Universität Bielefeld

Cognitive Interaction Technology Center of Excellence

Linear and Non-Linear Dimensionality Reduction

Alexander Schulz
aschulz(at)techfak.uni-bielefeld.de

University of Pisa, Pisa

04.05.2015 and 07.05.2015

Overview

Dimensionality Reduction

- Motivation
- Linear Projections
- Linear Mappings in Feature Space
- Neighbor Embedding
- Manifold Learning

Advances in Dimensionality Reduction

- Speedup for Neighbor Embeddings
- Quality Assessment of DR
- Feature Relevance for DR
- Visualization of Classifiers
- Supervised Dimensionality Reduction

Curse of Dimensionality[1]

[1] [Lee and Verleysen, 2007]

Curse of Dimensionality[1]

- high-dimensional spaces are almost empty

[1] [Lee and Verleysen, 2007]

Curse of Dimensionality[1]

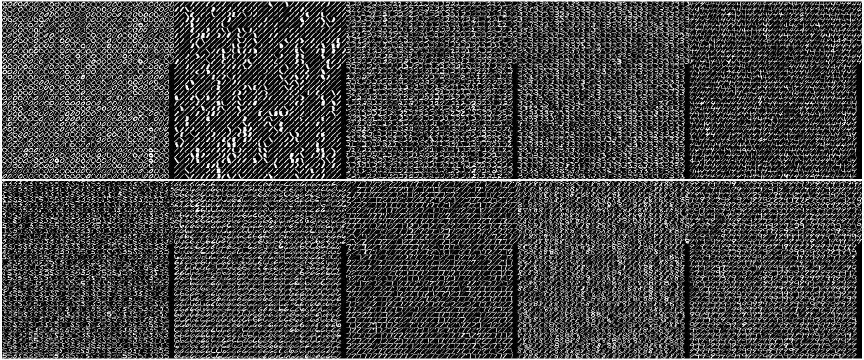
- high-dimensional spaces are almost empty
- Hypervolume concentrates in a thin shell close to the surface

[1] [Lee and Verleysen, 2007]

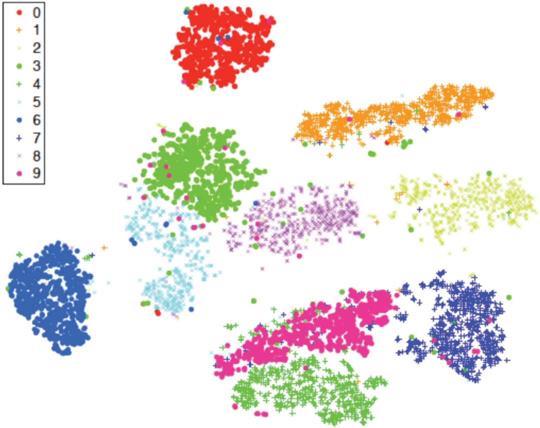
Why use Dimensionality Reduction?

0 0 0 0 0 0 0 0 0 0 0 0 0 0 81 34 0 0 0 0 0 0 0 0 0 0 0 0 3 104 247 153 0 0 0 0 0 0 0 0 0 0 0 0 70 255 255 175 0 0 0 0 0 0 0 0 0 0 14 54 227 255 255 98 0 0 0 0 0 0 0 0 0 0 168 255 255 255 179 7 0 0 0 0 0 0 0 0 33 157 254 255 247 162 13 0 0 0 0 0 0 0 0 72 207 255 255 255 124 0 0 0 0 0 0 0 0 29 149 250 255 255 228 160 2 0 0 0 0 0 0 0 6 148 255 255 255 245 112 0 0 0 0 0 0 0 0 56 195 255 255 255 169 15 0 0 0 0 0 0 0 0 5 205 255 254 208 61 10 0 0 0 0 0 0 0 0 39 84 255 219 127 0 0 0 0 0 0 0 0 0 0 3 186 234 211 41 0 0 0 0 0 0 0 0 0 0 0 105 255 230 15 0 0 0 0 0 0 0 0 0 0 0 0 175 224 48 0 0 0 0 0 0 0 0 0 0 0 0 0 63 52 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Why use Dimensionality Reduction?



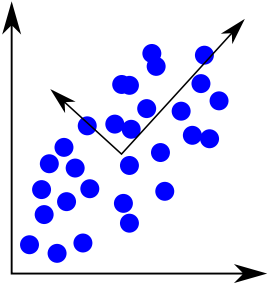
Why use Dimensionality Reduction?



Principal Component Analysis (PCA)

variance maximization

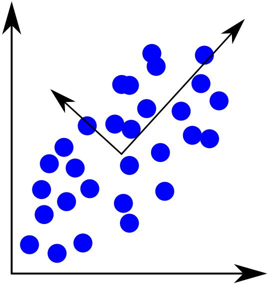
- $var\left(\mathbf{w}^\top \mathbf{x}_i\right)$ with $\|\mathbf{w}\| = 1$



Principal Component Analysis (PCA)

variance maximization

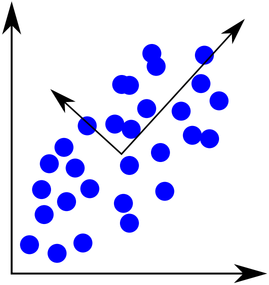
- $var\left(\mathbf{w}^\top \mathbf{x}_i\right)$ with $\|\mathbf{w}\| = 1$
- $= \frac{1}{N}\sum_i (\mathbf{w}^\top \mathbf{x}_i)^2$
- $= \frac{1}{N}\sum_i \mathbf{w}^\top \mathbf{x}_i {\mathbf{x}_i}^\top \mathbf{w}$



Principal Component Analysis (PCA)

variance maximization

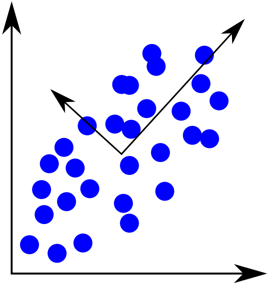
- $var\left(\mathbf{w}^\top \mathbf{x}_i\right)$ with $\|\mathbf{w}\| = 1$
- $= \frac{1}{N}\sum_i (\mathbf{w}^\top \mathbf{x}_i)^2$
- $= \frac{1}{N}\sum_i \mathbf{w}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}$
- $= \mathbf{w}^\top \mathbf{C} \mathbf{w}$



