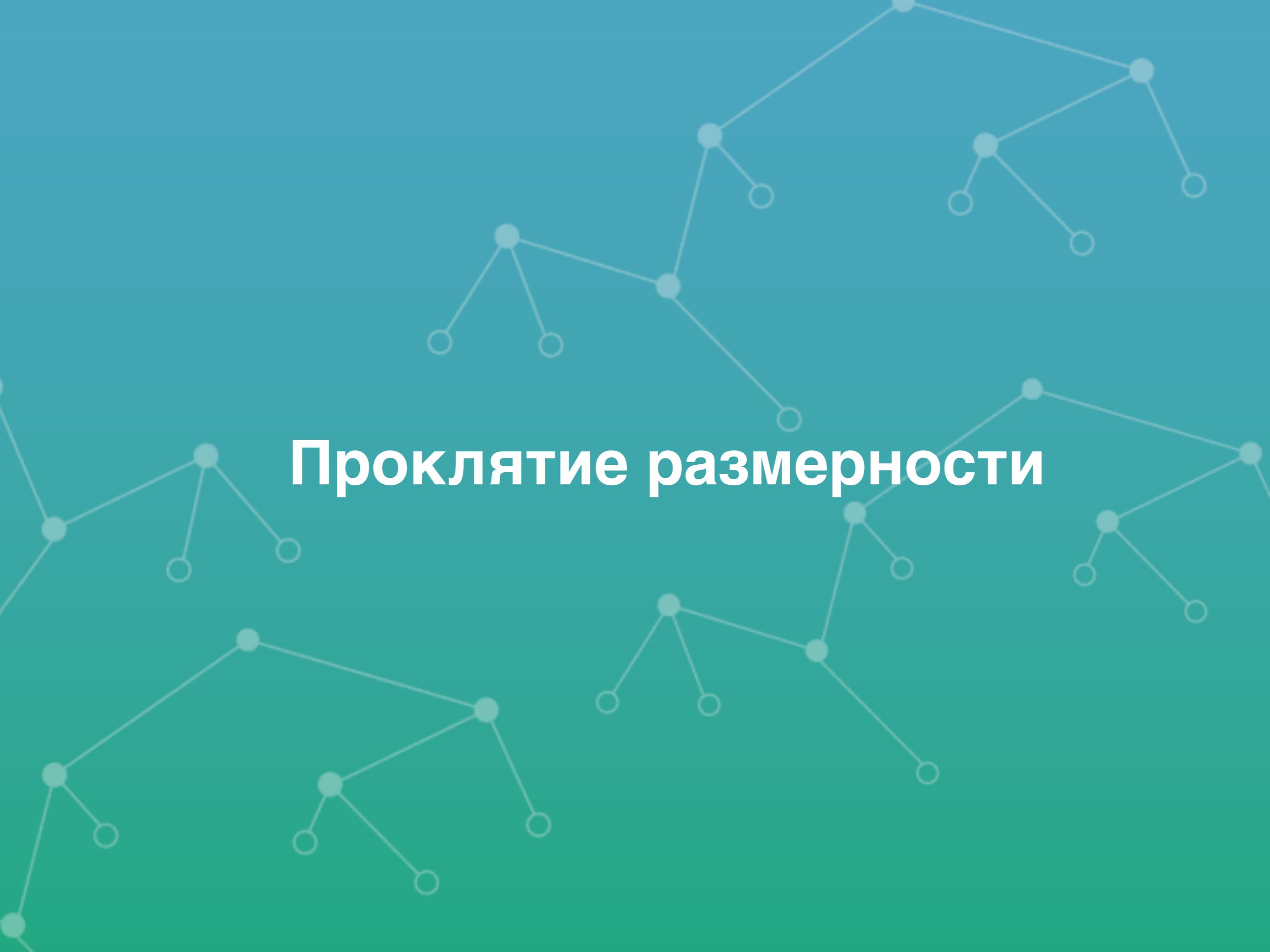




Методы снижения размерности

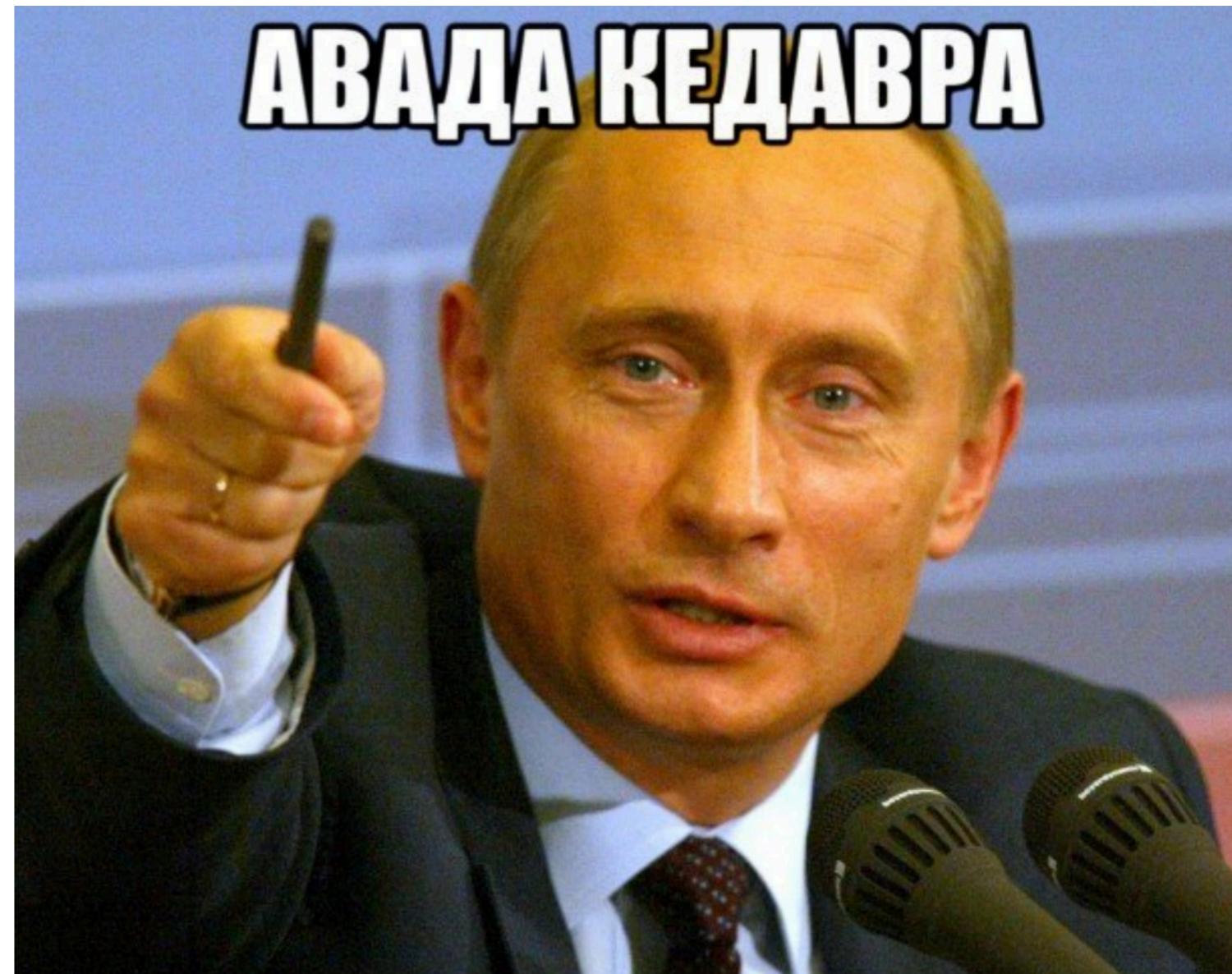
Парполов Дмитрий

Проклятие размерности



Проклятие размерности

Проблема, связанная с
экспоненциальным
возрастанием количества
данных из-за увеличения
размерности пространства

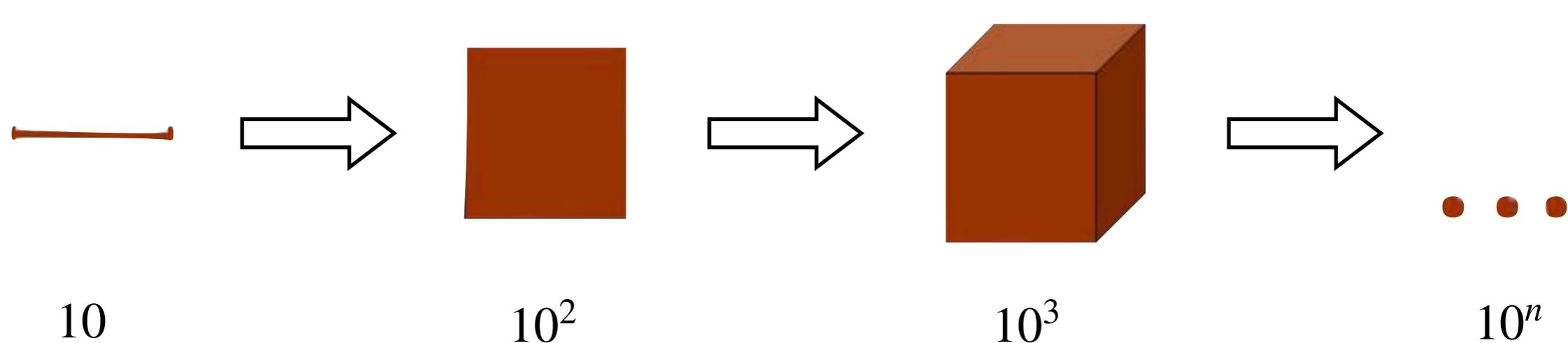


Пример



При увеличении размерности пространства для равномерного покрытия многомерного пространства требуется
экспоненциальное увеличение количества данных

Хотим равномерно покрыть n -мерный единичный куб точками с частотой не менее, чем 0,1



Почему это проблема



$$x_1 = (a_1, a_2, \dots, a_n)$$

$$x_2 = (a_1, a_2 + \Delta, \dots, a_N)$$

$$x_3 = (a_1 + \epsilon, a_2 + \epsilon, \dots, a_N + \epsilon), \epsilon \ll \Delta$$

В высокомерном пространстве $\rho(x_1, x_2) \approx \rho(x_1, x_3)$

То есть небольшие изменения ϵ в большом количестве координат вызывают сравнимые изменения в расстоянии, как и значительное изменение Δ в 1 координате

Расстояния во всех парах объектов стремятся (согласно ЗБЧ) к одному и тому же значению и становятся неинформативными - *проблема для метрических классификаторов*



