# Convex Optimization of the Sammon Transformation

## Final presentation

Susanne Westphal
02/17/2014
Computer Science Dept. 5 (Pattern Recognition)
Friedrich-Alexander University Erlangen-Nuremberg

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

## Convex Optimization of the Sammon Transformation

- Motivation
- Derivation
- Weighted Inner Product Objective Function
- Convexity
- Results
- Real Data
- Outlook & Conclusion
- Questions?

# Motivation

- In 1969 John Sammon published an article about a non-linear mapping for data structure analysis
- It is a mapping from a high-dimensional space to a lower-dimensional space
- The inner-point distances of the points are preserved as good as possible
- The Stress Function is an indicator for size of the difference of the inner-point distances in the different spaces
- For finding the best fitting points in the low-dimensional space we have to minimize this equation

# Sammon Stress Function:

$$E = \frac{1}{\sum_{i<j} d_{ij}} \sum_{i<j}^{N} \frac{(d_{ij} - ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2)^2}{d_{ij}}$$

$d_{ij}$ are the inner-point distances in the original space

$\boldsymbol{x}_i, \boldsymbol{x}_j$ are the projected points in the low-dimensional space

# Fields of Application

- face recognition
- speech recognition
- sensor localization
- shape matching
- and many more

# Objective of the Thesis

- Finding a convex function by using Lagrange Multipliers
- It should have the same properties as the Sammon Mapping
- And also a small Sammon Error

## Questions . . .

- Sounds a bit unlikely that there exists such a function . . .
- Nobody had the idea before . . .
- And I should be able to do it . . .

# Derivation

# Lagrange Multipliers

Optimization problem:

$$
\begin{aligned}
\text{minimize} \quad & f_0(\boldsymbol{x}) & \\
\text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0 \ , & i = 1, ..., m; \\
& h_i(\boldsymbol{x}) = 0 \ , & i = 1, ..., p;
\end{aligned}
$$

# Lagrange Multipliers

Optimization problem:

$$\begin{array}{lll} \text{minimize} & f_0(\boldsymbol{x}) & \\ \text{subject to} & f_i(\boldsymbol{x}) \leq 0 \,, & i = 1, ..., m; \\ & h_i(\boldsymbol{x}) = 0 \,, & i = 1, ..., p; \end{array}$$

The Lagrangian is defined as:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i h_i(\boldsymbol{x})$$

# Forming the Lagrangian

Objective function: $f_0(\boldsymbol{x}) = 0$

Constraint: $d_{ij}^2 = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 \; \forall i, j$

# Forming the Lagrangian

Objective function: $f_0(\boldsymbol{x}) = 0$

Constraint: $d_{ij}^2 = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 \; \forall i, j$

Then the Lagrangian is:

$$L(\boldsymbol{x}, \boldsymbol{\nu}) = \sum_{i,j} \nu_{ij}(d_{ij}^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2)$$

We can define matrices $A$ and $B$,

$$\boldsymbol{A} = (a_{ij}) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2$$

$$\boldsymbol{B} = (b_{ij}) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$$

so that

$$\boldsymbol{B} = -\frac{1}{2}\boldsymbol{HAH}$$

if the points are **centered around the origin.**

$\boldsymbol{H}$ is defined as:

$$\boldsymbol{H} = \boldsymbol{I}_n - n^{-1}\boldsymbol{J}_n$$

$\boldsymbol{I}_n$ is the identity matrix with size $(n \times n)$ and $\boldsymbol{J}_n$ is a $(n \times n)$-matrix of ones.

$\boldsymbol{H} \in \mathbb{R}^{n \times n}$ has the rank $n-1$. So we can solve the equation for $\boldsymbol{A}$ using the pseudo-inverse:

$$\boldsymbol{A} \cong -2 \cdot \boldsymbol{H}^{\dagger} \boldsymbol{B} \boldsymbol{H}^{\dagger}$$

for a **high number of points**:

$$\begin{aligned} &\cong -2 \cdot \boldsymbol{I}_n \boldsymbol{B} \boldsymbol{I}_n \\ &= -2 \cdot \boldsymbol{B} \end{aligned}$$

So our Lagrangian is:

$$L(\boldsymbol{x}, \boldsymbol{\nu}) = \sum_{ij} \nu_{ij} (d_{ij}^2 + 2 \cdot \boldsymbol{x}_i^\top \boldsymbol{x}_j)$$

$\boldsymbol{\nu}$ **has to be a symmetric matrix**, in our case it is defined as the constraint itself.