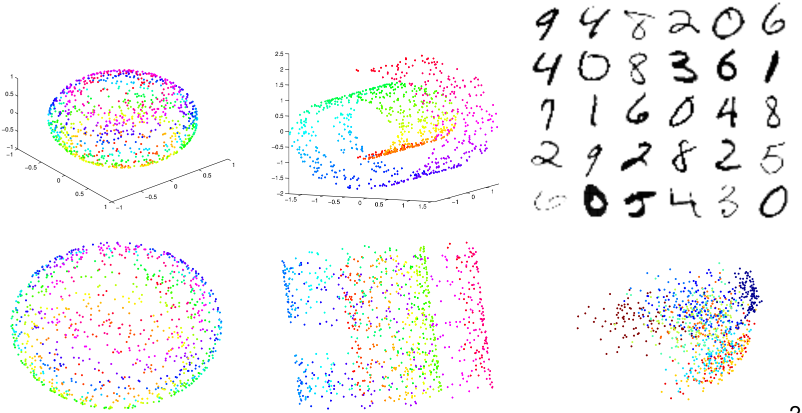
Principal Component Analysis (PCA)

variance maximization

- $var\left(\mathbf{w}^\top \mathbf{x}_i\right)$ with $\|\mathbf{w}\| = 1$
- $= \frac{1}{N}\sum_i (\mathbf{w}^\top \mathbf{x}_i)^2$
- $= \frac{1}{N}\sum_i \mathbf{w}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}$
- $= \mathbf{w}^\top \mathbf{C}\mathbf{w}$
- $\rightarrow$Eigenvectors of the covariance matrix are optimal



PCA in Action



[2]

---

[2] [Gisbrecht and Hammer, 2015]

Distance Preservation (CMDS)

- distances $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$
- objective $\delta_{ij} \approx d_{ij}$

Distance Preservation (CMDS)

- distances $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$
- objective $\delta_{ij} \approx d_{ij}$
- distances d_{ij} and similarities s_{ij} can be transformed into each other
- $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ matrix of pairwise similarities

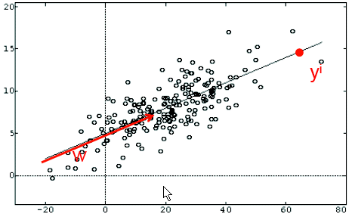
Distance Preservation (CMDS)

- distances $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$
- objective $\delta_{ij} \approx d_{ij}$
- distances d_{ij} and similarities s_{ij} can be transformed into each other
- $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ matrix of pairwise similarities
- best low rank approximation of $\mathbf{S}$ in Frobenius norm is $\mathbf{S} = \mathbf{U}\tilde{\boldsymbol{\Lambda}}\mathbf{U}^\top$ with the largest eigenvalue

Manifold Learning

learn linear manifold

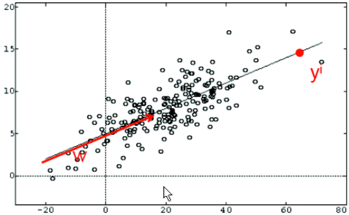
- represent data as projections on unknown **w**:
  $C = \frac{1}{2N} \sum_i (\mathbf{x}_i - y_i \mathbf{w})^2$



Manifold Learning

learn linear manifold

- represent data as projections on unknown **w**: $C = \frac{1}{2N} \sum_i (\mathbf{x}_i - y_i \mathbf{w})^2$
- What are the best parameters y_i and **w**?



PCA

three ways to obtain PCA

- maximize variance of a linear projection
- preserve distances
- find a linear manifold such that errors are minimal in an L2 sense

Kernel PCA[3]

Idea

- Apply a fixed nonlinear preprocessing $\phi(\mathbf{x})$
- Perform standard PCA in feature space

[3] [Schölkopf et al., 1998]

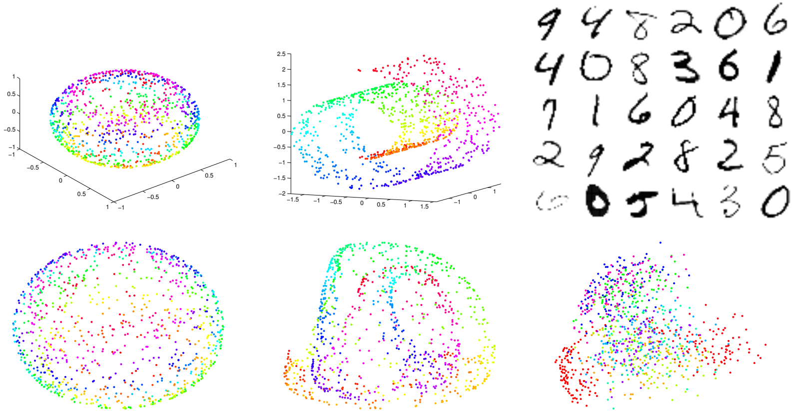
Kernel PCA[3]

Idea

- Apply a fixed nonlinear preprocessing $\phi(\mathbf{x})$
- Perform standard PCA in feature space
- How to apply the kernel trick here?

[3] [Schölkopf et al., 1998]

Kernel PCA in Action



[4]

[4][Gisbrecht and Hammer, 2015]

Stochastic Neighbor Embedding (SNE)[5]

- introduce a probabilistic neighborhood in the input space

$$p_{j|i} = \frac{\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_i^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_k\|^2/\sigma_i^2)}$$

[5] [Hinton and Roweis, 2002]

Stochastic Neighbor Embedding (SNE)[5]

- introduce a probabilistic neighborhood in the input space

$$p_{j|i} = \frac{\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_i^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_k\|^2/\sigma_i^2)}$$

- and in the output space

$$q_{j|i} = \frac{\exp(-0.5\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$$

[5] [Hinton and Roweis, 2002]

Stochastic Neighbor Embedding (SNE)[5]

- introduce a probabilistic neighborhood in the input space

$$p_{j|i} = \frac{\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_i^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{x}_i - \mathbf{x}_k\|^2/\sigma_i^2)}$$

- and in the output space

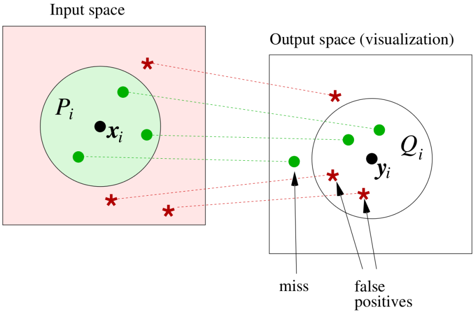
$$q_{j|i} = \frac{\exp(-0.5\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k,k\neq i}\exp(-0.5\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$$

- optimize the sum of Kullback-Leibler divergences

$$C = \sum_i KL(P_i, Q_i) = \sum_i \sum_{j\neq i} p_{j|i} \log\left(\frac{p_{j|i}}{q_{j|i}}\right)$$

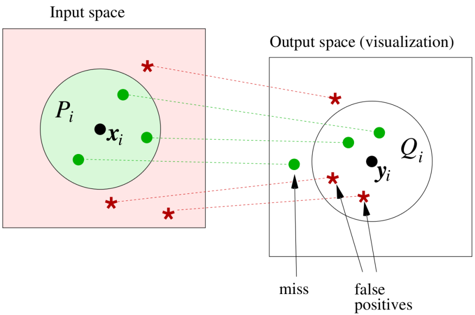
[5][Hinton and Roweis, 2002]

Neighbor Retrieval Visualizer (NeRV)[6]



[6][Venna et al., 2010]

Neighbor Retrieval Visualizer (NeRV)[6]



- $precision(i) = \frac{N_{TP,i}}{k_i} = 1 - \frac{N_{FP,i}}{k_i}, \quad recall(i) = \frac{N_{TP,i}}{r_i} = 1 - \frac{N_{MISS,i}}{r_i}$

