An Information Retrieval Approach to Visualization of High-dimensional Data

Jaakko Peltonen
Aalto University and University of Tampere

based on papers by Jaakko Peltonen, Jarkko Venna, Helena Aidos, Kristian Nybo, Nils Gehlenborg, Alvis Brazma, Zhirong Yang, Kerstin Bunte, Matti Järvisalo, Jeremias Berg, Max Sandholm, Ziyuan Lin, and Samuel Kaski


Tampere
Helsinki

Aalto University

UNIVERSITY OF TAMPERE

Machine learning for exploratory data analysis: supports making discoveries

When good hypotheses/models are not yet available...

Look at the data!

- how to find relevant data?
  see yesterday's talk!
- how to analyse the data?

Once you have found a set of relevant data, how to analyse it?

- We concentrate on methods for visual analysis

- Nonlinear dimensionality reduction (NLDR) often used to visualize data. Many methods are not designed for low-dim. visualization, and work poorly.

- **Information retrieval approach** : optimize visualization for a well-defined low-level analysis task: retrieval of which samples are similar!

-----> well performing NLDR family: **Neighbor Retrieval Visualizer**

- Variants: interpretable linear projections, projections supervised by annotation, projections learned from interactive feedback...

The Setting

You have:

- data about a complicated phenomenon
- possibly side information like labels, ontologies etc.

You want:

- to learn as much as possible about the phenomenon.
- ultimately you may want to do predictive tasks
- you are also interested in just exploring the data.

- Traditionally visualization by nonlinear dimensionality reduction has not been not approached rigorously
  – various methods/cost functions, no clear task, comparisons by looking at pictures
- Many NLDR methods are designed for **manifold learning:** Isomap, Locally Linear Embedding, Stochastic Neighbor Embedding, and others. It might seem attractive to simply use manifold learning methods for visualization.
- However, they have not been designed or optimized for visualization. **They may not work well if the inherent dimensionality of the data manifold is larger than the display dimension.**

Preservation Optimal for a Task?

What is a “good visualization”, is it the “nicest looking one”? Aesthetic considerations etc. are subjective different visualization methods best for different analysts?

Algorithmic approaches to preserve various things can be seen as guesswork about what will produce the most useful visualization for an analyst.

→ Useful for what?

→ For doing something (a task)?

Purpose of visualization (one possible definition): to generate insights about the data in the mind of the analyst.

→ hard to quantify (many kinds of possible “insights”, how to reach them depends on each analyst)

→ instead of “finding insight”, is there some simpler task that we could make visualizations for?

Preservation Optimal for a Task?

The analyst wants to achieve insight by analyzing something about the original data, based on what is visible on the display.

→ That is, analyzing some aspect of the data is a subtask of generating insight.

→ Then preservation approaches should focus on preserving the thing that the analyst wants to analyze.

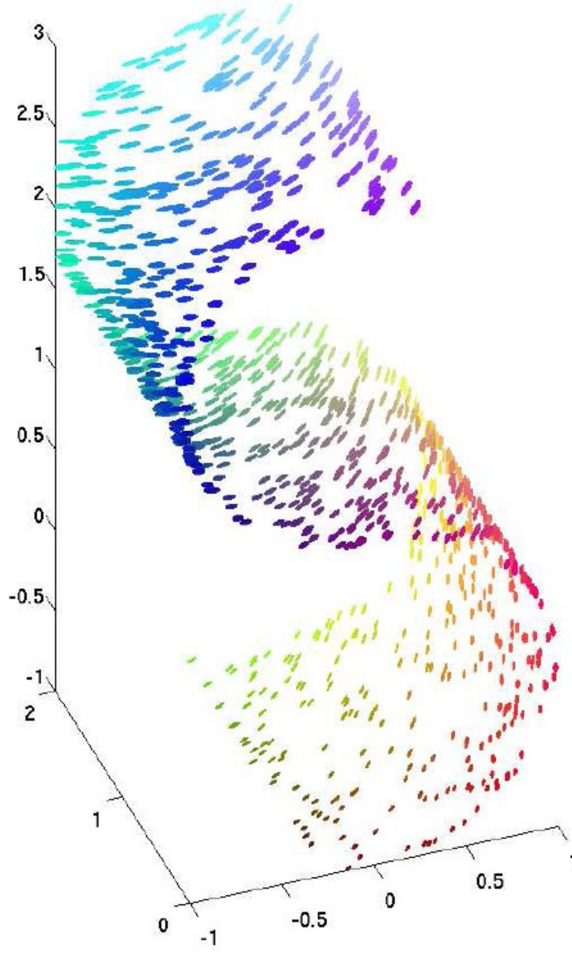
Preservation of distances is good in cases where the analyst wants to measure distances between data points (e.g. geographical maps).

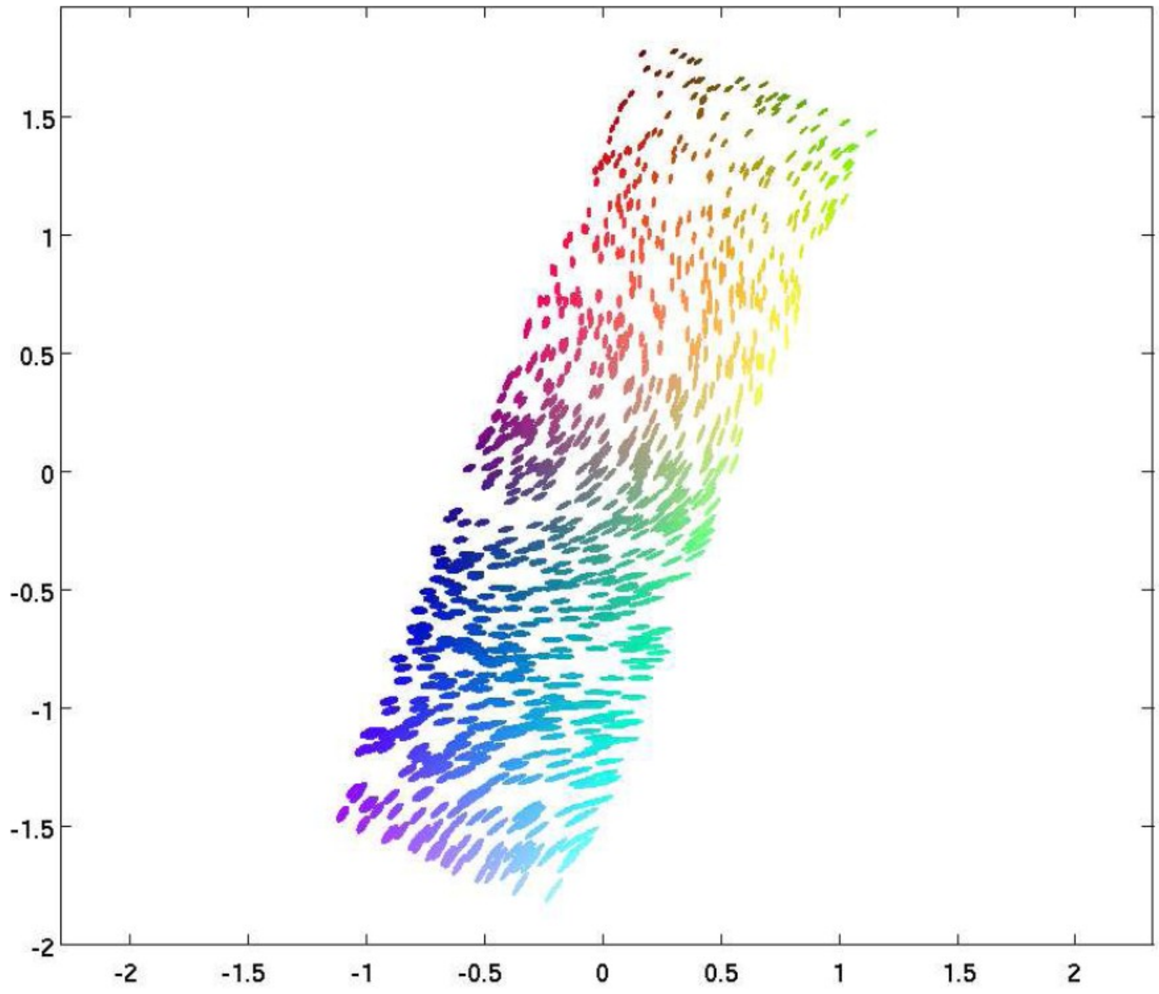
→ In general nonlinear dimensionality reduction, output axes don't have a simple meaning

→ distances are less informative.

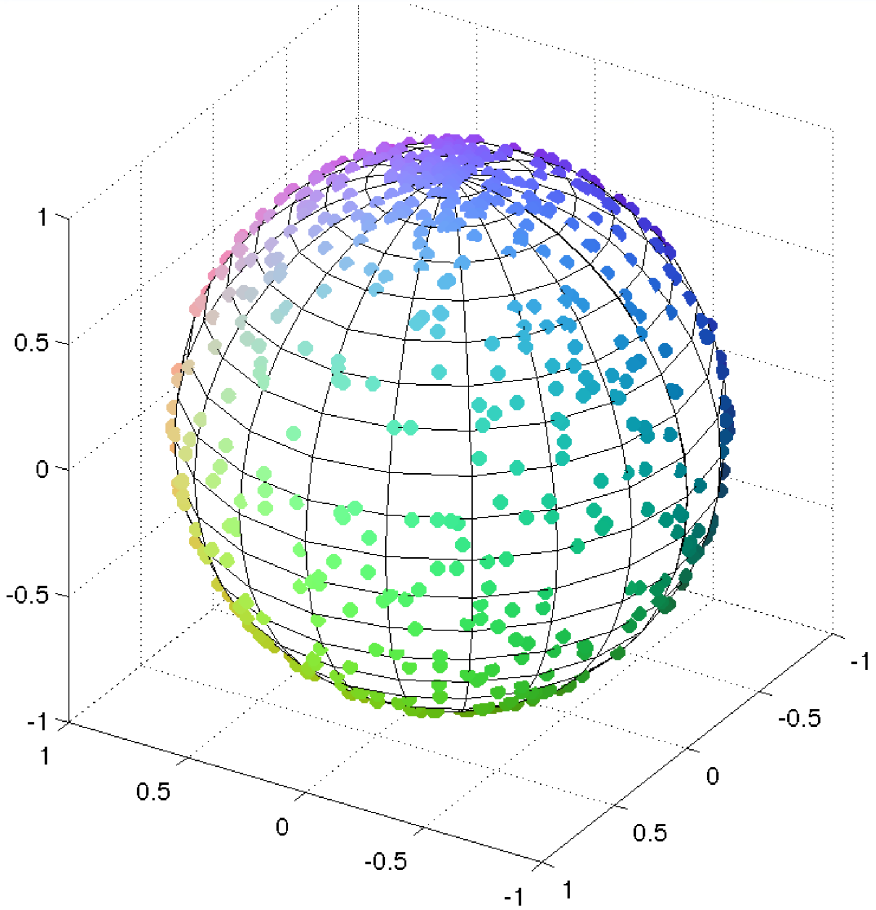
Novelties:

- Formalized the task of visualization as a **visual information retrieval task:** organize points on the display so that retrieving similar points based on the display maximizes accuracy of retrieving truly similar data.
- As in all information retrieval, the result is necessarily a **compromise between precision and recall**, of minimizing false positives and misses.
- Very well performing visualizer

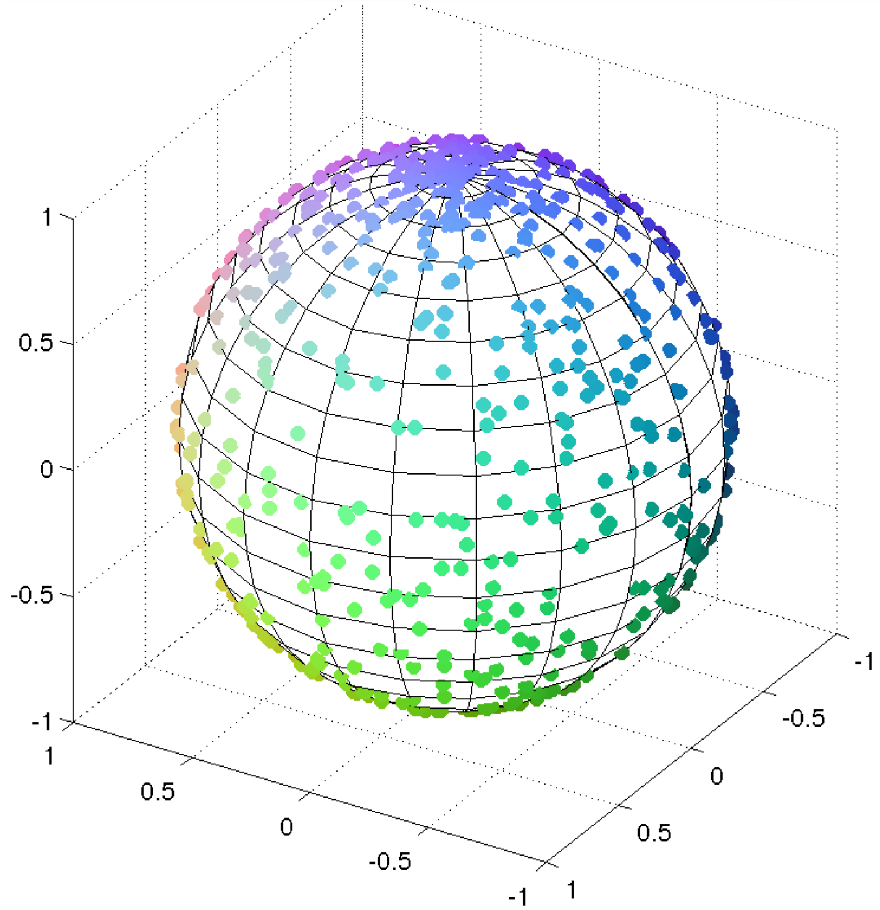


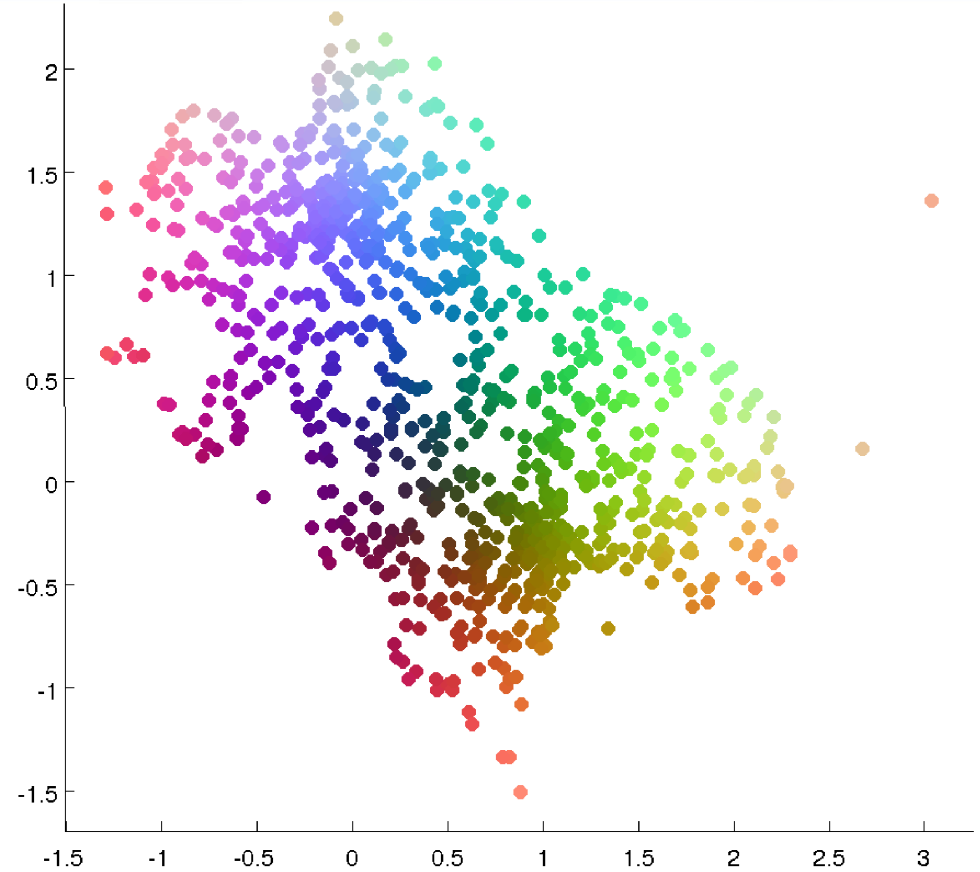


When the original data is complicated,**no low-dimensional visualization can represent it perfectly.** When visualization can't show everything, we must make a decision about what properties we want to show.

