Embedding of embeddings
Data Mining Lab, SS13
University of Bonn

Presented by
Priyanka Dank

Supervised by
Dr. Thomas Gärtner
Daniel Paurat

What is Embedding?

Embedding is mapping from one space to the other.

In data analysis, embedding is often mapping high dimensional data to a lower dimensional space

Also known as dimensionality reduction

Useful for

- Visualization: to understand the structure of the data
- Generalization: fewer dimensions allows better generalization
- Efficiency: compress data for efficiency
- Noise Reduction

Idea of Embedding of embeddings

Embedding techniques:

- try to keep the underlying structure of the data in the space that the data gets embedded into
- emphasize different aspects of the data

We applied some of the more common techniques on different datasets to find out how similar the different techniques are to each other

This lead to the idea of ***Embedding of embeddings***

Goal : to place the embedding techniques on a 2d map such that the techniques that deliver more similar embeddings should be placed closer together

Example

Consider the dataset <u>auto93</u> from the UCI repository that stores some automobile data

City_MPG,	City_MPG	Highway_MPG	Air_Bags_standard	Drive_train_type	Number_of_cylinders	Horsepower	RPM	Engine_revolutions_per_mile	Manual_transmission_available	Passenger_	Length	Wheelbase	Width	U-turn_space	Luggage_capacity	Weight	Domestic
25,31,0,1,4	25	31	0	1	4	140	6300	2890	1	5	177	102	68	37	11	2705	0
18,25,2,1,6	18	25	2	1	6	200	5500	2335	1	5	195	115	71	38	15	3560	0
20,26,1,1,6	20	26	1	1	6	172	5500	2280	1	5	180	102	67	37	14	3375	0
19,26,2,1,6	19	26	2	1	6	172	5500	2535	1	6	193	106	70	37	17	3405	0
22,30,1,0,4	22	30	1	0	4	208	5700	2545	1	4	186	109	69	39	13	3640	0
22,31,1,1,4	22	31	1	1	4	110	5200	2565	0	6	189	105	69	41	16	2880	1
19,28,1,1,6	19	28	1	1	6	170	4800	1570	0	6	200	111	74	42	17	3470	1
16,25,1,0,6	16	25	1	0	6	180	4000	1320	0	6	216	116	78	45	21	4105	1
19,27,1,1,6	19	27	1	1	6	170	4800	1690	0	5	198	108	73	41	14	3495	1

Application of different embedding techniques on the dataset resulted into:

dimensionality reduction:82 x 2 from original dimensions : 82 x 21

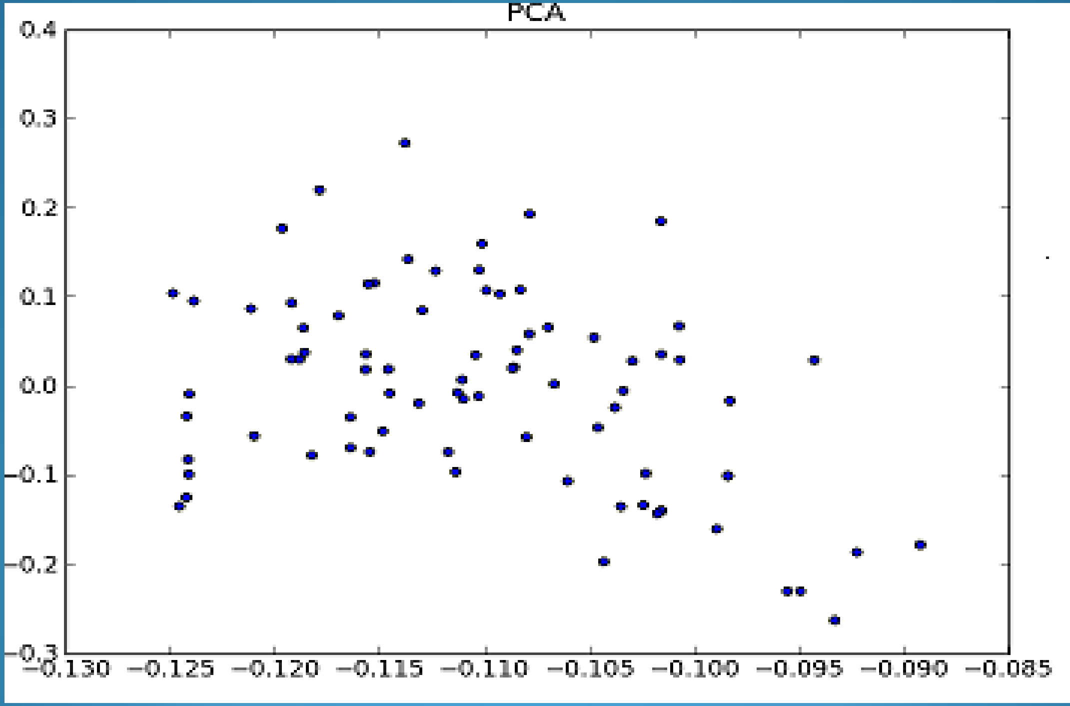
Let´s have a look at some of the techniques

Principle Component Analysis (PCA)

Finds orthogonal axes with the highest variance in high dimensional space

Uses the n most dominant ones to embed the data into a low dimensional space

PCA on auto93

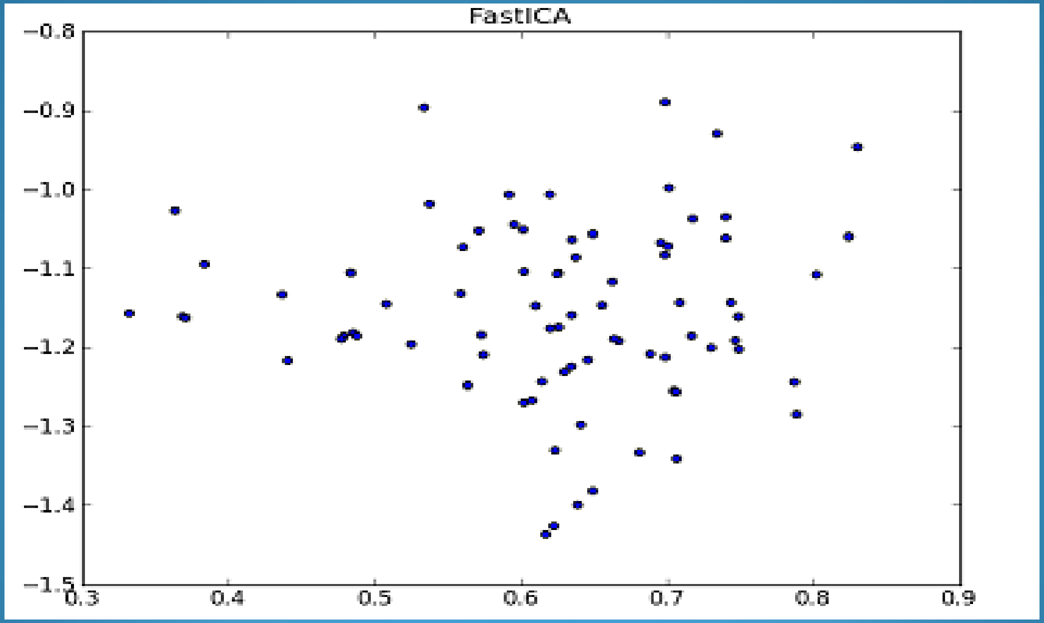


Fast Independent Component Analysis (FastICA)