Почему это проблема



... а также

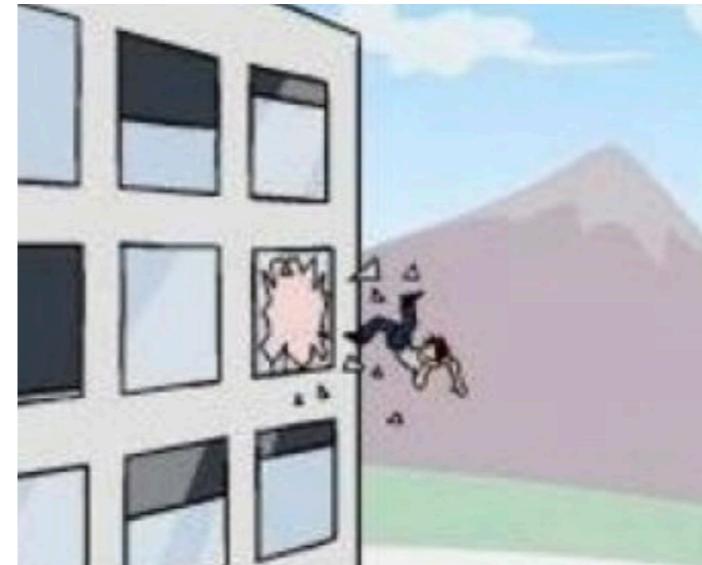
- повышается трудоемкость вычислений
- необходимость хранения большого количества данных
- увеличение доли шумов
- в линейных классификаторах приводит к мультиколлинеарности и переобучению

Как устраниТЬ

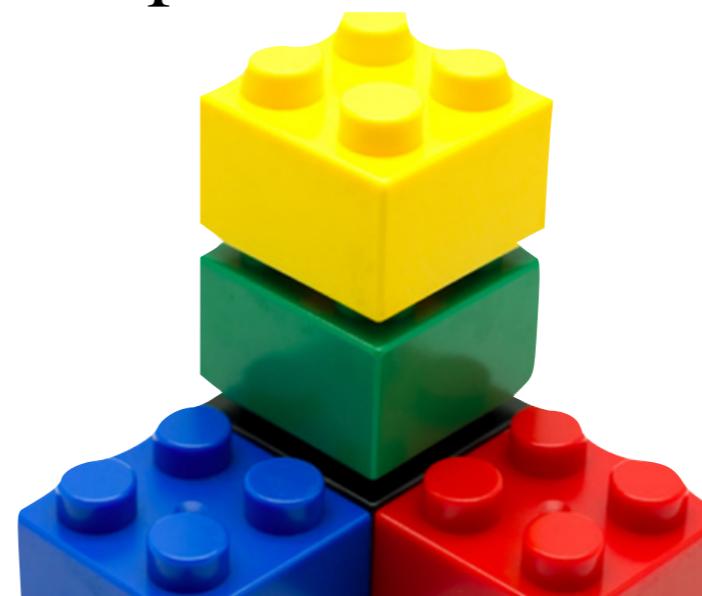


Уменьшить количество признаков

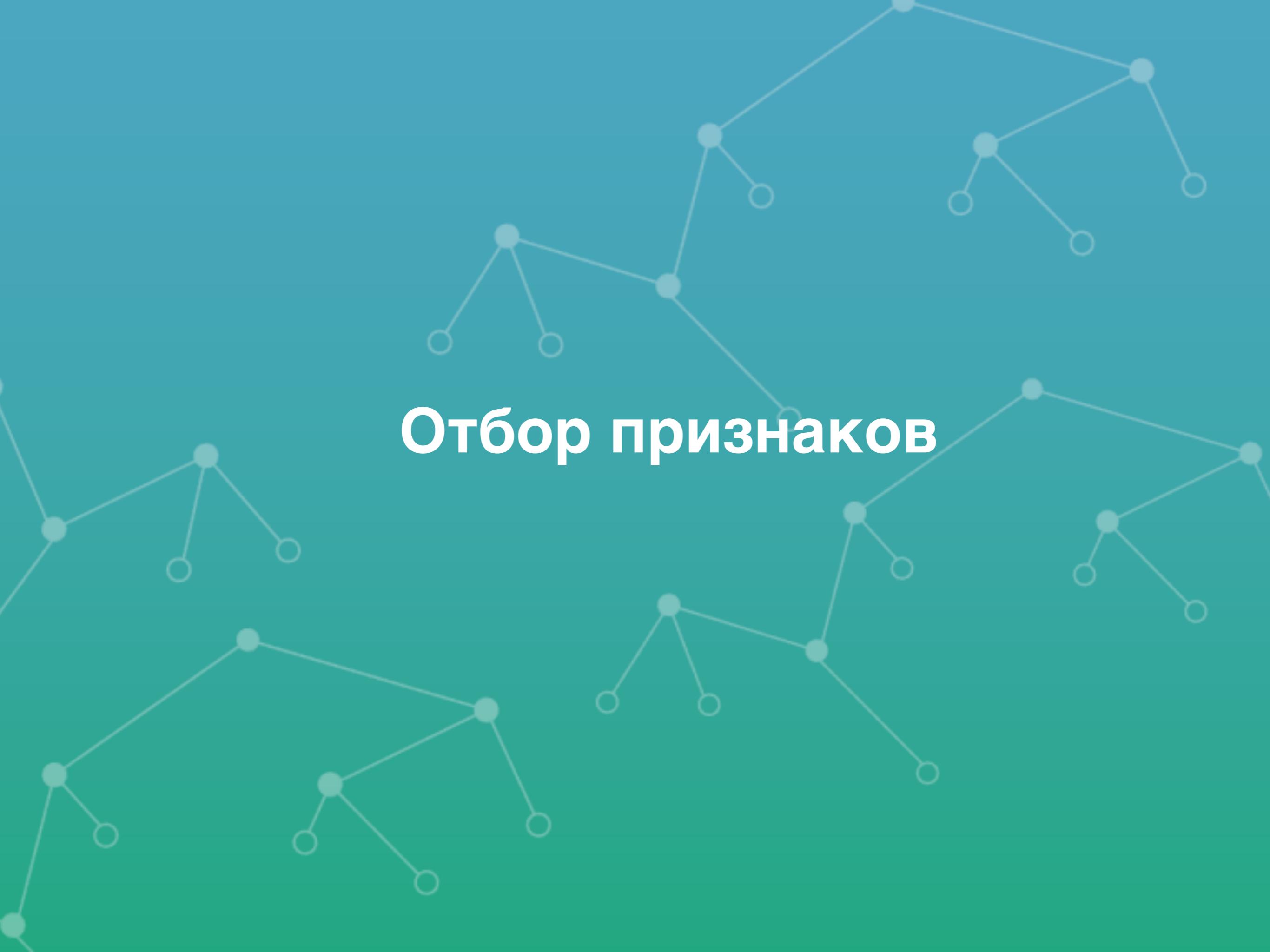
- Выбросить часть признаков



- Построить меньшее количество признаков на основе старых



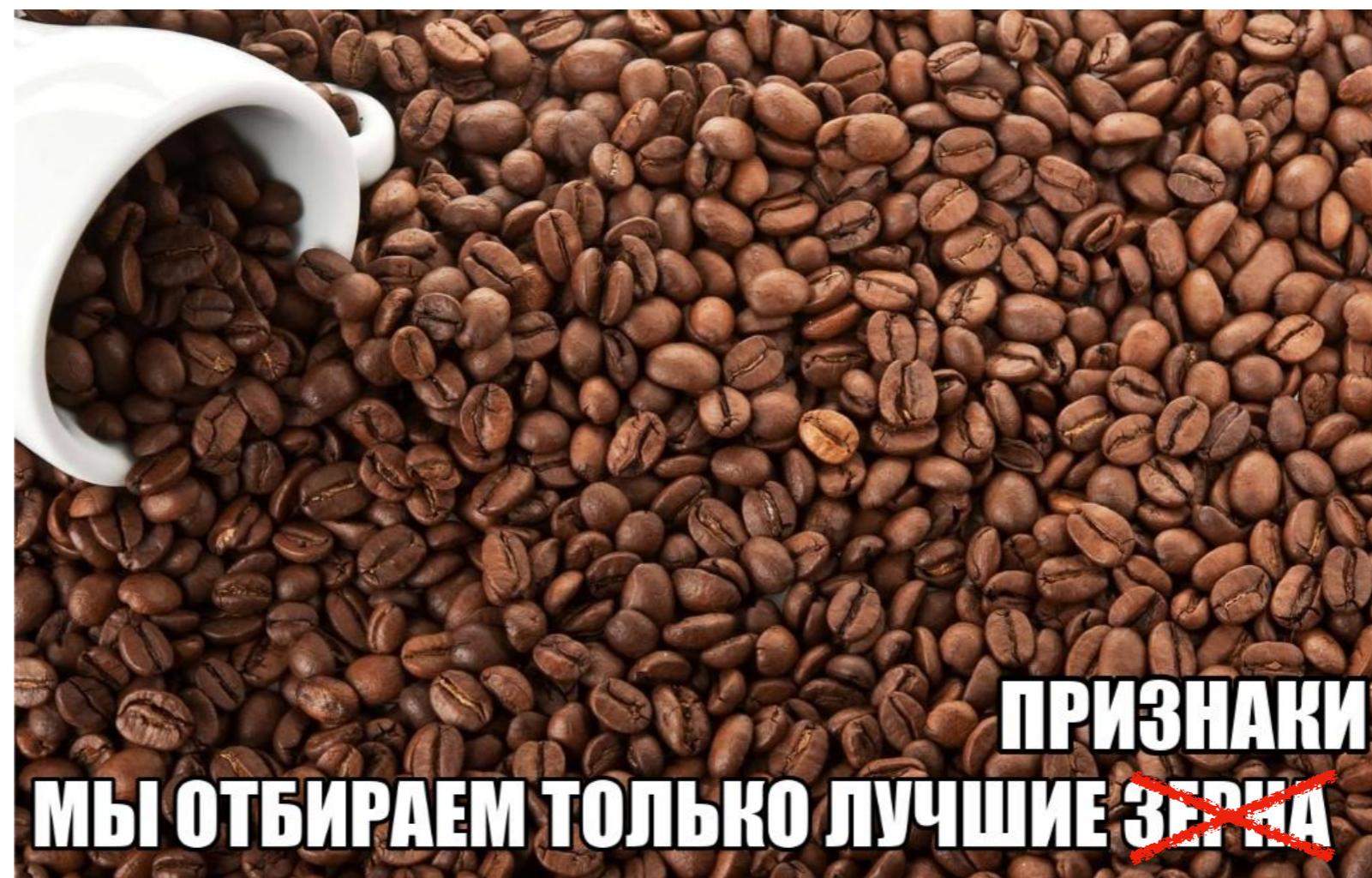
Отбор признаков



Зачем отбирать признаки



- боремся с проклятием размерности
- могут быть шумовые или коррелирующие признаки
- ограничение на ресурсы (модель/данные не влезают в память)
- ускорение обучения/инференса