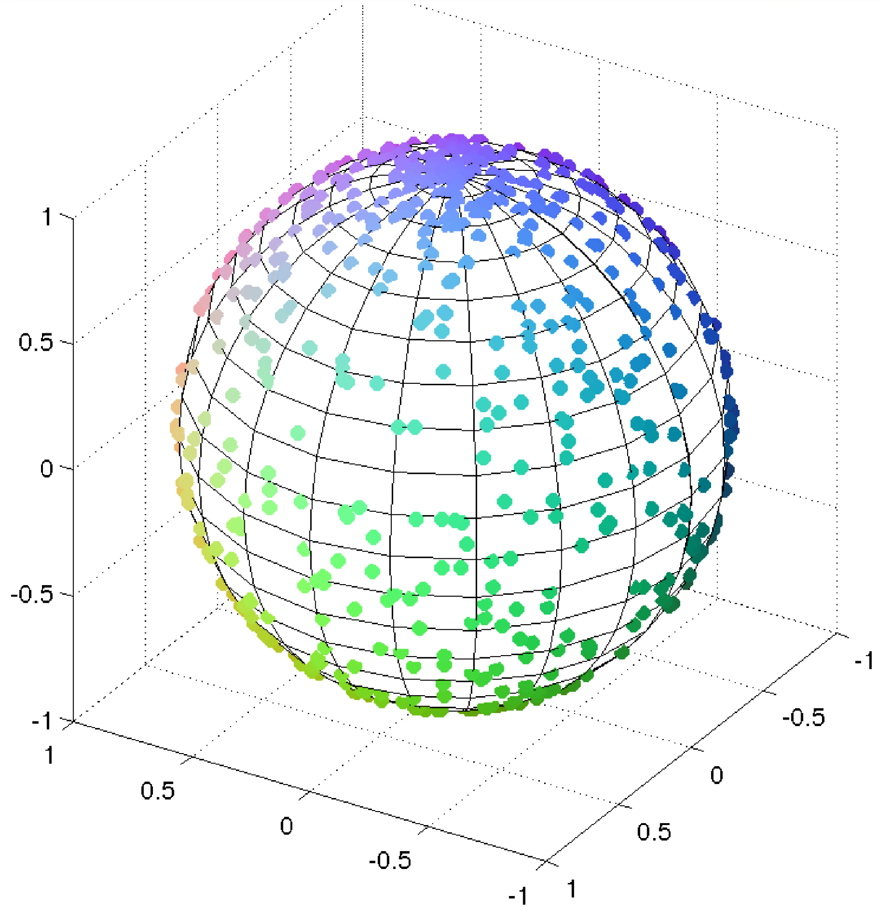


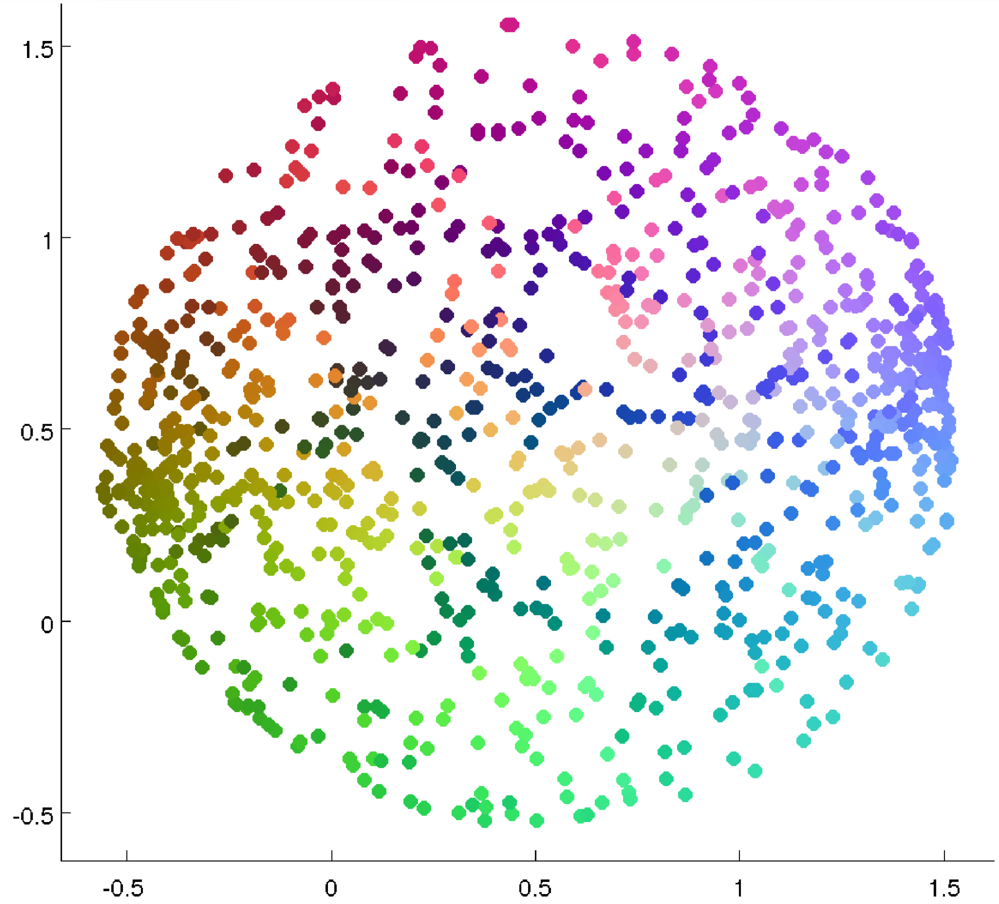
Example data set





“Orange-peel map”



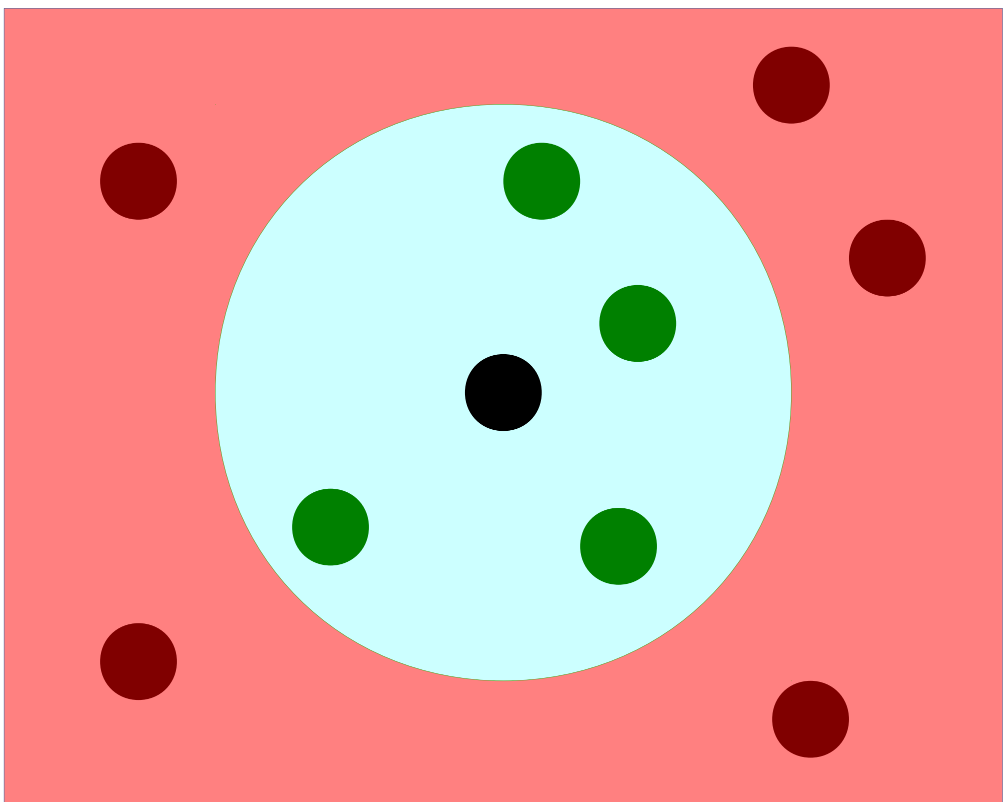


“Squashed-flat sphere”

Neighborhood Preservation

Neighbors are an important concept in many applications: neighboring cities, friends on social networks, followers of blogs, links between webpages.---> **Preserve neighbors** instead of distances?

In vectorial data, if nothing else is known, it is reasonable that close-by points in some metric can be considered neighbors.

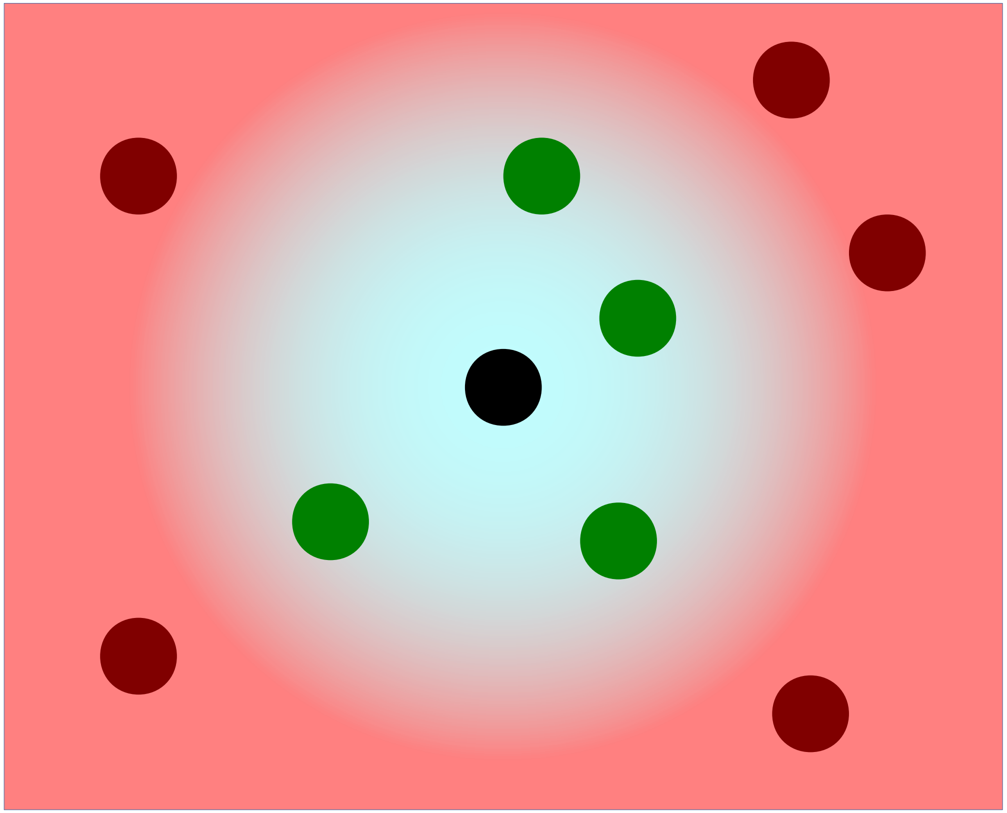


Hard neighborhood - each point is a neighbor or a non-neighbor

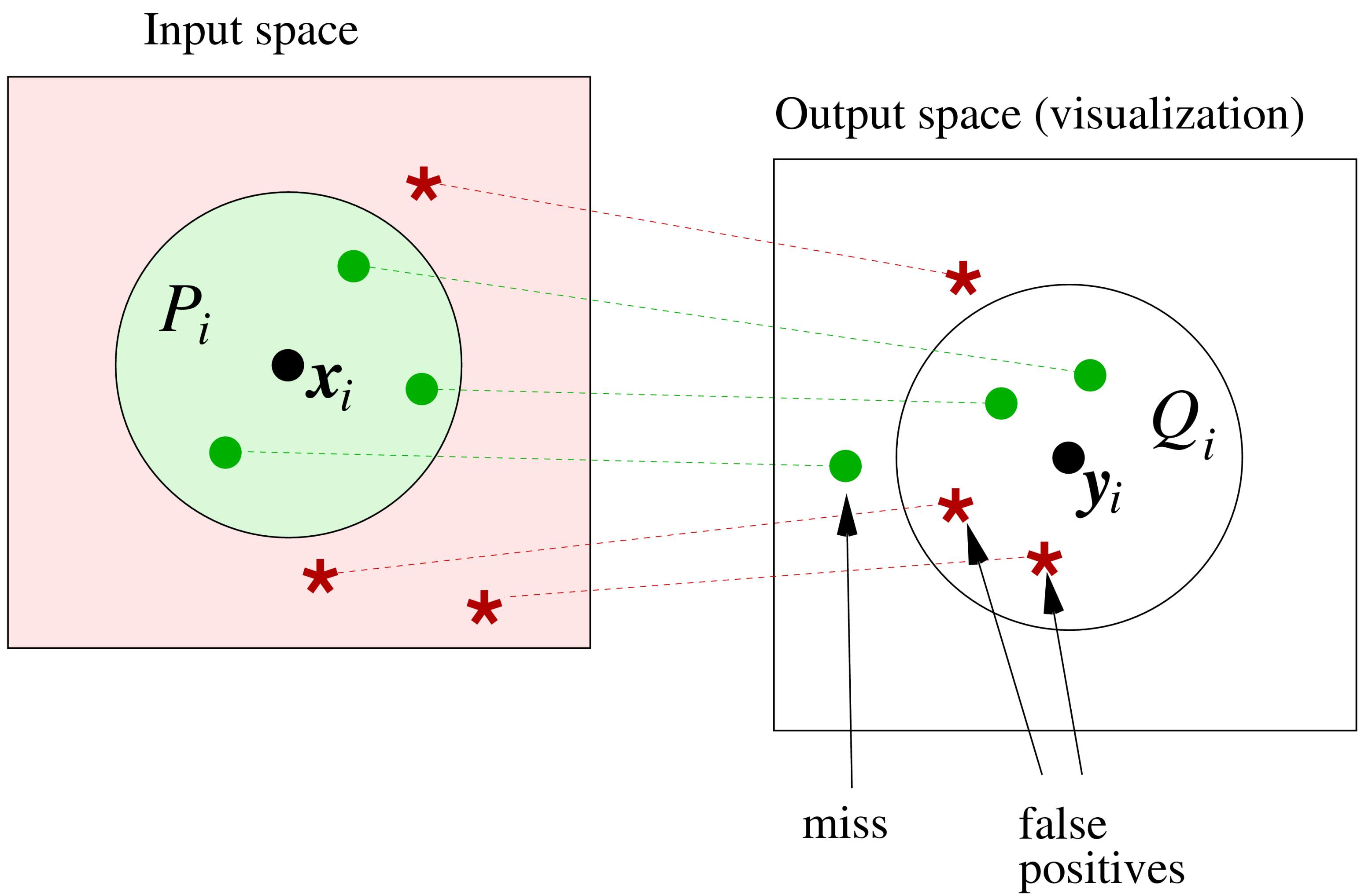
Neighborhood Preservation

Neighbors are an important concept in many applications: neighboring cities, friends on social networks, followers of blogs, links between webpages.---> **Preserve neighbors** instead of distances?

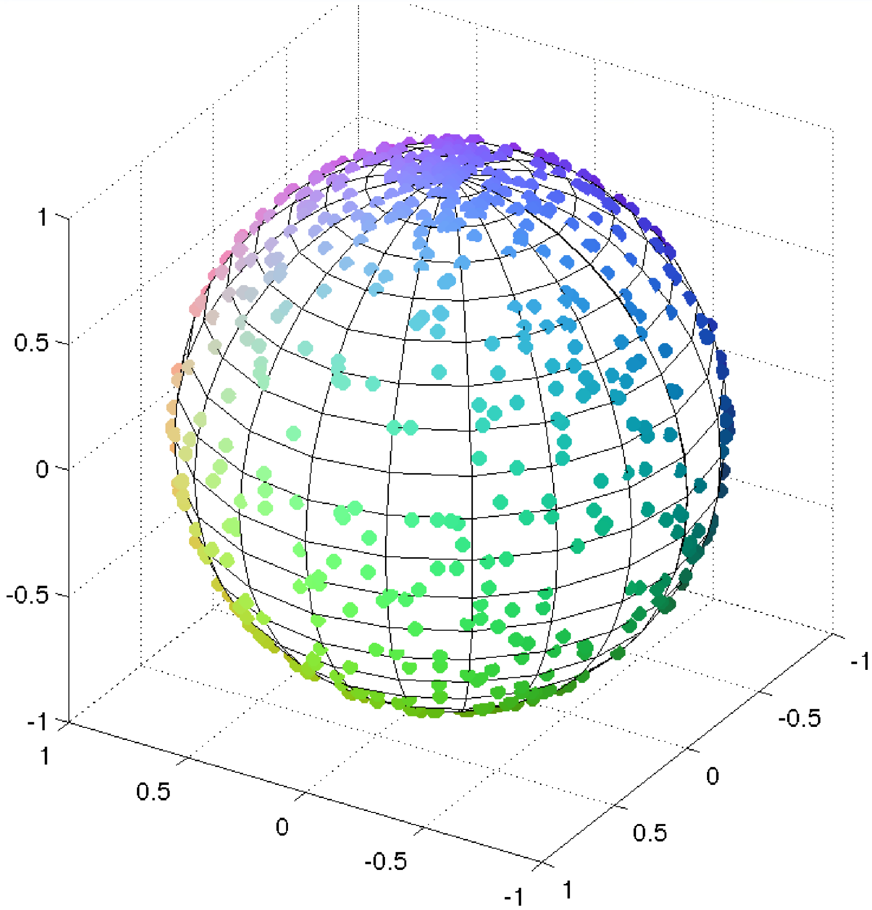
In vectorial data, if nothing else is known, it is reasonable that close-by points in some metric can be considered neighbors.