ICA represents multidimensional random vector as a linear combination of non-gaussian random variables i.e. independent components

FastICA is a computationally highly efficient method for performing the estimation of ICA

<u>FastICA on auto93</u>



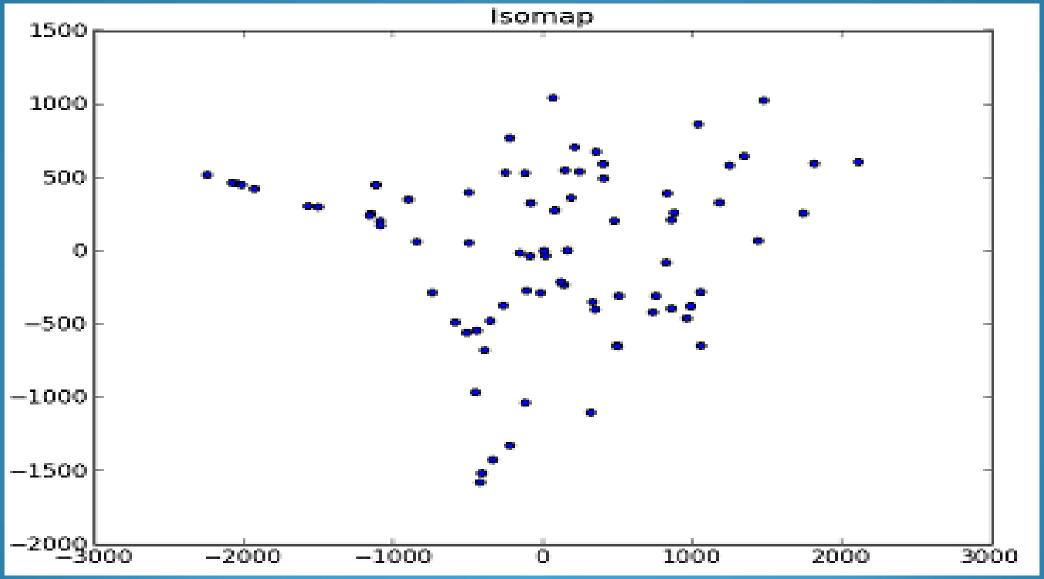
Isomap

Finds the neighbors of each data point in high dimensional data space

Computes the pairwise distances based on nearest-neighbors between all points

Embeds the data via Multidimensional Scaling which finds vectors $x_1, \ldots, x_I$ in low dimensions such that the distance relations are preserved

Isomap on auto93

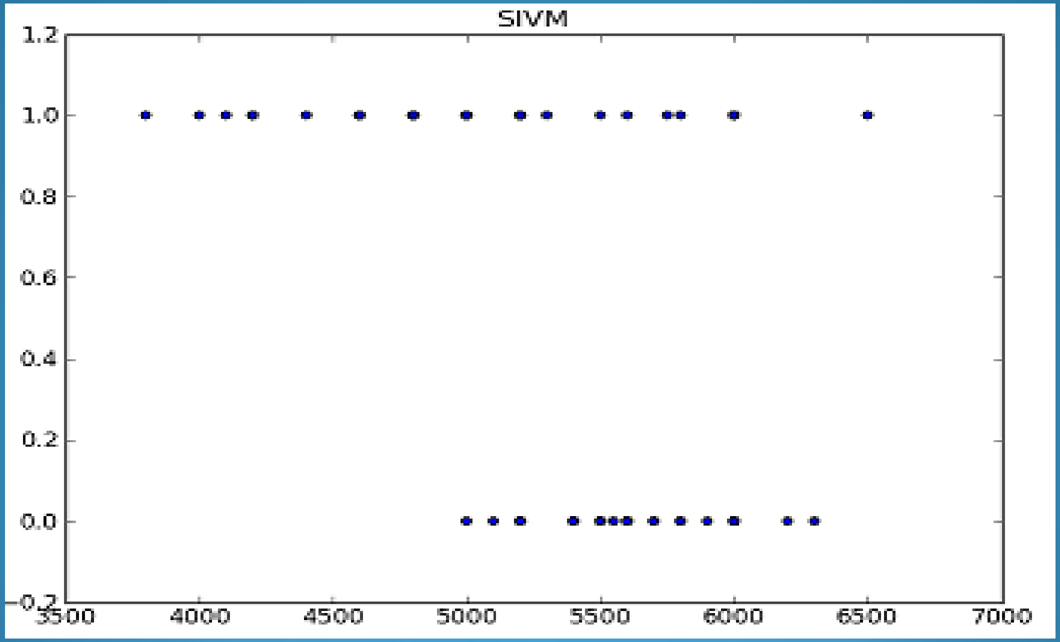


Simplex Volume Maximization (SiVM)

Selects opposing points of the dataset´s convex hull so that points couldn´t be further apart

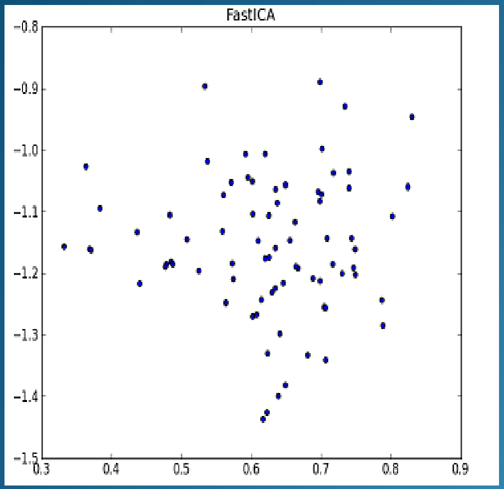
The embedding is done by expressing all data points as convex combination of these extreme points