Информативность признака



- корреляция/PMI с таргетом
- метрики классификатора на 1 признаке
- статистические критерии (chi-square)

$$\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)}$$

Одномерный отбор



- оцениваем информативность каждого признака отдельно
- отбираем к лучших или выше порога
 - + просто
 - не учитывается *взаимодействие* признаков между собой
(пропустим коллинеарные признаки)

Жадные методы

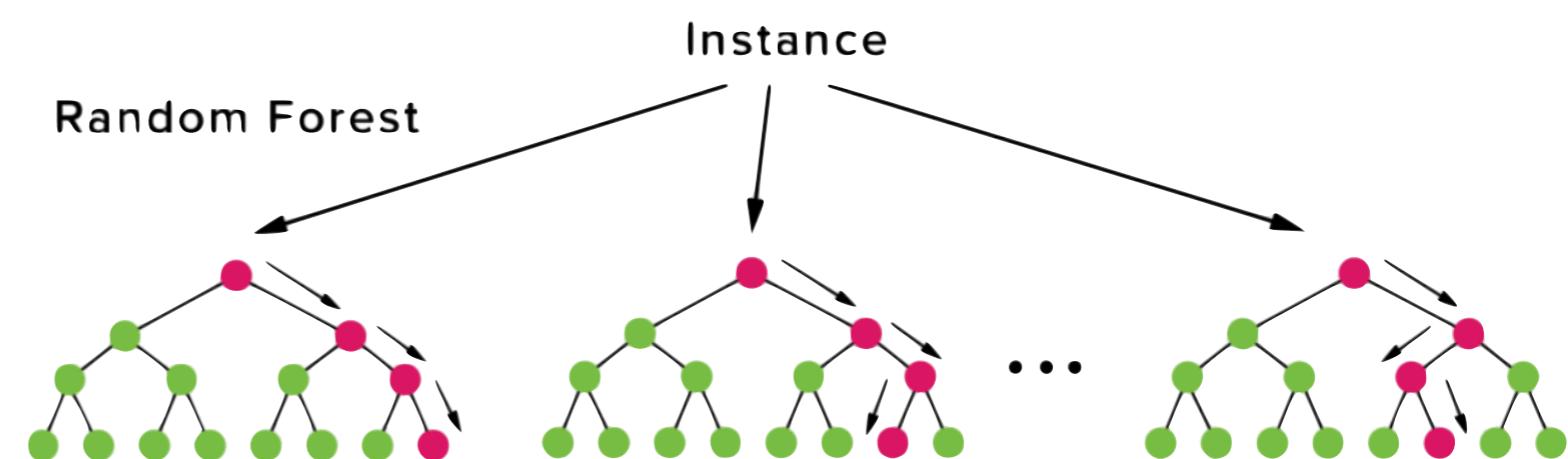
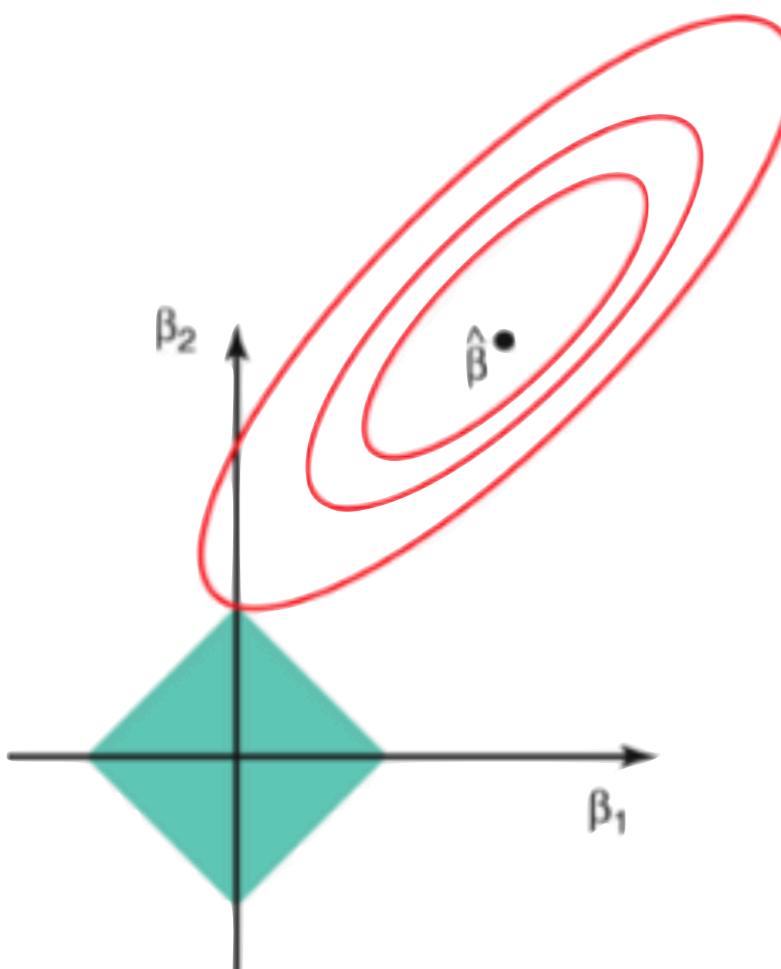


- Перебираем комбинации (переборные методы, жадные методы, ADD-DEL)
 - Для каждой комбинации обучаем модель
 - Смотрим метрики на валидации
 - Берем набор признаков с лучшим качеством
-
- + учитывает *взаимодействие* признаков
- затратно (много комбинаций, обучение на каждом наборе)
- жадные методы иногда слишком жадные $\text{\textbackslash}(\text{\texttt{\texttt{}}})\text{\textbackslash}/$

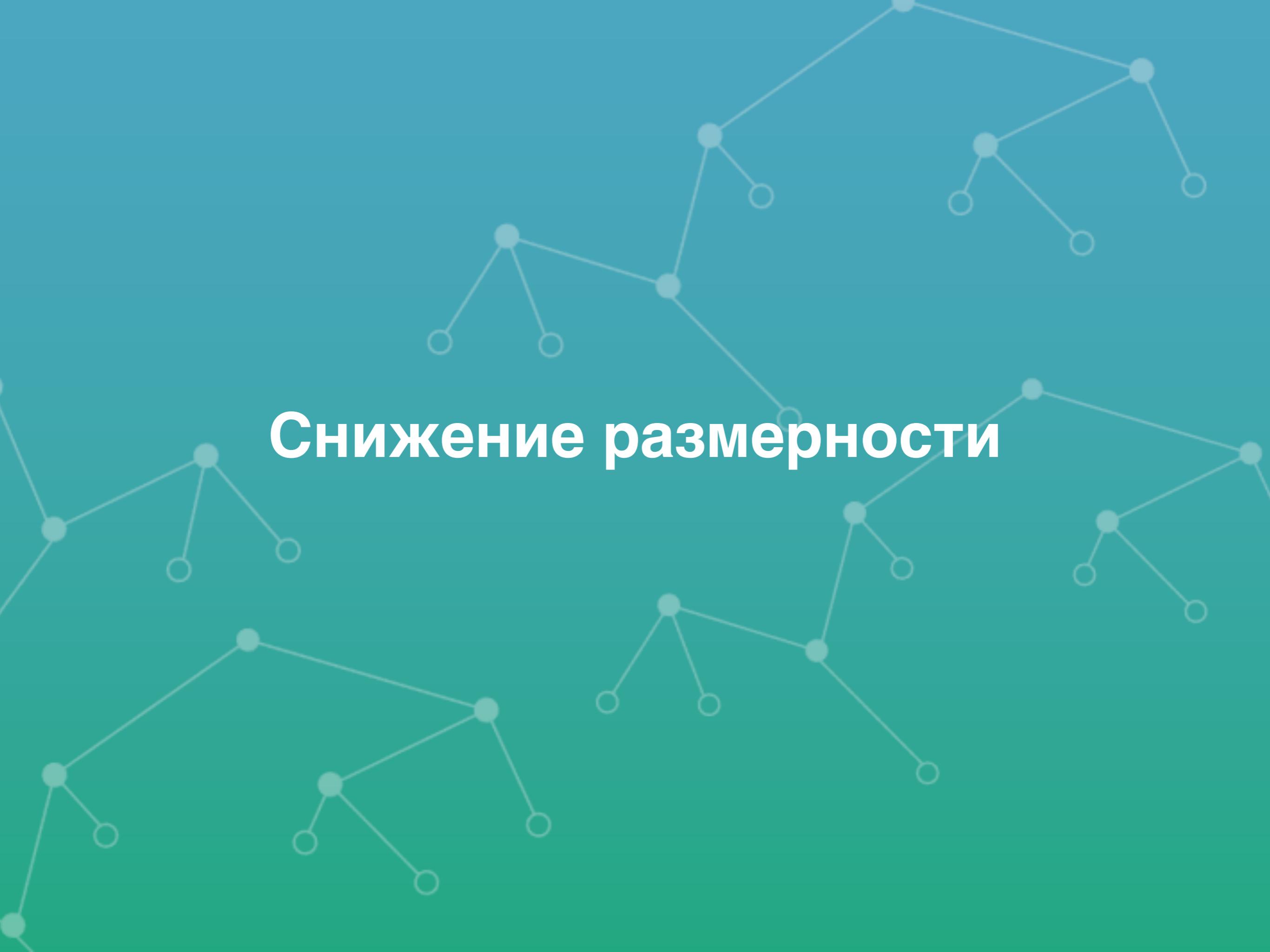
Отбор на основе моделей



- Линейные модели (веса, как показатель информативности)
- L1-регуляризация
- Деревья, случайный лес, бустинг над деревьями (изменение критерия информативности)



