

Введение в искусственный интеллект. Машинное обучение

Тема: Методы снижения размерности. Отбор признаков

Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем



① PCA

- SVD-разложение
- Kernel PCA
- Sparse PCA

② MDS

③ Isomap

④ Denoising autoencoder

⑤ Методы отбора признаков



- Многие задачи машинного обучения содержат тысячи или даже миллионы признаков



- Многие задачи машинного обучения содержат тысячи или даже миллионы признаков
- Типичный пример: если для восстановления плотности одномерной бинарной случайной величины нам потребуется $2 \times 100 = 200$ примеров, то для восстановления 100-мерной с той же точностью— $2^{100} \times 100$



- Многие задачи машинного обучения содержат тысячи или даже миллионы признаков
- Типичный пример: если для восстановления плотности одномерной бинарной случайной величины нам потребуется $2 \times 100 = 200$ примеров, то для восстановления 100-мерной с той же точностью— $2^{100} \times 100$

Вывод

Надо уменьшать размерность данных



Для чего нужно уменьшать размерность данных

- Сжатие данных

Для чего нужно уменьшать размерность данных

- Сжатие данных
- Если данные лежат на многообразии меньшей размерности, то локальные координаты могут оказаться более информативными

Для чего нужно уменьшать размерность данных

- Сжатие данных
- Если данные лежат на многообразии меньшей размерности, то локальные координаты могут оказаться более информативными
- Удаление шума из данных

Для чего нужно уменьшать размерность данных

- Сжатие данных
- Если данные лежат на многообразии меньшей размерности, то локальные координаты могут оказаться более информативными
- Удаление шума из данных
- Выделение главных признаков

Для чего нужно уменьшать размерность данных

- Сжатие данных
- Если данные лежат на многообразии меньшей размерности, то локальные координаты могут оказаться более информативными
- Удаление шума из данных
- Выделение главных признаков
- Визуализация данных



- Encoder — процедура сжатия

Формальная постановка задачи

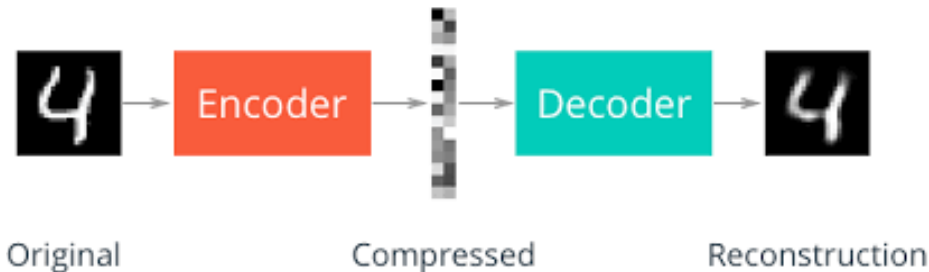
- Encoder — процедура сжатия
- Decoder — процедура восстановления

Формальная постановка задачи

- Encoder — процедура сжатия
- Decoder — процедура восстановления
- Оптимизационная задача $\|X - D(E(X))\|^2 \rightarrow \min$

Формальная постановка задачи

- Encoder — процедура сжатия
- Decoder — процедура восстановления
- Оптимизационная задача $\|X - D(E(X))\|^2 \rightarrow \min$

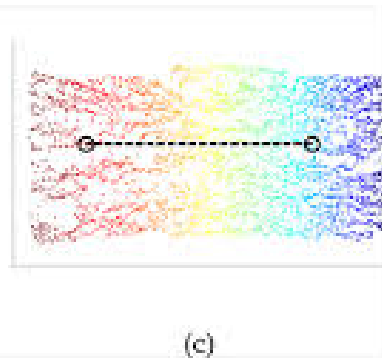
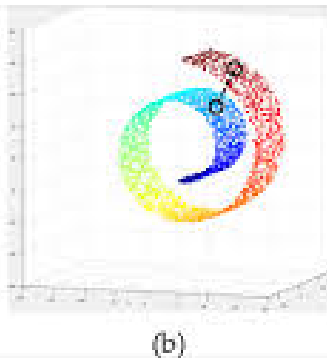
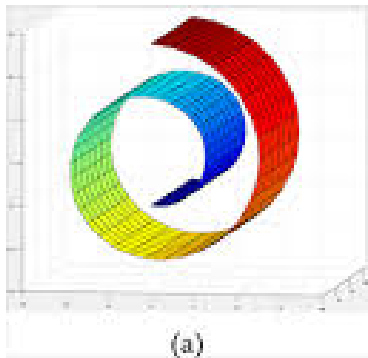


