Лекция 6
Нейросети для снижения размерности

Полыковский Даниил

27 марта 2016 г.

Задача снижения размерности

Dimensionality reduction

Дано. N обучающих D-мерных объектов $\mathbf{x}_i \in \mathcal{X}$, образующих тренировочный набор данных (training data set) $\mathbf{X}$.

Найти. Найти преобразование $A : \mathcal{X} \to \mathcal{P}$, $dim(\mathcal{P}) = d < D$, сохранив при этом большую часть “полезной информации” об $\mathcal{X}$.

Применение:

- Визуализация в 2D или 3D (поиск структуры и закономерностей)
- Уменьшение затрат на ресурсы (память, время)
- Снижение уровня шума в данных

Снижение размерности, примеры

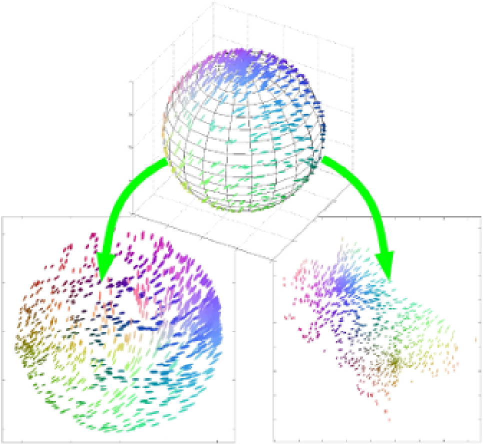


Рис.: Примеры данных[1]

[1] stats.stackexchange.com/questions/56589/visualizing-high-dimensional-data

Снижение размерности, примеры

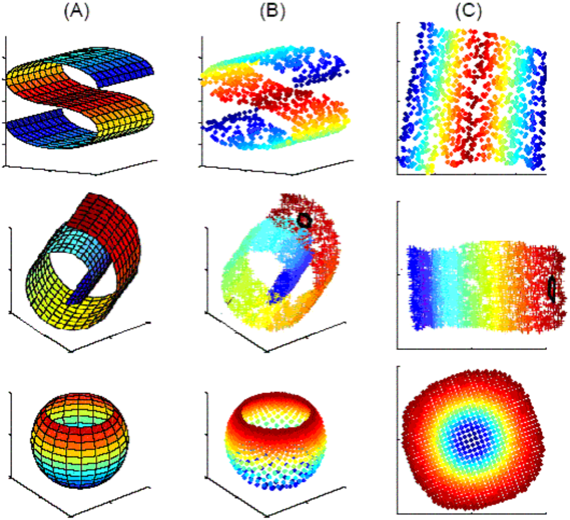


Рис.: Примеры данных[2]

[2]jntsai.blogspot.ru/2015/04/ammai-nonlinear-dimensionality.html

Подходы к снижению размерности

Сохранение расстояний

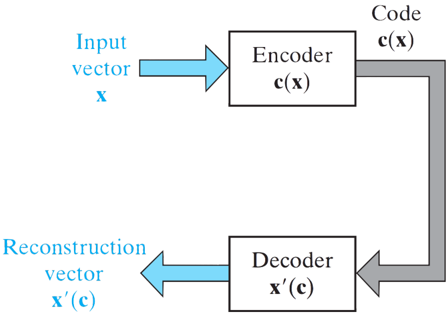
$$\sum_{i,j=1}^{N}(||X_i - X_j||_{\mathcal{X}} - ||y_i - y_j||_{\mathcal{P}})^2 \to min$$

На этом принципе основаны MDS (евклидово расстояние) и Isomap (топологическое расстояние)

Точность восстановления

$$\sum_{i=1}^{N}||X_i - X_i'||^2 \to min$$

На этом принципе построены Автокодировщики, SOM, PCA



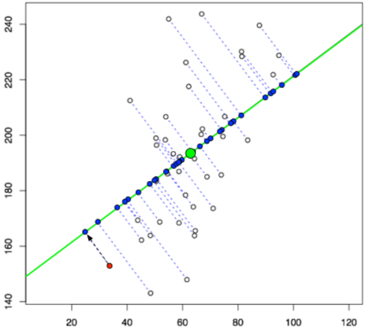
Метод главных компонент

PCA

Постановка задачи

Требуется найти гиперплоскость, задаваемую векторами $v_1, v_2, \dots v_d$, минимизирующую суммарное расстояние объектов до плоскости:

$$\sum_{n=1}^{N} ||h_n|| \to \min_{v_1, v_2, \dots v_d}$$



Решение задачи

План доказательства

1. Переход к задаче максимизации
   $||x||^2 = ||h||^2 + ||p||^2 \Rightarrow ||h|| \to min \Leftrightarrow ||p|| \to max$
   $\sum\limits_{n=1}^{N} ||h_n|| \to \min\limits_{v_1, v_2, \dots v_d} \Leftrightarrow \sum\limits_{n=1}^{N} ||p_n|| \to \max\limits_{v_1, v_2, \dots v_d}$

Решение задачи

План доказательства

1. Переход к задаче максимизации
   $||x||^2 = ||h||^2 + ||p||^2 \Rightarrow ||h|| \rightarrow min \Leftrightarrow ||p|| \rightarrow max$
   $\sum\limits_{n=1}^{N} ||h_n|| \rightarrow \min\limits_{v_1, v_2, \dots v_d} \Leftrightarrow \sum\limits_{n=1}^{N} ||p_n|| \rightarrow \max\limits_{v_1, v_2, \dots v_d}$
2. Жадное построение векторов
   $\begin{cases} ||Xa_k|| \rightarrow \max\limits_{a_k} \\ ||a_k|| = 1 \\ < a_i, a_k >= 0 \quad i = \overline{1, k-1} \end{cases}$

Решение задачи

План доказательства

1. Переход к задаче максимизации
$||x||^2 = ||h||^2 + ||p||^2 \Rightarrow ||h|| \to min \Leftrightarrow ||p|| \to max$
$\sum\limits_{n=1}^{N} ||h_n|| \to \min\limits_{v_1, v_2, \dots v_d} \Leftrightarrow \sum\limits_{n=1}^{N} ||p_n|| \to \max\limits_{v_1, v_2, \dots v_d}$
2. Жадное построение векторов
$$\begin{cases} ||Xa_k|| \to \max\limits_{a_k} \\ ||a_k|| = 1 \\ < a_i, a_k >= 0 \quad i = \overline{1, k-1} \end{cases}$$
3. Доказательство оптимальности жадного набора векторов
Сравним $L[a_1, a_2, \dots, a_{k-1}]$ с другим набором $L[b_1, b_2, \dots, b_{k-1}]$
$\sum\limits_{i=1}^{k-1} ||Xa_i|| \geq \sum\limits_{i=1}^{k-1} ||Xb_i||$ (индуктивное предположение)
$||Xa_k|| \geq ||Xb_k||$, т.к. a_k — решение оптимизационной задачи

PCA & SVD

Любая матрица может быть представлена в виде

$$X = U\Sigma V^T,$$

где
$U(N \times N)$ - ортогональная матрица левого сингулярного базиса (собственные вектора матрицы XX^T)
$V(D \times D)$ - ортогональная матрица правого сингулярного базиса (собственные вектора матрицы $X^T X$)
$\Sigma(N \times D)$ - диагональная матрица с сингулярными числами на главной диагонали (собственные значения $X^T X$)

Матрица главных компонет может быть вычислена:

$$XV = U\Sigma$$

Выбор размерности нового пространства

Критерий

Ошибка восстановления:

$$L(r) = ||X - \tilde{X}_r||_F^2 = \sum_{i=r}^{D} \sigma_i^2$$

Относительная ошибка восстановления:

$$\hat{L}(r) = \frac{||X - \tilde{X}_r||_F^2}{||X||_F^2} = \frac{\sum\limits_{i=r}^{D} \sigma_i^2}{\sum\limits_{i=1}^{D} \sigma_i^2}$$