[6] [Venna et al., 2010]

Neighbor Retrieval Visualizer (NeRV)

- $KL(P_i, Q_i)$ generalizes recall
- $KL(Q_i, P_i)$ generalizes precision

Neighbor Retrieval Visualizer (NeRV)

- $KL(P_i, Q_i)$ generalizes recall
- $KL(Q_i, P_i)$ generalizes precision
- NeRV optimizes

$$C = \lambda \sum_i KL(P_i, Q_i) + (1 - \lambda) \sum_i KL(Q_i, P_i)$$

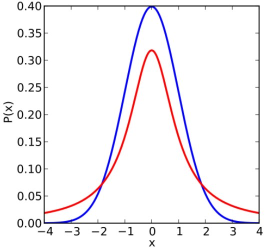
t-Distributed SNE (t-SNE)[7]

- symmetrized probabilities *p* and *q*
- uses a Student-t distribution in the output space

[7] [van der Maaten and Hinton, 2008]

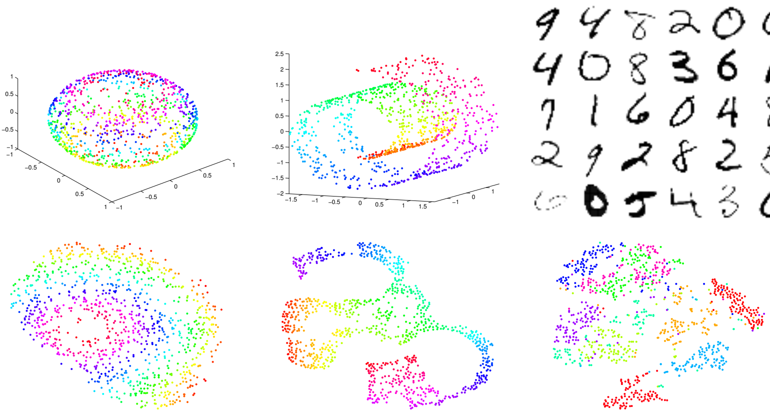
t-Distributed SNE (t-SNE)[7]

- symmetrized probabilities *p* and *q*
- uses a Student-t distribution in the output space



[7] [van der Maaten and Hinton, 2008]

Neighbor Embedding in Action



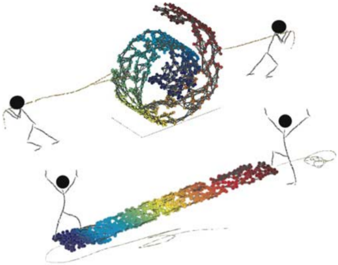
8

---

[8] [Gisbrecht and Hammer, 2015]

Maximum Variance Unfolding (MVU)[9]

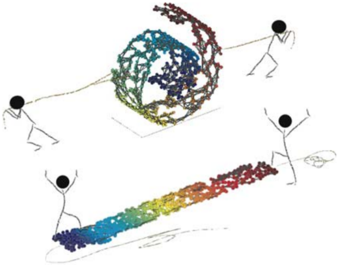
- goal: 'unfold' a given manifold while keeping all the local distances and angles fixed



[9][Weinberger and Saul, 2006]

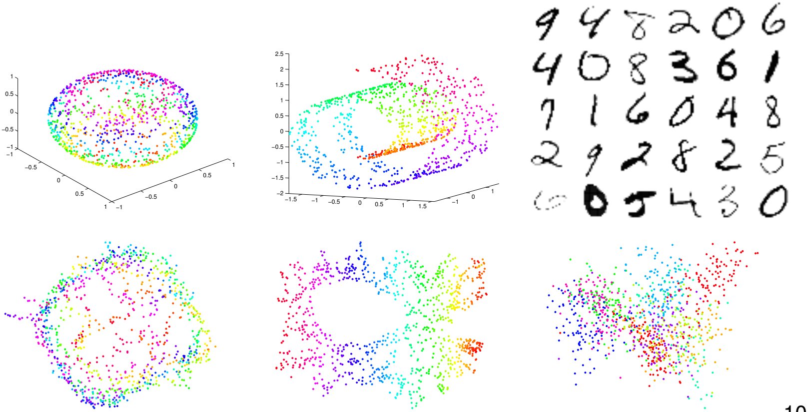
Maximum Variance Unfolding (MVU)[9]

- goal: 'unfold' a given manifold while keeping all the local distances and angles fixed
- maximize $\sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2$ s.t.
  $\sum_i \mathbf{y}_i = 0$
  $\|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, for all neighbors



[9] [Weinberger and Saul, 2006]

Manifold Learner in Action



[10]

[10] [Gisbrecht and Hammer, 2015]

Overview

Dimensionality Reduction
- Motivation
- Linear Projections
- Linear Mappings in Feature Space
- Neighbor Embedding
- Manifold Learning

Advances in Dimensionality Reduction
- Speedup for Neighbor Embeddings
- Quality Assessment of DR
- Feature Relevance for DR
- Visualization of Classifiers
- Supervised Dimensionality Reduction

Barnes Hut Approach[11]

Complexity of NE

- Neighbor Embeddings have the complexity $O(N^2)$

[11] [Yang et al., 2013, van der Maaten, 2013]

Barnes Hut Approach[11]

Complexity of NE

- Neighbor Embeddings have the complexity $O(N^2)$
- matrices P and Q are squared

[11] [Yang et al., 2013, van der Maaten, 2013]

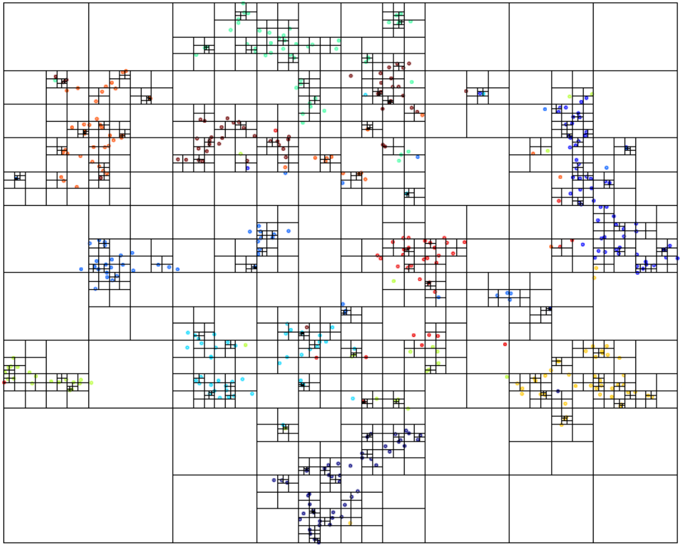
Barnes Hut Approach[11]

Complexity of NE

- Neighbor Embeddings have the complexity $O(N^2)$
- matrices P and Q are squared
- squared summation for the gradient

$$\frac{\partial C}{\partial \mathbf{y}_i} = \sum_{j \neq i} g_{ij}(\mathbf{y}_i - \mathbf{y}_j)$$

[11] [Yang et al., 2013, van der Maaten, 2013]



12

[12] [van der Maaten, 2013]