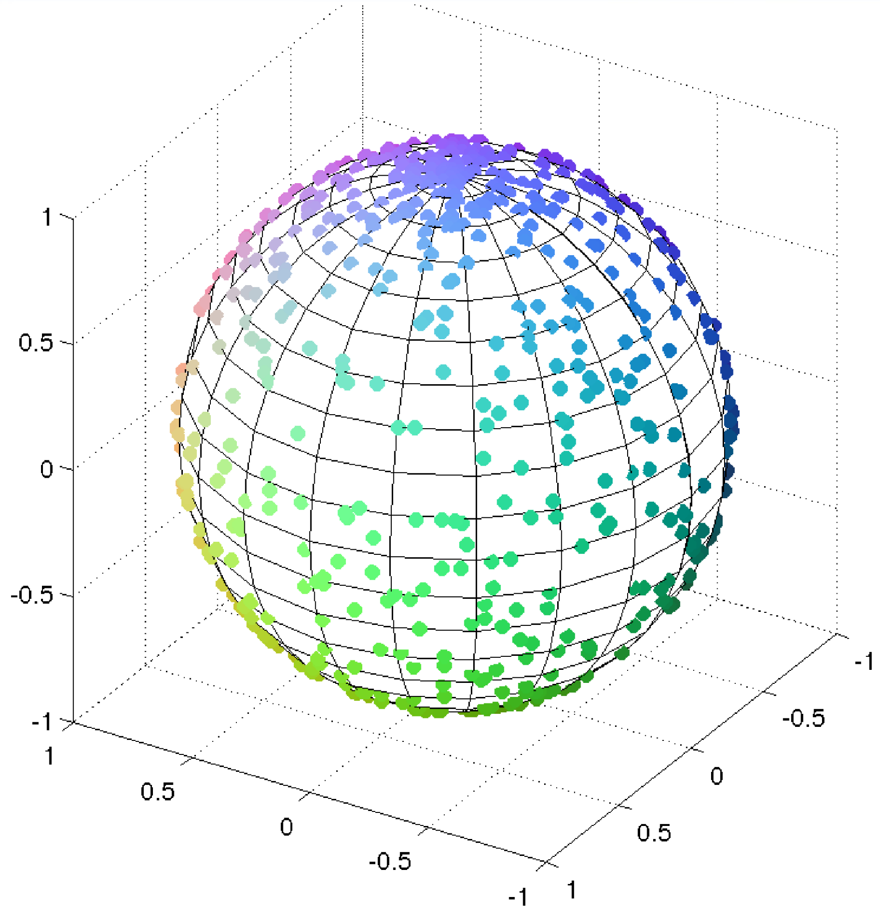
Soft neighborhood - each point is a neighbor with some weight and a non-neighbor with some weight

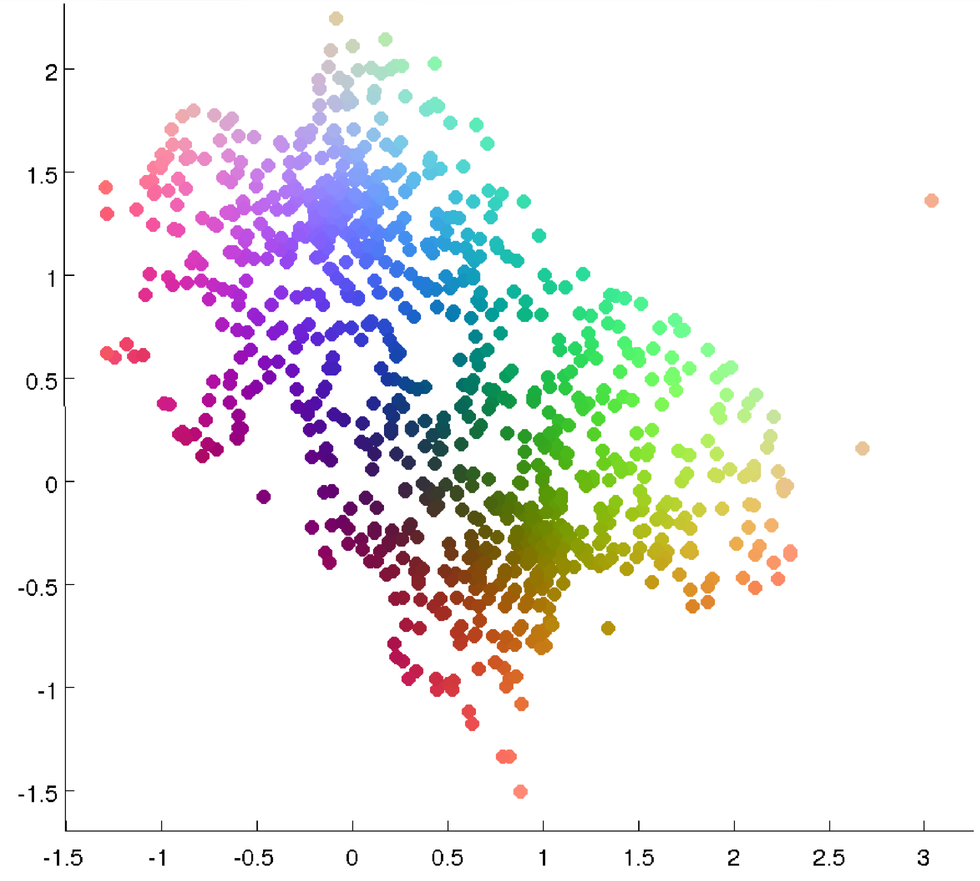


Minimize errors for best information retrieval.

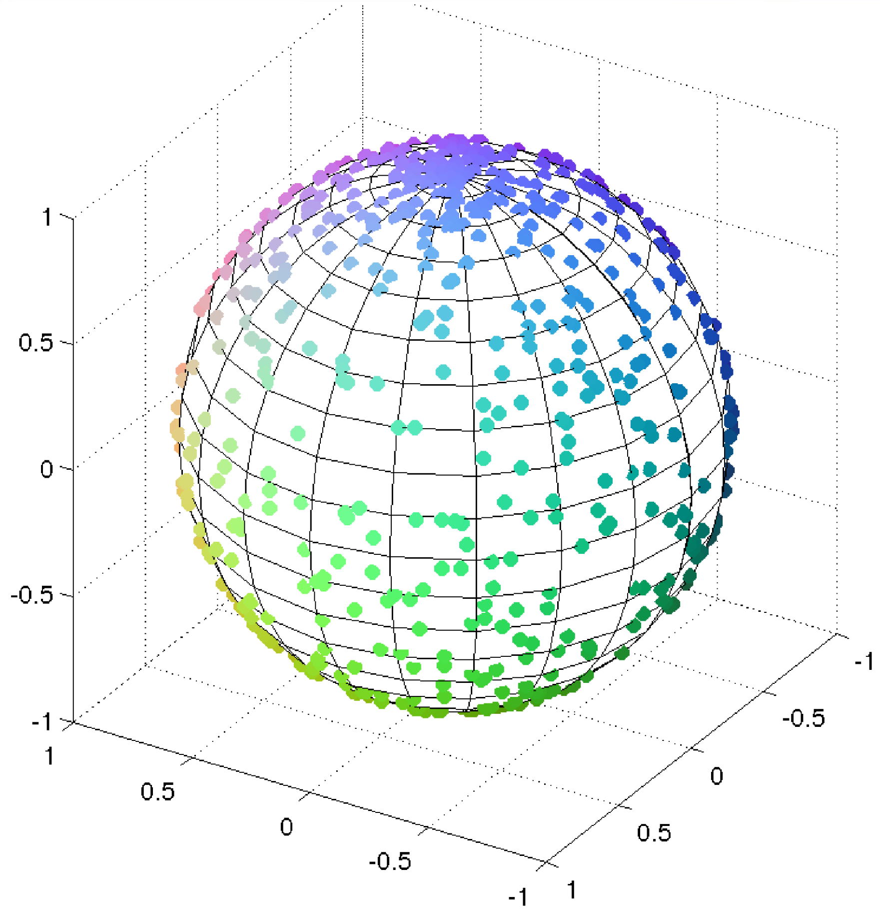


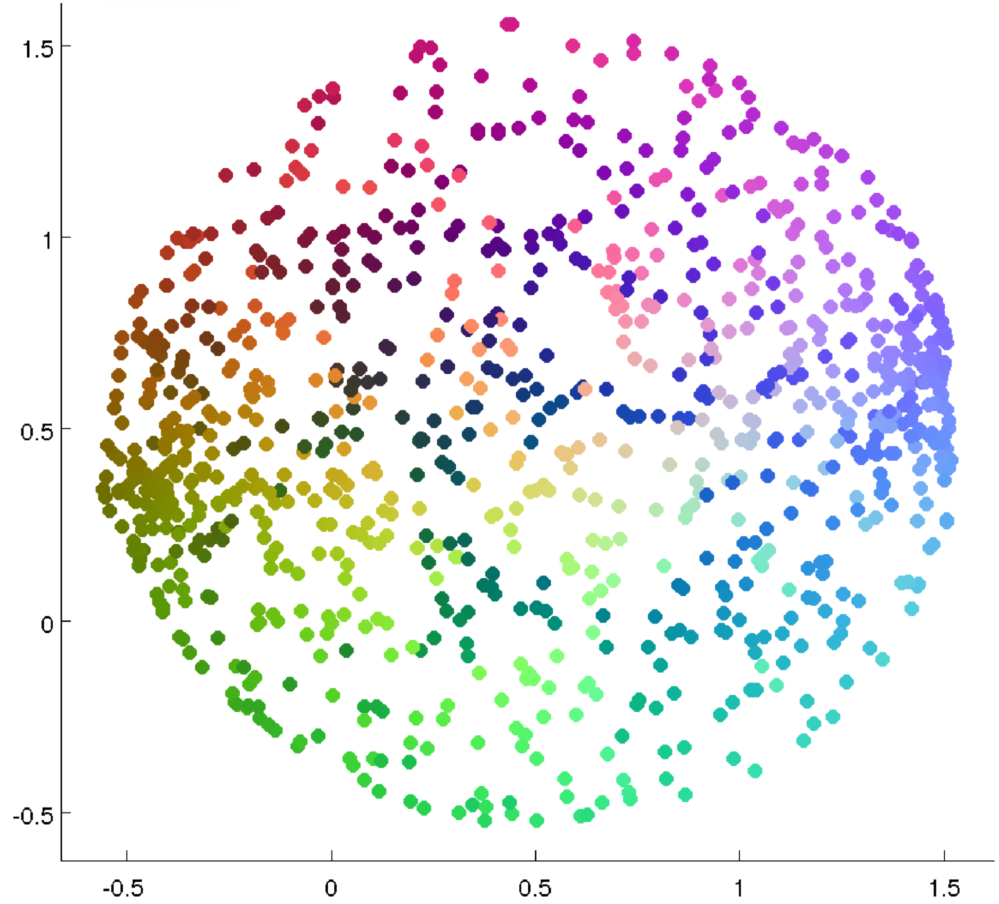
Example data set





Embedding mininizes false positives
(falsely retrieved neighbors)





Embedding mininizes misses (neighbors that were not retrieved)

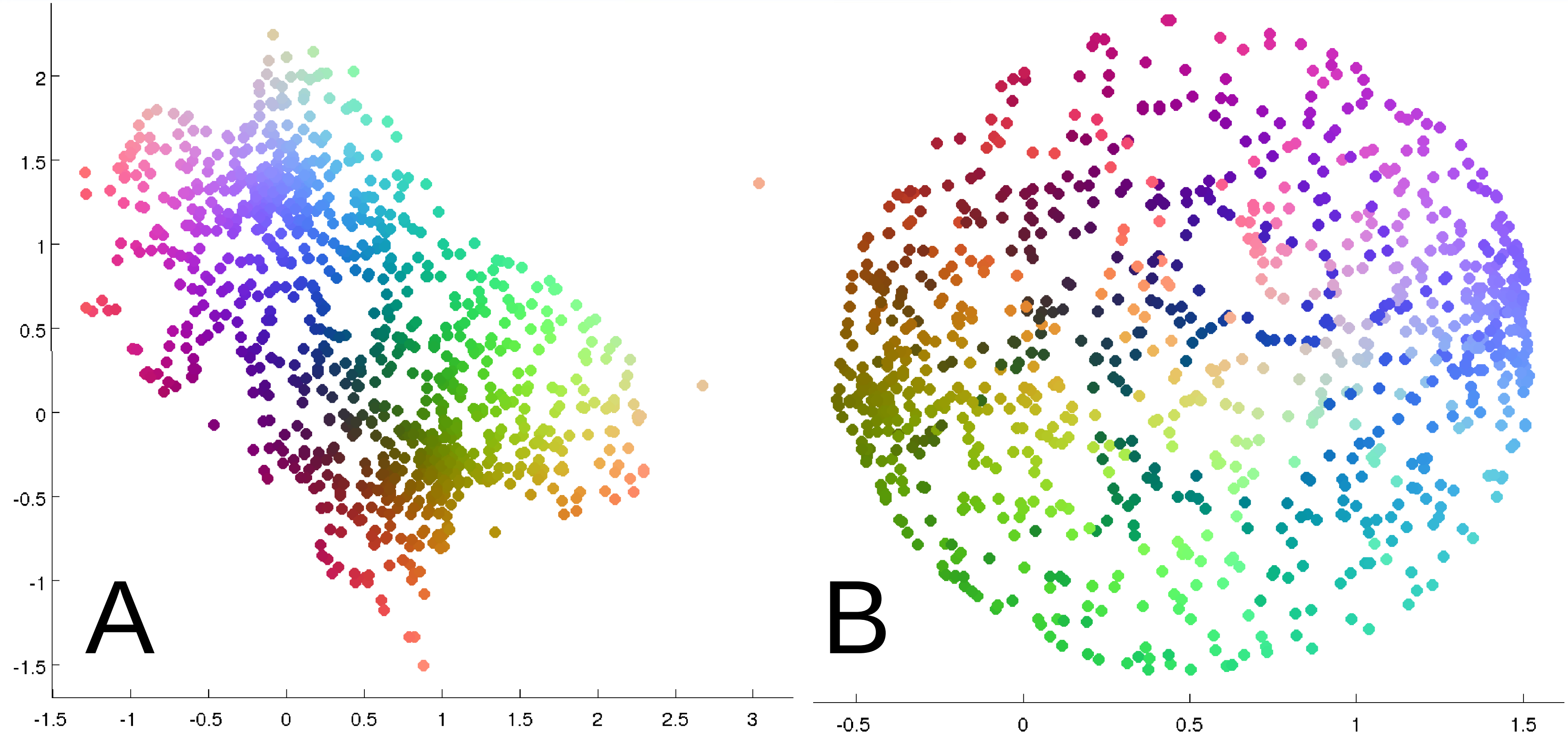
$$1 - precision = \frac{P_i^C \cap Q_i}{|Q_i|}$$

Proportion of false positives

$$1 - recall = \frac{Q_i^C \cap P_i}{|P_i|}$$

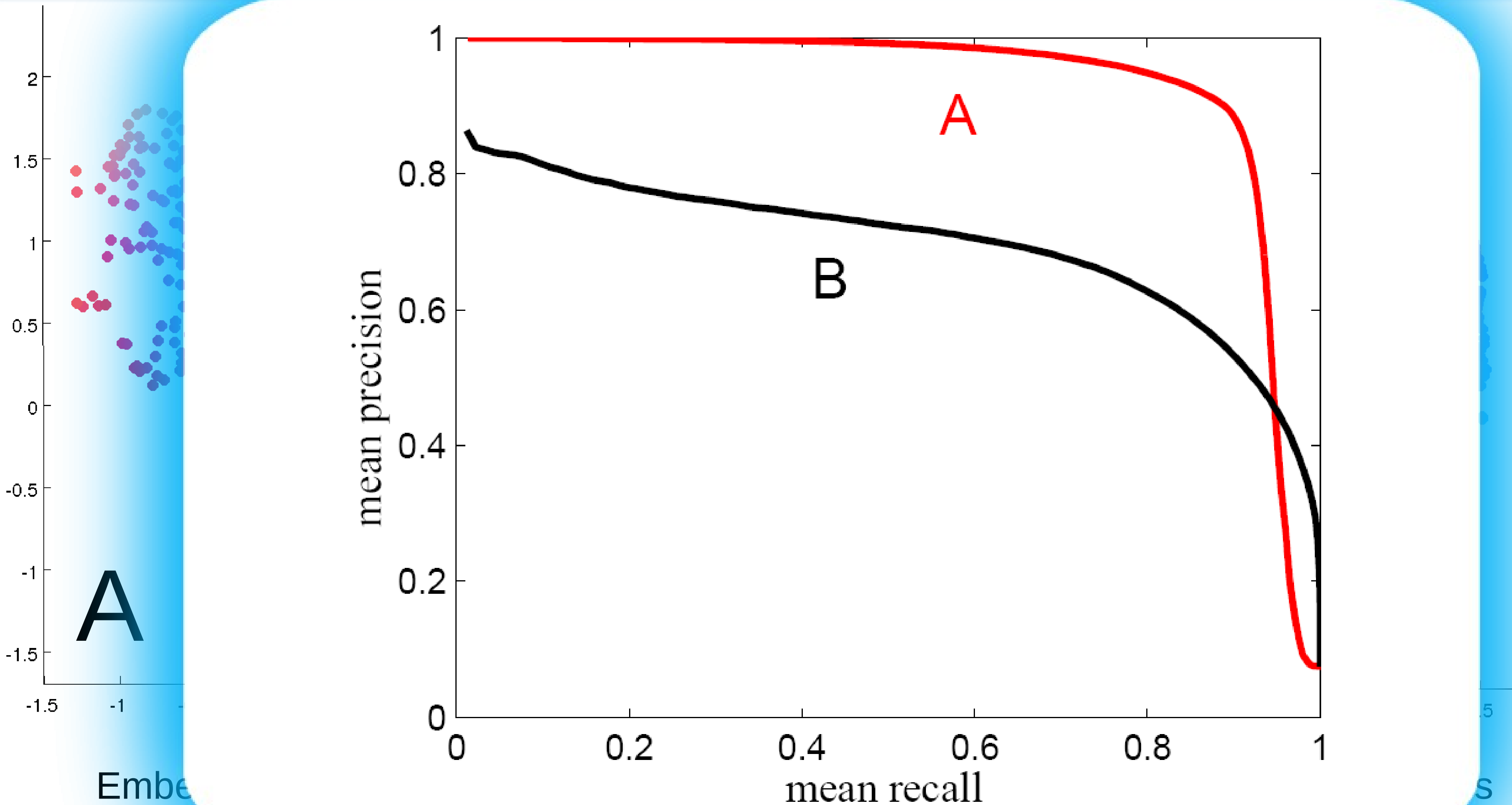
Proportion of missed neighbors

A visualization must make a **tradeoff** between false positives and misses. All methods end up with some tradeoff. A good visualization method should allow the user to specify the desired tradeoff.