- Многие алгоритмы снижения размерности имеют некоторые предположения о типе многообразия

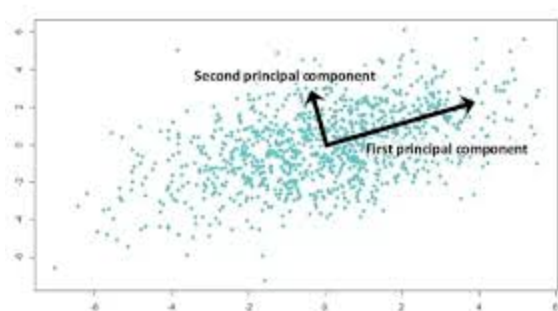
- Многие алгоритмы снижения размерности имеют некоторые предположения о типе многообразия
- Если есть априорные знания о данных, это может сильно помочь

Manifold learning

- Многие алгоритмы снижения размерности имеют некоторые предположения о типе многообразия
- Если есть априорные знания о данных, это может сильно помочь
- Типичный пример



- Основное предположение о многообразии — гиперплоскость ¹
- Оптимизационные задачи
 - Наименьшее отклонение от плоскости: $\sum_i dist^2(x_i, L_k) \rightarrow \min$
 - Наибольшее среднеквадратическое отклонение проекции на плоскость
 - Encoder и decoder — линейные функции



¹Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". Philosophical

Обозначения

Пусть x_1, \dots, x_ℓ — наблюдения из \mathbb{R}^n . Пусть $\bar{X} = 0$.

Пусть u_1, \dots, u_k — ортонормированный базис некоторого подпространства L_k , $k < n$.

Обозначения

Пусть x_1, \dots, x_ℓ — наблюдения из \mathbb{R}^n . Пусть $\bar{X} = 0$.

Пусть u_1, \dots, u_k — ортонормированный базис некоторого подпространства L_k , $k < n$.

Задача для $k=1$

Для текущих наблюдений найти такой u_1 , что $\|u_1\| = 1$ и выполнено

$$\sum_i \|u_1^T x_i\|^2 \rightarrow \max$$

Обозначения

Пусть x_1, \dots, x_ℓ — наблюдения из \mathbb{R}^n . Пусть $\bar{X} = 0$.

Пусть u_1, \dots, u_k — ортонормированный базис некоторого подпространства L_k , $k < n$.

Задача для $k=1$

Для текущих наблюдений найти такой u_1 , что $\|u_1\| = 1$ и выполнено

$$\sum_i \|u_1^T x_i\|^2 \rightarrow \max$$

Замечание

$$\|x_i - u_1(u_1, x_i)\|^2 = \|x_i\|^2 - (u_1, x_i)^2$$

Поэтому последнее условие эквивалентно $\sum_i \|x_i - u_1(u_1, x_i)\|^2 \rightarrow \min$

Решение оптимизационной задачи

① $\frac{1}{\ell} \sum_i \|u_1^T x_i\|^2 = u_1^T S u_1$, где $S = \frac{1}{\ell} \sum_i x_i x_i^T$ — матрица ковариаций.

Решение оптимизационной задачи

- 1 $\frac{1}{\ell} \sum_i \|u_1^T x_i\|^2 = u_1^T S u_1$, где $S = \frac{1}{\ell} \sum_i x_i x_i^T$ — матрица ковариаций.
- 2 Так как оптимизационная задача решается при условии $u_1^T u_1 = 1$, то перейдём к безусловной задаче максимизации (метод множителей Лагранжа):

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1)$$



Решение оптимизационной задачи

- 1 $\frac{1}{\ell} \sum_i \|u_1^T x_i\|^2 = u_1^T S u_1$, где $S = \frac{1}{\ell} \sum_i x_i x_i^T$ — матрица ковариаций.
- 2 Так как оптимизационная задача решается при условии $u_1^T u_1 = 1$, то перейдём к безусловной задаче максимизации (метод множителей Лагранжа):

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1)$$

- 3 Дифференцируя по параметру и приравнявая к нулю, получаем: $S u_1 = \lambda_1 u_1$ и $u_1^T S u_1 = \lambda_1$
- 4 u_1 — собственный вектор матрицы S , соответствующий максимальному собственному значению.