SiVM on auto93

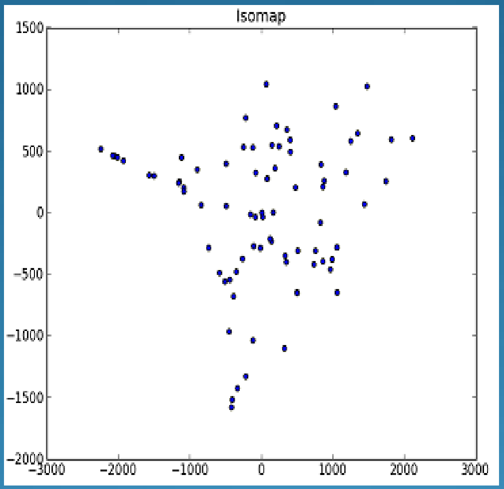


Observations

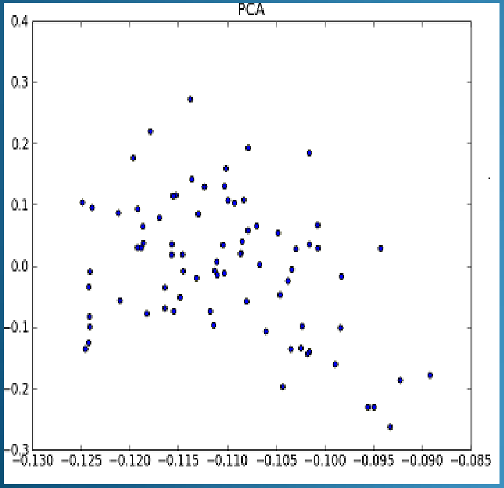
FastICA



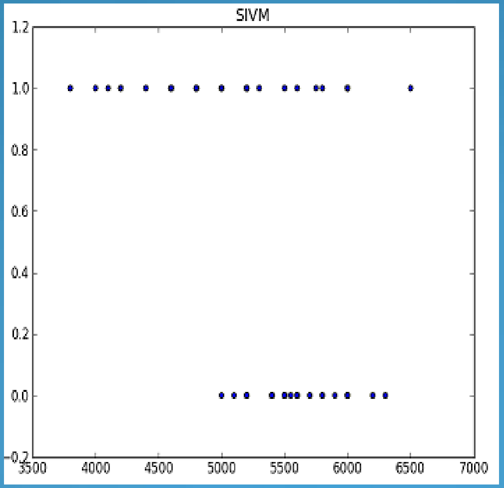
Isomap



PCA



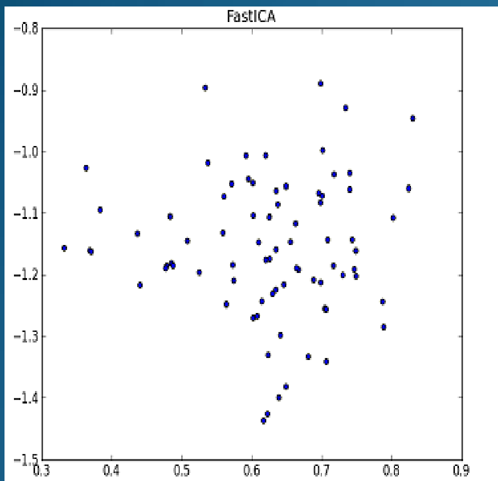
SiVM



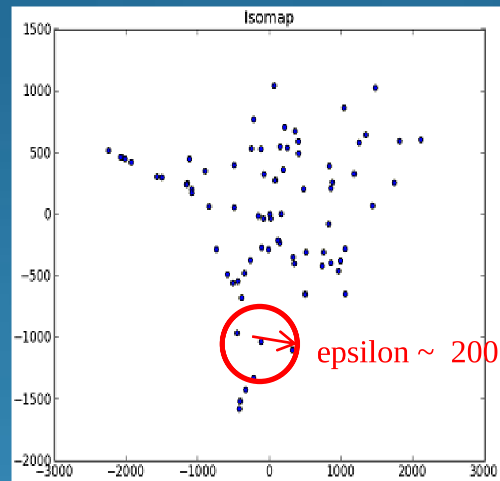
How to compare these techniques?

Analysis(1)

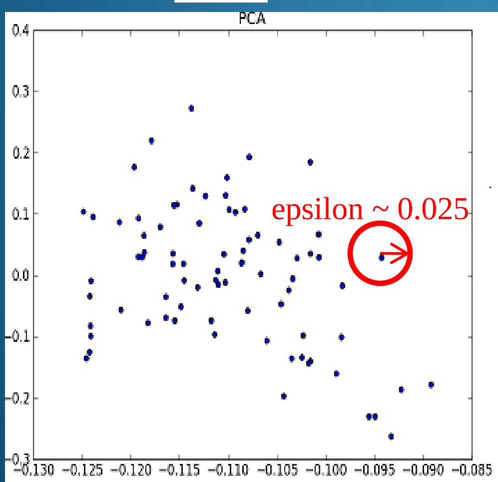
FastICA



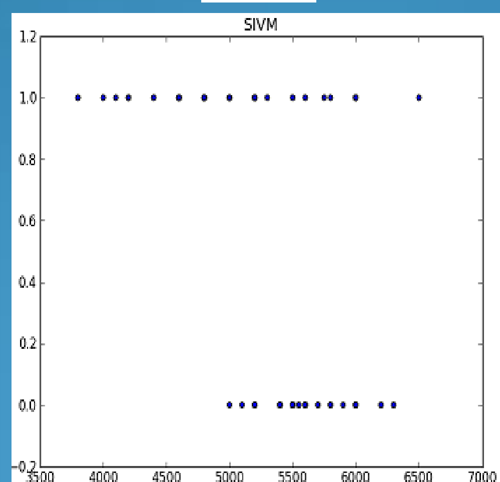
Isomap



PCA



SiVM



Different Scales:
FastICA [-1.5 : -0.8]
Isomap[-2000 :1500]
PCA [-0.3 : 0.4]
SiVM [-0.2 : 1.2]

Different epsilon-radius for each data point :bring all embeddings to one scale

Feasible option : Scale independent measure e.g. k-nearest- neighbors

Analysis (2)

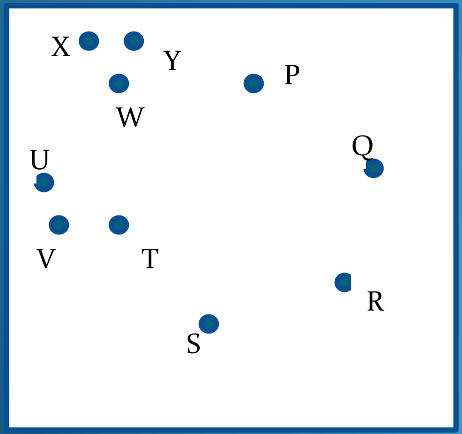
We decided to build a similarity measure based on the k-nearest-neighbors

Let´s simplify the problem to find similarity measure by formulating an example with less number of data points.