Снижение размерности

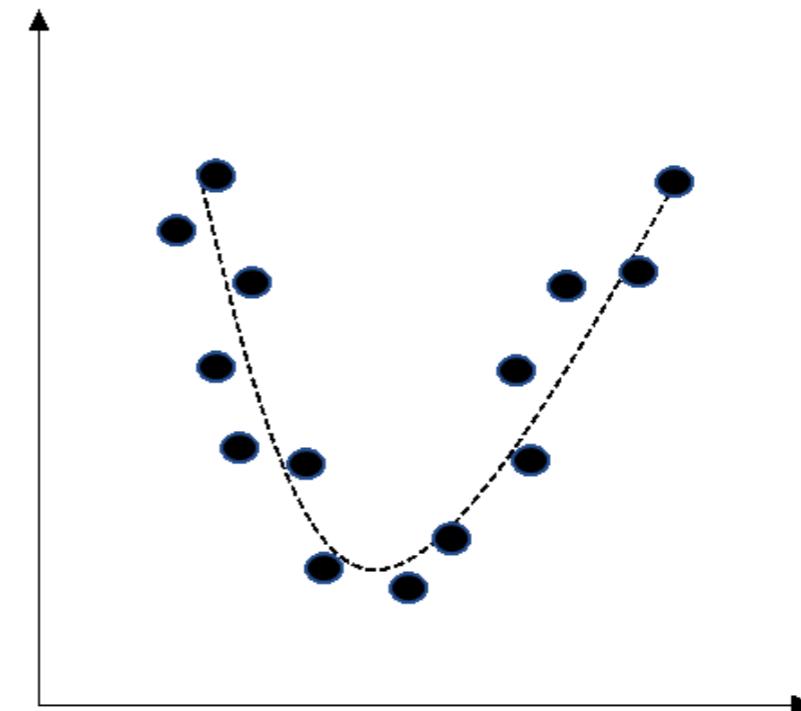
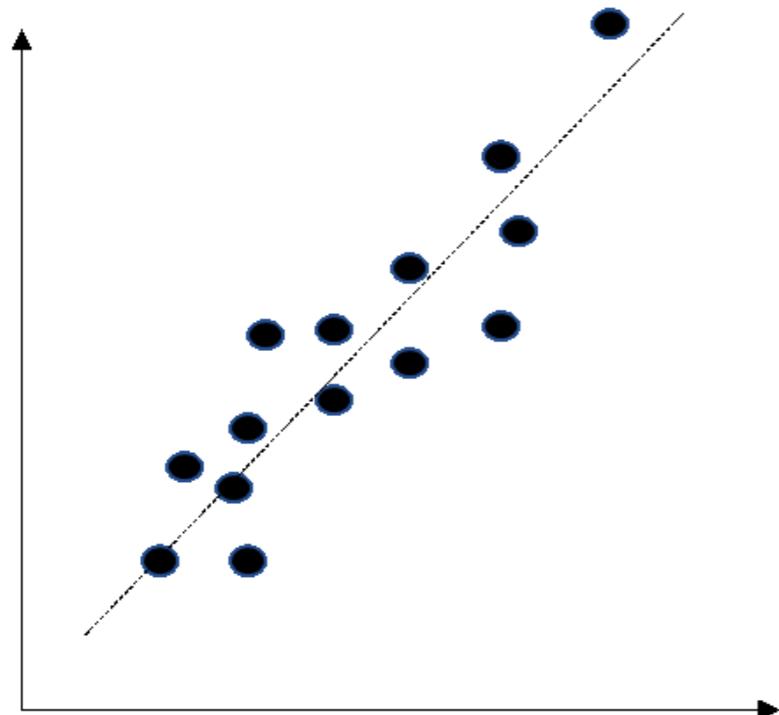


Снижение размерности



Цель: построить меньшее количество новых признаков, которые содержат максимум информации из исходных

- Линейные методы (PCA)
- Нелинейные методы (Isomap, TSNE)



Principal Component Analysis (PCA)

Постановка задачи



Пусть есть матрица $F_{l,n}$ где l - количество объектов, n - количество признаков

Хотим сократить количество признаков с n до m

При этом старые признаки должны как можно точнее **линейно восстанавливаться по новым** на обучающей выборке

$$\hat{f}_j(x) = \sum_{s=1}^m g_s(x) u_{js}, \quad j = 1, \dots, n, \quad \forall x \in X$$



PCA. Постановка задачи



Старые признаки \hat{f}_j - линейная комбинация новых g_s

$$\hat{f}_j(x) = \sum_{s=1}^m g_s(x) u_{js}, \quad j = 1, \dots, n, \quad \forall x \in X$$

Хотим восстанавливать старые признаки из новых, как
можно точнее

$$\sum_{i=1}^{\ell} \sum_{j=1}^n (\hat{f}_j(x_i) - f_j(x_i))^2 \rightarrow \min_{\{g_s(x_i)\}, \{u_{js}\}}$$

PCA. Постановка задачи



Матрицы «объекты–признаки», старая и новая:

$$F_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}; \quad G_{\ell \times m} = \begin{pmatrix} g_1(x_1) & \dots & g_m(x_1) \\ \dots & \dots & \dots \\ g_1(x_\ell) & \dots & g_m(x_\ell) \end{pmatrix}$$

Матрица линейного преобразования новых признаков в старые:

$$U_{n \times m} = \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \dots & \dots & \dots \\ u_{n1} & \dots & u_{nm} \end{pmatrix}$$

$$\hat{F} = GU^\top \quad \text{хотим} \quad \approx F$$

Найти: и новые признаки G , и преобразование U :

$$\sum_{i=1}^{\ell} \sum_{j=1}^n (\hat{f}_j(x_i) - f_j(x_i))^2 = \|GU^\top - F\|^2 \rightarrow \min_{G, U}$$

PCA. Решение



Теорема:

Если $m \leq \text{rk } F$, то минимум $\|GU^\top - F\|^2$ достигается, когда столбцы U — это с.в. матрицы $F^\top F$, соответствующие m максимальным с.з. $\lambda_1, \dots, \lambda_m$, а матрица $G = FU$.

при этом:

- матрица U ортонормирована: $U^\top U = I_m$;
- матрица G ортогональна: $G^\top G = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$
- $U\Lambda = F^\top FU$; $G\Lambda = FF^\top G$;
- $\|GU^\top - F\|^2 = \|F\|^2 - \text{tr } \Lambda = \sum_{j=1}^n \lambda_j - \sum_{j=1}^m \lambda_j = \sum_{j=m+1}^n \lambda_j$.