Индукция по числу компонент

Применив индукцию получаем, что u_1, \dots, u_k — собственные векторы матрицы S соответствующие максимальным собственным значениям.

Индукция по числу компонент

Применив индукцию получаем, что u_1, \dots, u_k — собственные векторы матрицы S соответствующие максимальным собственным значениям.

Определение

Направления соответствующие u_1, \dots, u_k называются главными

Теорема

Если $k < rkX$, то минимум $\|GU^T - X\|^2$ достигается, когда столбцы U — это собственные векторы матрицы $X^T X$ соответствующие максимальным значениям $\lambda_1, \dots, \lambda_k$, а матрица $G = XU$. При этом выполнено:

- 1 $U^T U = I_k$
- 2 матрица G ортогональна: $G^T G = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_k)$
- 3 $U\Lambda = X^T XU$, $G\Lambda = XX^T G$
- 4 $\|GU^T - X\|^2 = \sum_{i=k+1}^n \lambda_i$



Следствие

Если в предыдущей теореме взять $k = n$, то

$$X = V\sqrt{\Lambda}U^T,$$

где $U^T U = I_k$, $V^T V = I_k$.

Следствие

Если в предыдущей теореме взять $k = n$, то

$$X = V\sqrt{\Lambda}U^T,$$

где $U^T U = I_k$, $V^T V = I_k$.

SVD

Как правило большинство реализаций PCA используют SVD разложения, для нахождения главных компонент.



Вероятностная модель

$$p(z) = N(z|0, I)$$

$$p(x|z) = N(x|Wz + \mu, \sigma^2 I)$$



Вероятностная модель

$$p(z) = N(z|0, I)$$

$$p(x|z) = N(x|Wz + \mu, \sigma^2 I)$$

$$x = Wz + \mu + \varepsilon$$

Вероятностная интерпретация PCA

Вероятностная модель

$$p(z) = N(z|0, I)$$
$$p(x|z) = N(x|Wz + \mu, \sigma^2 I)$$

$$x = Wz + \mu + \varepsilon$$

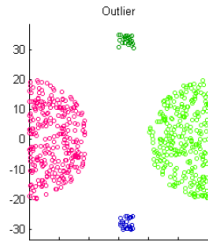
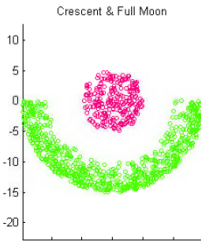
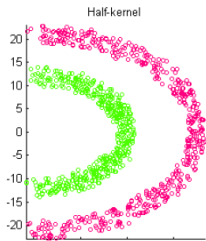
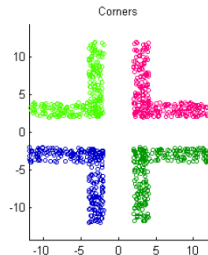
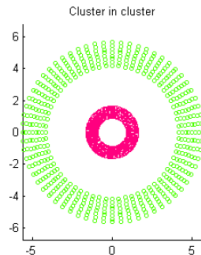
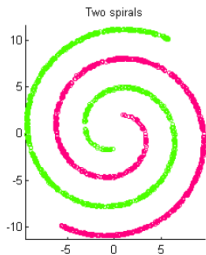
Следствие

Вероятностная интерпретация позволяет обобщить метод PCA и применять к нему вероятностные техники (например, EM-алгоритм)



- Kernel PCA
- Sparse PCA

Когда линейные методы не работают



Ядерный PCA (kernel trick)

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности² $\varphi : X \rightarrow H$ и применить линейный PCA там.

²Schölkopf, Bernhard (1998). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". Neural Computation. 10 (5): 1299–1319



Ядерный PCA (kernel trick)

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности² $\varphi : X \rightarrow H$ и применить линейный PCA там.

Ядро – функция $K : X \times X \rightarrow \mathbb{R}$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$ при некотором $\varphi : X \rightarrow H$, где H – гильбертово пространство.