Barnes Hut Approach

Barnes Hut

- approximate the gradient $\frac{\partial C}{\partial \mathbf{y}_i} = \sum_{j \neq i} g_{ij}(\mathbf{y}_i - \mathbf{y}_j)$ as

$$\sum_{j \neq i} g_{ij}(\mathbf{y}_i - \mathbf{y}_j) \approx \sum_t |G_t^i| \cdot g_{ij}(\mathbf{y}_i - \hat{\mathbf{y}}_t^i)$$

Barnes Hut Approach

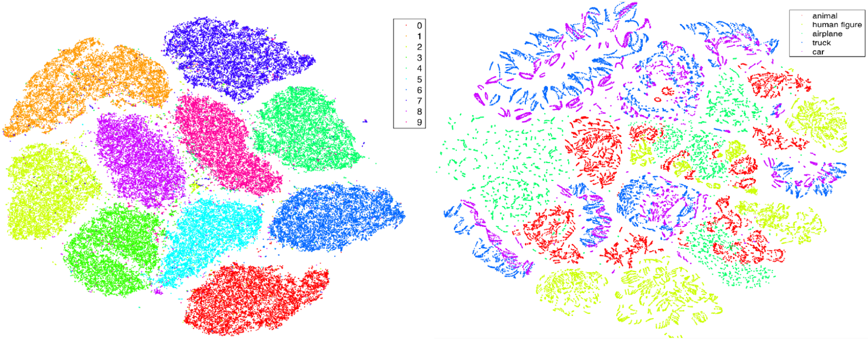
Barnes Hut

- approximate the gradient $\frac{\partial C}{\partial \mathbf{y}_i} = \sum_{j \neq i} g_{ij}(\mathbf{y}_i - \mathbf{y}_j)$ as

$$\sum_{j \neq i} g_{ij}(\mathbf{y}_i - \mathbf{y}_j) \approx \sum_t |G_t^i| \cdot g_{ij}(\mathbf{y}_i - \hat{\mathbf{y}}_t^i)$$

- approximate P as sparse matrix
- results in a $O(N \log N)$ algorithm

Barnes Hut SNE



13

[13] [van der Maaten, 2013]

Kernel t-SNE[14]

- $O(N)$ algorithm: apply NE to a fixed subset, map remainder with out of sample projection

[14] [Gisbrecht et al., 2015]

Kernel t-SNE[14]

- $O(N)$ algorithm: apply NE to a fixed subset, map remainder with out of sample projection
- how to obtain an out of sample extension?

[14] [Gisbrecht et al., 2015]

Kernel t-SNE[14]

- $O(N)$ algorithm: apply NE to a fixed subset, map remainder with out of sample projection
- how to obtain an out of sample extension?
- use kernel mapping

$$\mathbf{x} \mapsto \mathbf{y}(\mathbf{x}) = \sum_j \alpha_j \cdot \frac{k(\mathbf{x}, \mathbf{x}_j)}{\sum_l k(\mathbf{x}, \mathbf{x}_l)} = \mathbf{A}\mathbf{k}$$

[14] [Gisbrecht et al., 2015]

Kernel t-SNE[14]

- $O(N)$ algorithm: apply NE to a fixed subset, map remainder with out of sample projection
- how to obtain an out of sample extension?
- use kernel mapping

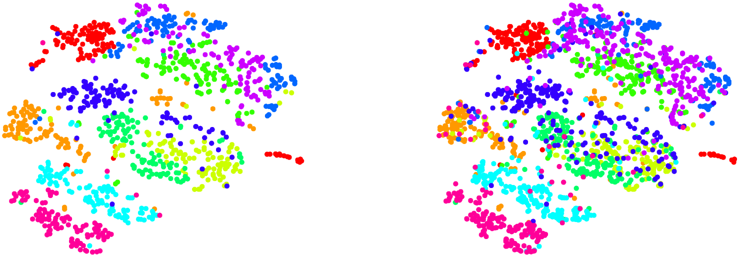
$$\mathbf{x} \mapsto \mathbf{y}(\mathbf{x}) = \sum_j \alpha_j \cdot \frac{k(\mathbf{x}, \mathbf{x}_j)}{\sum_l k(\mathbf{x}, \mathbf{x}_l)} = \mathbf{A}\mathbf{k}$$

- minimization of

$$\sum_i \|\mathbf{y}_i - \mathbf{y}(\mathbf{x}_i)\|^2 \quad \text{yields} \quad \mathbf{A} = \mathbf{Y} \cdot \mathbf{K}^{-1}$$

[14] [Gisbrecht et al., 2015]

Kernel t-SNE



15

[15] [Gisbrecht et al., 2015]

Quality Assessment of DR[16]

- most popular measure

$$Q_k(X, Y) = \sum_i \left(N_k(\vec{x}^i) \cap N_k(\vec{y}^i) \right) / (Nk)$$

[16] [Lee et al., 2013]

Quality Assessment of DR[16]

- most popular measure

$$Q_k(X, Y) = \sum_i \left(N_k(\vec{x}^i) \cap N_k(\vec{y}^i) \right) / (Nk)$$

- rescaling added recently

[16] [Lee et al., 2013]

Quality Assessment of DR[16]

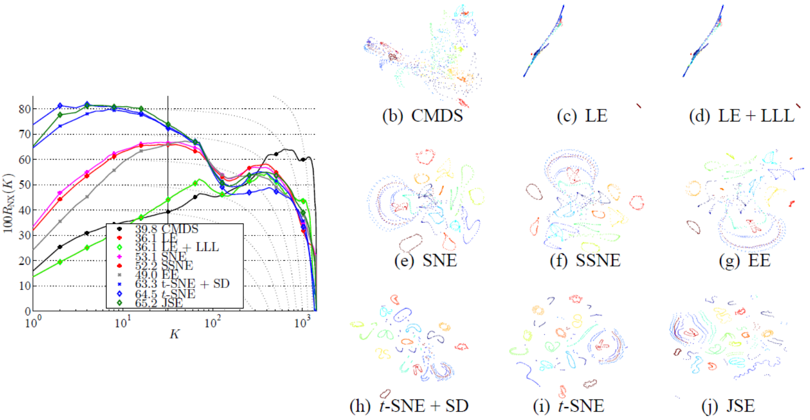
- most popular measure

$$Q_k(X, Y) = \sum_i \left(N_k(\vec{x}^i) \cap N_k(\vec{y}^i) \right) / (Nk)$$

- rescaling added recently
- recently used to compare many DR techniques

[16][Lee et al., 2013]

Quality Assessment of DR



[17]

[17] [Peluffo-Ordóñez et al., 2014]

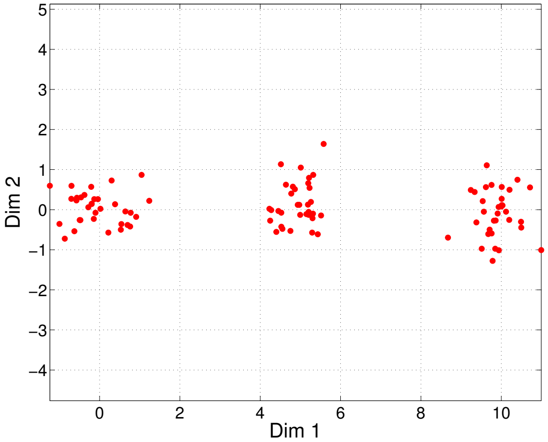
Feature Relevance for DR

- Which features are important for a given projection?