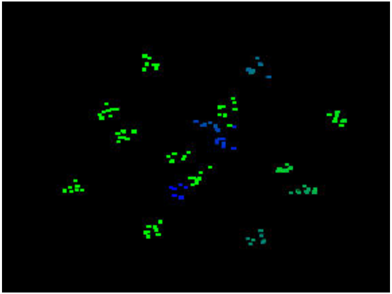
# The new target function

$$L(\boldsymbol{x}) = \sum_i \sum_j (2 \cdot \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + d_{ij}^2)^2$$

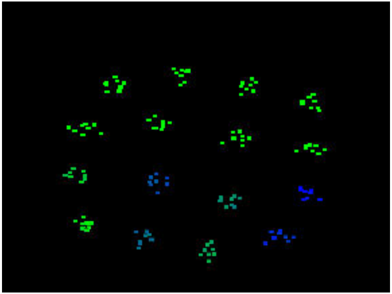
$d_{ij}$ are the inner-point distances in the original space.
$\boldsymbol{x}_i, \boldsymbol{x}_j$ are the projected points in the low-dimensional space.

# It really works!!!

Our objective function

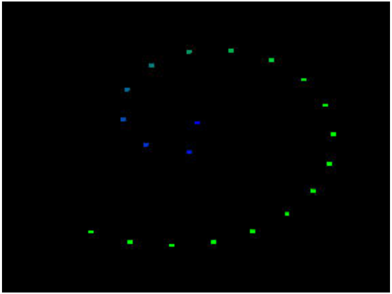


Sammon Objective Function

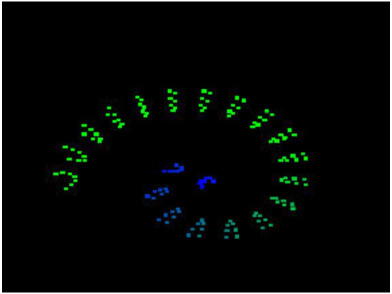


4D Cube

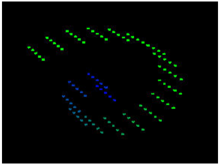
Our objective function



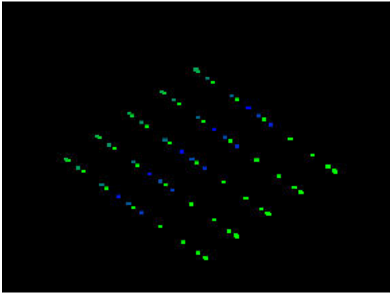
Sammon Objective Function



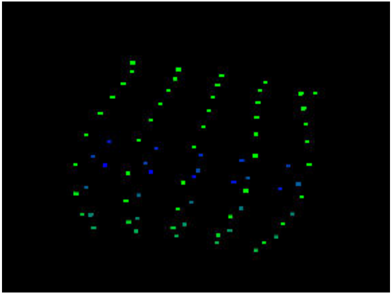
Swiss Roll



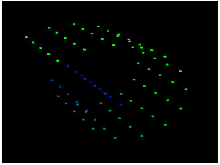
Our objective function



Sammon Objective Function



Swiss Roll



# Weighted Inner Product Objective Function

# Improvement of the actual target function

We want to have:

- Non-linear projection
- Smaller distances should be weighted stronger

# Improvement of the actual target function

We want to have:

- Non-linear projection
- Smaller distances should be weighted stronger

So we introduce a weighting factor as it is done in the Sammon Objective Function.

The new target function looks like this:

$$L(\boldsymbol{x}) = \sum_{p} \sum_{q>p} \frac{(2 \cdot \langle \boldsymbol{x}_p, \boldsymbol{x}_q \rangle + d_{pq}^2)^2}{d_{pq}}$$

This leads to instabilities due to small inner-point distances. So we add a factor $k$:

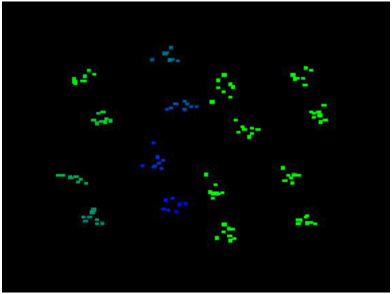
$$L(\boldsymbol{x}) = \sum_{p} \sum_{q>p} \frac{(2 \cdot \langle \boldsymbol{x}_p, \boldsymbol{x}_q \rangle + d_{pq}^2)^2}{d_{pq} + k}$$

We can weight the smaller distances stronger by a quadratic denominator:

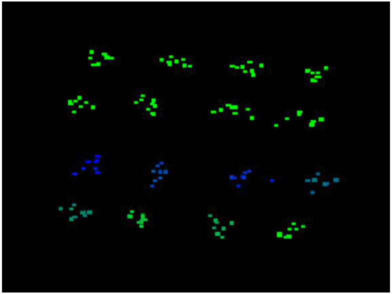
$$L(\boldsymbol{x}) = \sum_{p} \sum_{q>p} \left( \frac{2 \cdot \langle \boldsymbol{x}_p, \boldsymbol{x}_q \rangle + d_{pq}^2}{d_{pq} + k} \right)^2$$

# 4D Cube

linear denominator, $k = 3.5$

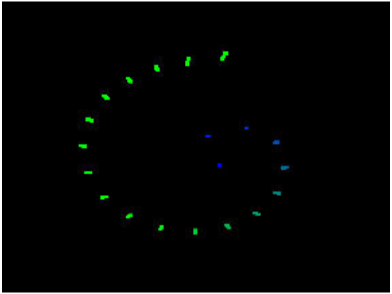


quadratic denominator, $k = 7$

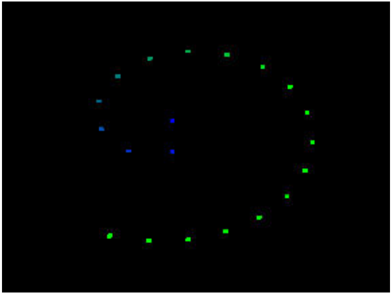


# Swiss Roll 1

linear denominator, $k = 1.0$

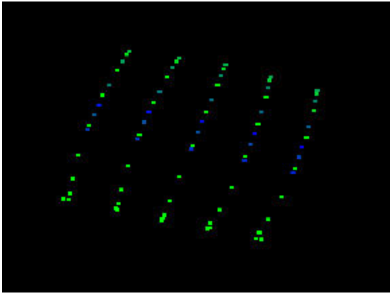


quadratic denominator, $k = 10$

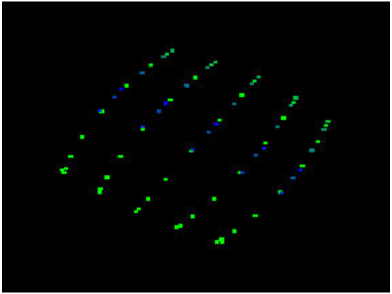


# Swiss Roll 2

linear denominator, $k = 0.2$



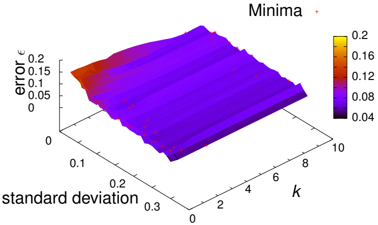
quadratic denominator, $k = 0.5$



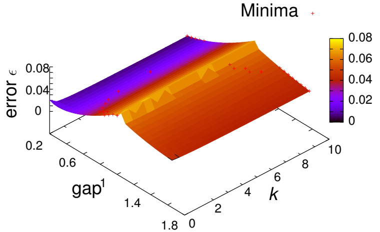
# Which *k*-factor is the best one?

# Which $k$-factor is the best one?

Linear denominator:



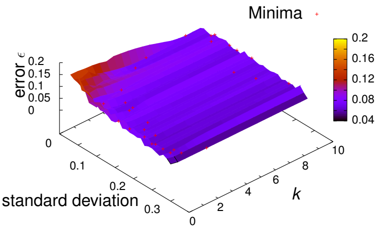

3D Cube



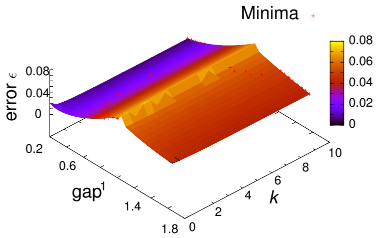

Swiss Roll

# Which *k*-factor is the best one?

Linear denominator:




3D Cube

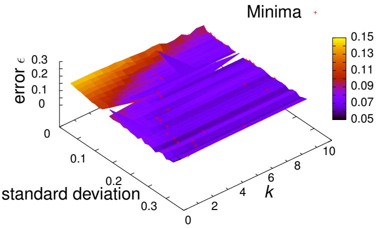



Swiss Roll

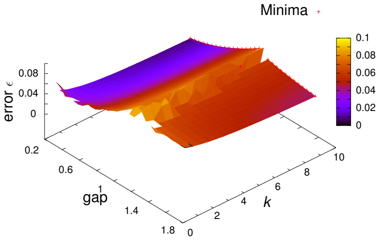
- If the principal components have about the same length a smaller *k* is better
- Otherwise a PCA is a good solution, and the solution of the WIPOF with a big *k* is similar to a PCA

