч. $\mu_{j_1} \neq \mu_{j_2}$

Оценка контраста

Пусть H_{rs} отвергается $\Longrightarrow$ оцениваем контраст $\Delta_{rs} = \mu_r - \mu_s$.

$V_{rs} = med\{X_{ir} - X_{ls}, i = 1..n_r, j = 1..n_s\}$ — первичная оценка

$W_r = \frac{1}{N} \sum\limits_{s=1}^{k} n_s V_{rs}$, где $V_{rr} = 0$

$\widehat{\Delta} = W_r - W_s$ — уточненная оценка контраста

Свойства:

1. Первичные оценки могут быть несогласованными: $V_{12} \neq V_{13} + V_{32}$
2. Уточненные оценки согласованы и состоятельны.
3. Уточненные оценки зависят от всех выборок.

Для применения данных методов нужно знать

теорию вероятностей и **математическую статистику**.

Результат

Теперь мы знаем, в какой стране завести маркетинговую кампанию!

Мы принесли компании деньги, мы молодцы!

Примеры прикладных задач

Рекомендательная система

Продуктовая аналитика

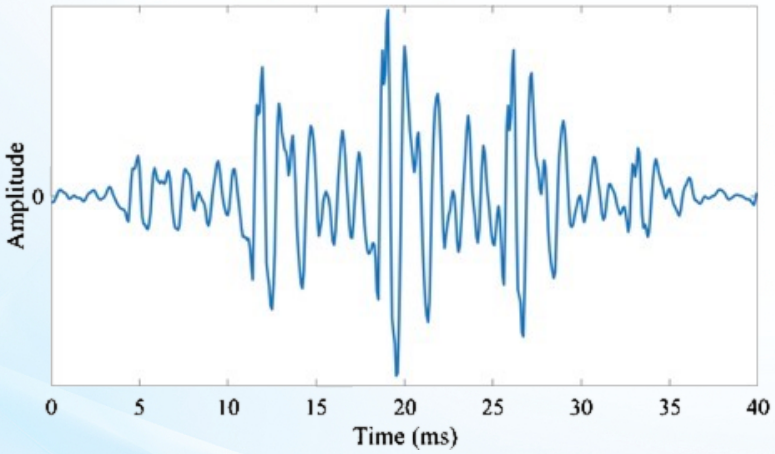
Синтез речи

Где еще?

Синтез речи

Что такое звук?

Звук



Волна

Звук — композиция волн с разными амплитудами и частотой.

Волна — периодич. ф-ия, имеющая амплитуду, период и частоту.

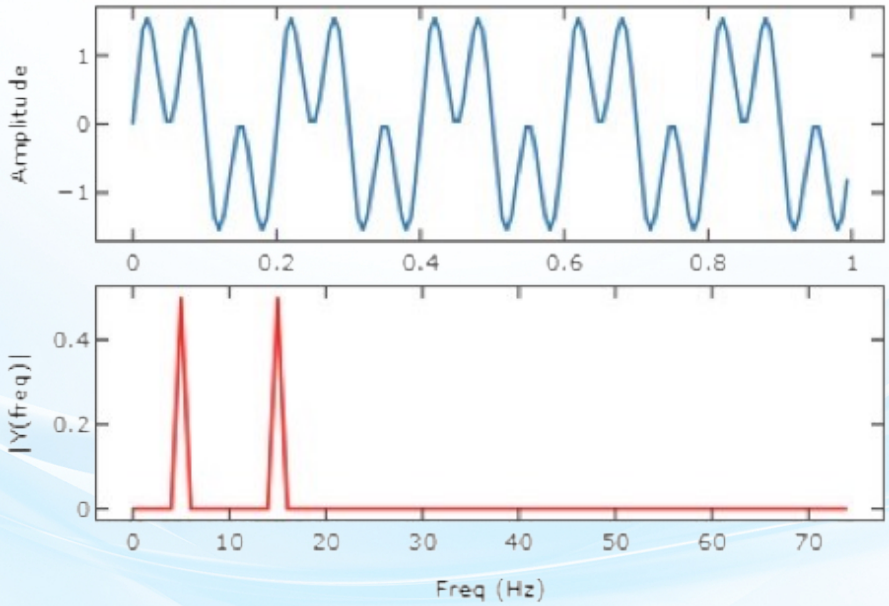
Синтез речи

Как работать со звуком?