Критерий: $\hat{L}(r) \geq \eta$, где $\eta \sim 0.95$

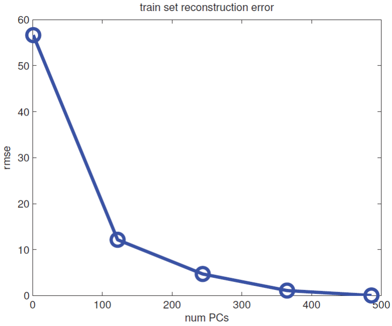
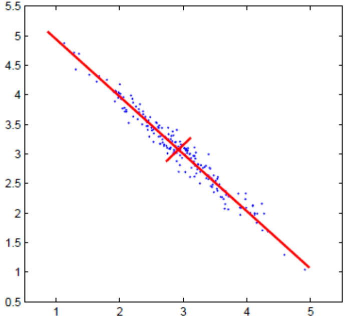
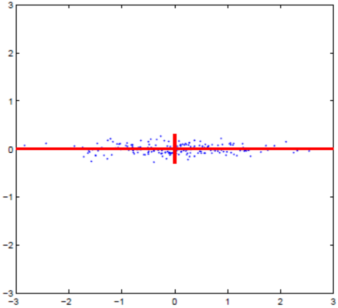


Иллюстрация PCA



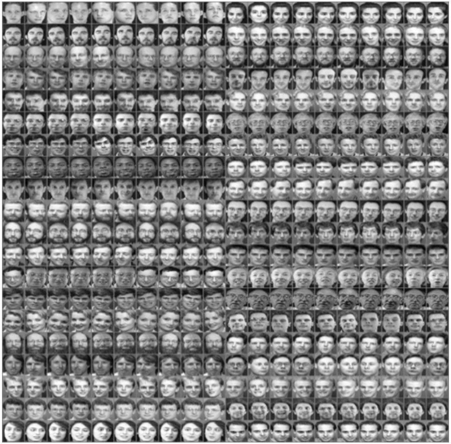
(a) Исходное пространство



(b) Итоговое пространство

- Сдвигаем начало координат в центр выборки
- Поворачиваем оси, чтобы признаки не коррелировали
- Избавляемся от координат с малой дисперсией

PCA для лиц[3]



PCA применяется для идентификации лиц людей. Для этого каждое лицо можно представить вектором координат в пространстве главных компонент.

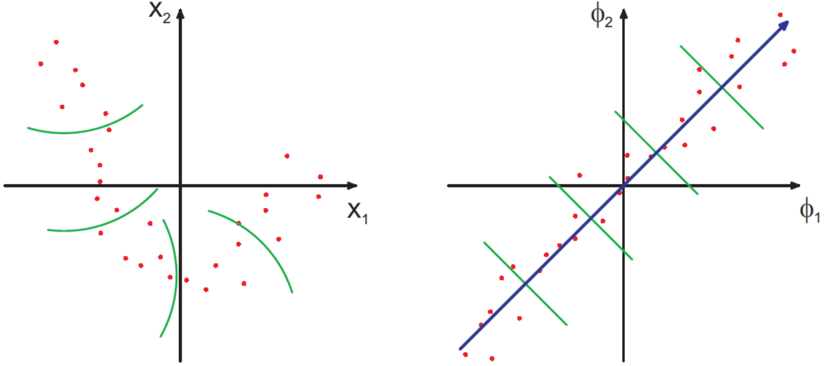
[3]https://en.wikipedia.org/wiki/Eigenface

Eigenfaces

Рис.: “Собственные” лица

Kernel PCA

Выберем некоторое нелинейное преобразование $\phi : R^D \to H$, при котором в новом простренстве нелинейное многообразие выборки переходит в гиперплоскость.



Ядра

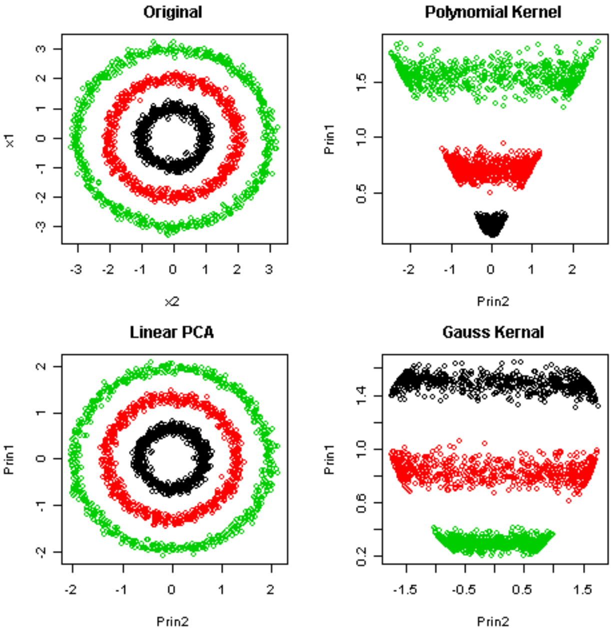
Ядро — скалярное произведение в новом пространстве $K(x, y) =< \phi(x), \phi(y) >$. Его использование не требует перехода в пространство H.