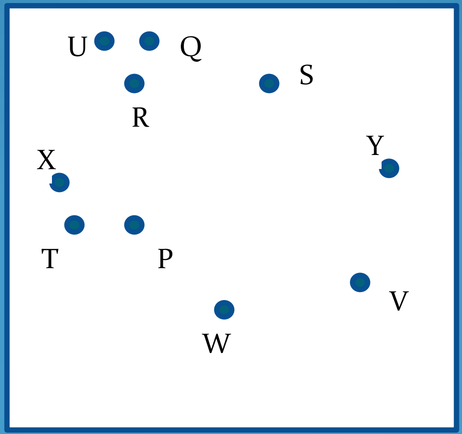
Consider three embedding techniques A,B,C which map 10 data points {P,Q,R,S,T,U,V,W,X,Y} to lower dimensions

The mappings by each technique might look like as follows:

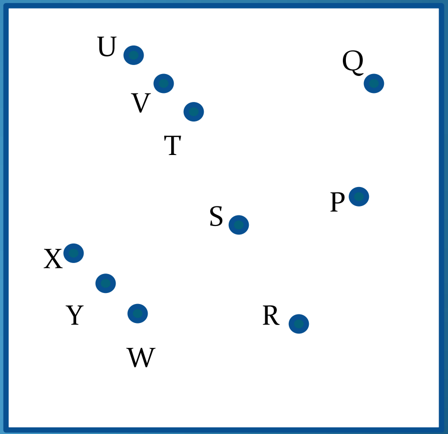
A



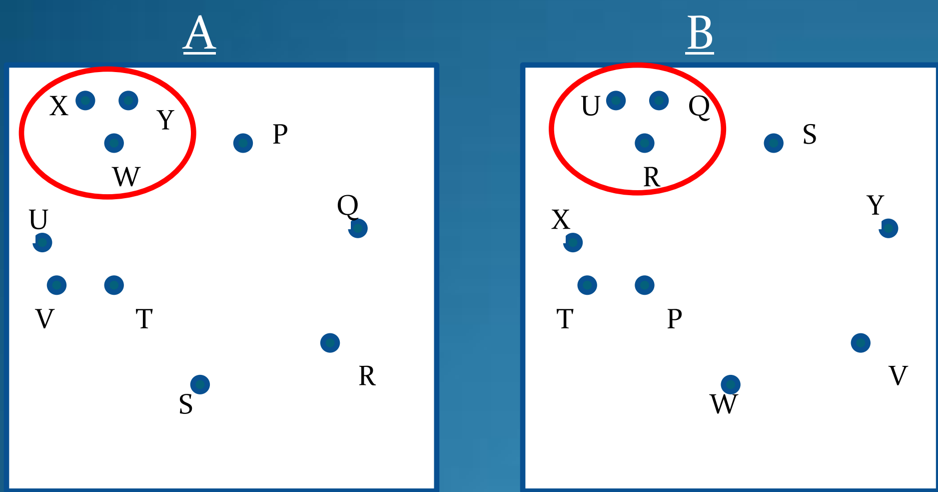
B



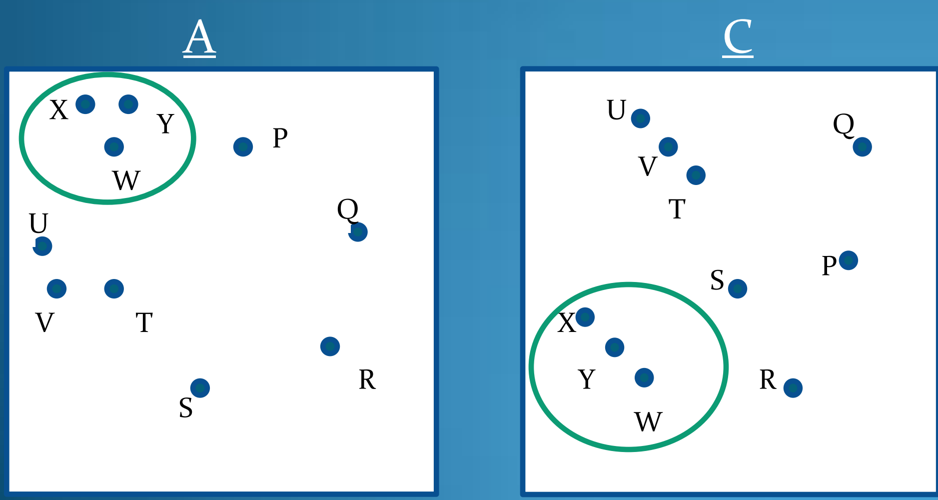
C



Analysis (3)



The output of A and B look similar but the neighborhood of the mappings are different



The output of A and C look different but the neighborhood of the mappings are similar

Similarity Measure (1)

Let´s call the data point and its k neighbors as a knn set

We run our program for different values of k

We found that for the size of our datasets:

- A setting of k=5 performs well
- Below k=5, the effect of noise becomes more dominant
- Much above k=5, the overall similarities tend to become more and more similar

Similarity Measure (2)

Compared each i^{th} data point´s knn set of one method with i^{th} datapoint´s knn set of remaining methods

Computed their intersection and union

Formulated the equation for similarity measure as

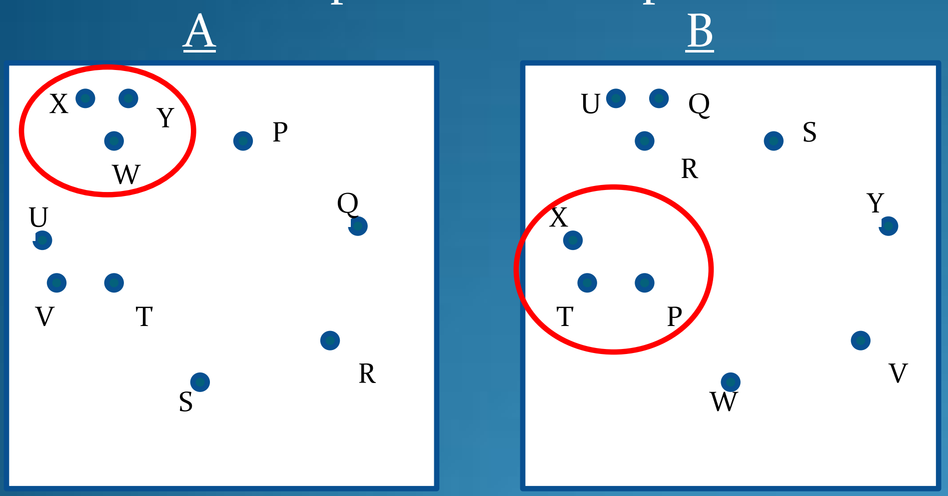
$$sim\_k(m1, m2) = \frac{1}{|D|} \sum_{x \in D} \frac{|(knn(m1(x))) \cap (knn(m2(x)))|}{|(knn(m1(x))) \cup (knn(m2(x)))|}$$