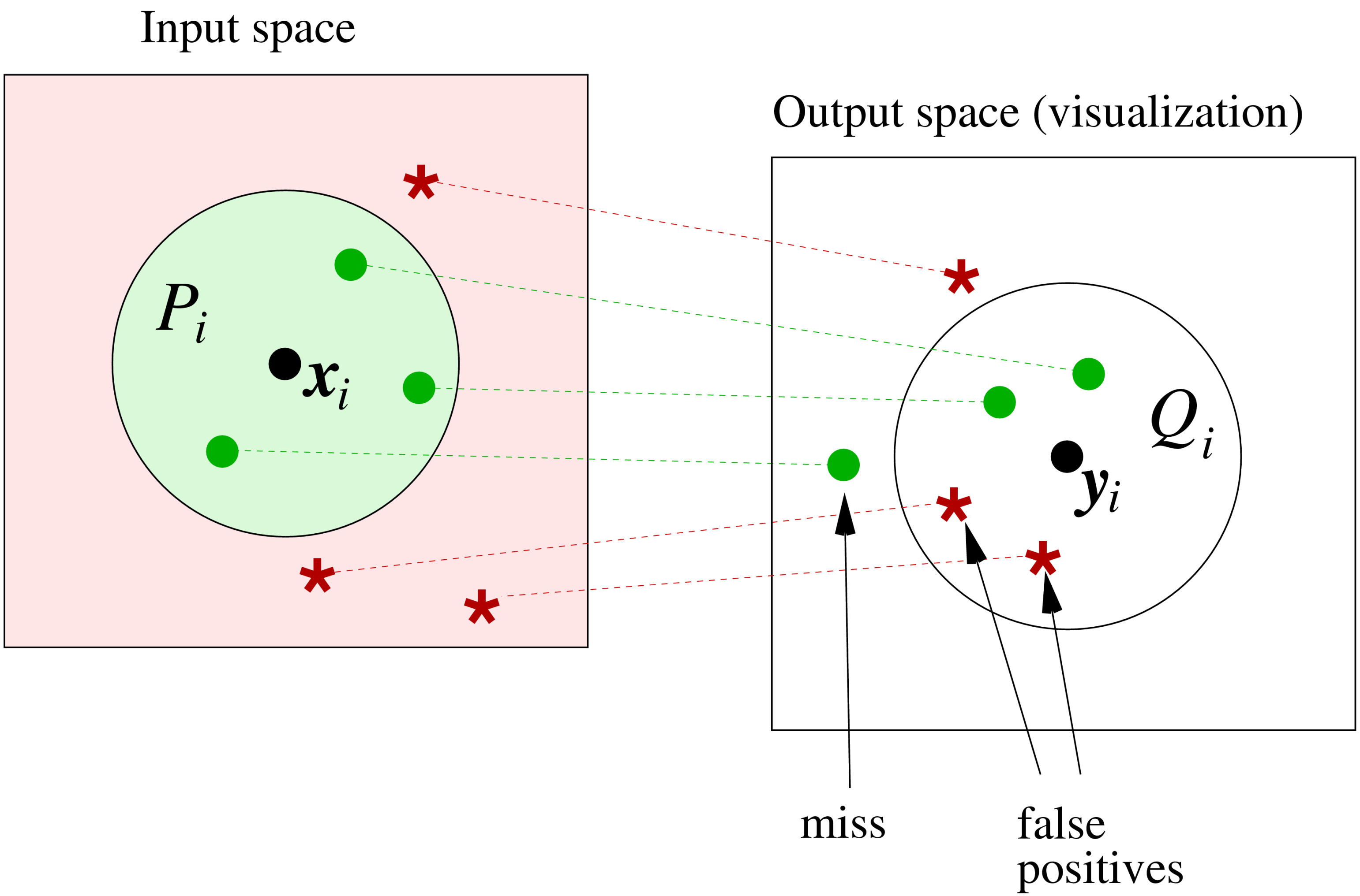


Embedding mininizes false positives (falsely retrieved neighbors)

Embedding mininizes misses (neighbors that were not retrieved)



Embe … s
(falsely r

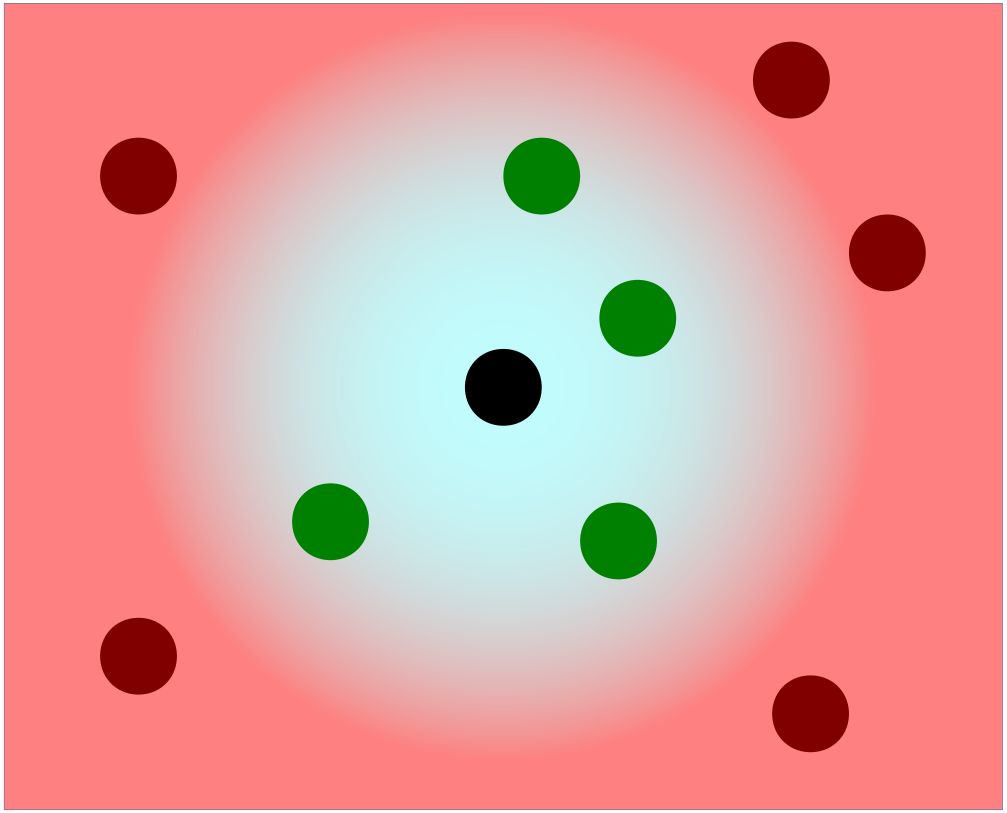


What about continuous neighborhoods?

Neighborhood Preservation

Neighbors are an important concept in many applications: neighboring cities, friends on social networks, followers of blogs, links between webpages.---> **Preserve neighbors** instead of distances?

In vectorial data, if nothing else is known, it is reasonable that close-by points in some metric can be considered neighbors.



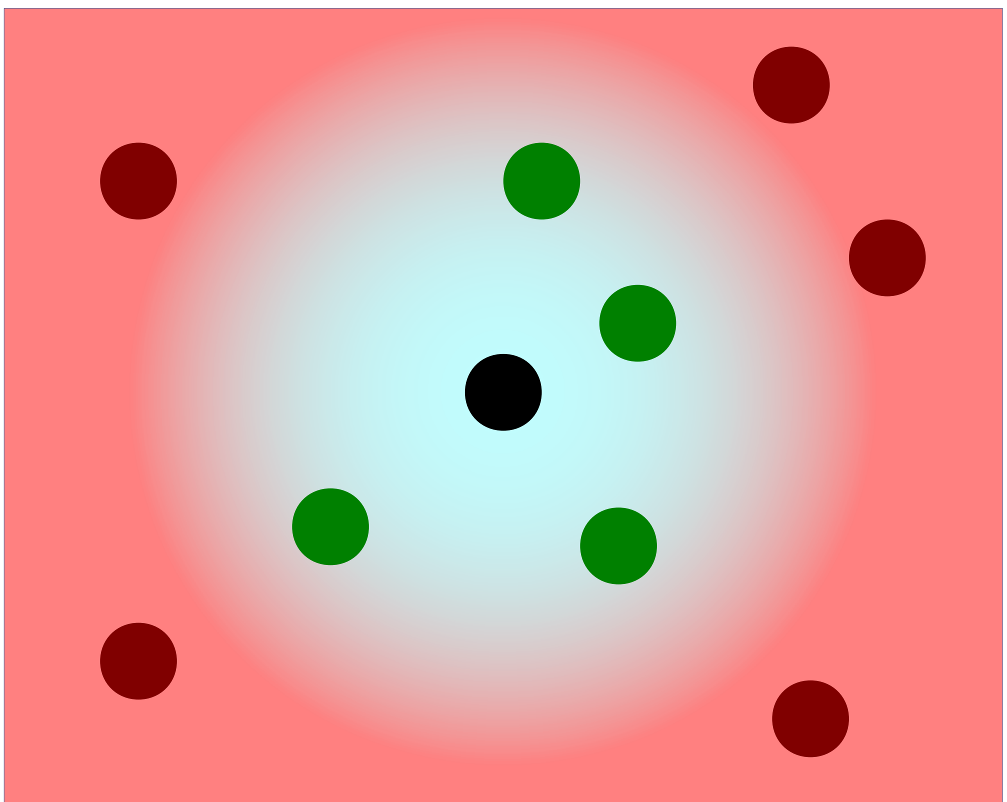
Soft neighborhood - each point is a neighbor with some weight and a non-neighbor with some weight

Some images and equations on the following slides are from the SNE paper (Roweis & Hinton '02).

Neighborhood Preservation

Neighbors are an important concept in many applications: neighboring cities, friends on social networks, followers of blogs, links between webpages.---> **Preserve neighbors** instead of distances?

In vectorial data, if nothing else is known, it is reasonable that close-by points in some metric can be considered neighbors.



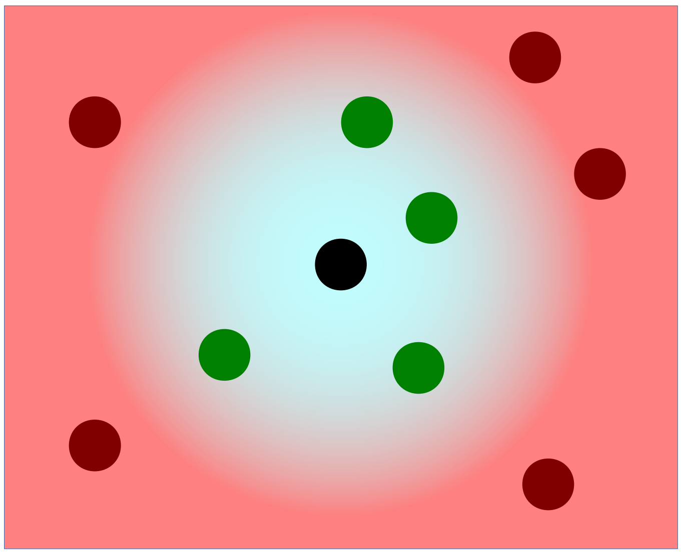
Probabilistic neighborhood

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$$

(probability to be picked as a neighbor **in input space**.)

Neighborhood Preservation

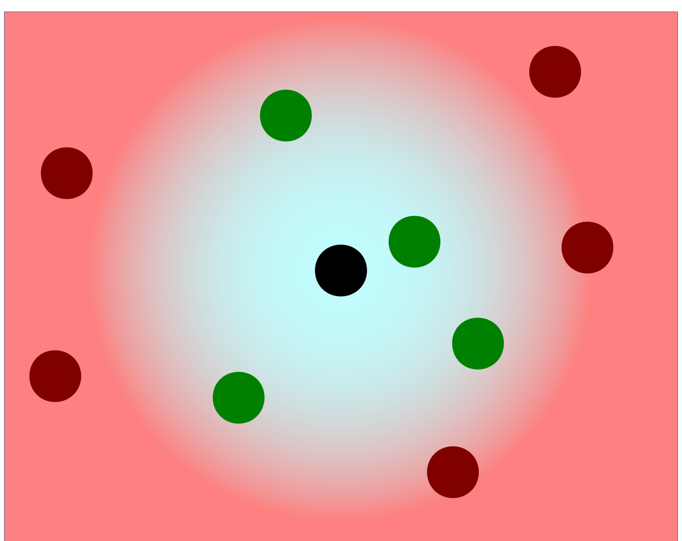
In vectorial data, if nothing else is known, it is reasonable that close-by points in some metric can be considered neighbors.



Probabilistic input neighborhood

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$$

(probability to be picked as a neighbor)



Probabilistic output neighborhood

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

(probability based on display coords.)

Input neighborhood

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

Output neighborhood

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

Input neighborhood

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

Output neighborhood

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

Recall:

$$\sum_i$$

Input neighborhood

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

Output neighborhood

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

Recall:

$$\sum_i \sum_{j \neq i}$$

Input neighborhood

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

Output neighborhood

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

Recall:

$$\sum_i \sum_{j\neq i} p_{j|i}$$

Input neighborhood

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

Output neighborhood

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

Recall:

$$\sum_i \sum_{j\neq i} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Input neighborhood

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

Output neighborhood

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

Precision:

$$\sum_i \sum_{j \neq i}$$

Input neighborhood

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

Output neighborhood

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

Precision:

$$\sum_{i} \sum_{j \neq i} q_{j|i}$$

Input neighborhood

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

Output neighborhood

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

Precision:

$$\sum_i \sum_{j\neq i} q_{j|i} \log \frac{q_{j|i}}{p_{j|i}}$$

Input neighborhood

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

Output neighborhood

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

Tradeoff measure

$$E_{\mathrm{NeRV}} = \lambda\mathbb{E}_i[D(p_i,q_i)] + (1-\lambda)\mathbb{E}_i[D(q_i,p_i)]$$

Minimize with respect to output coordinates y_i

Neighbor Retrieval Visualizer

Input neighborhood

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

Output neighborhood

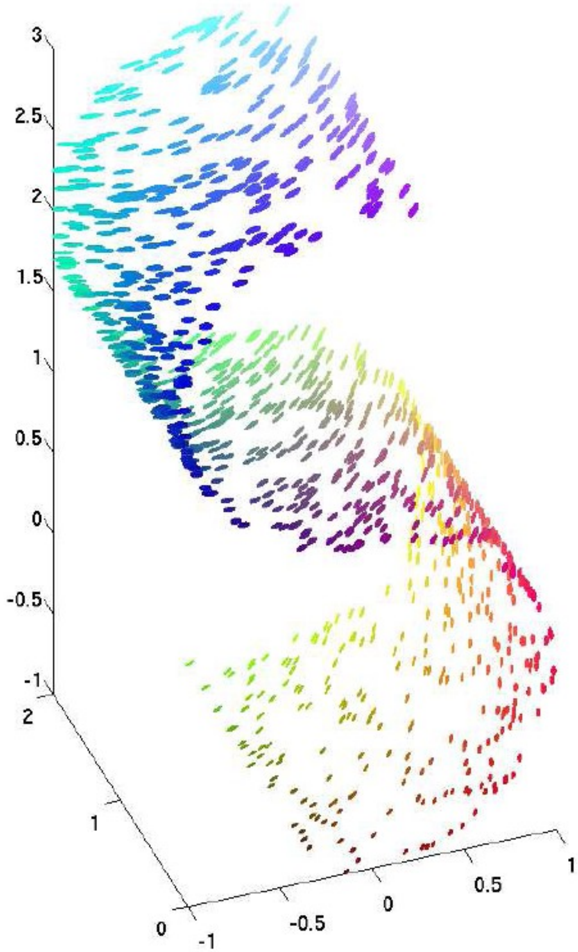
$$q_{j|i} = \frac{\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_k\|^2}{\sigma_i^2}\right)}$$