PCA: требуется знять только $X^T X = K(X, X)$

Примеры ядер

- Линейное: $x^T y$
- Полиномиальное: $(1 + x^T y)^d$
- Гаусово: $exp(-||x - y||^2/\sigma^2)$

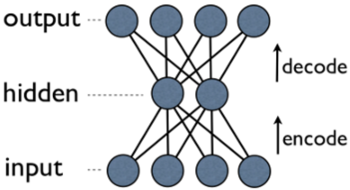
Пример Kernel PCA



Автокодировщики

Структура

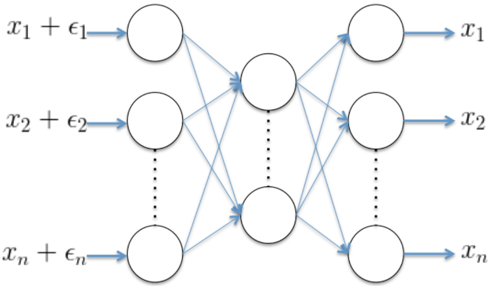
- Рассматривается сеть, обучаемая на отображении $f(x) = x$
- Внутри сети есть bottleneck слой, активации которого — представление объектов в низкоразмерном пространстве
- В сверточных сетях bottleneck достигается комбинацией pooling и unpooling



Применение

- Выделение признаков для других алгоритмов
- Снижение размерности
- Предобучение на неразмеченных данных

Denoising autoencoder



Примеры шума

- Нормальный шум: $\mathcal{N}\left(\mu, \sigma^2 I\right)$
- Маскирующий шум: часть элементов обнуляется
- Соль и перец: часть элементов принимают максимальное/минимальное допустимое значение

Denoising autoencoder

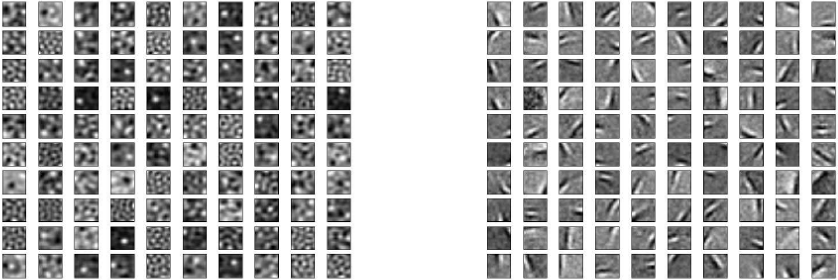
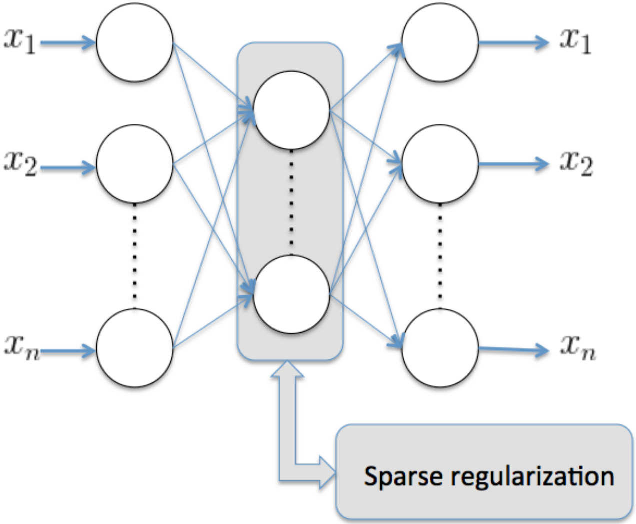


Рис.: Слева автоенкодер, справа автоенкодер с гауссовым шумом[4]

[4] Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, 2010, P. Vincent, Y. Bengio, and others

Разреженный автокодировщик



Sparse autoencoder

KL дивергенция

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- $KL(p||q) \geq 0$
- $KL(p||q) = 0 \Leftrightarrow p(x) = q(x)$ п.в