Where
m1, m2 are compared embeddings
D is the set of datapoints

Resulting sim_k(m1,m2) will lie between 0 to 1

Illustration of Similarity Measure (1)

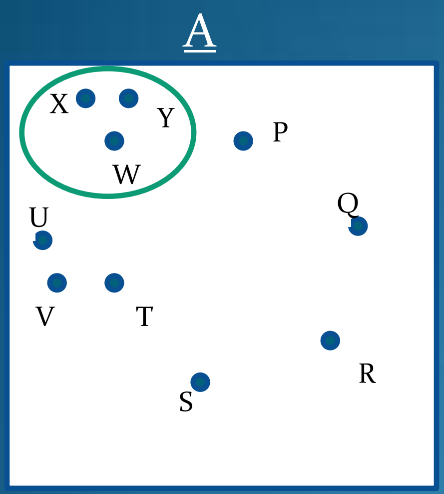
Consider our previous example

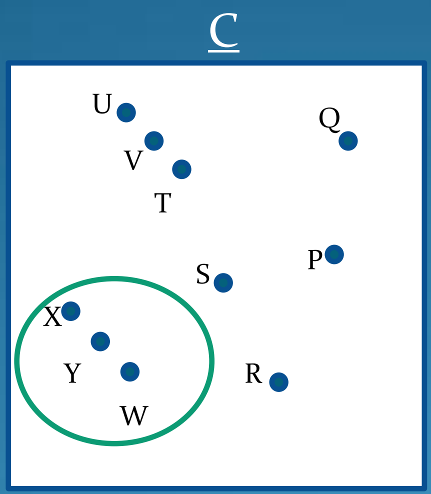


$$\text{sim\_3(A, B)} = \frac{1}{10}\left[\frac{|(X,Y,W)\cap(X,T,P)|}{|(X,Y,W)\cup(X,T,P)|} + \frac{|(Y,X,W)\cap(Y,S,V)|}{|(Y,X,W)\cup(Y,S,V)|} + .. + \frac{|(R,Q,S)\cap(R,Q,U)|}{|(R,Q,S)\cup(R,Q,U)|}\right]$$

$$\text{sim\_3(A, B)} = 0.2$$

Illustration of Similarity Measure (2)





$$\mathrm{sim\_3(A,C)} = \frac{1}{10}\left[\frac{|(X,Y,W)\cap(X,Y,W)|}{|(X,Y,W)\cup(X,Y,W)|} + \frac{|(Y,X,W)\cap(Y,X,W)|}{|(Y,X,W)\cup(Y,X,W)|} + .. + \frac{|(R,Q,S)\cap(R,S,P)|}{|(R,Q,S)\cup(R,S,P)|}\right]$$

$$\mathrm{sim(A,C)\_3} = 0.7$$

Thus, sim_3(A,B) which is 0.2 is less than sim_3(A,C)
A and C are more similar embedding techniques

Used Similarity Measures

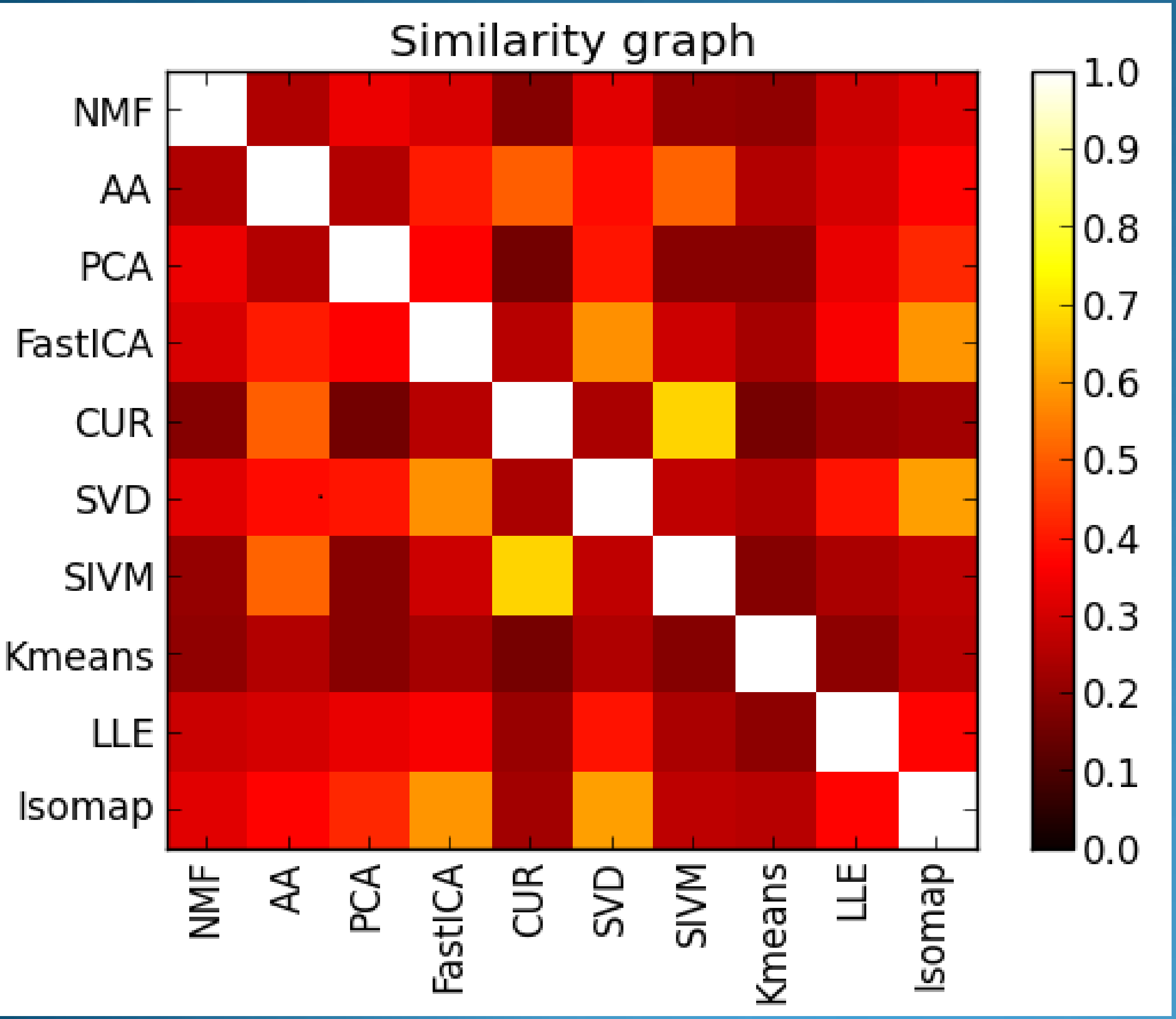
We applied 10 different embedding techniques on auto93