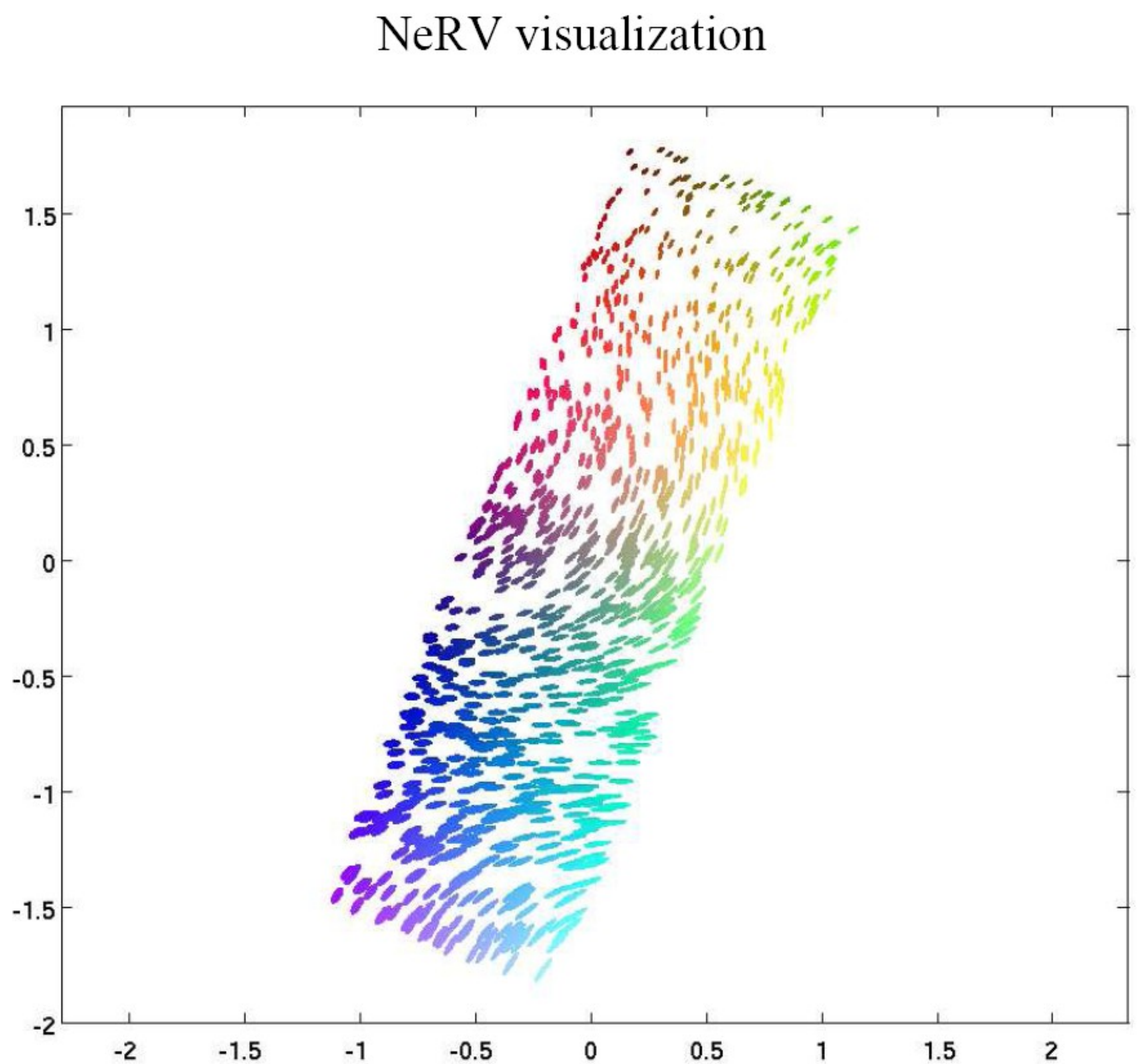
Tradeoff measure

$$E_{\mathrm{NeRV}} = \lambda \mathbb{E}_i[D(p_i, q_i)] + (1-\lambda)\mathbb{E}_i[D(q_i, p_i)]$$

Minimize with respect to output coordinates y_i

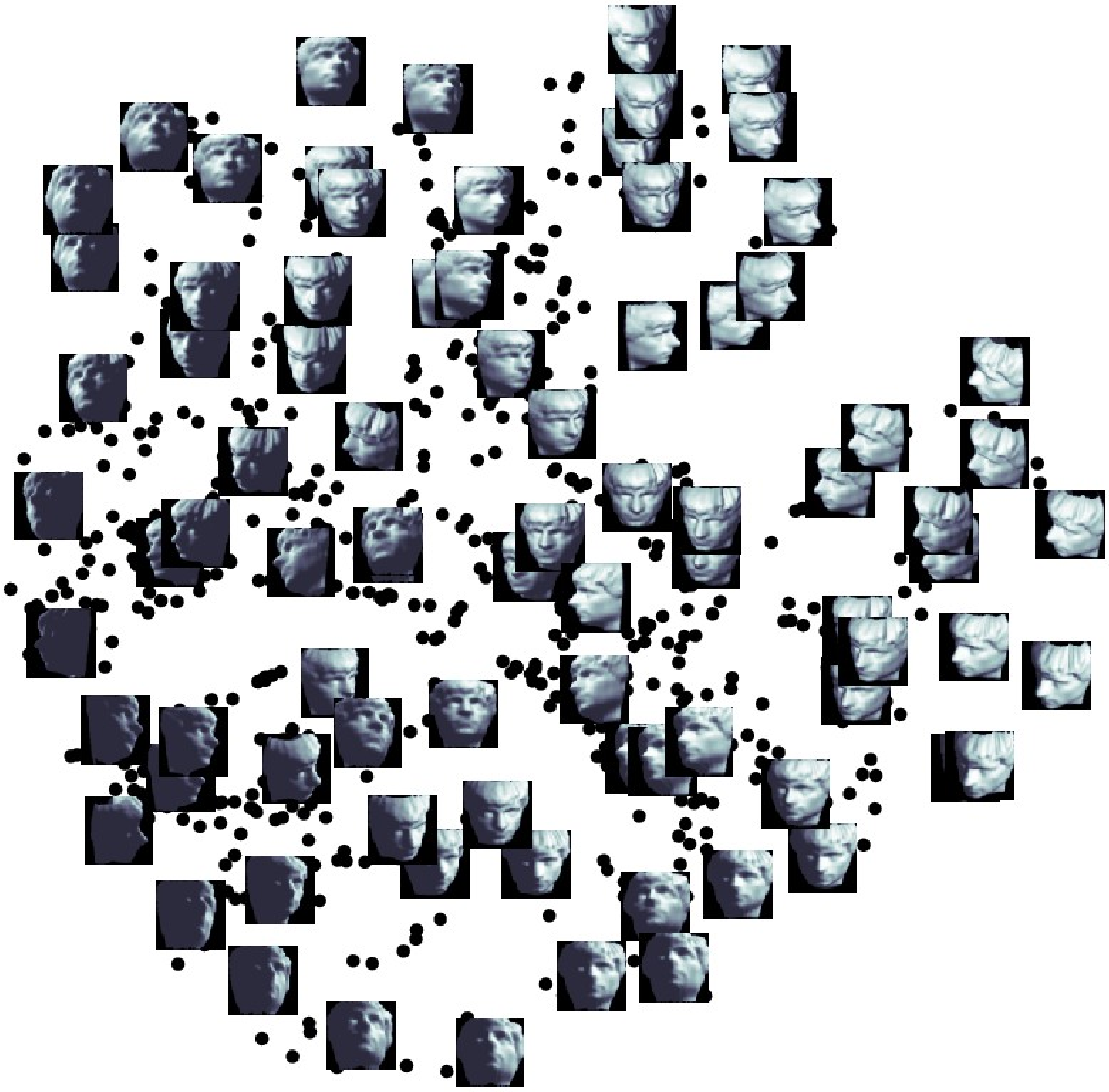
Neighbor Retrieval Visualizer





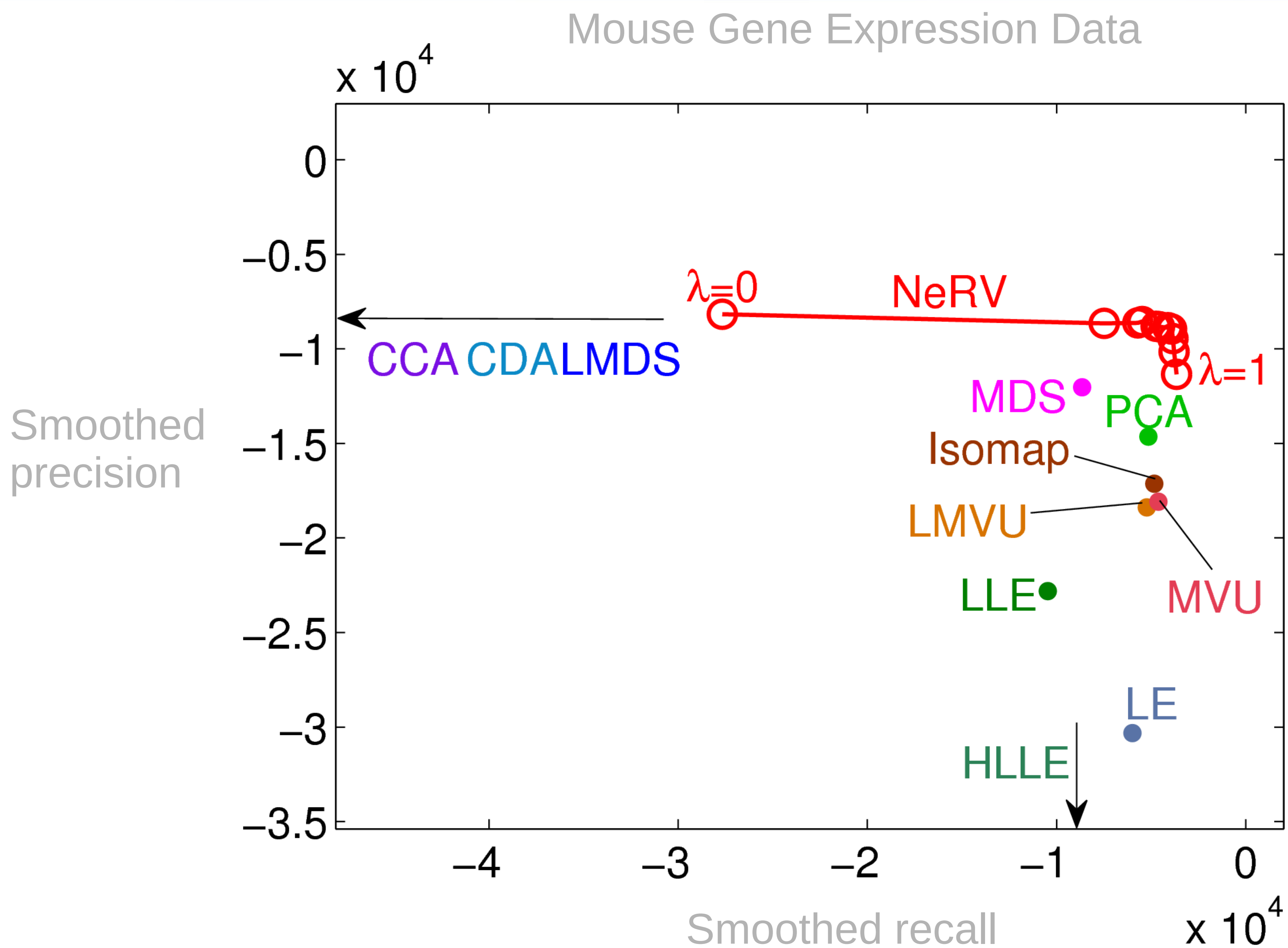
Of course NeRV can unfold the simple cases.