Регуляризатор разреженности

- Хотим, чтобы каждый нейрон в среднем активировался в ρ случаях ($\rho = 0.05$)
- Пусть средняя активация нейрона $\hat{\rho}$
- Регуляризатор: $\mathrm{KL}\left(\rho \| \hat{\rho}\right) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$

Sparse autoencoder[5]

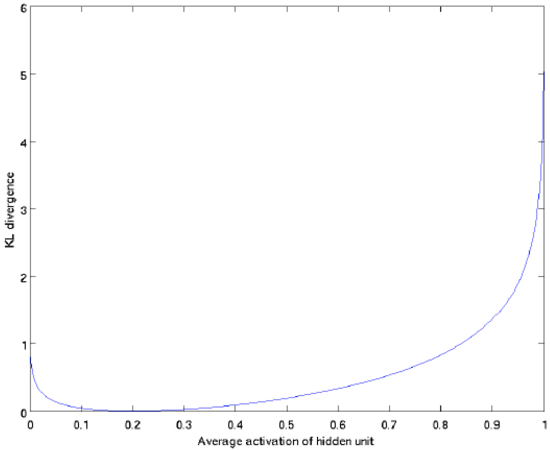
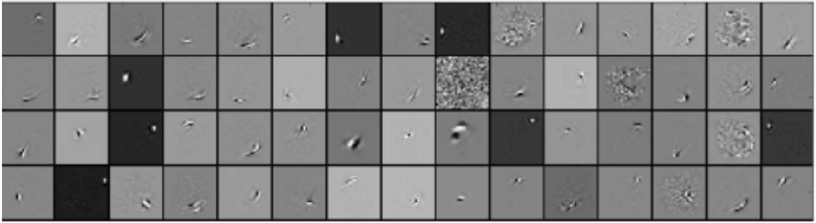
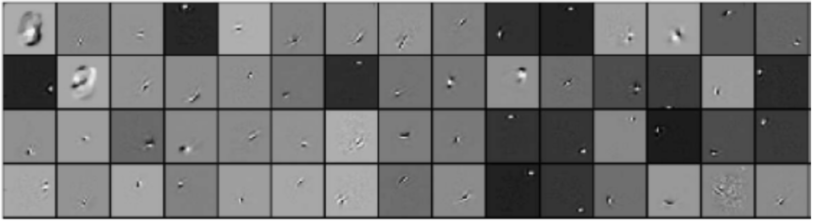


Рис.: KL достигает минимального значение в точке $\hat{\rho}_j = \rho$

[5] Sparse autoencoder, CS294A Lecture notes, Andrew Ng

Sparse autoencoder



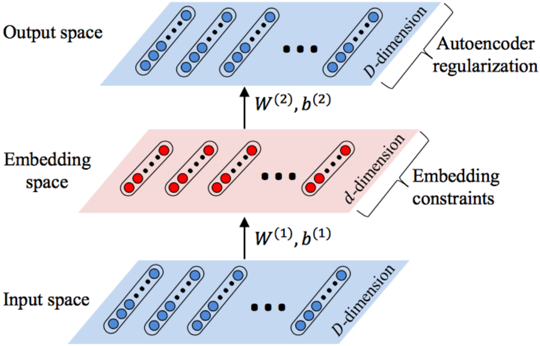


Embedding regularized AutoEncoder [6]

Совмещаем идею MDS и AE:

$$\begin{cases} \sum\limits_i ||X_i - \hat{X}_i||^2 \to min \\ \sum\limits_{i,j} (||X_i - X_j|| - ||E_i - E_j||)^2 \to min \end{cases}$$

В результате получаем более качественный embedding.