Посмотрим на распределение частот волн в звуке.

Для этого применяется дискретное преобразование Фурье:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N}kn\right) - i\sin\left(\frac{2\pi}{N}kn\right)\right]$$



Пример для звука из двух волн

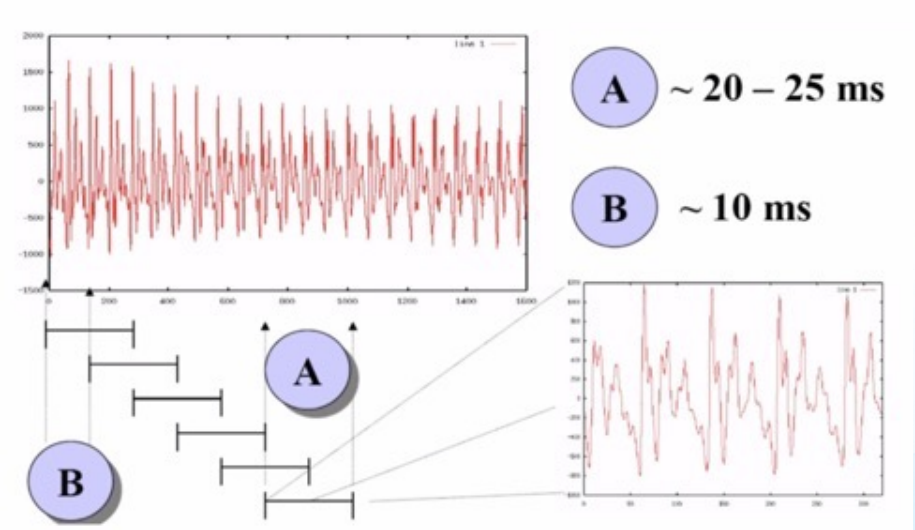
Синтез речи

Как работать со звуком?

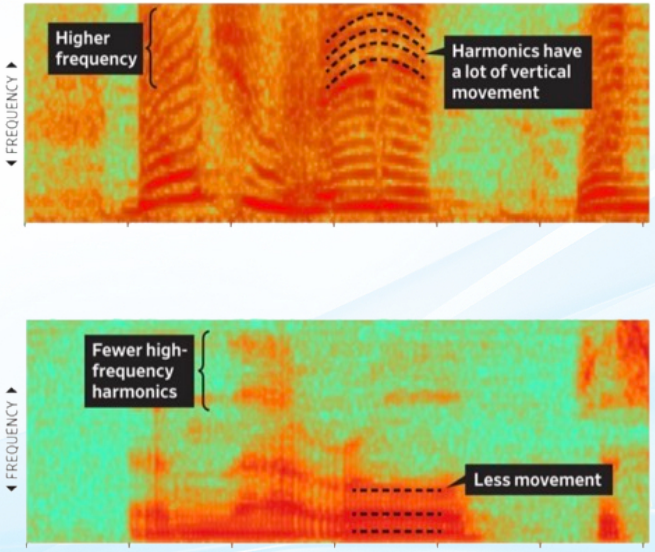
У результата преобразования Фурье ко всем данным нет времени.

Посчитаем распределение на окнах из звука.

Окна



Спектограмма



- Берем маленькие окна (20-25 мс) от исходных данных.
- К каждому окну применяем преобразования Фурье.
- Стакаем распределения частот вместе, получаем спектрограмму.

Синтез речи

Постановка задачи синтеза речи

Для введенного получить соответствующее аудио.