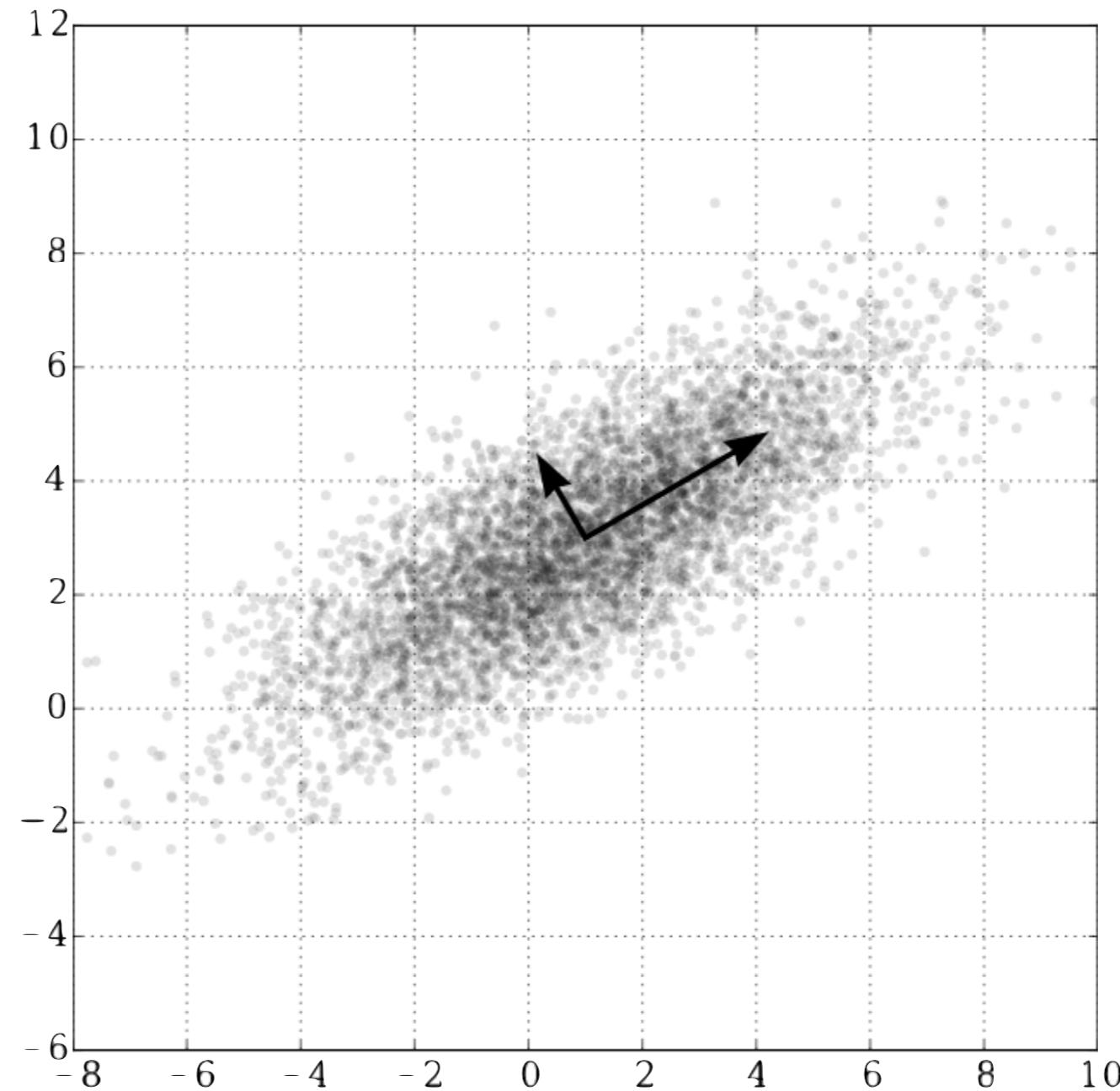


PCA - декоррелирующее преобразование

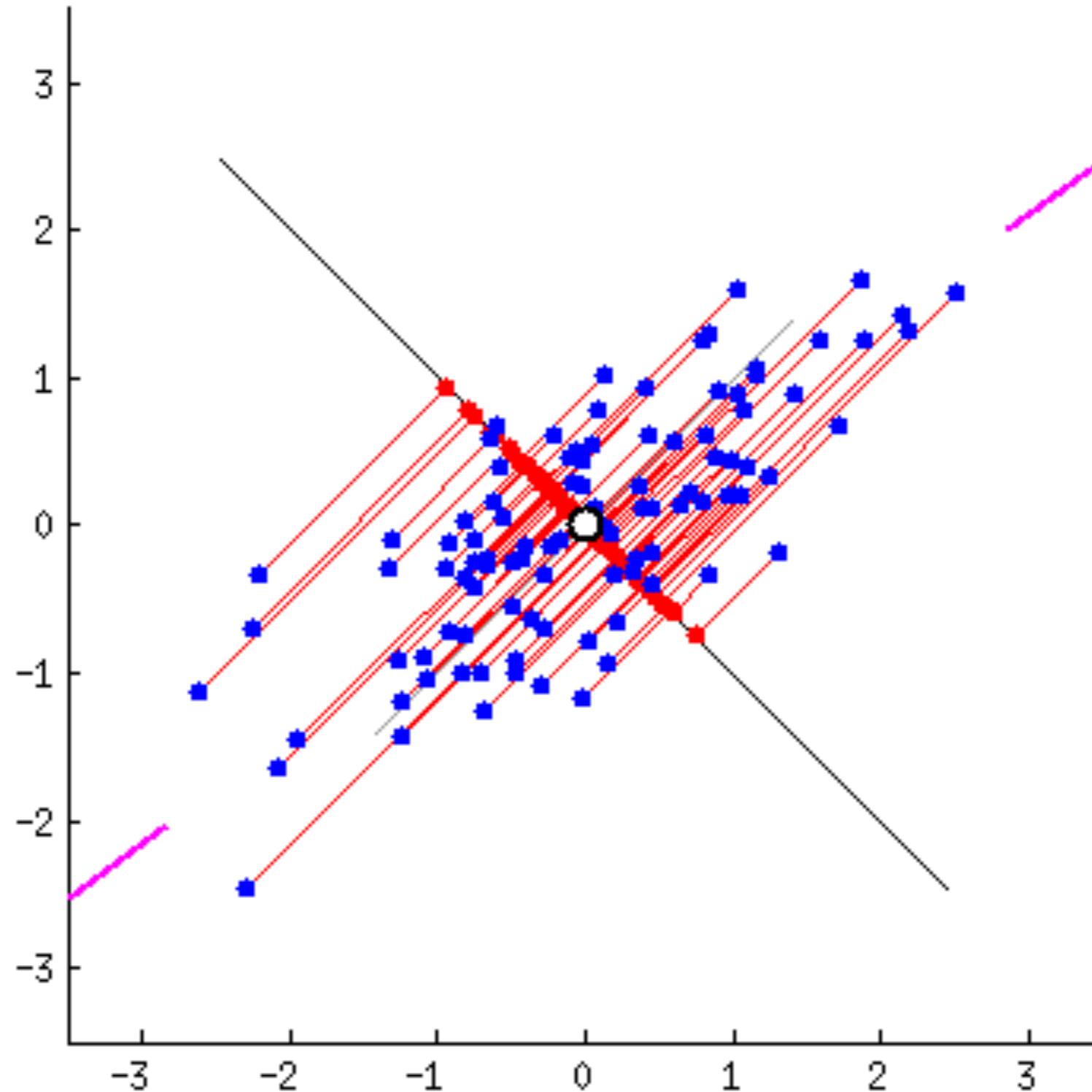


Матрица G ортогональна $G^T G = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$

-> Новые признаки
образуют **базис**



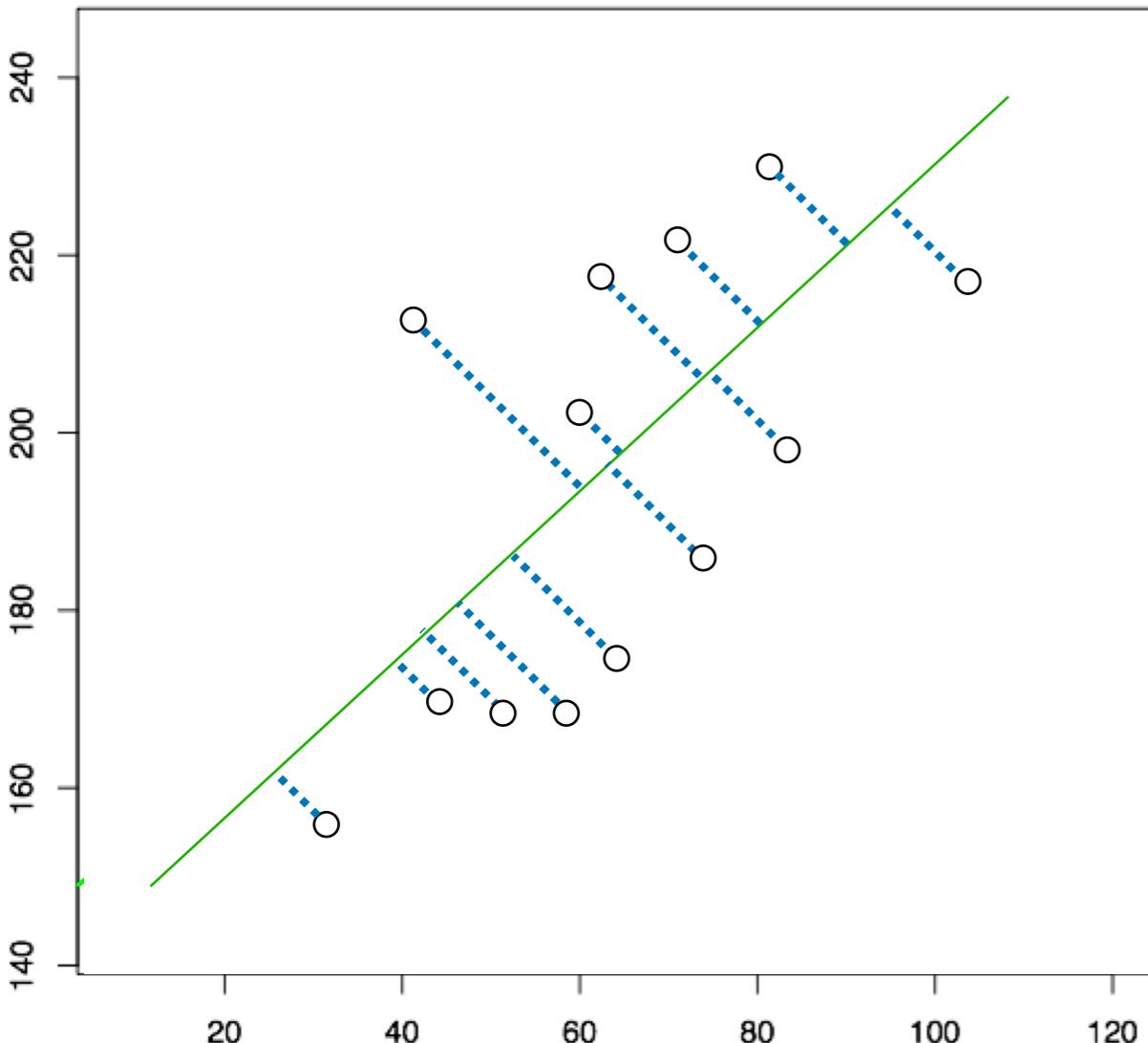
PCA. Интуиция



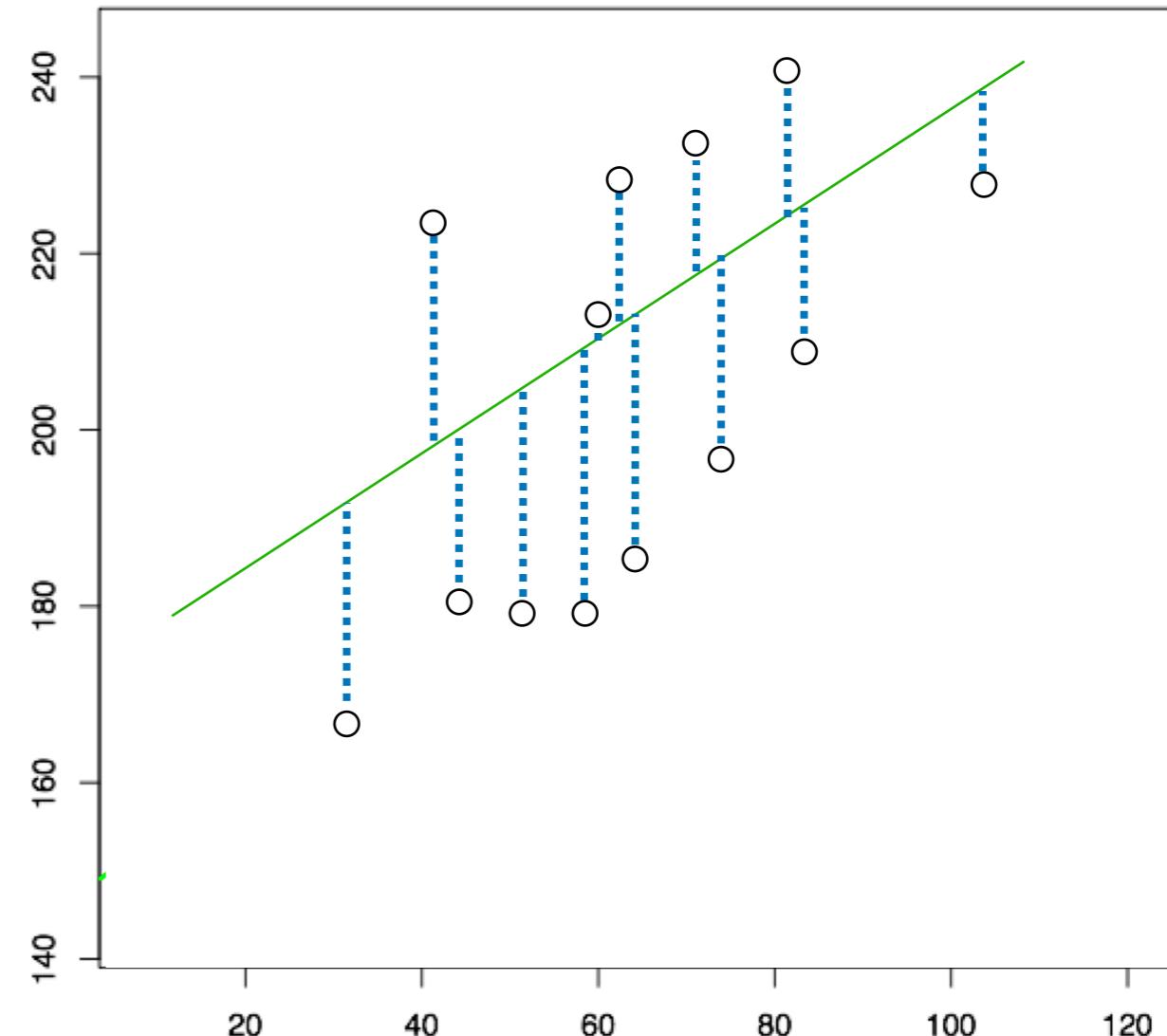
PCA. Интуиция



PCA



Regression



PCA. Recap



- PCA ищет такую гиперплоскость, суммарное расстояние от точек выборки до которых будет минимально
- PCA ищет такие ортогональные проекции, дисперсия вдоль которых для точек выборки будет максимальна
- PCA строит такой базис, в котором новые признаки ортогональны