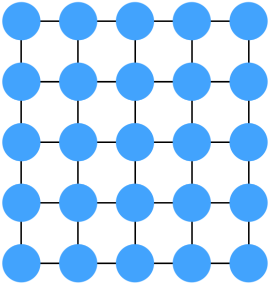
[6] http://www.ecmlpkdd2013.org/wp-content/uploads/2013/07/196.pdf

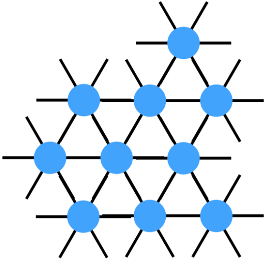
Самоорганизующиеся карты Кохонена

Описание структуры

Построим отображение узлов решетки (прямоугольной или шестиугольной) в пространство данных.
$\phi : \mathcal{A} \to \mathbb{R}^D$
$\phi(i) = m_i,\ i \in \mathcal{A}$

- Соседние узлы решетки должны быть близки после отображения
- Образ узлов должен "хорошо" приближать данные





Визуализация

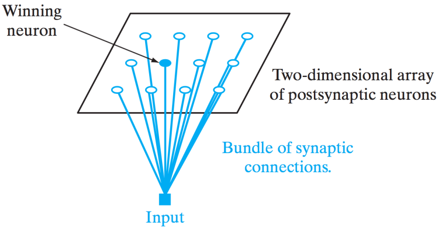
- 2D визуализация
- 3D визуализация

Шаги обучения SOM

- Соревнование — нейроны борются за право быть активированными (только один победитель)
- Кооперация — соседние с активным нейроны также активируются
- Адаптация — изменение положений образов

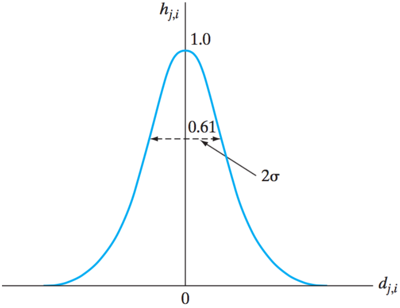
Соревнование

- $i(x) = \arg\min_{i \in \mathcal{A}} ||x - w_i||$
- $i(x)$ может быть рассмотрен как механизм внимания
- Эта часть алгоритма — кодирование, т.е. $\mathbb{R}^D \to \mathcal{A}$



Коопрация

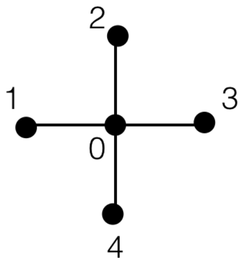
- Зададим значение активаций нейронов
- Более далекие от победителя нейроны получают меньшую активацию
- Расстояние на решетке: $d_{i,j}$
- Активация: $h_{j,i(x)} = \exp(-d_{ji}^2/2\sigma^2)$
- $\sigma(n) = \sigma_0 \exp(-n/\tau_1)$

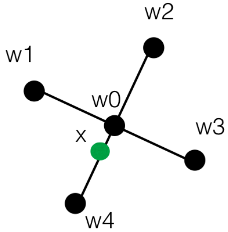


Адаптация

- $w_j = (1 - \eta h_{j,i(x)}) w_j + \eta h_{j,i(x)} x$
- $\eta(n) = \eta_0 exp(-n/\tau_2)$

Пример:
$||x - w_0|| = 0.1$
$||w_0 - w_1|| = ||w_0 - w_2|| = ||w_0 - w_3|| =$
$||w_0 - w_4|| = 1$
$\sigma = \eta = 1$
$\exp(-1) \simeq 0.4$
$\exp(-0.5) \simeq 0.6$
Найти вектор обновлений образов
w_0, w_1, w_2, w_3, w_4

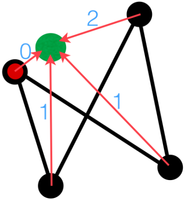




Фазы обучения

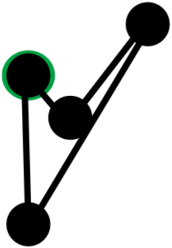
Этап сортировки

Карта производит топологическую сортировку своих узлов



Количество итераций: 1000
$\eta_0 = 0.1$, $\tau_2 = 1000$
$h_{j,i(x)}$ содержит почти все нейроны

Этап сходимости



Карта ищет оптимальное квантование исходного пространства

Количество итераций: $\sim 500 \cdot |\mathcal{A}|$
$0 < \eta \leq 0.01$
$h_{j,i(x)}$ содержит малую окрестность

Отображение модели

- U-matrix: каждому нейрону приписывается среднее расстояние до его топологических соседей