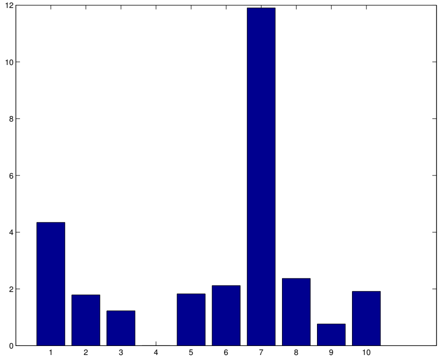
data set: ESANN participants

Partic.	University	ESANN paper	#publications	...	likes beer	...
1	A	1	15		1	...
2	A	0	8		-1	...
3	B	1	22	...	-1	...
4	C	1	9	...	0	...
5	C	0	15		-1	...
6	D		...			
⋮	⋮				⋮	

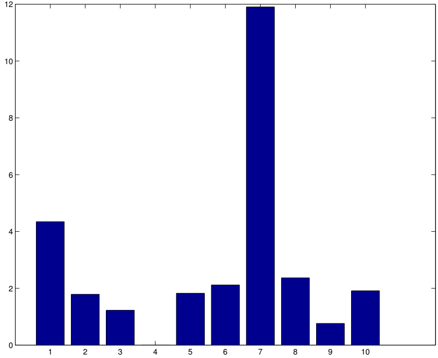
Visualization of ESANN participants



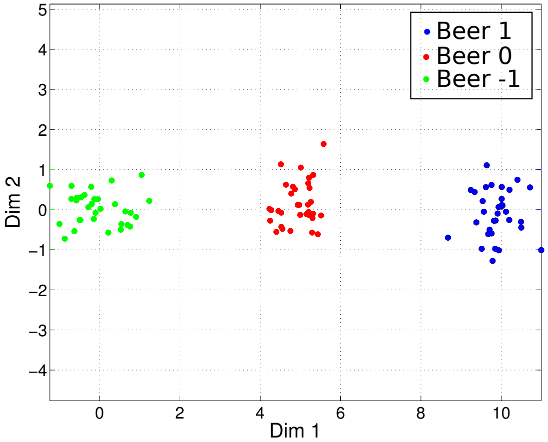
Relevance of features for the projection



Relevance of features for the projection



Visualization of ESANN participants



Aim

- estimate the relevance of single features for non linear dimensionality reductions

Aim

- estimate the relevance of single features for non linear dimensionality reductions

Idea

- change the influence of a single feature and observe the change in the quality

Evaluation of Feature Relevance for DR[19]

NeRV cost function[18] Q_k^{NeRV}

- interpretation from an information retrieval perspective

[18] [Venna et al., 2010]
[19] [Schulz et al., 2014a]

Evaluation of Feature Relevance for DR[19]

NeRV cost function[18] Q_k^{NeRV}

- interpretation from an information retrieval perspective
- $d(\vec{x}^i, \vec{x}^j)^2 = \sum_l (x_l^i - x_l^j)^2$ becomes $\sum_l \lambda_l^2 (x_l^i - x_l^j)^2$

[18][Venna et al., 2010]
[19][Schulz et al., 2014a]

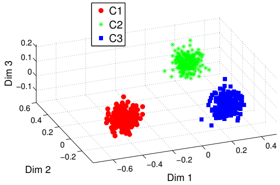
Evaluation of Feature Relevance for DR[19]

NeRV cost function[18] Q_k^{NeRV}

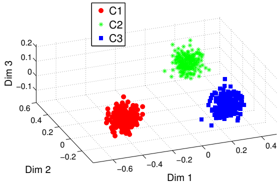
- interpretation from an information retrieval perspective
- $d(\vec{x}^i, \vec{x}^j)^2 = \sum_l (x_l^i - x_l^j)^2$ becomes $\sum_l \lambda_l^2 (x_l^i - x_l^j)^2$
- $\lambda_{\mathrm{NeRV}}^k(l) := \lambda_l^2$ where λ optimizes $Q_k^{\mathrm{NeRV}}(X_\lambda, Y) + \delta \sum_l \lambda_l^2$

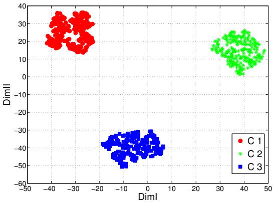
[18][Venna et al., 2010]
[19][Schulz et al., 2014a]