# Which *k*-factor is the best one?

Quadratic denominator:



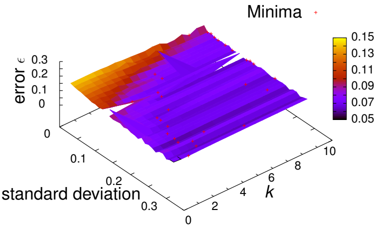

3D Cube



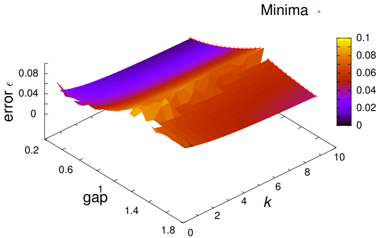

Swiss Roll

# Which *k*-factor is the best one?

Quadratic denominator:




3D Cube

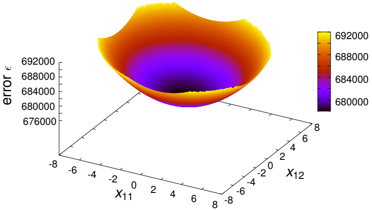



Swiss Roll

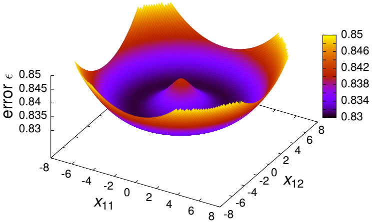
- In general a big *k*-factor is better
- But exceptions exist

# Convexity

# Convexity




Weighted Inner Product Objective Function




Sammon Objective Function

# Results

# Results in numbers

Linear denominator:

| function | SOF | PCA | WIPOF (best $k$) |
|---|---|---|---|
| Swiss(0.4, 20, 5) | 0.0115 | 0.0160 | **0.0159** |
| Swiss(1.0, 20, 8) | 0.0360 | 0.0390 | **0.0387** |
| Swiss(0.1, 20, 8) | 0.00266 | 0.00508 | **0.00479** |
| Cube(1.0, 3, 0.05, 10) | 0.0677 | **0.0759** | 0.0934 |
| Cube(1.0, 4, 0.25, 10) | 0.0960 | 0.105 | **0.101** |
| Cube(1.0, 5, 0.1, 5) | 0.131 | 0.139 | **0.136** |

Tab.: Comparison of WIPOF with PCA and SOF

# Results in numbers

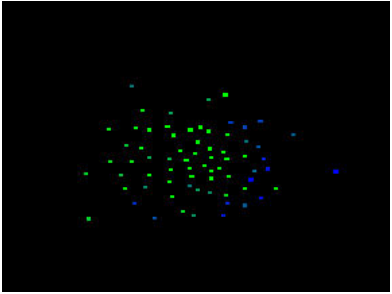
Quadratic denominator:

| function | SOF | PCA | WIPOF (best $k$) |
|---|---|---|---|
| Swiss(0.4, 20, 5) | 0.0115 | **0.0160** | 0.0173 |
| Swiss(1.0, 20, 8) | 0.0366 | **0.0390** | 0.0436 |
| Swiss(0.1, 20, 8) | 0.00266 | **0.00508** | 0.00747 |
| Cube(1.0, 3, 0.05, 10) | 0.0650 | 0.0763 | **0.0734** |
| Cube(1.0, 4, 0.25, 10) | 0.0884 | 0.115 | **0.109** |
| Cube(1.0, 5, 0.1, 5) | 0.124 | 0.151 | **0.140** |

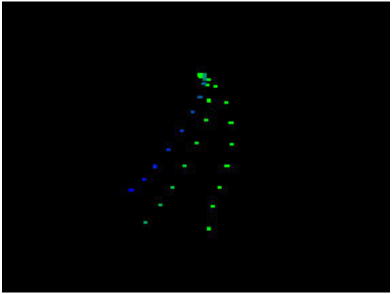
Tab.: Comparison of WIPOF with PCA and SOF

# Real Data

Pathological speech data



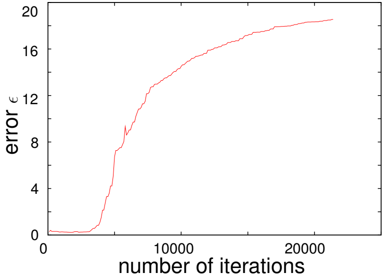
Movement fields of MR data



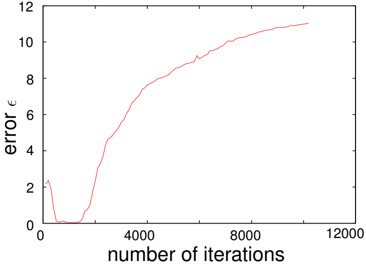
2D presentation with the Sammon Objective Function