Как выбрать количество новых признаков



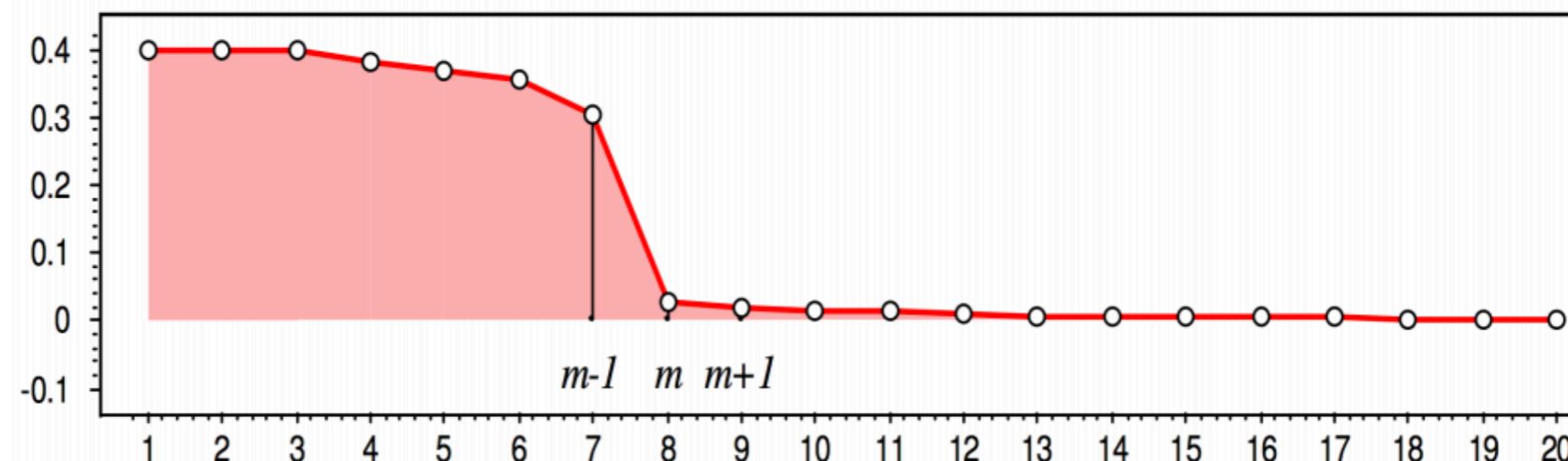
Собственные числа упорядочены по убыванию:

$$\lambda_1 \geq \dots \geq \lambda_n \geq 0.$$

$$E_m = \frac{\|GU^\top - F\|^2}{\|F\|^2} = \frac{\lambda_{m+1} + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_n} \leq \varepsilon$$

$$\begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \lambda_m & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_{n-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

Критерий «крутого склона»: находим m : $E_{m-1} \gg E_m$:

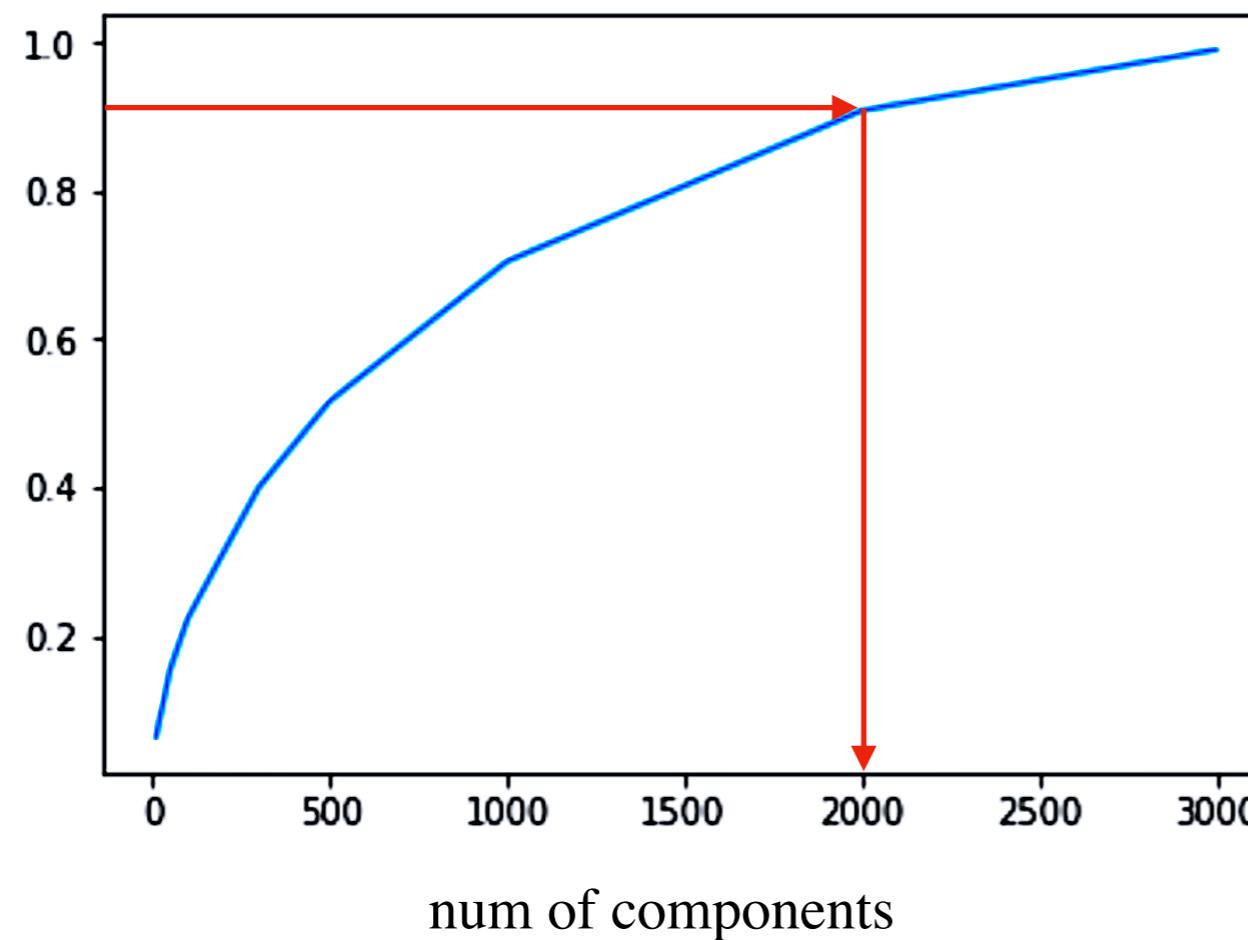


Как выбрать количество новых признаков



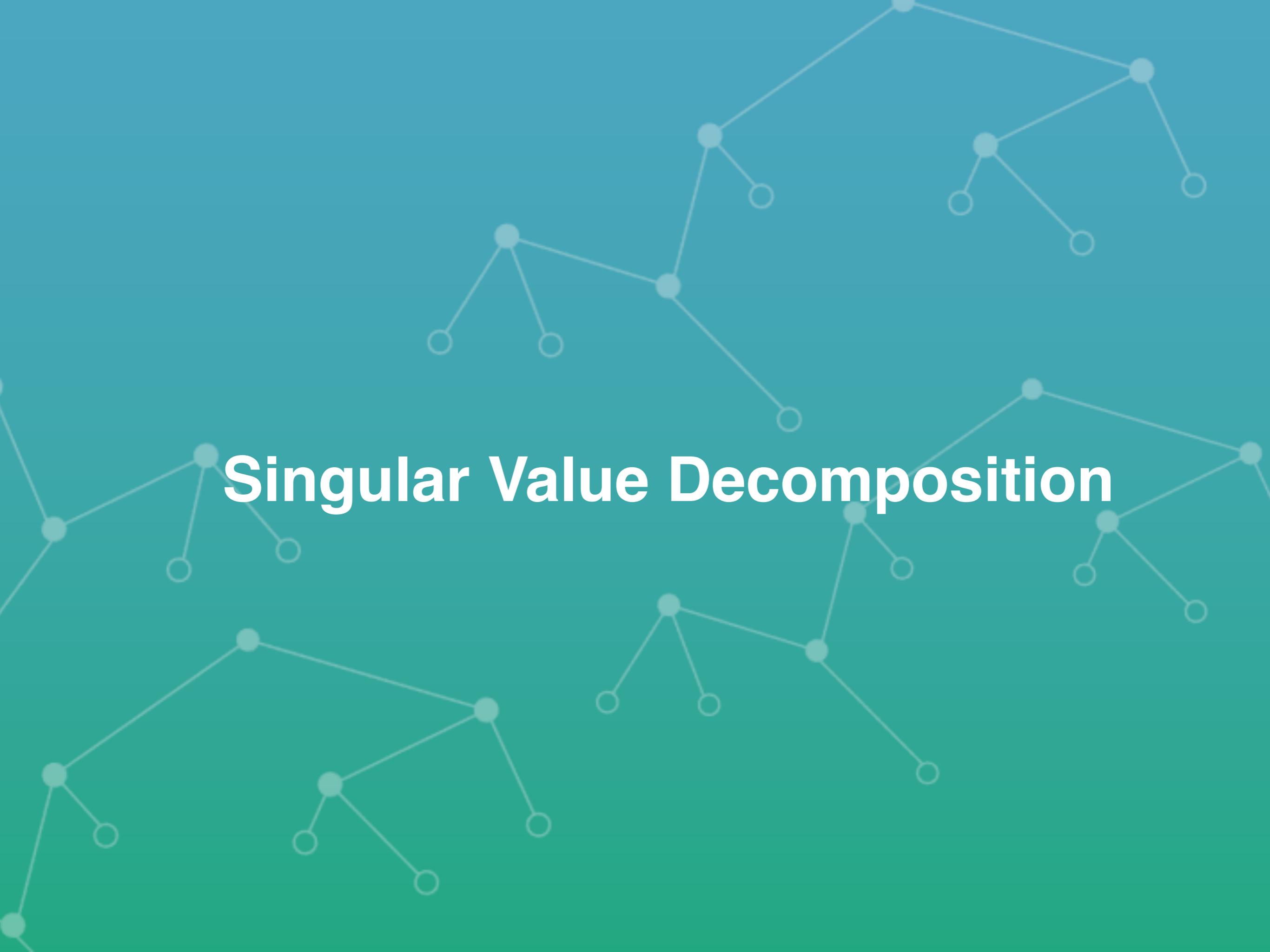
$$EVR = \frac{\lambda_1 + \dots + \lambda_m}{\lambda_1 + \dots + \lambda_n} > thr$$

Explained variance ratio



$$\begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \lambda_m & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_{n-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