Neighbor Retrieval Visualizer

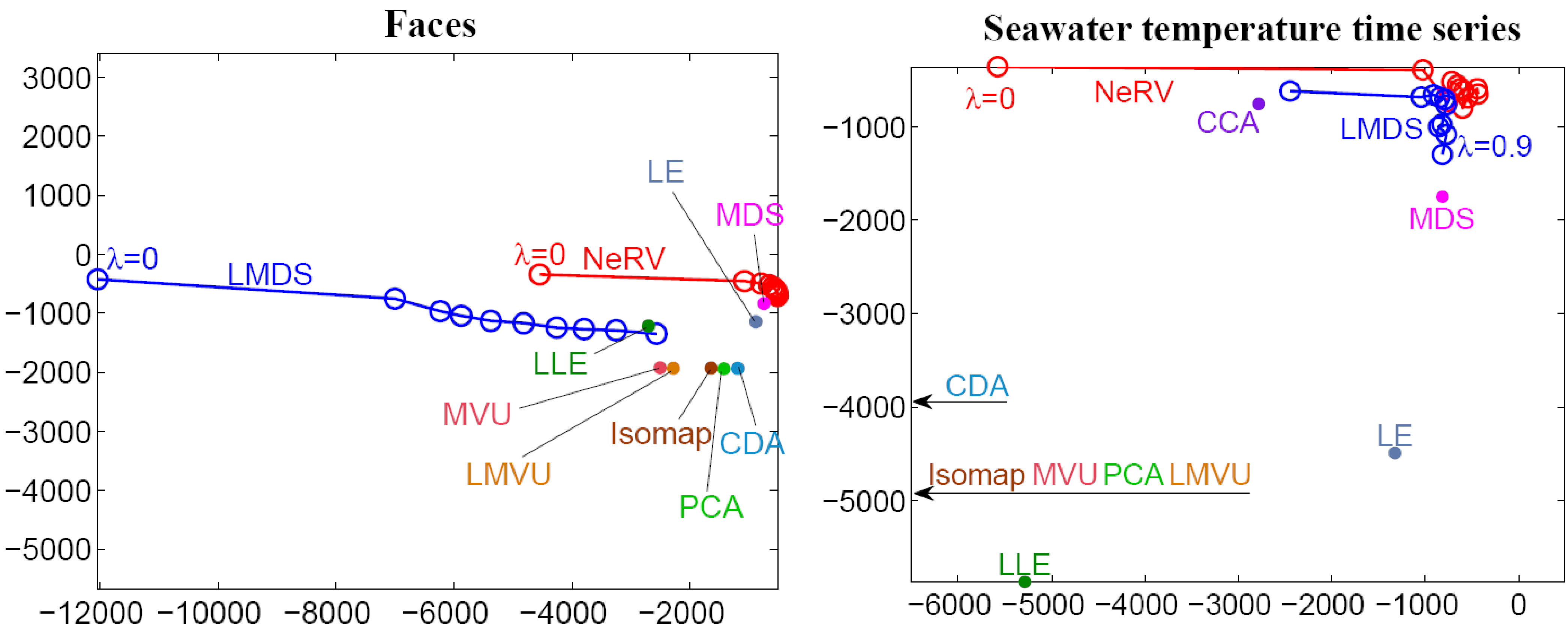


NeRV visualization of a complicated face image data set

Neighbor Retrieval Visualizer

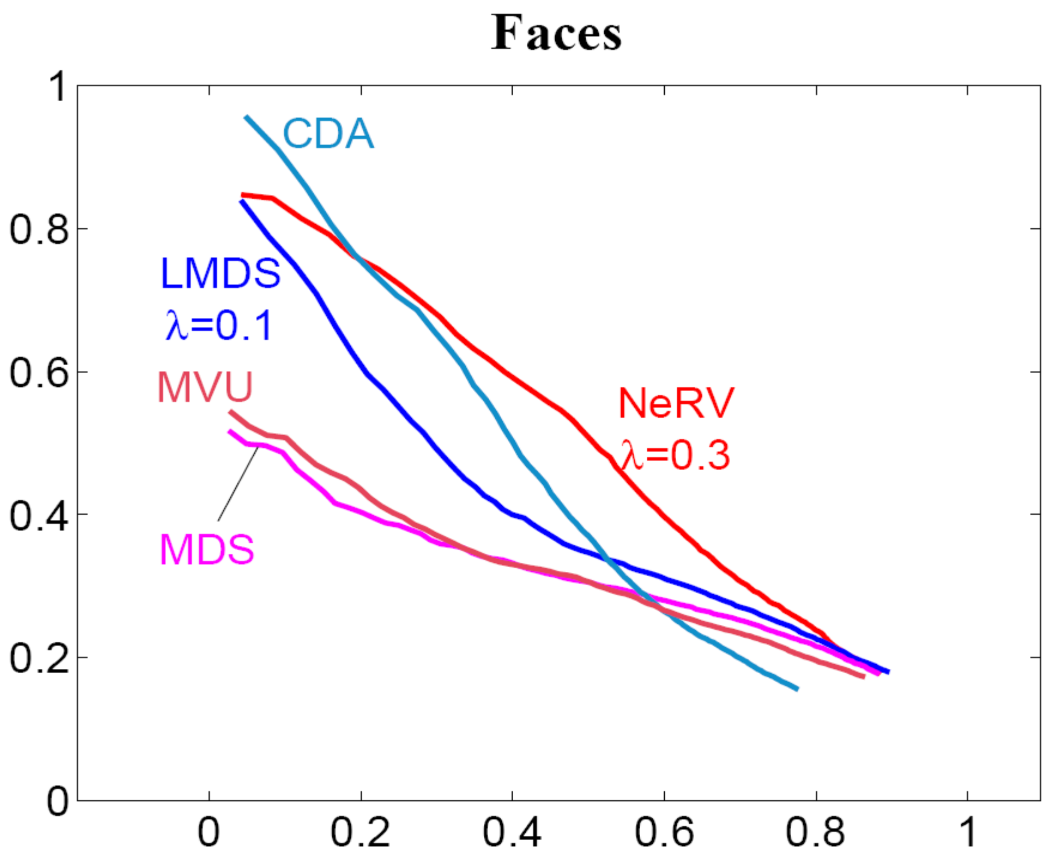


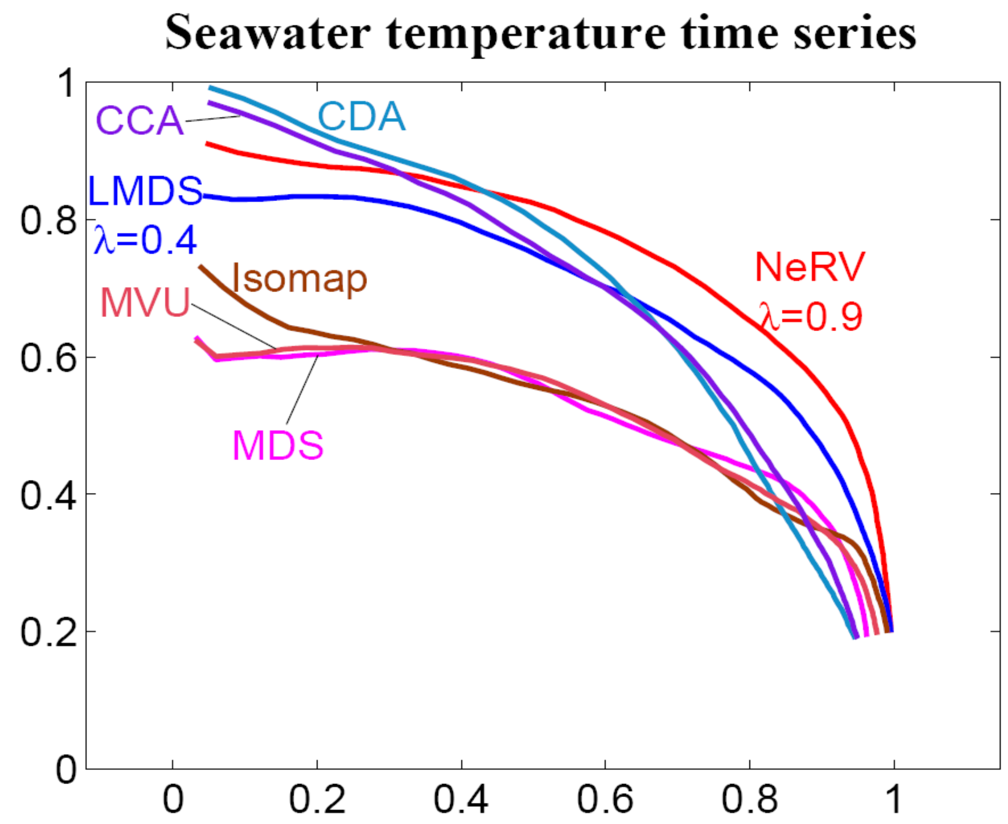
Neighbor Retrieval Visualizer



New measures: smoothed precision / recall

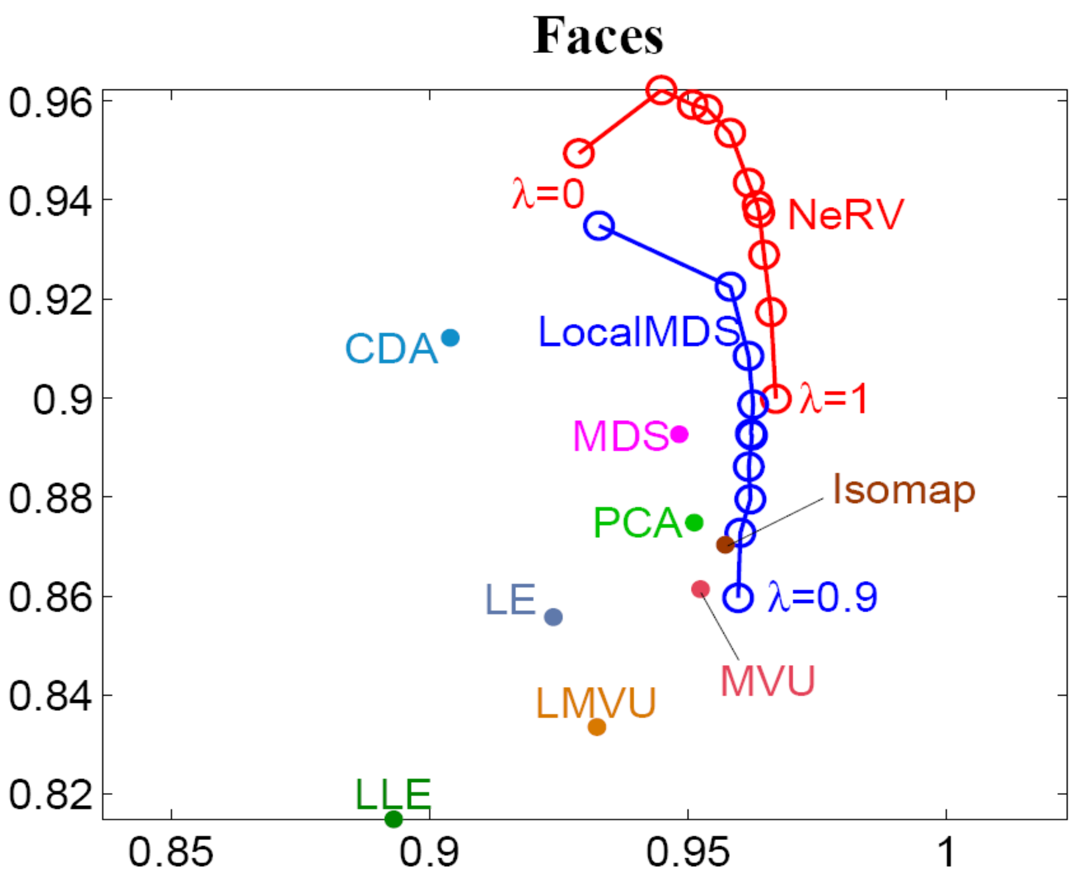
Neighbor Retrieval Visualizer

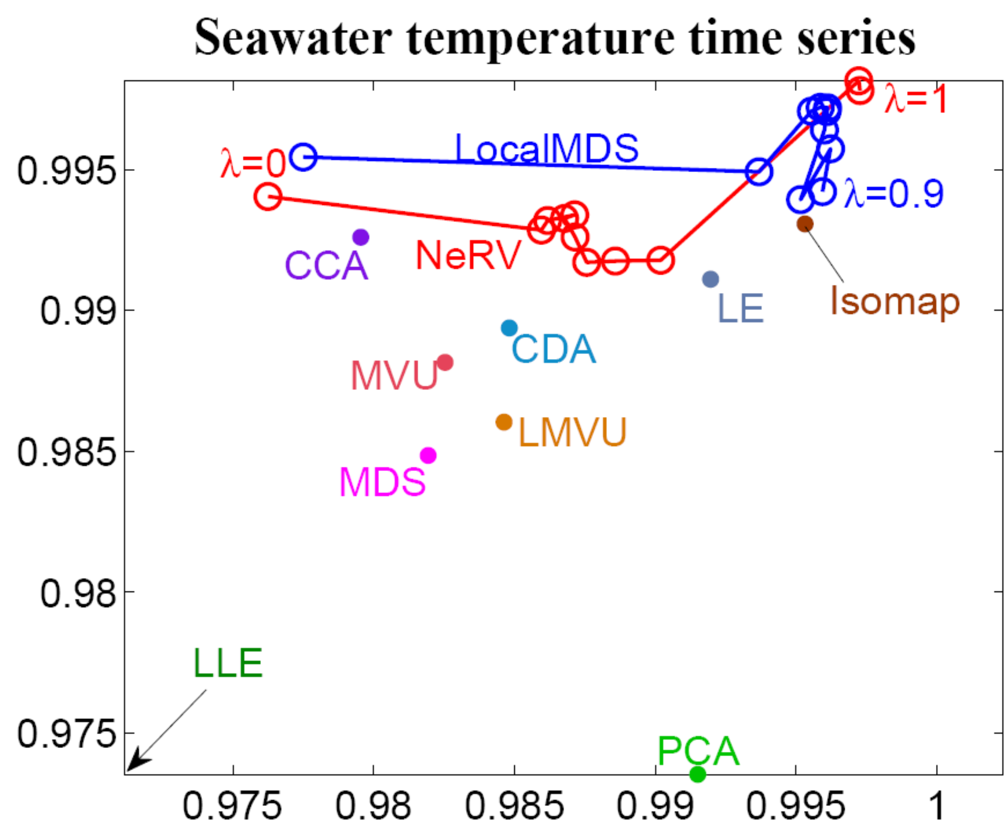




Standard precision / recall curves (novel for visualization!)

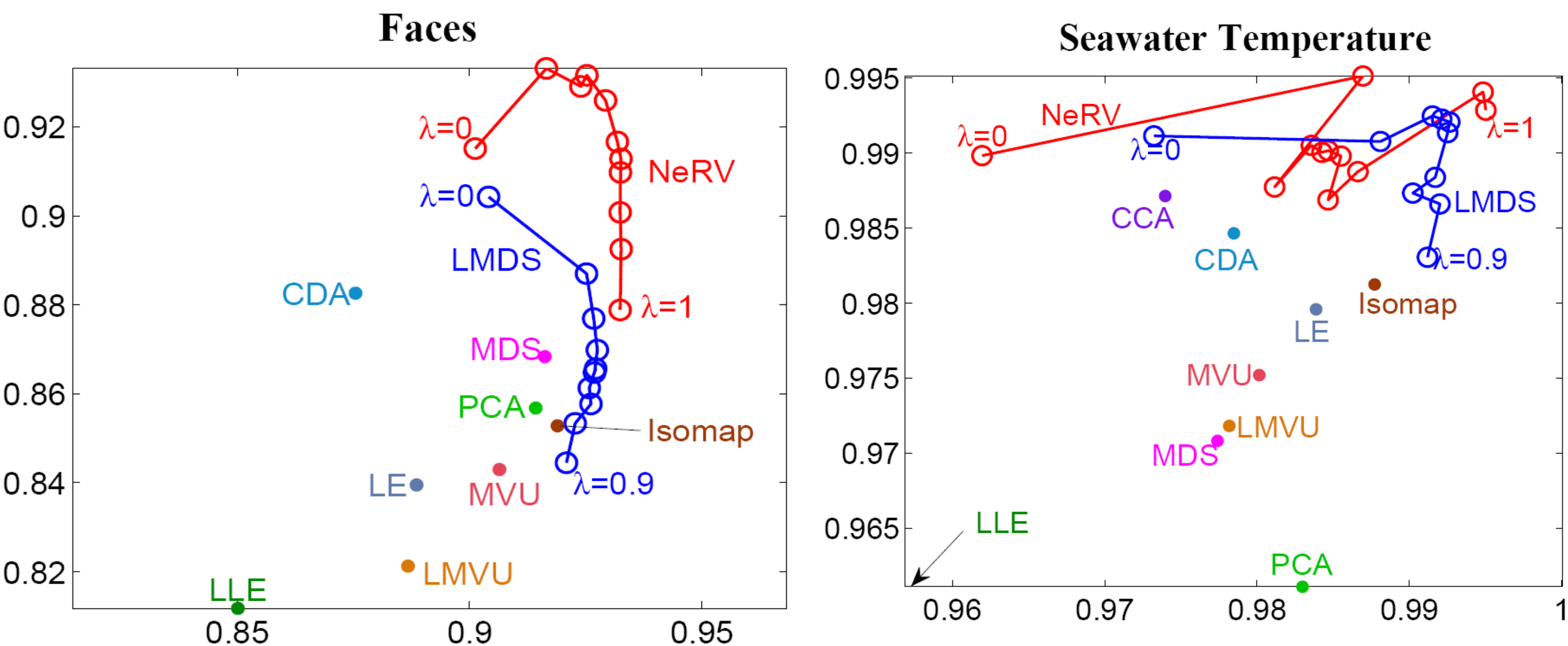
Neighbor Retrieval Visualizer





New measures: rank-based smoothed precision / recall

Neighbor Retrieval Visualizer



Earlier measures: trustworthiness / continuity

Extension 1: t-distributed NeRV

Input neighborhood

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

$$p_{i,j} = \frac{1}{2n}(p_{i|j} + p_{j|i})$$

Output neighborhood

$$q_{i,j} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}$$

Tradeoff measure = t-NeRV cost function

$$E_{\text{t-NeRV}} = \lambda \sum_i \sum_{j \neq i} p_{i,j} \log \frac{p_{i,j}}{q_{i,j}} + (1-\lambda) \sum_i \sum_{j \neq i} q_{i,j} \log \frac{q_{i,j}}{p_{i,j}} = \lambda D(p,q) + (1-\lambda) D(q,p)$$

Extension 2: NeRV with a linear projection

Even if the objective function of visualization is an advanced concept like neighborhood preservation, there is still interest in keeping the mapping function simple.

An example of a simple mapping: linear projection.

Idea: use the well-justified NeRV objective function, which has an information retrieval interpretation, but constrain the mapping to be a linear projection.

Advantage: the mapping is easy to interpret, can e.g. find out what are the main original features affecting each axis of the visualization.

Disadvantage: linear projection is more constrained --> cannot preserve neighborhoods as well as a nonlinear mapping.

Extension 2: NeRV with a linear projection

Input neighborhood

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

Output neighborhood

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

Tradeoff measure = NeRV cost function

$$E_{\text{NeRV}} = \lambda\mathbb{E}_i[D(p_i,q_i)] + (1-\lambda)\mathbb{E}_i[D(q_i,p_i)]$$

Restrict $\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i$

Minimize cost with respect to projection W

Extension 2: NeRV with a linear projection

Input neighborhood

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

Output neighborhood

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

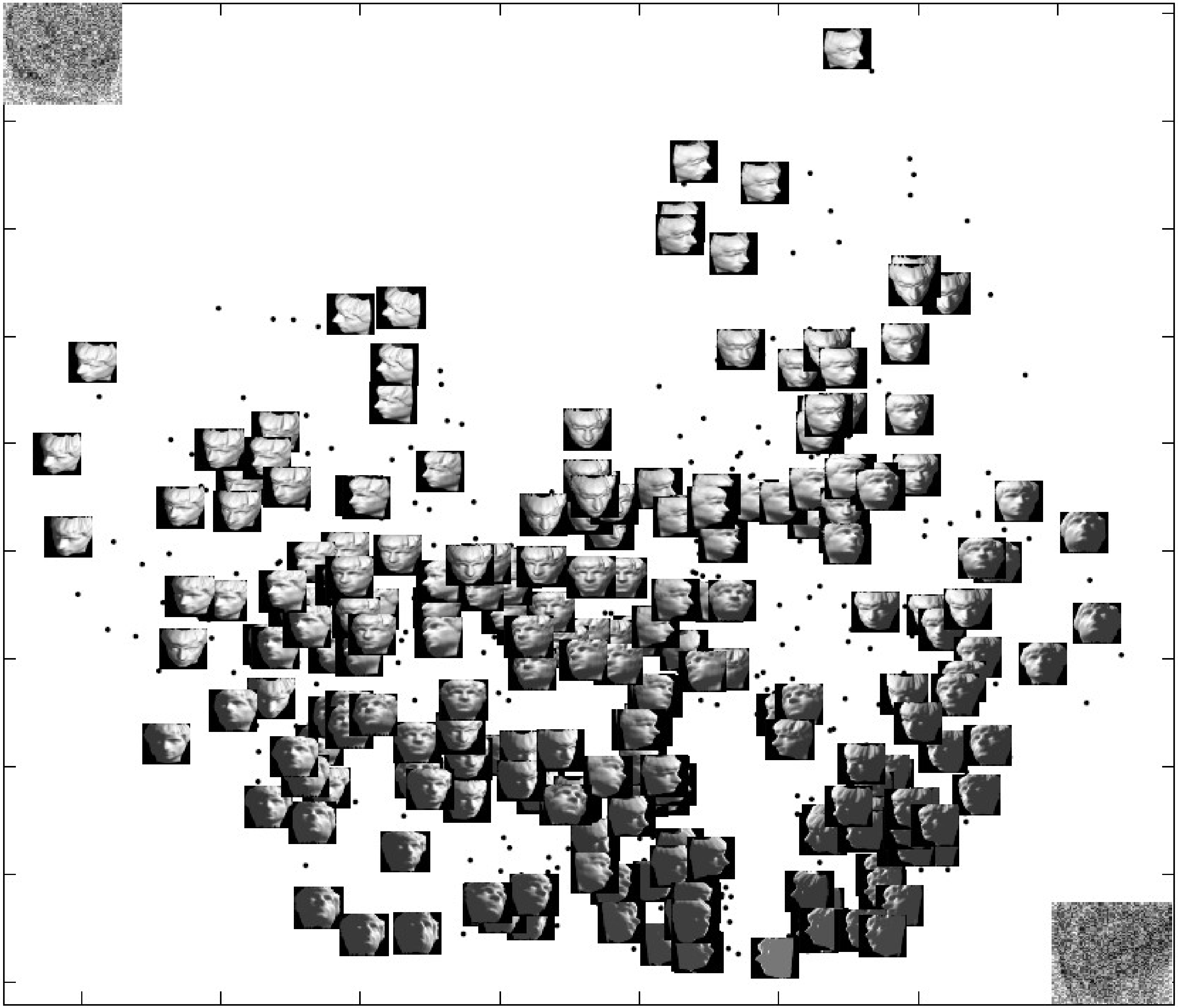
Tradeoff measure = NeRV cost function

$$E_{\text{NeRV}} = \lambda\mathbb{E}_i[D(p_i,q_i)] + (1-\lambda)\mathbb{E}_i[D(q_i,p_i)]$$

Restrict $\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i$

Minimize cost with respect to projection W

Extension 2: NeRV with a linear projection.
Neighborhoods and features can be given separately.