Toy data set 1

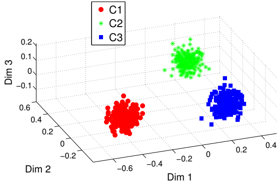


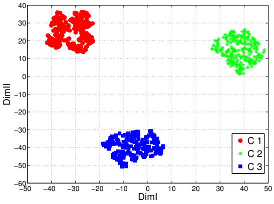
Toy data set 1

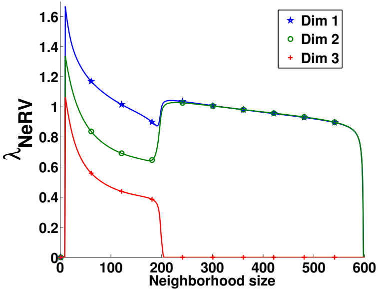




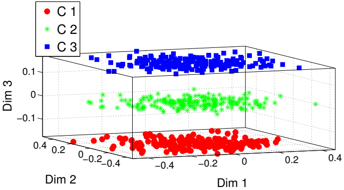
Toy data set 1



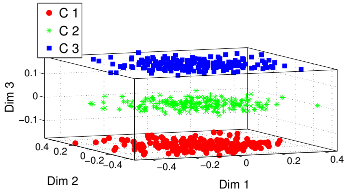


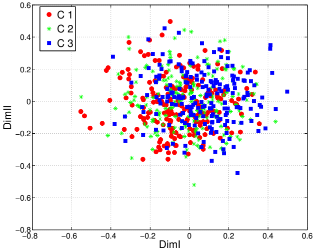


Toy data set 2

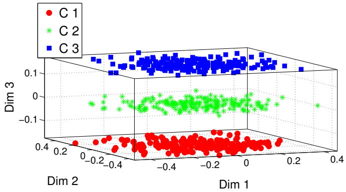


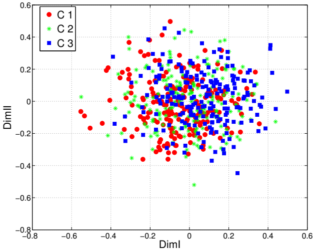
Toy data set 2

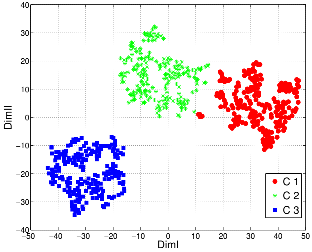




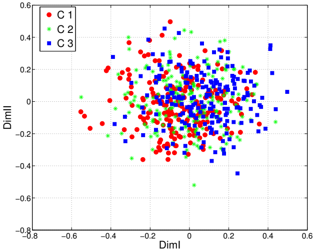
Toy data set 2

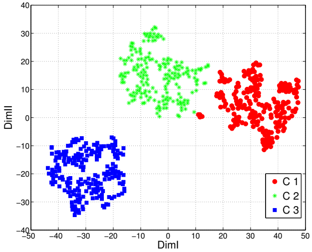


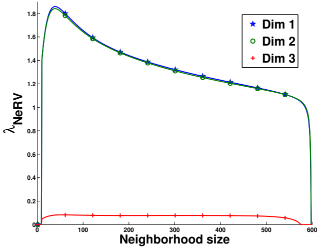




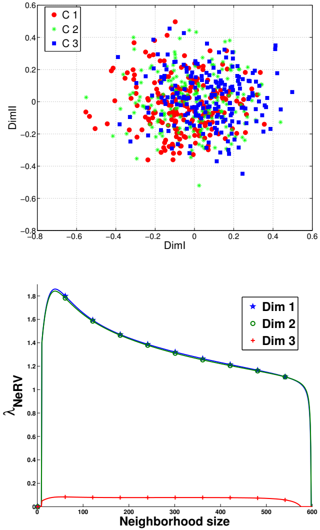
Relevances for different projections





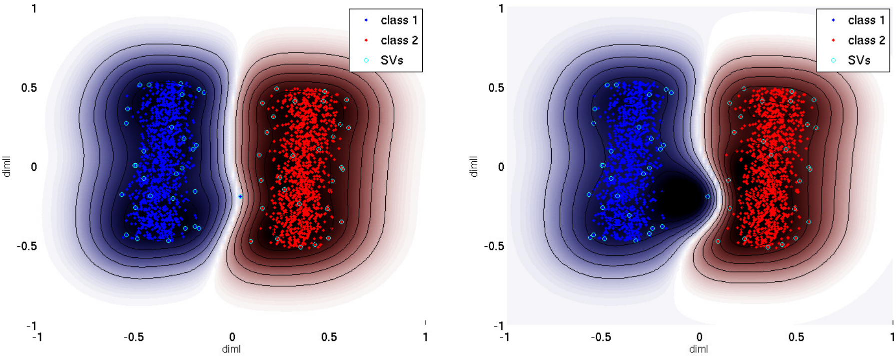


Relevances for different projections

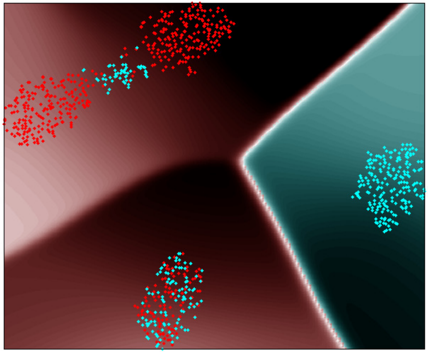


98.7%

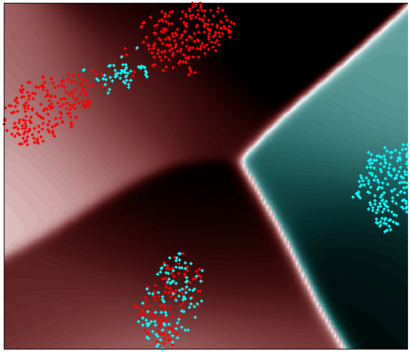
Why visualize Classifiers?

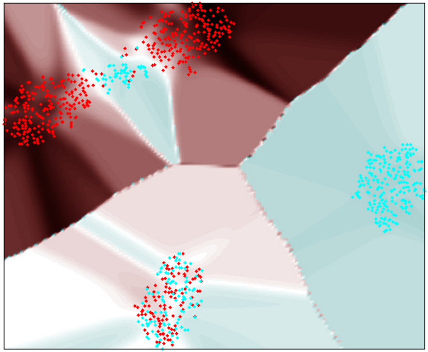


Why visualize Classifiers?



Why visualize Classifiers?



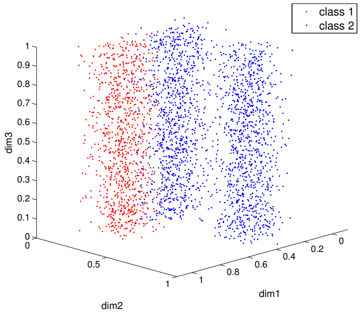


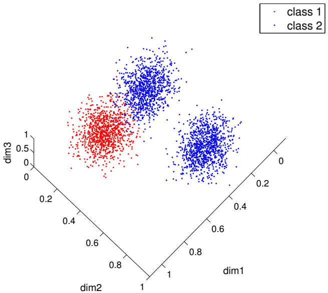
Where is the Difficulty?

Class borders are

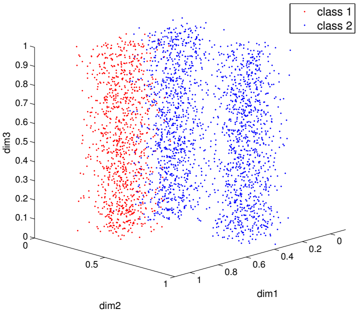
- often non linear
- often not given in an explicit functional form (e.g. SVM)
- high dimensional which makes it non feasible to sample them for a projection

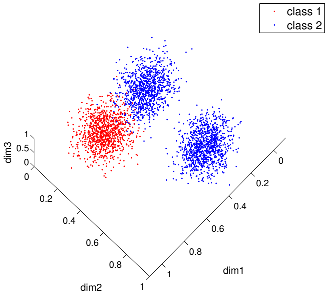
An illustration the approach





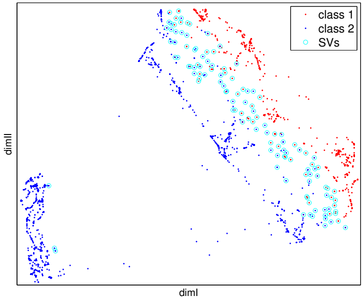
An illustration the approach



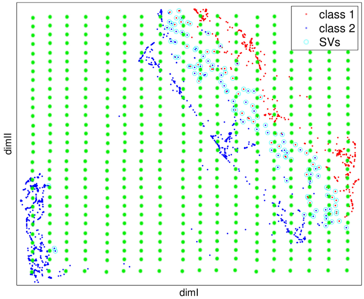


- project data to 2D

An illustration: dimensionality reduction

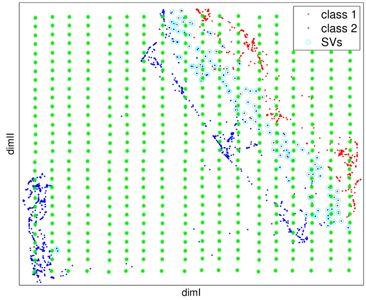


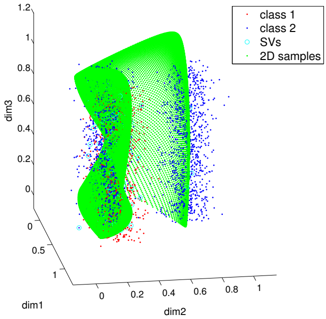
An illustration: dimensionality reduction