Singular Value Decomposition



Singular Value Decomposition (SVD)



Любая вещественная матрица $F_{l,n}$ разложима следующим образом:

$$F_{l,n} = V_{l,n} * \Sigma_{n,n} * U_{n,n}^T$$

где V и U - ортогональные матрицы (состоят из левых и правых сингулярных векторов), Σ - на главной диагонали сингулярные числа

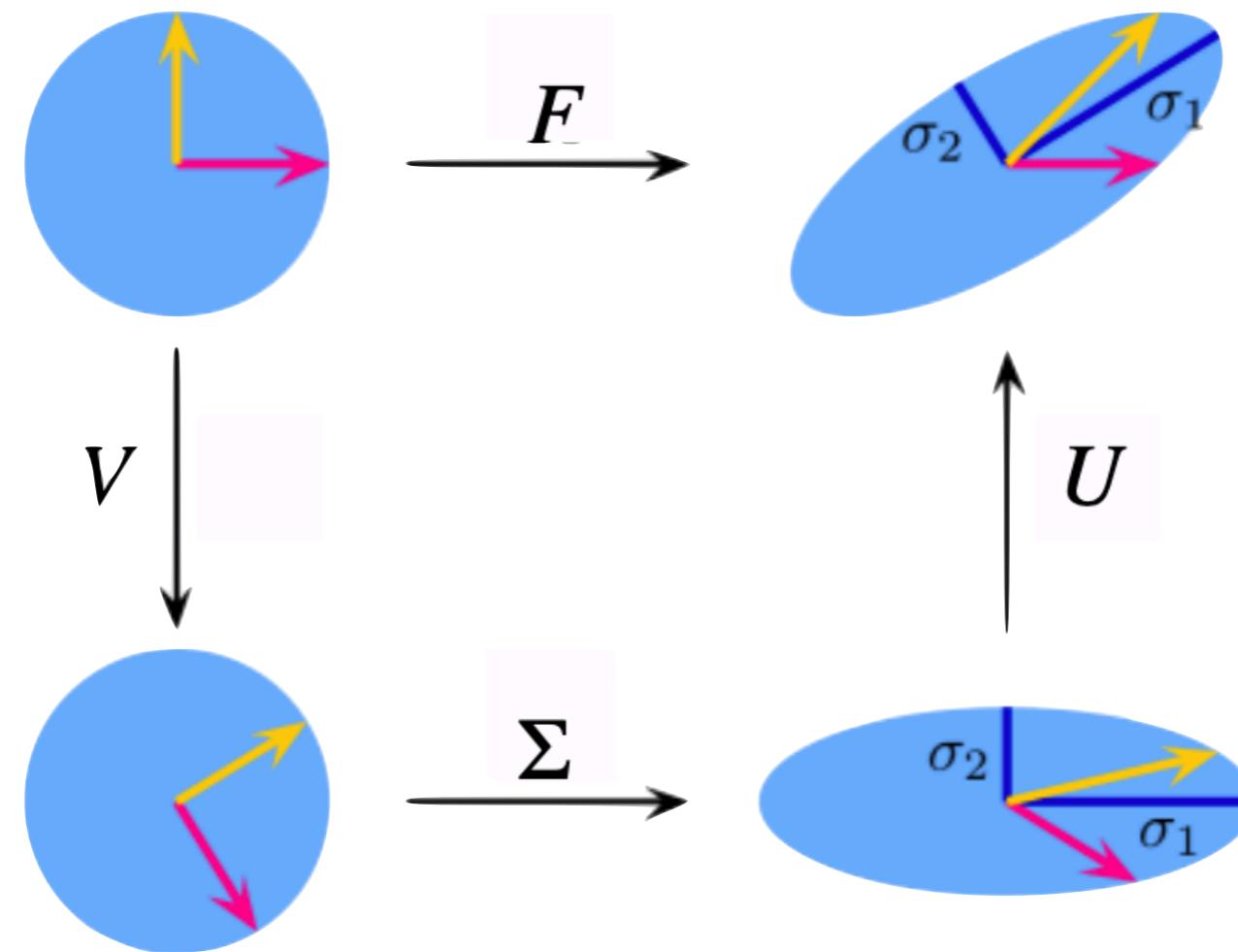
$$\begin{matrix} n \\ l & F \end{matrix} = \begin{matrix} n \\ l & V \end{matrix} \begin{matrix} n \\ \Sigma \end{matrix} \begin{matrix} n \\ n & U \end{matrix}$$

SVD - геометрическое объяснение



Пусть исходная матрица F задает некоторое линейное преобразование исходного пространства.

Тогда это линейное преобразование можно разбить на последовательность более простых линейных преобразований

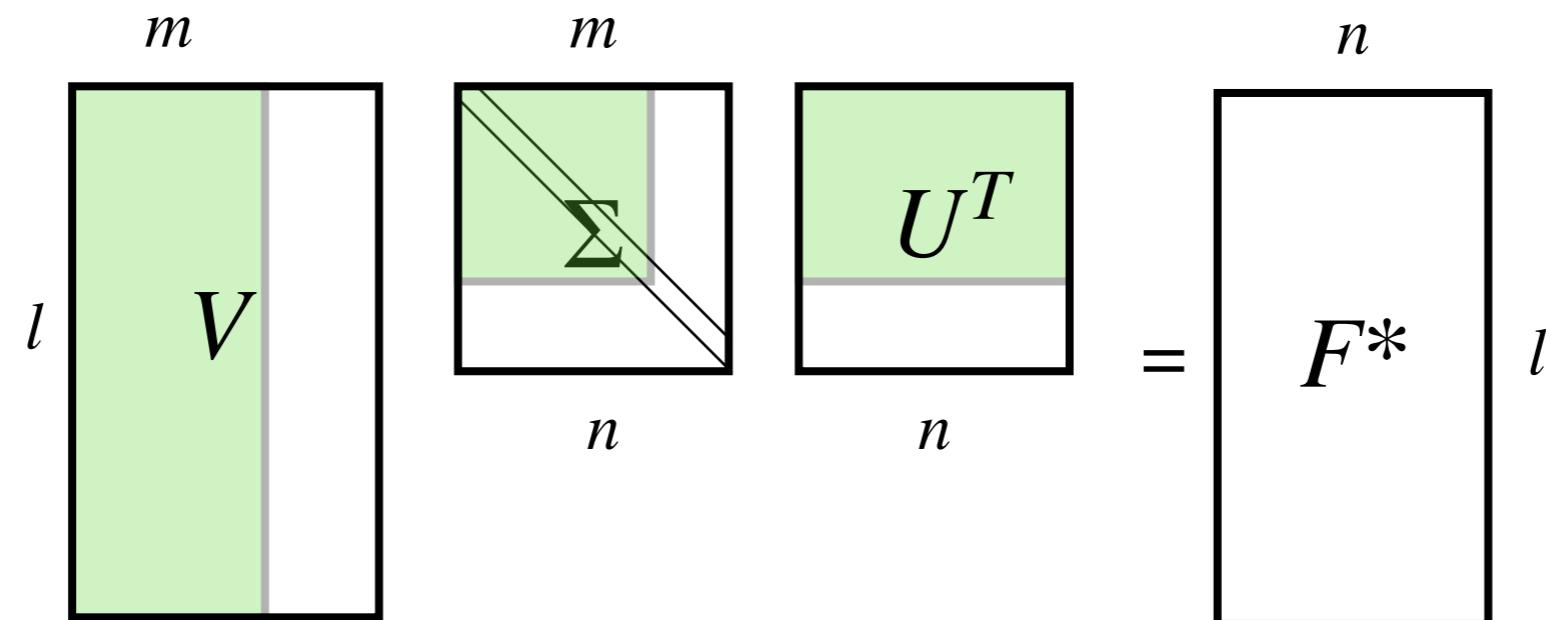


Truncated SVD



Можно взять m строк матрицы Σ , соответствующие наибольшим сингулярным числам, m столбцов матриц V и $U \rightarrow$ получить приближенное представление матрицы F

$$F *_{l,n} = V_{l,m} * \Sigma_{m,m} * U_{m,n}^T$$



SVD vs PCA



PCA:

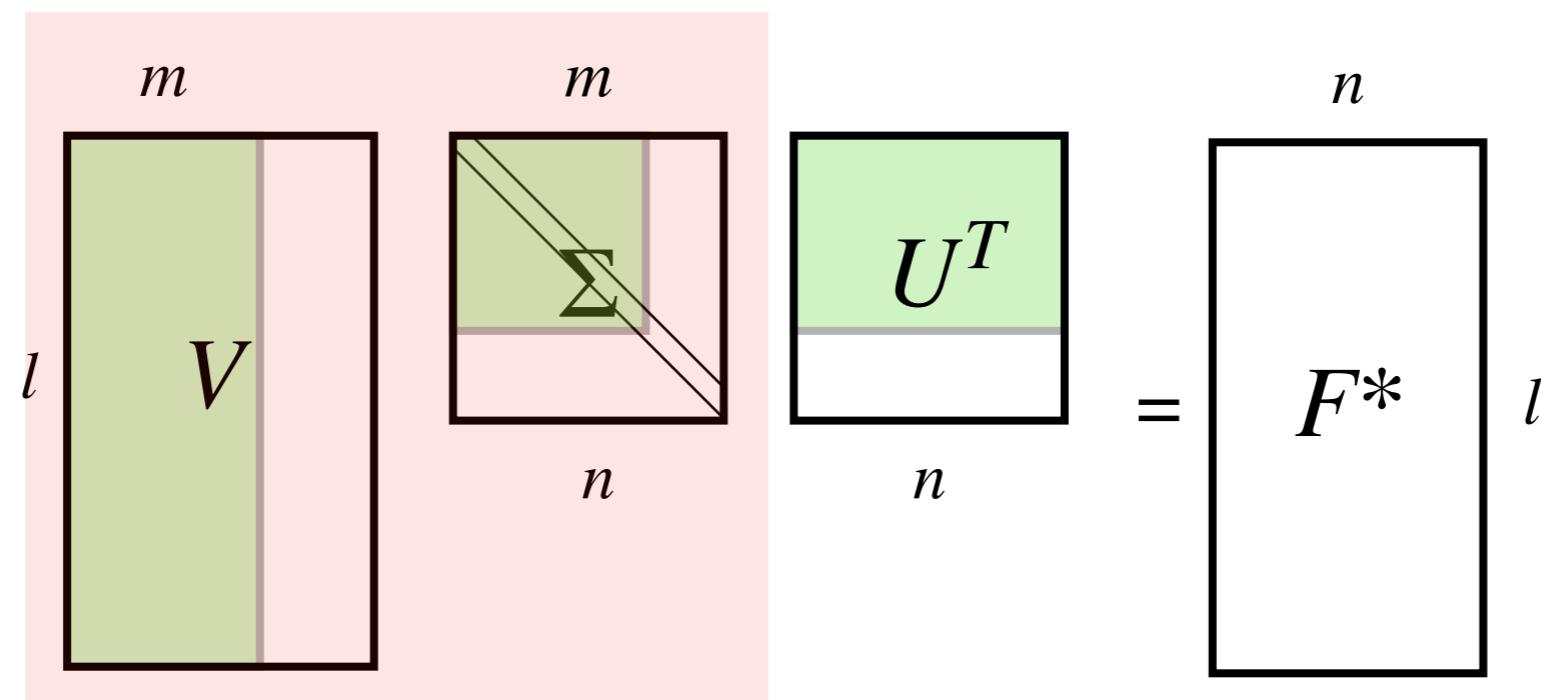
$$F_{l,m} = G_{l,m} * U_{m,n}^T$$

SVD:

$$F^*_{l,n} = V_{l,m} * \Sigma_{m,m} * U_{m,n}^T$$

В качестве новых признаков можно взять произведение первых двух
матриц

$$\underline{G_{l,m}} = \underline{V_{l,m}} * \Sigma_{m,m}$$



Manifold learning



Manifold Learning



PCA - линейный метод снижения размерности признакового пространства

Но никто нас не ограничивает в выборе преобразования.