Extension 3: NeRV by generative modeling

Stochastic Neighbor Embedding can be seen as a generative model, but it only focuses on recall (misses) because its cost function is **dominated by misses.**

Idea: **change the retrieval model so that misses become less dominant, so that the model can also focus on false positives**.

New retrieval distribution: mixture of the **user model** and an **explaining away model**.

$$q_{ij} \propto r_{ij} + \gamma p_{ij}$$

new retrieval distribution

plain user model

explaining away model = true neighborhood distribution

amount of explaining away

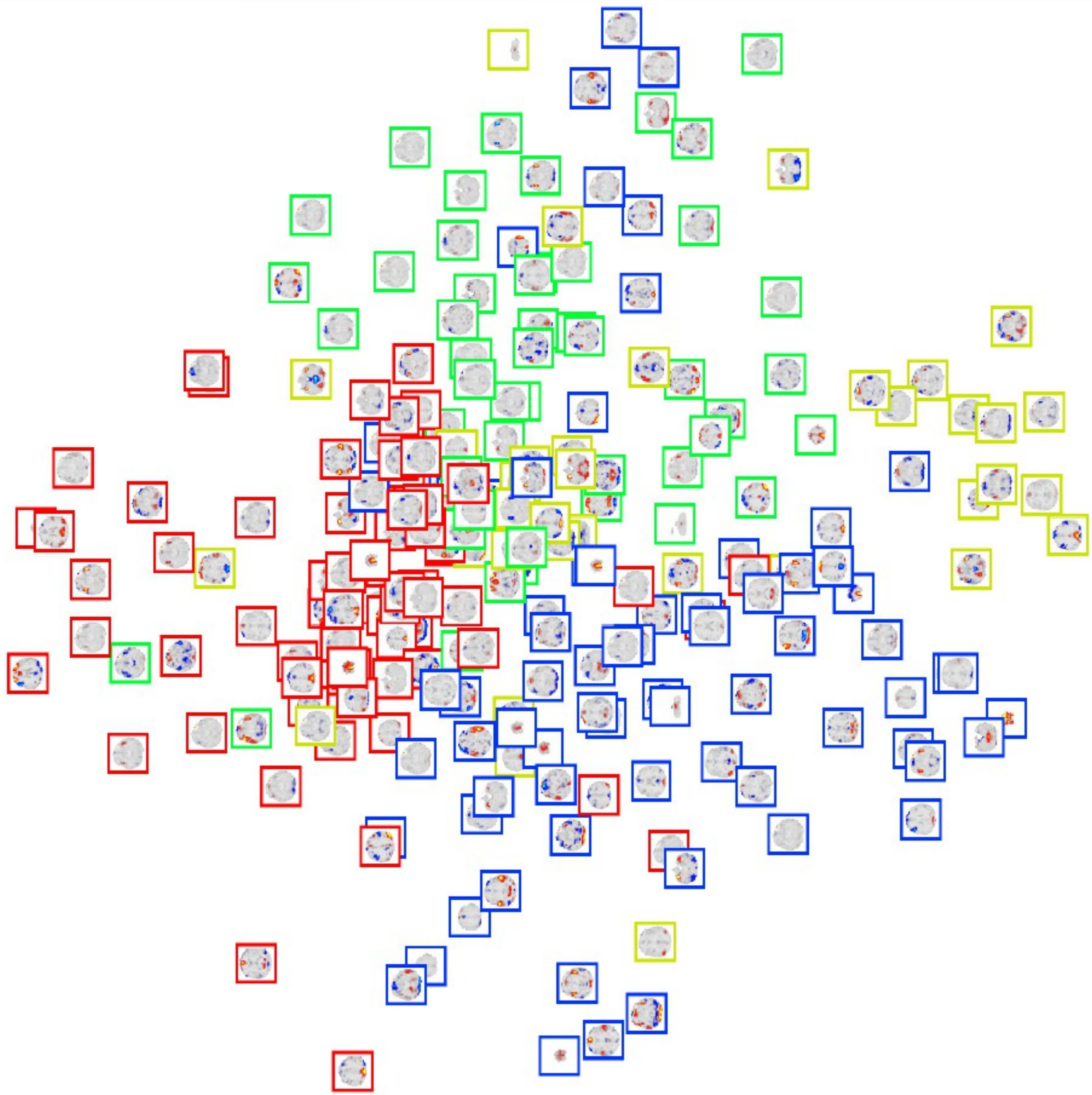
Cost function is log-likelihood (generative modeling):

$$L = \sum_i \sum_{j \neq i} p_{ij} \log q_{ij}$$

Extension 3: NeRV by generative modeling

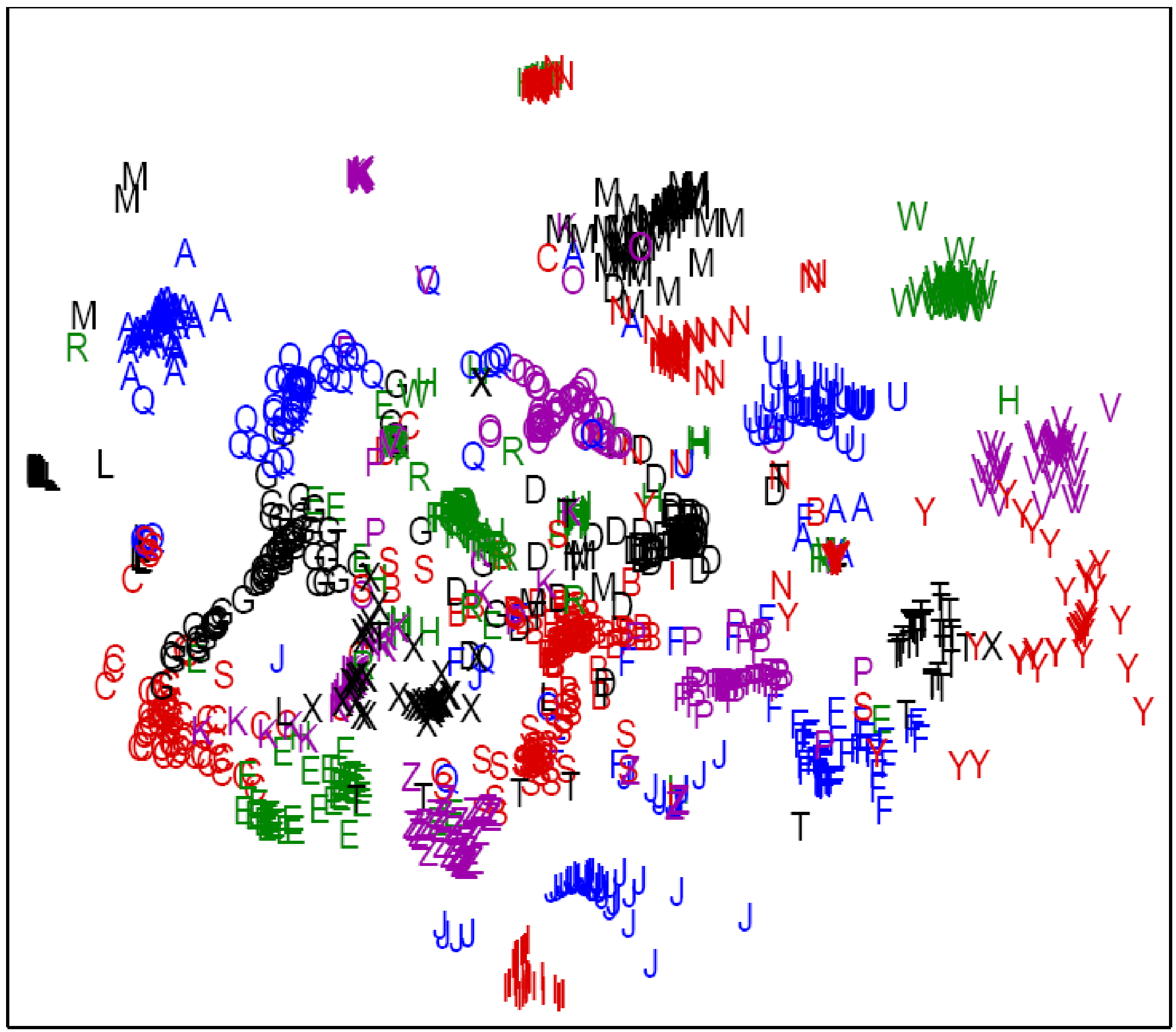
fMRI measurements of 6 adults who received four types of stimuli:
- tactile (red)
- auditory tone (yellow)
- auditory voice (green)
- visual (blue).

Visualization by the new method, strong explaining away used during training. Different stimuli types become separated in the (unsupervised) visualization.



Extension 4: Supervised NeRV

When we have data labels, how can we take them into account while still being able to visualize also unlabeled data? How do we avoid visualizing trivial label similarity?



Extension 4: Supervised NeRV

NeRV works based on a distance metric. We can learn it from data and labels. The Learning Metric is a supervised “topology-preserving” distance metric learned from class probabilities

- a Riemannian metric
- assume we have a class estimator $p(c|\mathbf{x})$

- local distance:

$$d_L^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) \equiv D_{KL}(p(c|\mathbf{x})||p(c|\mathbf{x} + d\mathbf{x})) = d\mathbf{x}^T \mathbf{J}(\mathbf{x}) d\mathbf{x}$$

$$\mathbf{J}(\mathbf{x}) = E_{p(c|\mathbf{x})} \left\{ \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right)^T \right\}$$

- global distance:

$$d(p, q) = \inf_{\{\gamma|\gamma(0)=p, \gamma(1)=q\}} \int_0^1 d_L(\gamma(t), \gamma(t+dt)) dt$$

Extension 4: Supervised NeRV

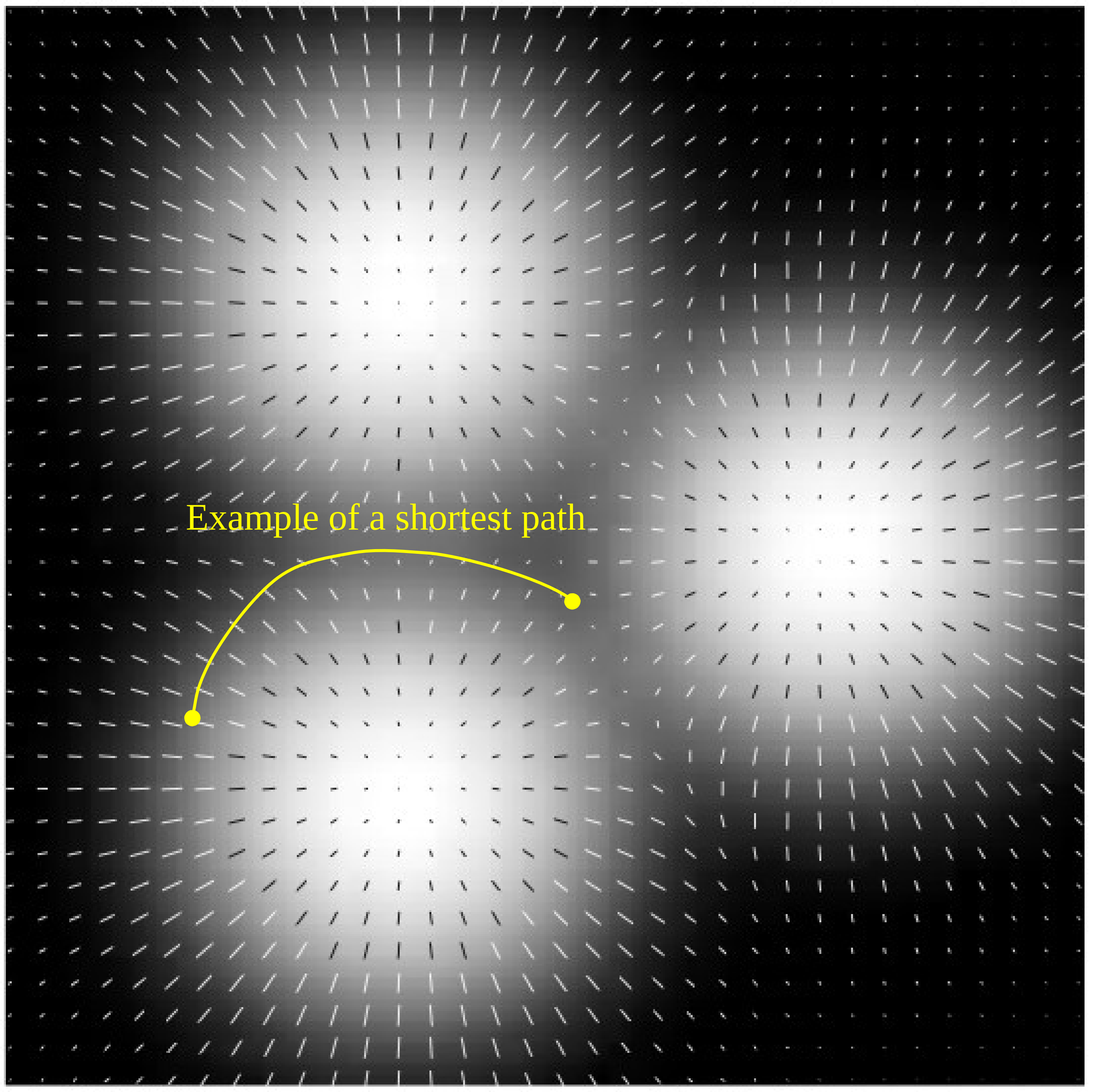


Image from
J. Peltonen's
D.Sc. thesis

Extension 4: Supervised NeRV

Simply compute the input neighborhoods based on distances in the learning metric.