- sample the 2D data space
- project the samples up

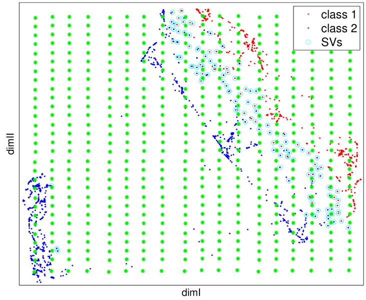
An illustration: dimensionality reduction

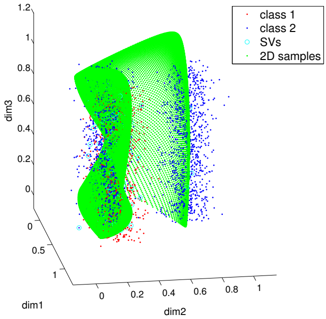




- sample the 2D data space
- project the samples up

An illustration: dimensionality reduction

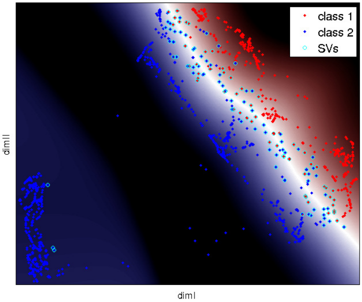




- sample the 2D data space
- project the samples up
- classify them

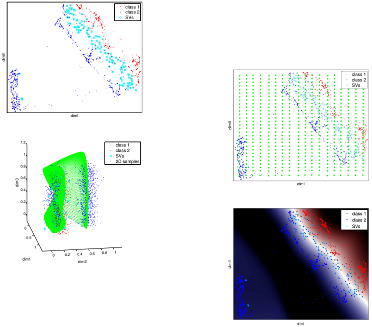
An illustration: border visualization

- color intensity codes the certainty of the classifier



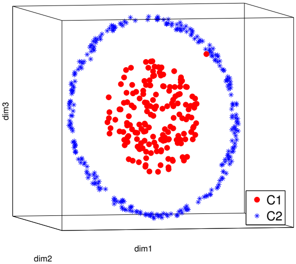
Classifier visualization[20]

- project data to 2D
- sample the 2D data space
- project the samples up
- classify them

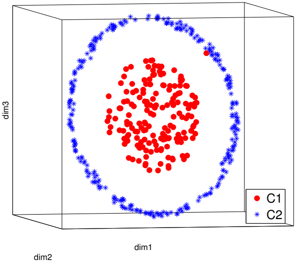


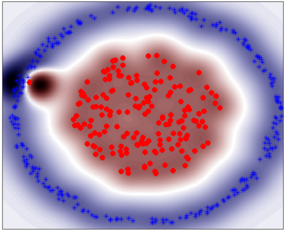
[20] [Schulz et al., 2014b]