# With the WIPOF..... very bad results..... but why??????

Graph of the Sammon Error over the optimization:

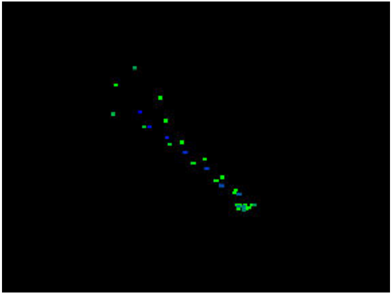



Pathological Speech Data

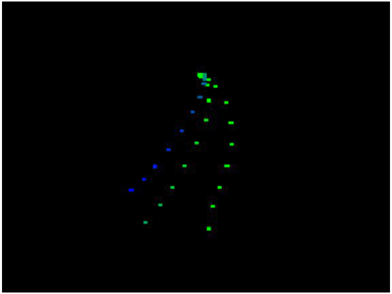



Movement Fields of MR

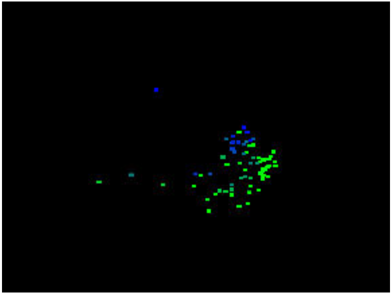
# Best results of all iteration steps



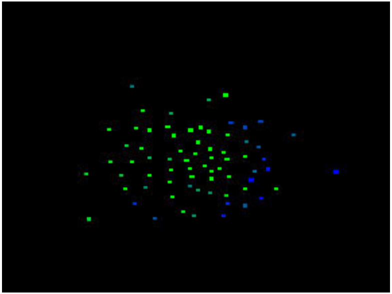
Weighted Inner Product Objective Function (quadratic denominator)
Error: 0.0342



Sammon Objective Function
Error: 0.000448



Weighted Inner Product Objective Function (linear denominator)
Error: 0.30170



Sammon Objective Function
Error: 0.1064

# Possible causes (1)

**Small size of principal components.**

# Possible causes (1)

**Small size of principal components.**

- The principal components of the real data are on average smaller.

# Possible causes (1)

**Small size of principal components.**

- The principal components of the real data are on average smaller.

But:

# Possible causes (1)

**Small size of principal components.**

- The principal components of the real data are on average smaller.

But:

- A Hypercube, with additional dimensions of noise, still converges to a good result

# Possible causes (1)

**Small size of principal components.**

- The principal components of the real data are on average smaller.

But:

- A Hypercube, with additional dimensions of noise, still converges to a good result
- A PCA of the real data, before the optimization, does not affect the result

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu***

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu*** ✓

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu*** ✓
- Many points

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu*** ✓
- Many points ✓

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu*** ✓
- Many points ✓
- Mean value equal to zero

# Possible causes (2)

**The three conditions from the derivation have to be fulfilled.**

- Symmetric ***nu*** ✓
- Many points ✓
- Mean value equal to zero ✗

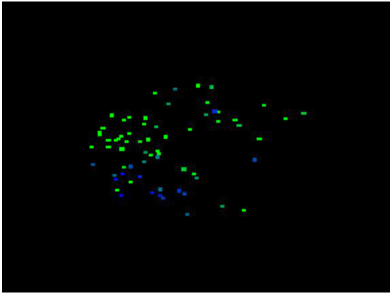
# Possible causes (3)

## Outliers

# Possible causes (3)

## Outliers

Some points have a bigger distance to the origin than the others.
Ignoring them leads to good results with the pathological speech data.



Error: 0.1440

# Outlook & Conclusion

## Still existing questions:

Still existing questions:

- Why are outliers a problem?
- How can we make the function more stable?
- Is there a possibility to predict a good $k$-factor?

Still existing questions:

- Why are outliers a problem?
- How can we make the function more stable?
- Is there a possibility to predict a good $k$-factor?

What we have:

## Still existing questions:

- Why are outliers a problem?
- How can we make the function more stable?
- Is there a possibility to predict a good $k$-factor?

## What we have:

- A new target function
- A logical derivation
- Convexity
- Quite good results

# Questions?

The End