²Schölkopf, Bernhard (1998). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". *Neural Computation*. 10 (5): 1299–1319



Ядерный PCA (kernel trick)

Если в исходном пространстве сложно разделить выборку, то попробуем перейти в пространство большей размерности² $\varphi : X \rightarrow H$ и применить линейный PCA там.

Ядро – функция $K : X \times X \rightarrow \mathbb{R}$, т.ч. $K(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$ при некотором $\varphi : X \rightarrow H$, где H – гильбертово пространство.

Теорема Мерсера

Функция $K(x_1, x_2)$ является ядром \Leftrightarrow 1) Она симметрична $K(x_1, x_2) = K(x_2, x_1)$ и 2) Неотрицательно определена $\int_X \int_X K(x_1, x_2) f(x_1) f(x_2) dx_1 dx_2 \geq 0$ для любой функции $f : X \rightarrow \mathbb{R}$.

²Schölkopf, Bernhard (1998). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". Neural Computation. 10 (5): 1299–1319

- Скалярное произведение $K(x_1, x_2) = \langle x_1, x_2 \rangle$ – ядро
- Константа $K(x_1, x_2) = c$ – ядро
- Произведение ядер $K(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$ – ядро
- Для любой $\varphi : X \rightarrow \mathbb{R}$ сепарабельная $K(x_1, x_2) = \varphi(x_1)\varphi(x_2)$ – ядро
- Линейная $K(x_1, x_2) = \alpha_1 K_1(x_1, x_2) + \alpha_2 K_2(x_1, x_2)$ – ядро при $\alpha_1, \alpha_2 > 0$, K_1, K_2 – ядрах
- Для любой $\varphi : X \rightarrow X$ подстановка $K(x_1, x_2) = K_1(\varphi(x_1), \varphi(x_2))$ – ядро при K_1 – ядро



- Полиномиальное ядро с мономами степени d : $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$



- Полиномиальное ядро с мономами степени d : $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$



- Полиномиальное ядро с мономы степени d : $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономы степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2)$ (наиболее универсальное)

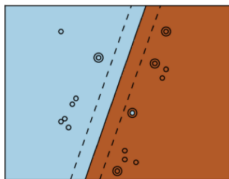


Примеры ядер

- Полиномиальное ядро с мономами степени d : $K(x_1, x_2) = \langle x_1, x_2 \rangle^d$
- Полиномиальное ядро с мономами степени $\leq d$: $K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^d$
- Радиальное ядро (RBF): $K(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2)$ (наиболее универсальное)

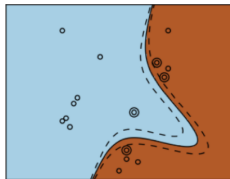
Линейное ядро

$$K(x_1, x_2) = \langle x_1, x_2 \rangle$$



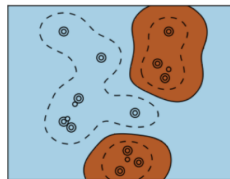
Полиномиальное ядро

$$K(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^3$$

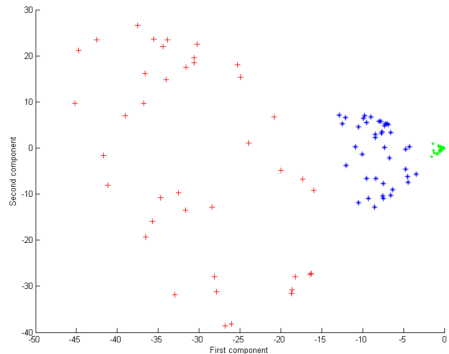
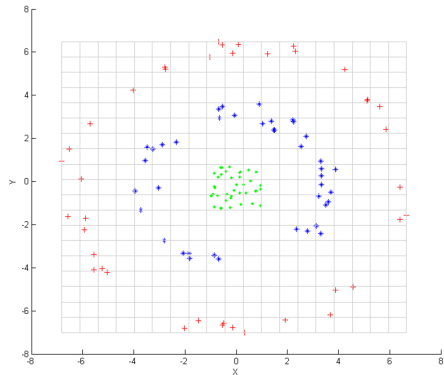


Радиальное ядро

$$K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2)$$



Пример работы с ядром $(x^T y + 1)^2$



Недостаток PCA

При применении PCA обычно получаются компоненты с небольшим числом нулей. Обычно это затрудняет интерпретируемость компонент.