Другие задачи, связанные со звуком:

- Определить, была ли речь на записи.
- Споттер — распознавание определенной фразы или определенного события.
  Например, “Ok, Google”, “Алиса” или определение выстрела.
- Идентификация говорящего или его признаков.
  Например, мужчина/женщина, ребенок/взрослый.
- Распознавание речи
  По аудио вернуть полную его расшифровку в виду текста.

Синтез речи

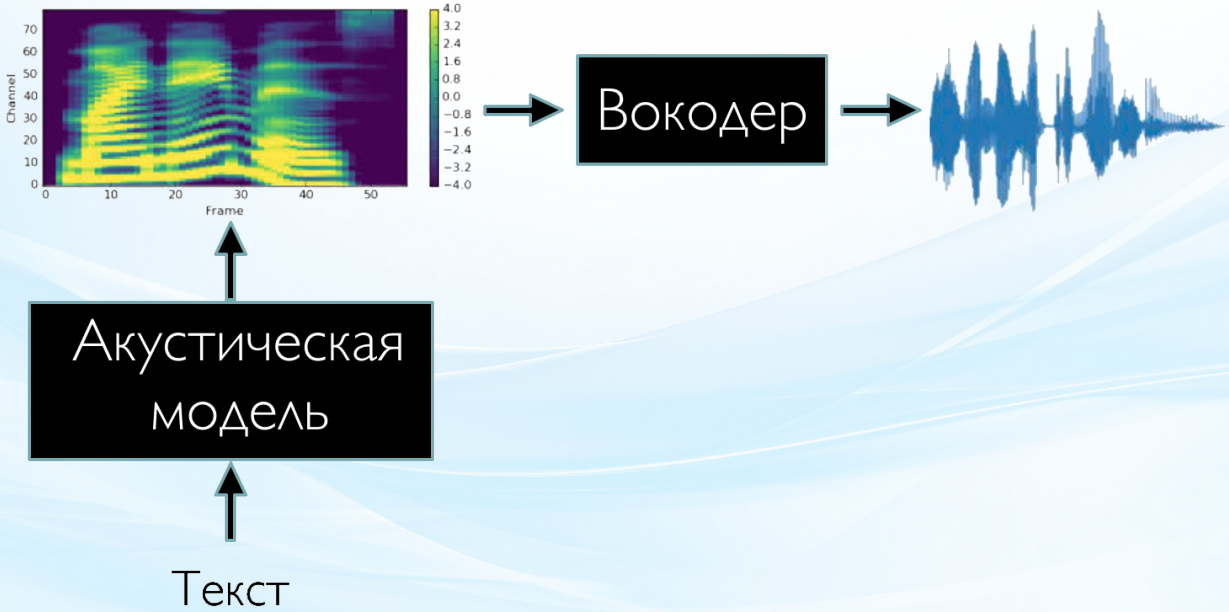
1. Акустическая модель по тексту предсказывает мел-спектограмму.

   Акустическая модель — это какая-то нейросеть.

2. Вокодер по мел-спектограмме предсказывает аудио.

   Вокодер может быть как алгоритмом, так и нейросетью.

   Некоторые известные вокодеры используют байесовские методы.



Примеры прикладных задач

Рекомендательная система

Продуктовая аналитика

Синтез речи

Где еще?

- Беспилотные автомобили
- Ранжирование результатов в поисковой системе
- Поиск по картинкам и видео на основе содержания
- Распознавание спама/фрода
- Прогноз погоды
- Прокладывание маршрутов в навигаторах
- AI-камеры в телефонах
- Генерация картин подобно художникам
- Оплата проезда в метро взглядом
- Решение о выдаче кредита
- Персонализированная реклама
- Определение причин оттока клиентов
- Прогнозирование спроса на товар
- Определение месторасположения новой торговой точки
- Расстановка полок в магазинах
- Автоопределение свежести товаров в магазинах
- Распознавание патологий на медицинских снимках
- Персонализированная медицина
- Прогнозирование поломок оборудования
- Автоматическая система оптимального управления оборудованием


ВСЁ!