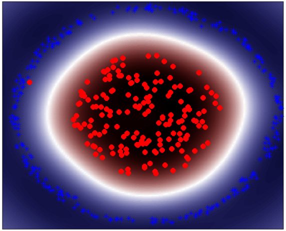
Toy Data Set 1



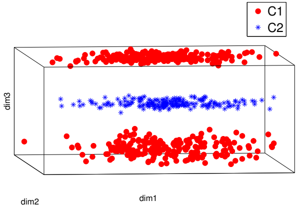
Toy Data Set 1



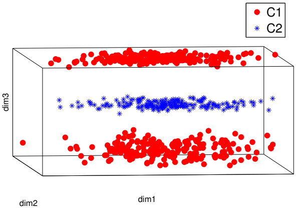


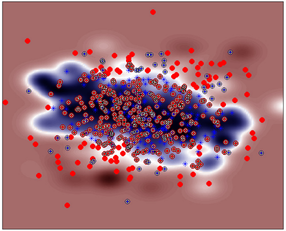


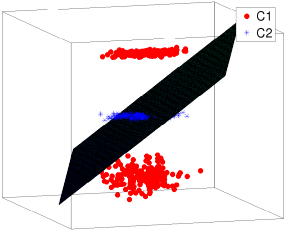
Toy Data Set 2



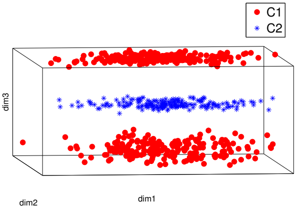
Toy Data Set 2

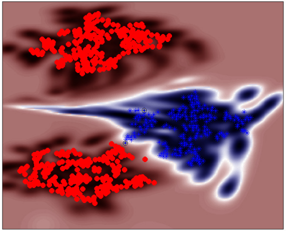




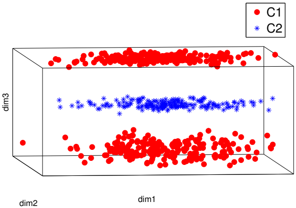


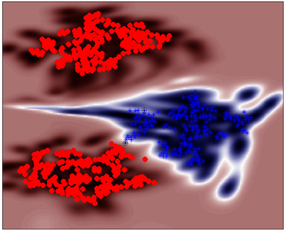
Toy Data Set 2 with NE

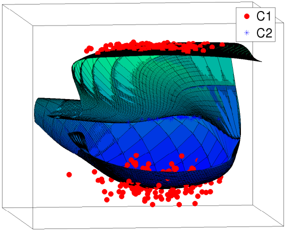




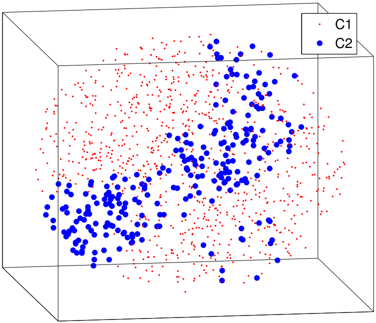
Toy Data Set 2 with NE

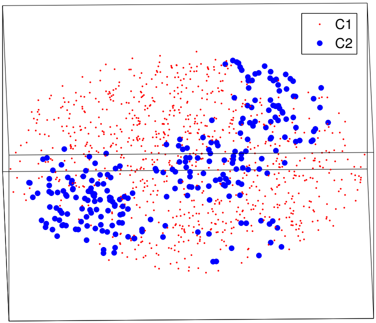




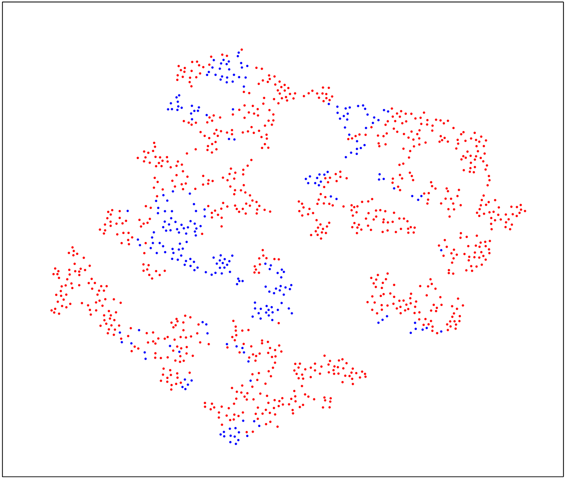


DR: intrinsically 3D data





DR: NE projection



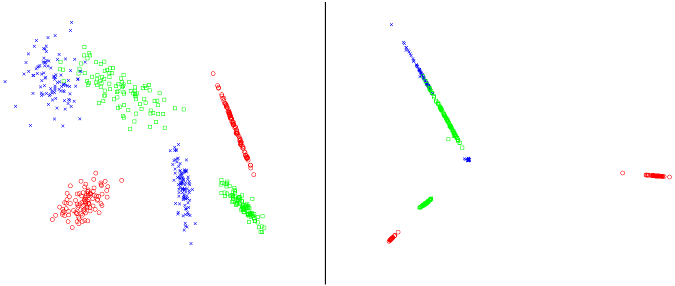
Supervised dimensionality reduction

- Use the Fisher metric[21] $d(\mathbf{x}, \mathbf{x} + d\mathbf{x}) = \mathbf{x}^\top \mathbf{J}(\mathbf{x})\mathbf{x}$
- $\mathbf{J}(\mathbf{x}) = \mathrm{E}_{p(c|\mathbf{x})} \left\{ \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right)^\top \right\}$

[21] [Peltonen et al., 2004, Gisbrecht et al., 2015]

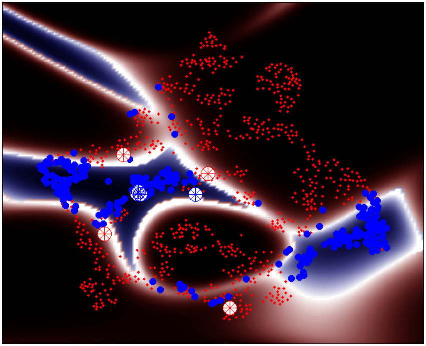
Supervised dimensionality reduction

- Use the Fisher metric[22] $d(\mathbf{x}, \mathbf{x} + d\mathbf{x}) = \mathbf{x}^\top \mathbf{J}(\mathbf{x})\mathbf{x}$
- $\mathbf{J}(\mathbf{x}) = \mathrm{E}_{p(c|\mathbf{x})} \left\{ \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right)^\top \right\}$

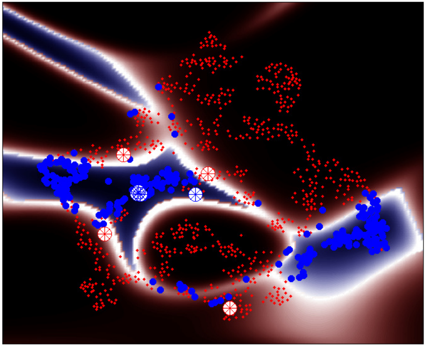


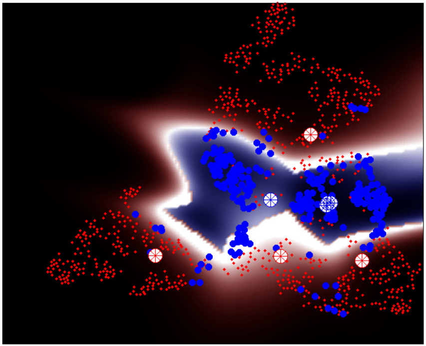
[22] [Peltonen et al., 2004, Gisbrecht et al., 2015]

DR: supervised NE projection

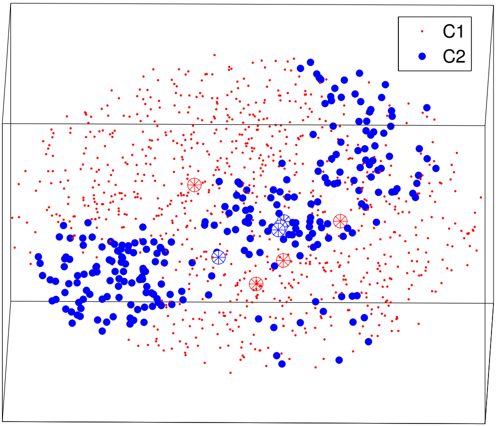


DR: supervised NE projection





prototypes in original space



Summary

- different objectives of dimensionality reduction

Summary

- different objectives of dimensionality reduction
- new approach to get insight into trained classification models
- discriminative information can yield major imrpovements

Thank You For Your Attention!

Alexander Schulz
aschulz (at) techfak.uni-bielefeld.de

Literature I

Gisbrecht, A. and Hammer, B. (2015).
Data visualization by nonlinear dimensionality reduction.
Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(2):51–73.

Gisbrecht, A., Schulz, A., and Hammer, B. (2015).
Parametric nonlinear dimensionality reduction using kernel t-sne.
Neurocomputing, 147:71–82.

Hinton, G. and Roweis, S. (2002).
Stochastic neighbor embedding.
In *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press.

Lee, J. A., Renard, E., Bernard, G., Dupont, P., and Verleysen, M. (2013).
Type 1 and 2 mixtures of kullback-leibler divergences as cost functions in dimensionality reduction based on similarity preservation.
Neurocomput., 112:92–108.

Lee, J. A. and Verleysen, M. (2007).
Nonlinear dimensionality reduction.
Springer.

Peltonen, J., Klami, A., and Kaski, S. (2004).
Improved learning of riemannian metrics for exploratory analysis.
Neural Networks, 17:1087–1100.

Literature II

Peluffo-Ordóñez, D. H., Lee, J. A., and Verleysen, M. (2014).
Recent methods for dimensionality reduction: A brief comparative analysis.
In *22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014*.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998).
Nonlinear component analysis as a kernel eigenvalue problem.
Neural Comput., 10(5):1299–1319.

Schulz, A., Gisbrecht, A., and Hammer, B. (2014a).
Relevance learning for dimensionality reduction.
In *22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014*.

Schulz, A., Gisbrecht, A., and Hammer, B. (2014b).
Using discriminative dimensionality reduction to visualize classifiers.
Neural Processing Letters, pages 1–28.

van der Maaten, L. (2013).
Barnes-hut-sne.
CoRR, abs/1301.3342.

van der Maaten, L. and Hinton, G. (2008).
Visualizing high-dimensional data using t-sne.
Journal of Machine Learning Research, 9:2579–2605.

Venna, J., Peltonen, J., Nybo, K., Aidos, H., and Kaski, S. (2010).
Information retrieval perspective to nonlinear dimensionality reduction for data visualization.
Journal of Machine Learning Research, 11:451–490.

Literature III

Weinberger, K. Q. and Saul, L. K. (2006).
An introduction to nonlinear dimensionality reduction by maximum variance unfolding.
In *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI.

Yang, Z., Peltonen, J., and Kaski, S. (2013).
Scalable optimization of neighbor embedding for visualization.
In Dasgupta, S. and Mcallester, D., editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 127–135. JMLR Workshop and Conference Proceedings.