Недостаток PCA

При применении PCA обычно получаются компоненты с небольшим числом нулей. Обычно это затрудняет интерпретируемость компонент.

Решение

Добавить l1-регуляризацию

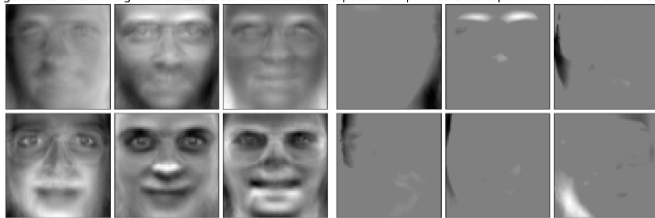
Недостаток PCA

При применении PCA обычно получаются компоненты с небольшим числом нулей. Обычно это затрудняет интерпретируемость компонент.

Решение

Добавить l_1 -регуляризацию

genfaces - PCA using randomized SVD - Train time 0.1 Sparse comp. - MiniBatchSparsePCA - Train time 0.8s



Multidimensional Scaling (MDS)

Дано

Дана матрица попарных расстояний между объектами d_{ij}

Задача

Найти z_1, z_2, \dots, z_n удовлетворяющие:

$$\min_{z_1, \dots, z_n} \sum_{i,j} (d_{ij} - \|z_i - z_j\|)^2$$



Least squares / Kruskal-Shephard scaling

$$\min_{z_1, \dots, z_n} \sum_{i,j} (d_{ij} - \|z_i - z_j\|)^2$$

Sammon mapping

$$\min_{z_1, \dots, z_n} \sum_{i \neq j} \frac{(d_{ij} - \|z_i - z_j\|)^2}{d_{ij}}$$

Classical scaling

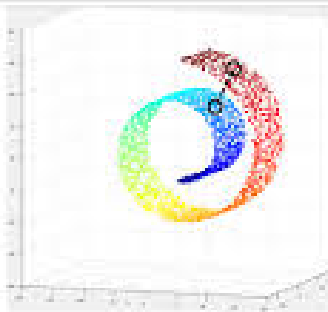
$$d_{ij} = (x_i - \bar{x}, x_j - \bar{x})$$
$$\min_{z_1, \dots, z_n} \sum_{i \neq j} (d_{ij} - (z_i - \bar{z}, z_j - \bar{z}))^2$$

Идея

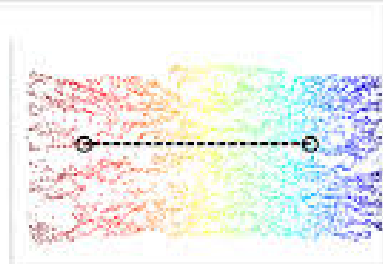
Вместо расстояний использовать геодезические



(a)



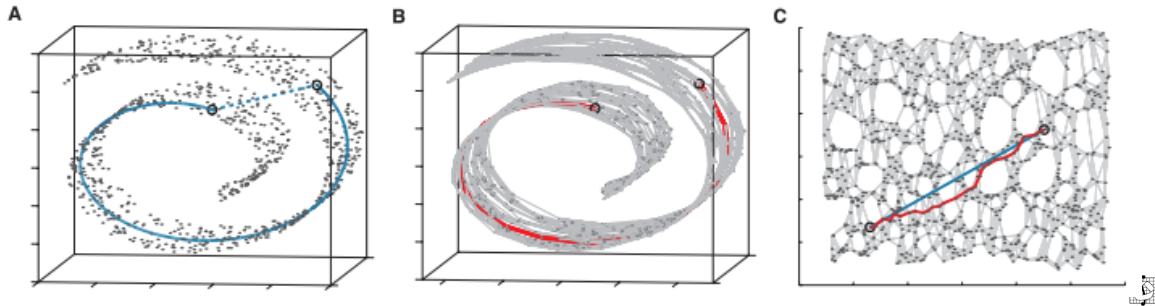
(b)



(c)