1. PCA [Pearson 1901]
2. FastICA [Hyvärinen 2000]
3. Isomap [Tenenbaum 2000]
4. SiVM [Thurau, Kersting, Bauckhage 2010]
5. CUR matrix decomposition [Drineas 2006]
6. SVD (Singular Value Decomposition) [Yang, Ma, Buja 2011]
7. NMF (Non-negative Matrixfactorization) [Cho, Saul 2011]
8. Kmeans [Ding 2007]
9. AA(Archetypal Analysis) [Cutler 1994]
10. LLE (Locally Linear Embedding) [Roweis, Saul 2000]

Obtained the lower dimensional mappings

Compared the results using our similariry measures and obtained the pairwise similarity graph

Similarity Comparison on auto93



Each method is similar to itself and hence have highest simlarity index

Isomap and SVD are similar to each other and hence have high similarity index

PCA and CUR are totally different and hence have less similarity index

Average Similarity

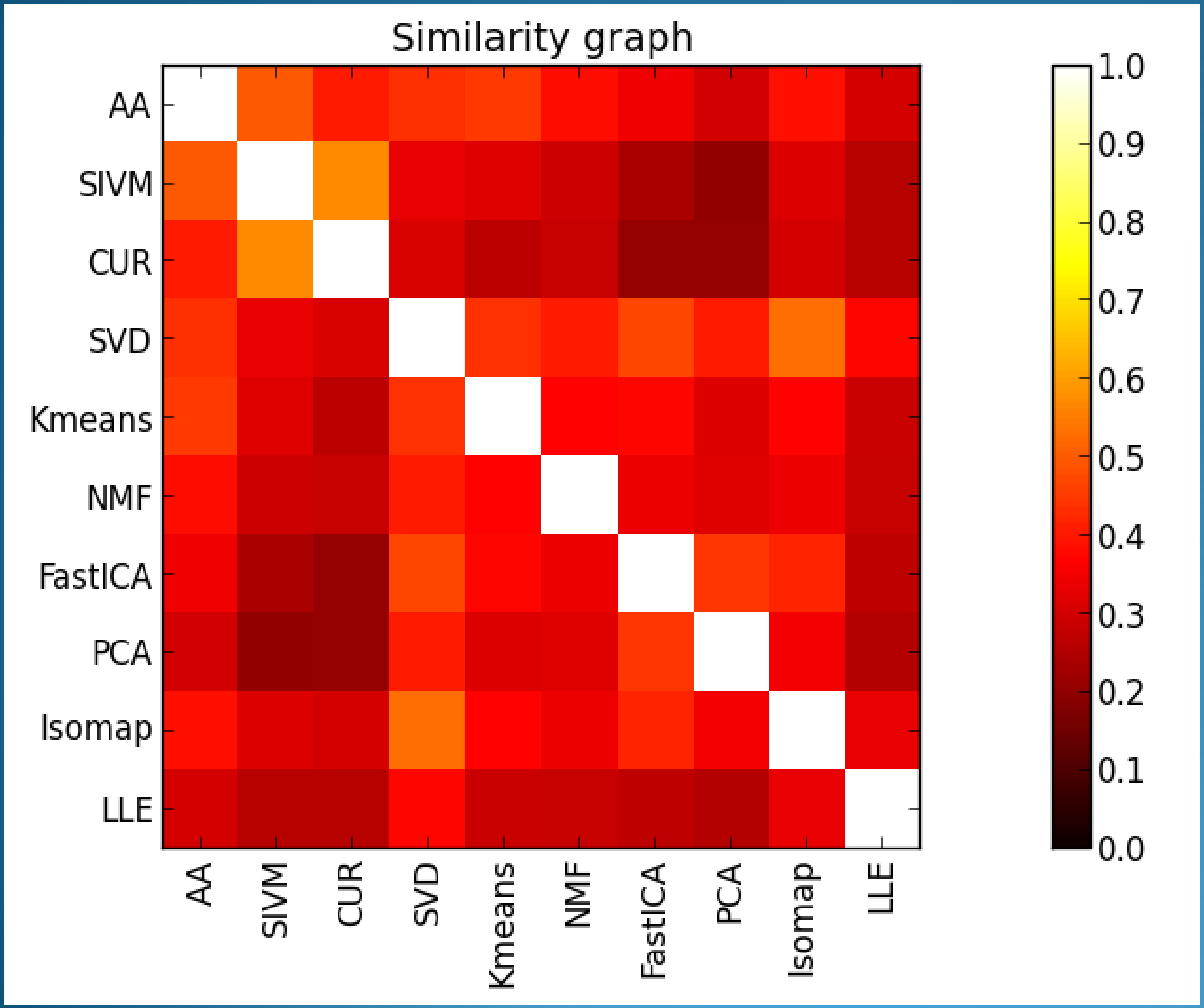
In order to generalize, we applied our algorithm of similarity measurement on 20 databases from UCI repository

Took average of all obtained similarity indices using formula

$$\text{avg\_sim\_k}(m1, m2) = \frac{1}{no.\ of\ datasets} \sum_{\text{for each dataset}} \sum_{\text{for each (m1,m2)}} sim(m1, m2)$$

Plotted similarity graph which illustrates the comparison between embedding techniques

Average Similarity Comparison



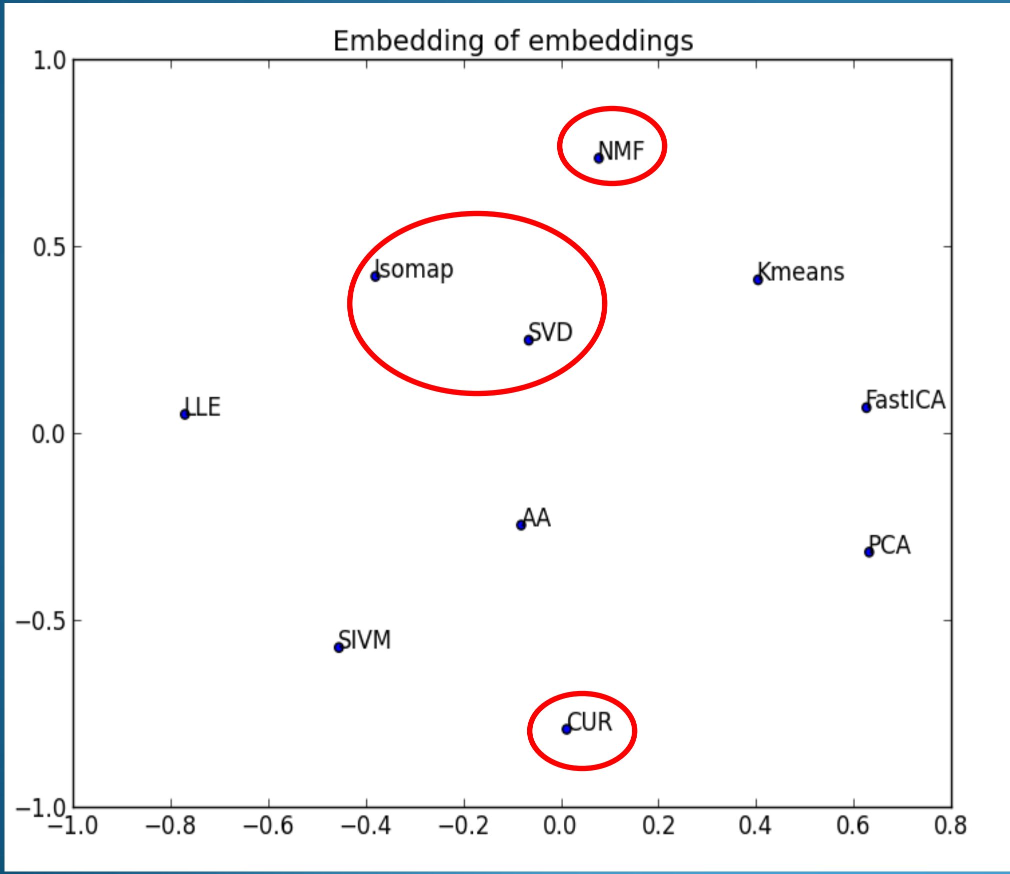
Each method is similar to itself and hence have highest simlarity index