Есть ряд **нелинейных** методов снижения размерности: Isomap, t-SNE, MDS

Кроме того, с PCA отлично работает kernel trick.

Multidimensional scaling (MDS)



Идея: при снижении размерности **сохранять попарные расстояния** между объектами

x_1, \dots, x_ℓ - исходные объекты

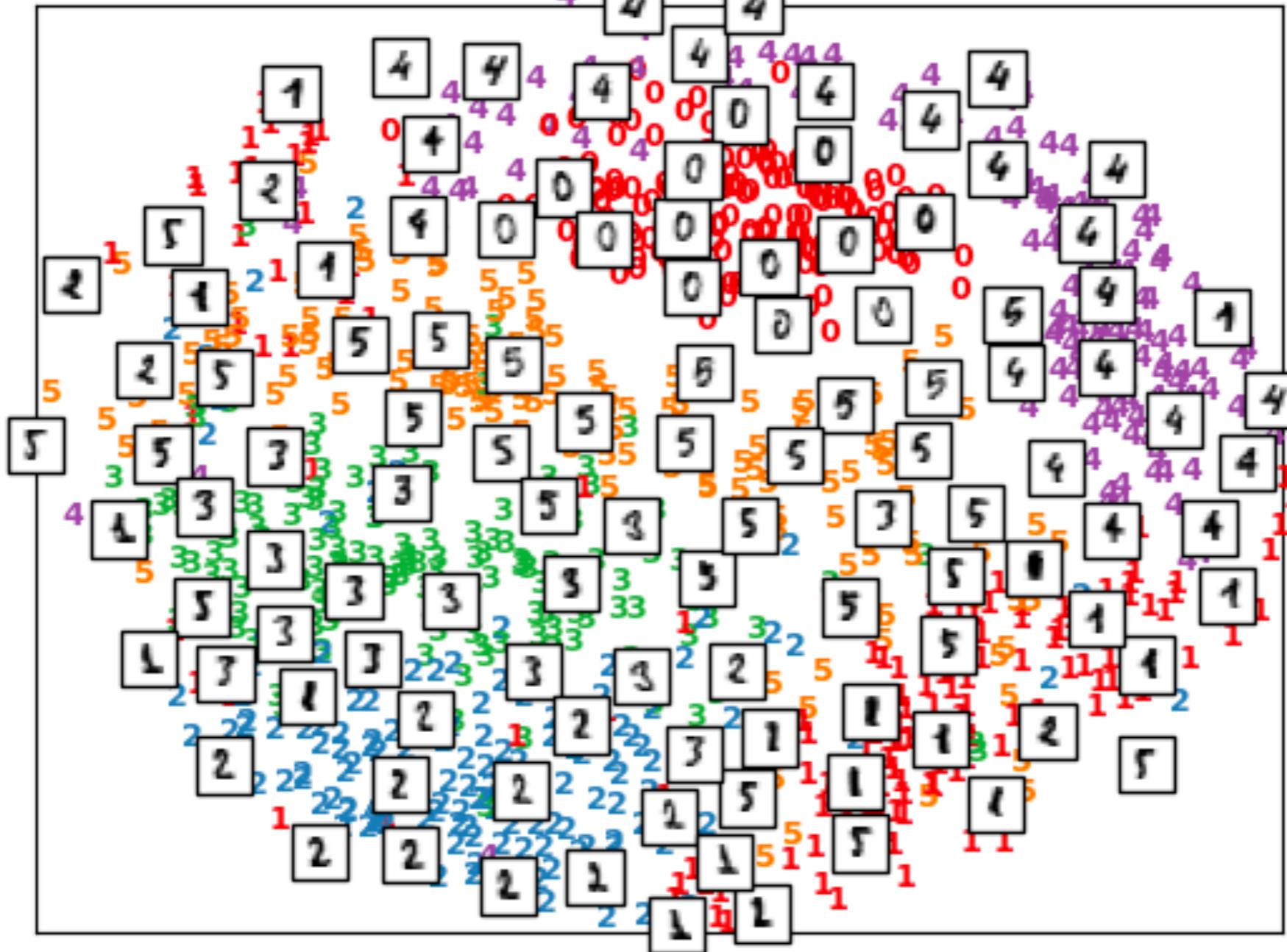
$\tilde{x}_1, \dots, \tilde{x}_\ell$ - объекты в новом признаков пространстве

$$\sum_{i < j}^{\ell} (\tilde{d}_{ij} - d_{ij})^2 \rightarrow \min_{\tilde{x}_1, \dots, \tilde{x}_\ell}$$

Multidimensional scaling (MDS)



MDS embedding of the digits (time 1.61s)



t-Stochastic Neighbour Embedding (t-SNE)



При решении задачи уменьшения размерности очень сложно сохранить попарные расстояния в случае сильного снижения размерности

Идея: при снижении размерности сохранять *пропорции расстояний* между объектами

$\rho(x_i, x_j) = \alpha \rho(x_i, x_k)$ - в исходном пространстве

$\rho(\tilde{x}_i, \tilde{x}_j) = \alpha \rho(\tilde{x}_i, \tilde{x}_k)$ - в новом пространстве

t-SNE



Условная вероятность соседства двух точек в исходном пространстве (гауссова плотность):

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

В новом пространстве

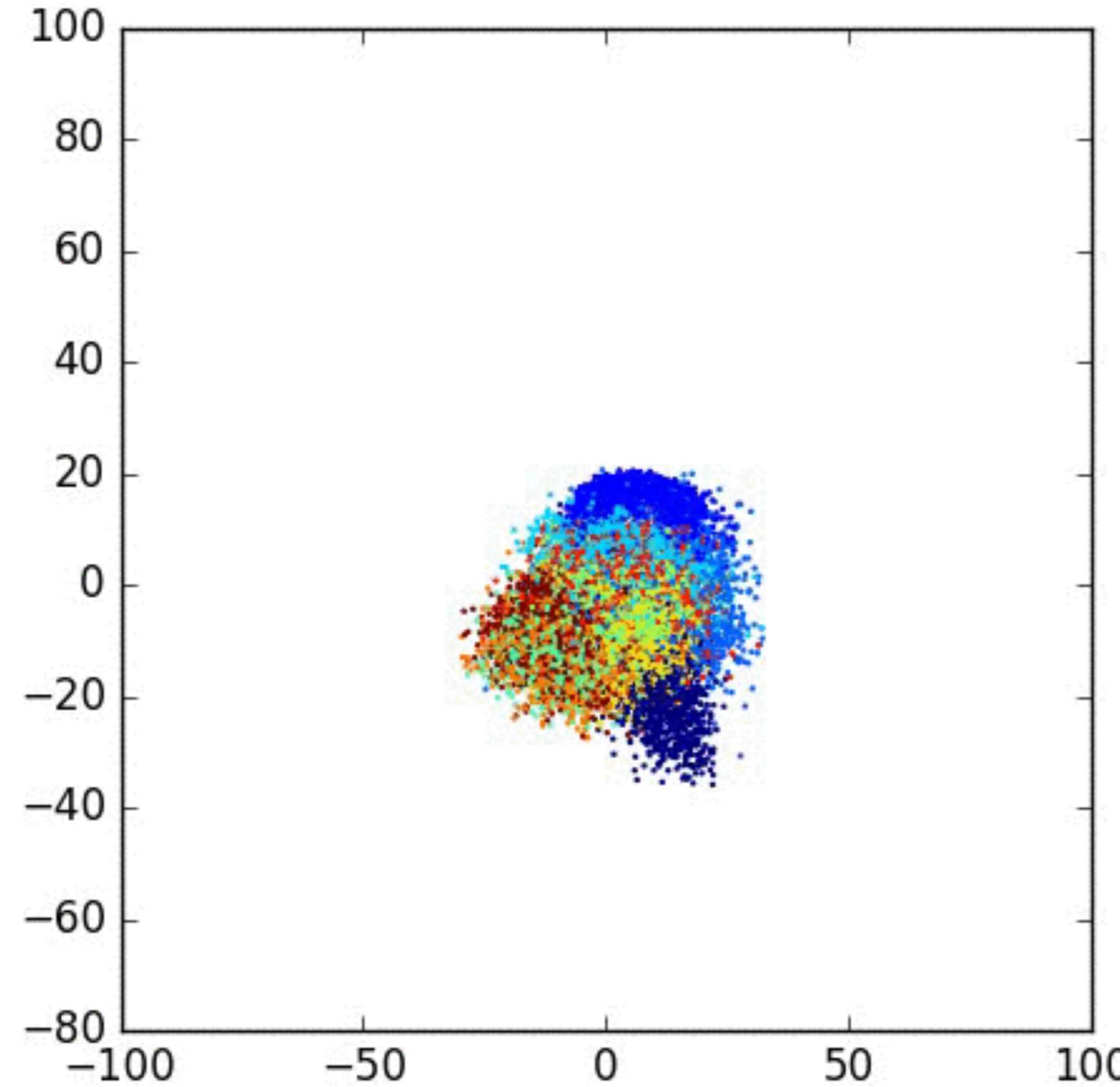
$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{y}_i - \mathbf{y}_k\|^2)^{-1}}$$

слабее штраф за увеличение
расстояний (решаем
проблему скученности)

Минимизируем

$$Cost = \sum_i KL(P_i \| Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

t-SNE - визуализация



NEVER TRAVEL WITH THIS GUY



T.Hanks for attention