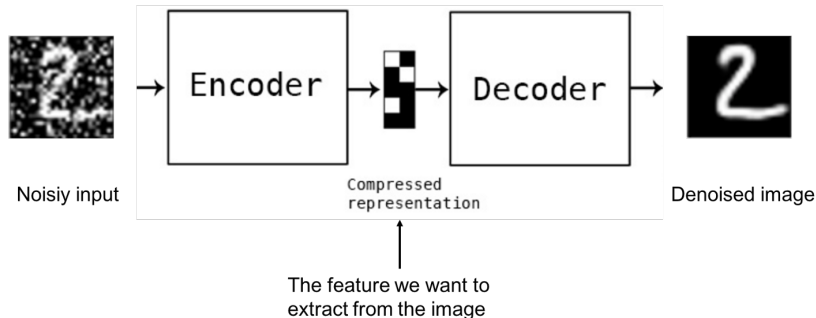
Схема алгоритма

- Построения графа соседства
- Вычисление геодезических на графе
- MDS



Denoising autoencoder

- Модель автоэнкодера — довольно общая архитектура снижения размерности данных
- Может использоваться для генерации новых признаков
- Может использоваться для отчистки данных от шума



- Статистические
- Основанные на важности признаков для конкретного алгоритма машинного обучения
- Переборные

Алгоритм

Для каждой сложности наборов искать лучший набор признаков.

Методы отбора признаков: полный перебор

Алгоритм

Для каждой сложности наборов искать лучший набор признаков.

Преимущества

- простота реализации
- гарантированный результат

Методы отбора признаков: полный перебор

Алгоритм

Для каждой сложности наборов искать лучший набор признаков.

Преимущества

- простота реализации
- гарантированный результат

Недостатки

- очень долго работает
- переобучение



Алгоритм

На каждой итерации алгоритма добавляется/удаляется наиболее выгодный признак

Методы отбора признаков: жадные алгоритмы перебор

Алгоритм

На каждой итерации алгоритма добавляется/удаляется наиболее выгодный признак

Преимущества

- простота реализации
- работает быстро

Методы отбора признаков: жадные алгоритмы перебор

Алгоритм

На каждой итерации алгоритма добавляется/удаляется наиболее выгодный признак

Преимущества

- простота реализации
- работает быстро

Недостатки

- склонен включать в набор лишние признаки



Методы отбора признаков: генетический алгоритм

Вход: множество F , критерий Q , параметры: d , p_m ,
 B — размер популяции, T — число поколений;

- 1: инициализировать случайную популяцию из B наборов:
 $B_1 := B$; $R_1 := \{J_1^1, \dots, J_1^{B_1}\}$; $Q^* := Q(\emptyset)$;
- 2: **для всех** $t = 1, \dots, T$, где t — номер поколения:
- 3: ранжирование индивидов: $Q(J_t^1) \leq \dots \leq Q(J_t^{B_t})$;
- 4: **если** $B_t > B$ **то**
- 5: селекция: $R_t := \{J_t^1, \dots, J_t^B\}$;
- 6: **если** $Q(J_t^1) < Q^*$ **то** $t^* := t$; $Q^* := Q(J_t^1)$;
- 7: **если** $t - t^* \geq d$ **то вернуть** $J_{t^*}^1$;
- 8: породить $t+1$ -е поколение путём скрещиваний и мутаций:
 $R_{t+1} := \{\sim(J' \times J'') \mid J', J'' \in R_t\} \cup R_t$;



Эвристики генетического алгоритма

- Увеличивать вероятности перехода признаков от более успешного родителя к потомку
- Накапливать оценки информативности признаков. Чем более информативен признак, тем выше вероятность его включения в набор во время мутации
- Скрещивать только лучшие индивиды (элитаризм)
- Переносить лучшие индивиды в следующее поколение
- В случае стагнации увеличивать количество мутаций
- Параллельно выращивать несколько изолированных популяций



- Метод главных компонент — рабочий инструмент по уменьшению размерности
- Метод главных компонент имеет огромное число обобщений, но не всегда они работают на реальных данных
- Автоэнкодер — универсальная модель для уменьшения размерности
- Отбор признаков и их ранжирование по важности — ключ к пониманию данных
- Точные алгоритмы по отбору признаков не работают на реальных данных, надо использовать эвристики



На основе материалов сайта <http://www.machinelearning.ru>.