Isomap and SVD are similar to each other and hence have high similarity index

PCA and CUR are different and hence have less similarity index

Embedding of Embeddings

As project title suggests, we embedded the compared embedding techniques in 2D map



<u>NMF and CUR are at large distance</u>

<u>Isomap and SVD are nearby</u>

Normalization (1)

Provides an easy way to compare the values that are measured using different scales (for example degrees Celsius and degrees Fahrenheit)

As Embeddings are done on different scales, we decided to normalize the input data before processing for better comparison.

Used two common normalization techniques:

- Min-Max normalization
- Mean shift and divide by variance

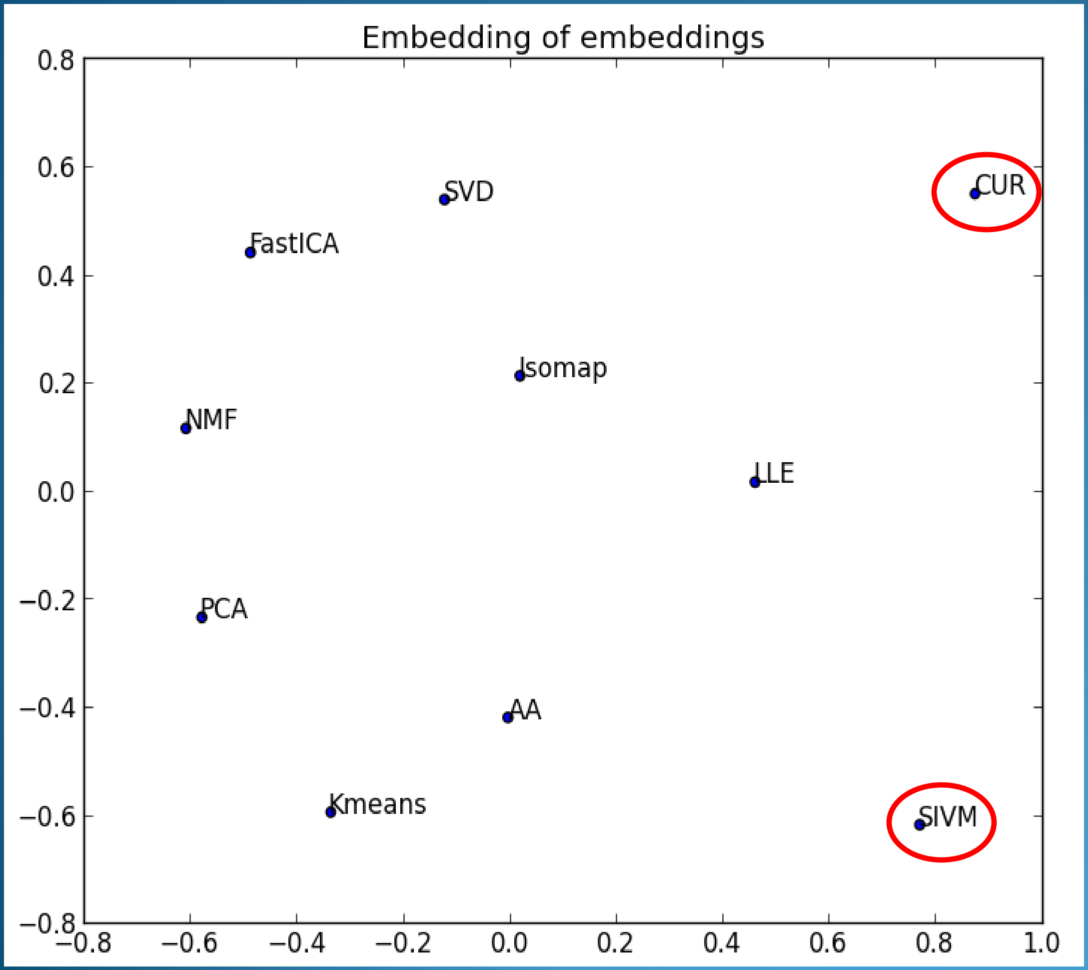
Min-Max normalization :data is fitted in the scale [0,1] and formula for the normalization of point A as

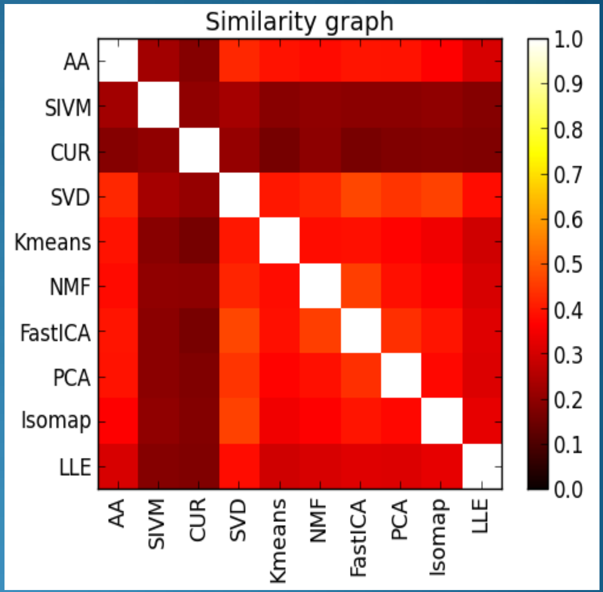
$$B = \frac{A - \min(A)}{\max(A - \min(A))}$$