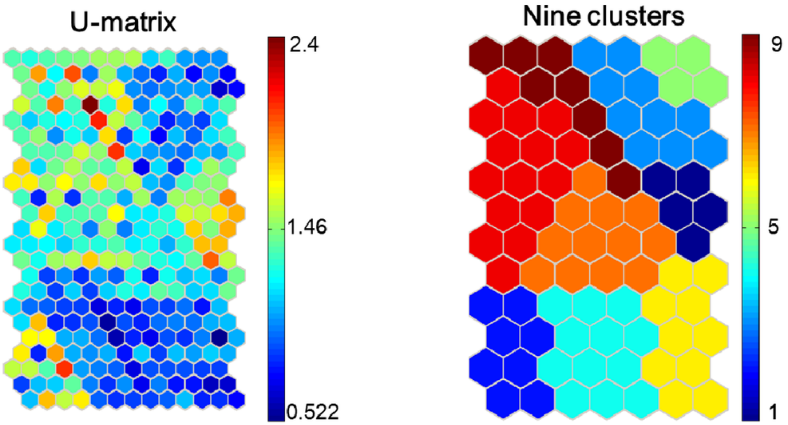


Рис.: U-matrix и k-means кластеризация в новом пространстве[7]

[7] http://www.mdpi.com/1660-4601/11/4/3618/htm

Уровень бедности

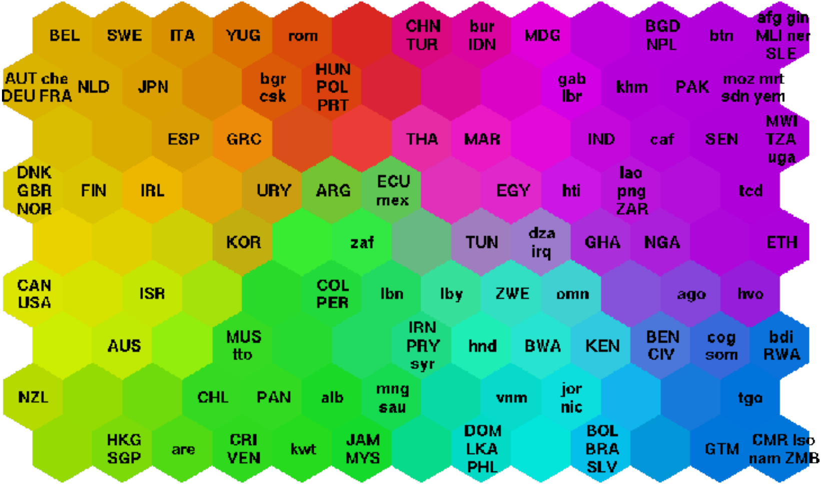


Рис.: SOM стран по 39 показателям World Bank[8]

[8]http://www.cis.hut.fi/research/som-research/worldmap.html

Уровень бедности

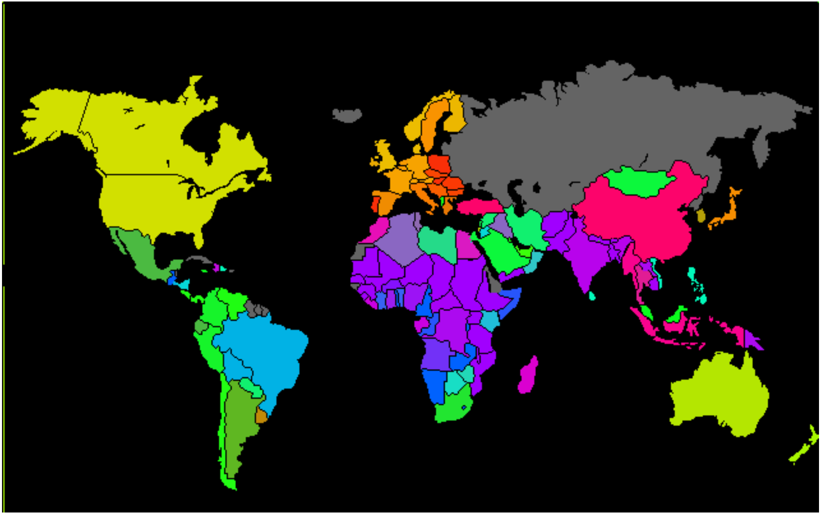


Рис.: Цвета кластеров перенесены на реальную карту[9]

[9] http://www.cis.hut.fi/research/som-research/worldmap.html

Вопросы