where B is the value of A after normalization

Min-Max Normalization

After applying Min-Max normalization we got the plot as

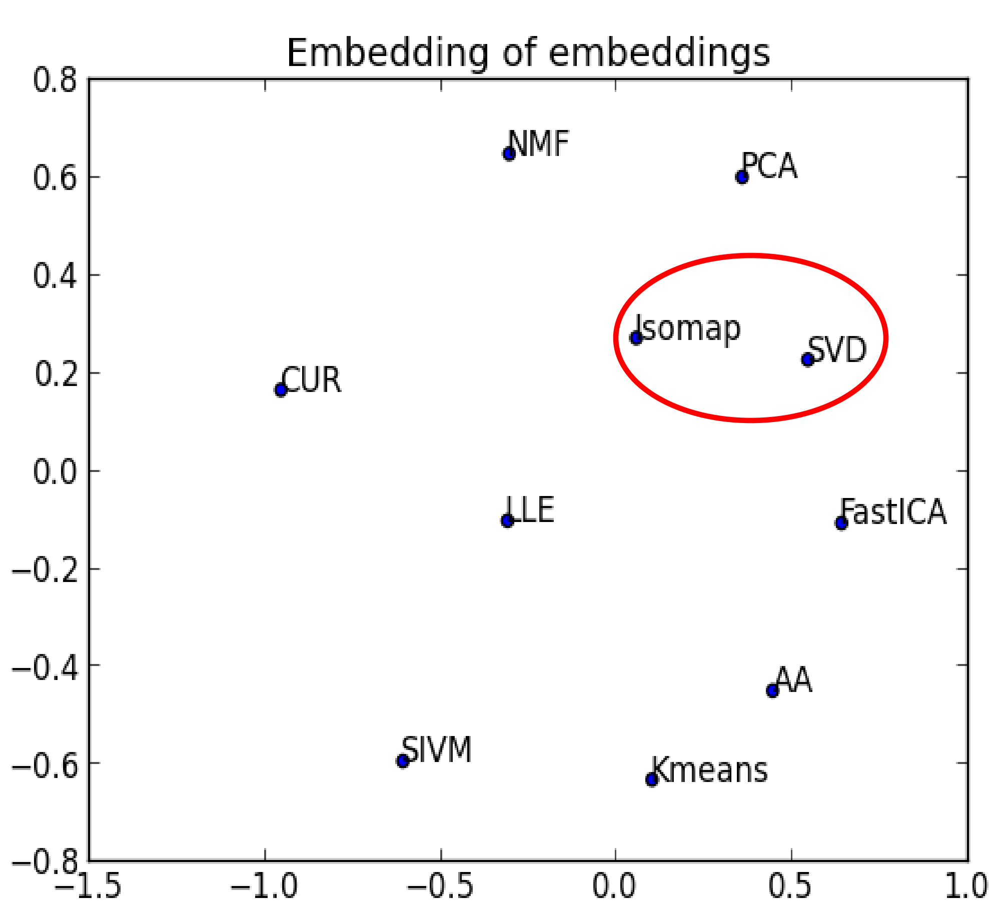


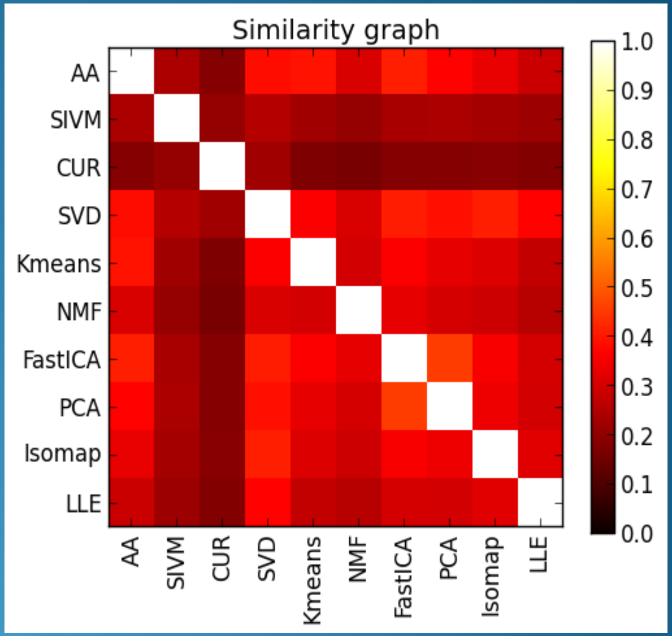


CUR and SiVM seem to be much further apart from the rest

Mean Shift & Divide by Variance Normalization

In Mean shift, divide by variance, formula used $$B = \frac{A - \mathrm{Mean}(A)}{\mathrm{variance}(A)}$$





This normalization seems to make Isomap and SVD closer

Summary

Embedding techniques in data mining map higher dimensional data to lower dimensions using different scales

Embedding techniques try to keep the underlying structure of the data in the space that the data gets embedded into

We embedded the embedding techniques in 2D map such that similar techniques are placed close to each other

The type of normalization seems to have a significant impact on the similarity of the different techniques

References (1)

Pearson, K. (1901), 'On Lines and Planes of Closest Fit to Systems of Points in Space', *Philosophical Magazine* **2**, 559-572.

Hyvärinen, A. & Oja, E. (2000), 'Independent component analysis: algorithms and applications', *Neural Networks* **13** (4-5), 411-430.

Tenenbaum, J.; Silva, V. & Langford, J. (2000), 'A global geometric framework for nonlinear dimensionality reduction', *Science* **290** (5500), 2319--2323.

Thurau, C.; Kersting, K. & Bauckhage, C., Yes We Can - Simplex Volume Maximization for Descriptive Web Scale Matrix Factorization. In CIKM, 2010.

Yang, D., Ma, Z., & Buja, A. (2011) A sparse SVD method for high-dimensional data arXiv: 1112.2433.

References (2)

Drineas, P.; Kannan, R. & Mahoney, M. W. (2006), 'Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition', *SISC* **36** (1), 184-206.

Cho, Y. & Saul, L. K. (2011), 'Nonnegative Matrix Factorization for Semi-supervised Dimensionality Reduction', *CoRR* **abs/1112.3714**.

Ding, C. H. Q. & Li, T. (2007), Adaptive dimension reduction using discriminant analysis and K-means clustering, *in* Zoubin Ghahramani, ed., 'ICML' , ACM, pp. 521-528

Cutler, A. & Breiman, L. (1994), 'Archetypal Analysis', *Technometrics* Vol. 36, No. 4

Roweis, S. T. & Saul, L. K. (2000), 'Nonlinear Dimensionality Reduction by Locally Linear Embedding',*Science* **290** (5500), 2323-2326.

Thank You
for your attention