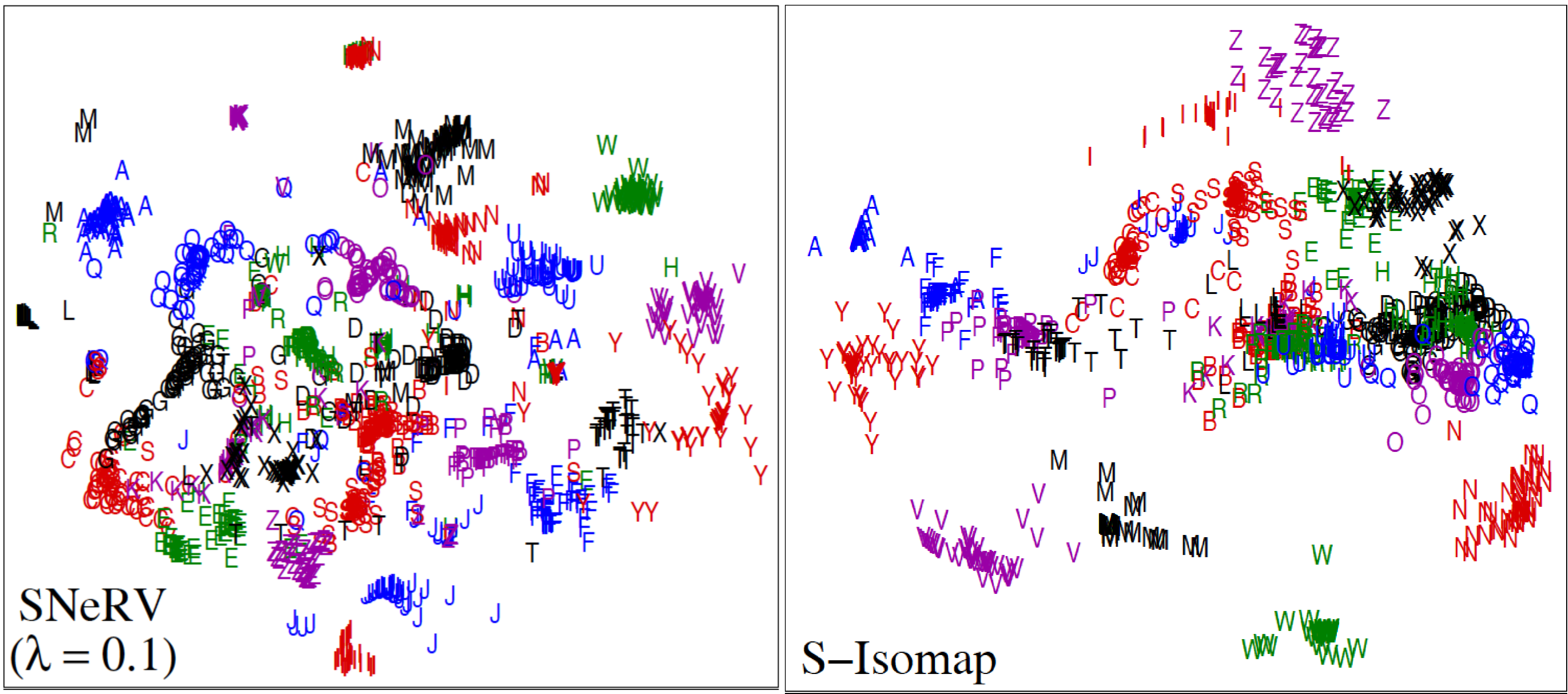
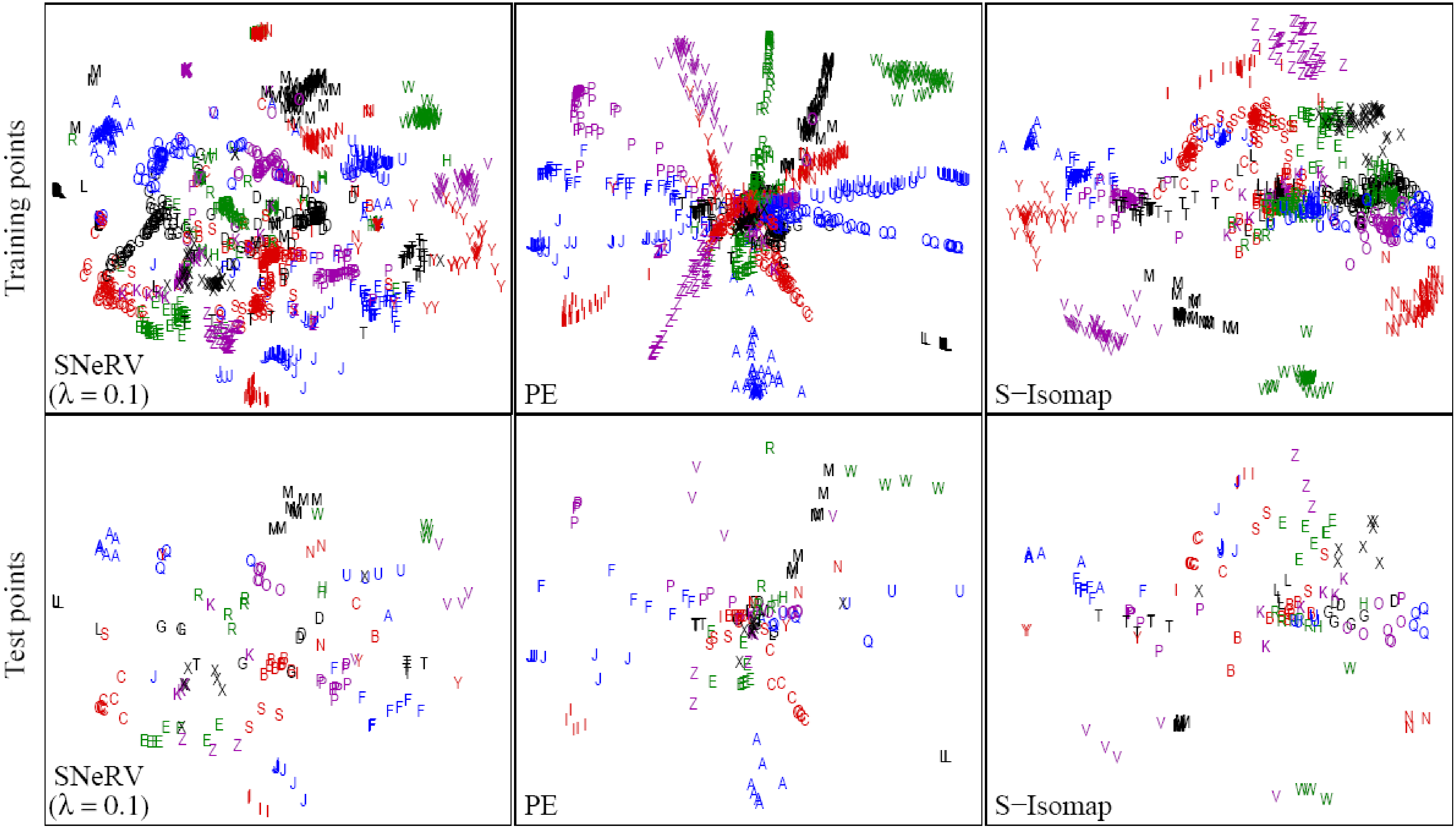
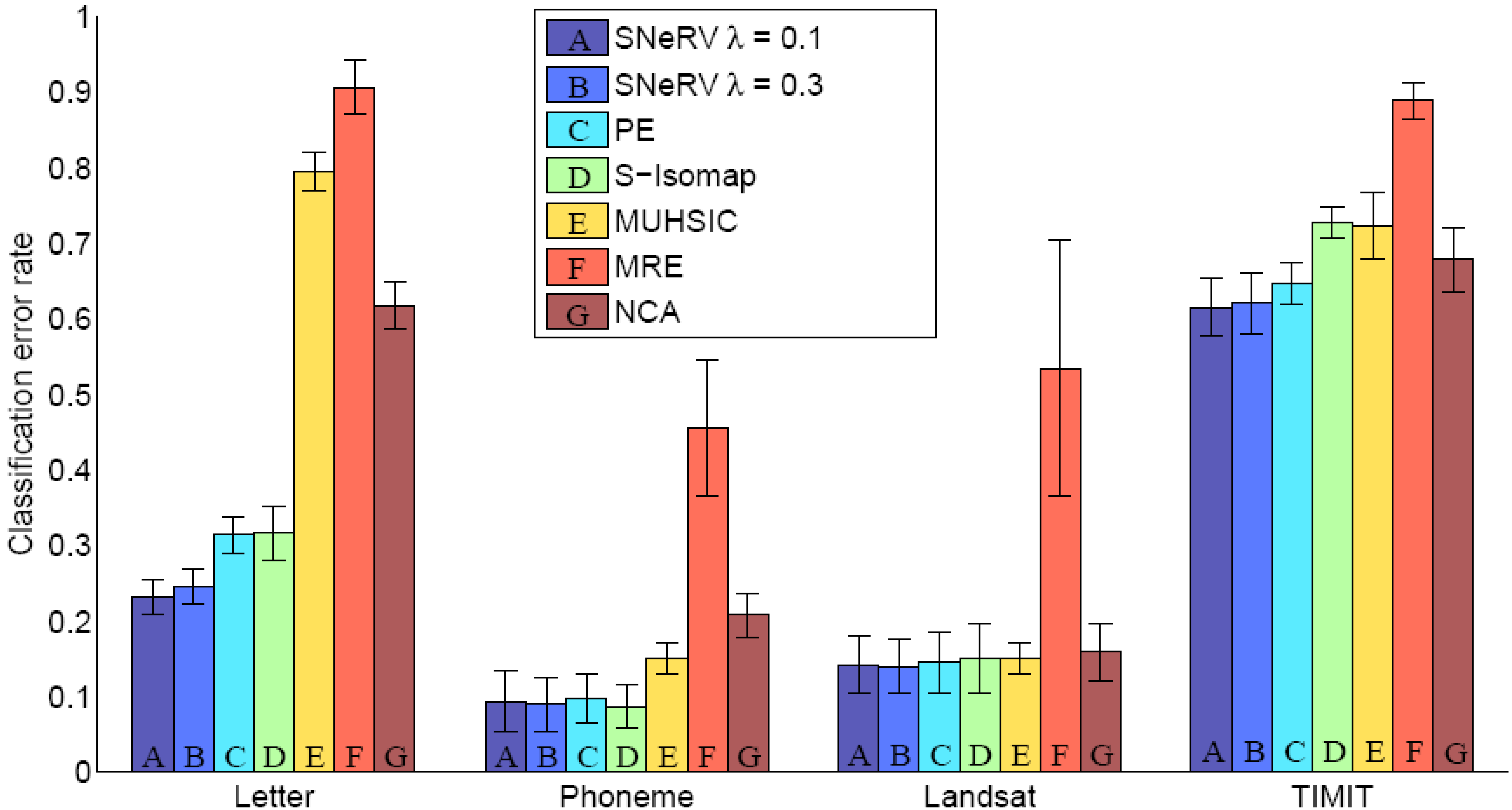


Image from ref. [4]

Extension 4: NeRV with a supervised topology-preserving metric. Visualizes labeled data better than existing methods.



Extension 4: NeRV with a supervised topology-preserving metric. Visualizes labeled data better than existing methods.



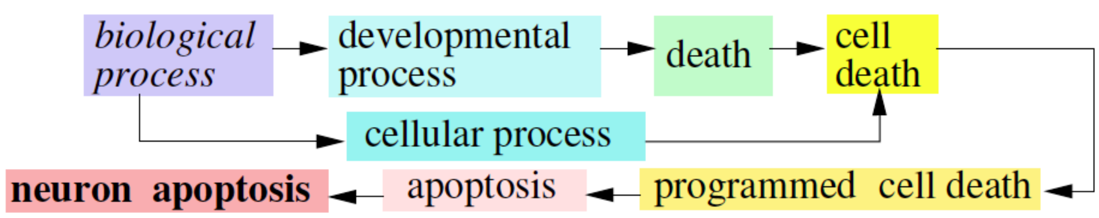
Case 1: Visualization Supervised by Ontologies

(Peltonen, Aidos, Gehlenborg, Brazma, and Kaski, ICASSP 2010)

Suppose you have measurements and annotations. Then...

Expression distance: any suitable distance between measured activity, e.g. simply euclidean distance between gene expression profiles as vectors, or any more advanced distance (e.g. time series distance messures if the profiles are over time).

Ontology distance:
Given ontology annotations of two genes, compute Jaccard distance between their **true paths** (paths from annotations to ontology root)



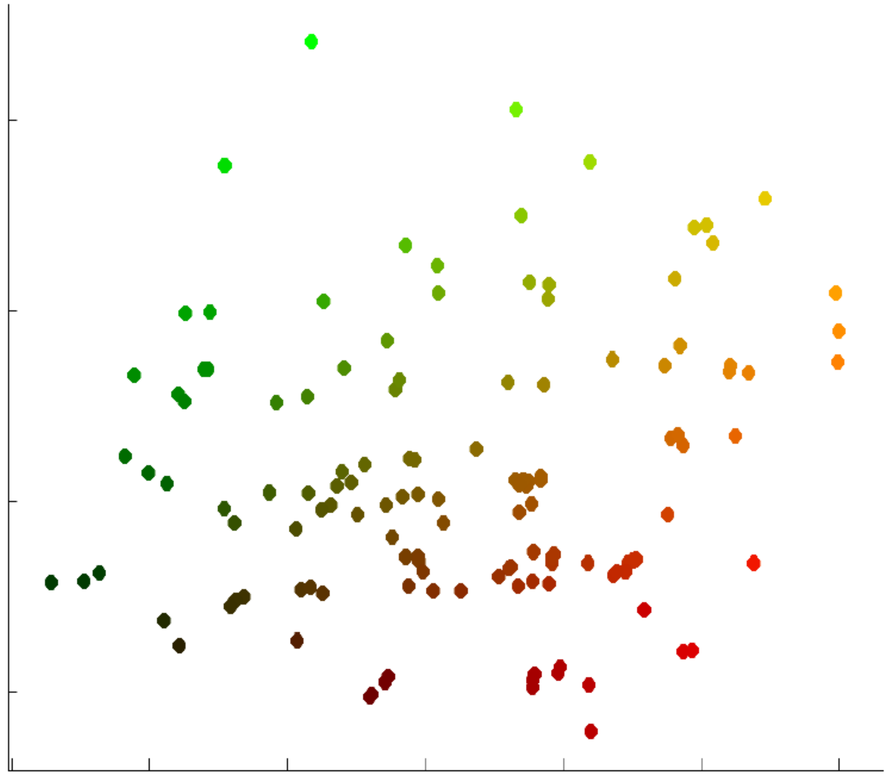
one of the 19 GO true paths for human gene AIFM1

$$J(S_i, S_j) = (|S_i \cup S_j| - |S_i \cap S_j|)/|S_i \cup S_j|$$

Case 1: Visualization Supervised by Ontologies

Example: Yeast genes significantly expressed in a study of 300 comparisons of mutant yeast strains to wild-type (normal) strain

To visualize regularities in annotation, give the Jaccard distances as input to NeRV ---> visualizes which genes are neighbors in terms of annotation.



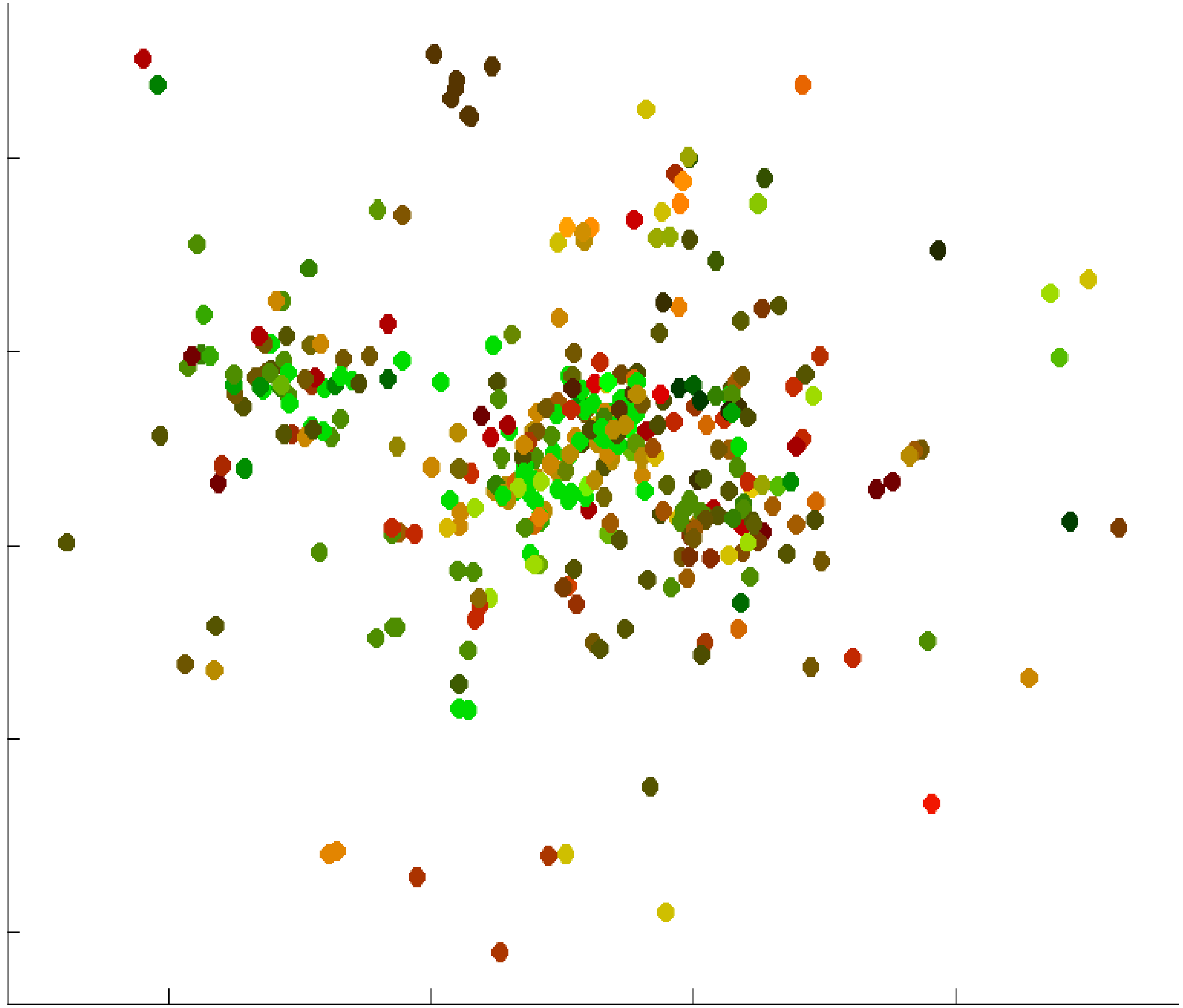
Case 1: Visualization Supervised by Ontologies

To visualize regularities in gene expression, give the distances of gene expression profiles as input to NeRV ---> visualizes which genes are neighbors in terms of gene expression.

300 comparisons of strains

---> 300 dim. gene expression profile for each gene.

Visualize similarities of expression. **Color** by ontology similarity.

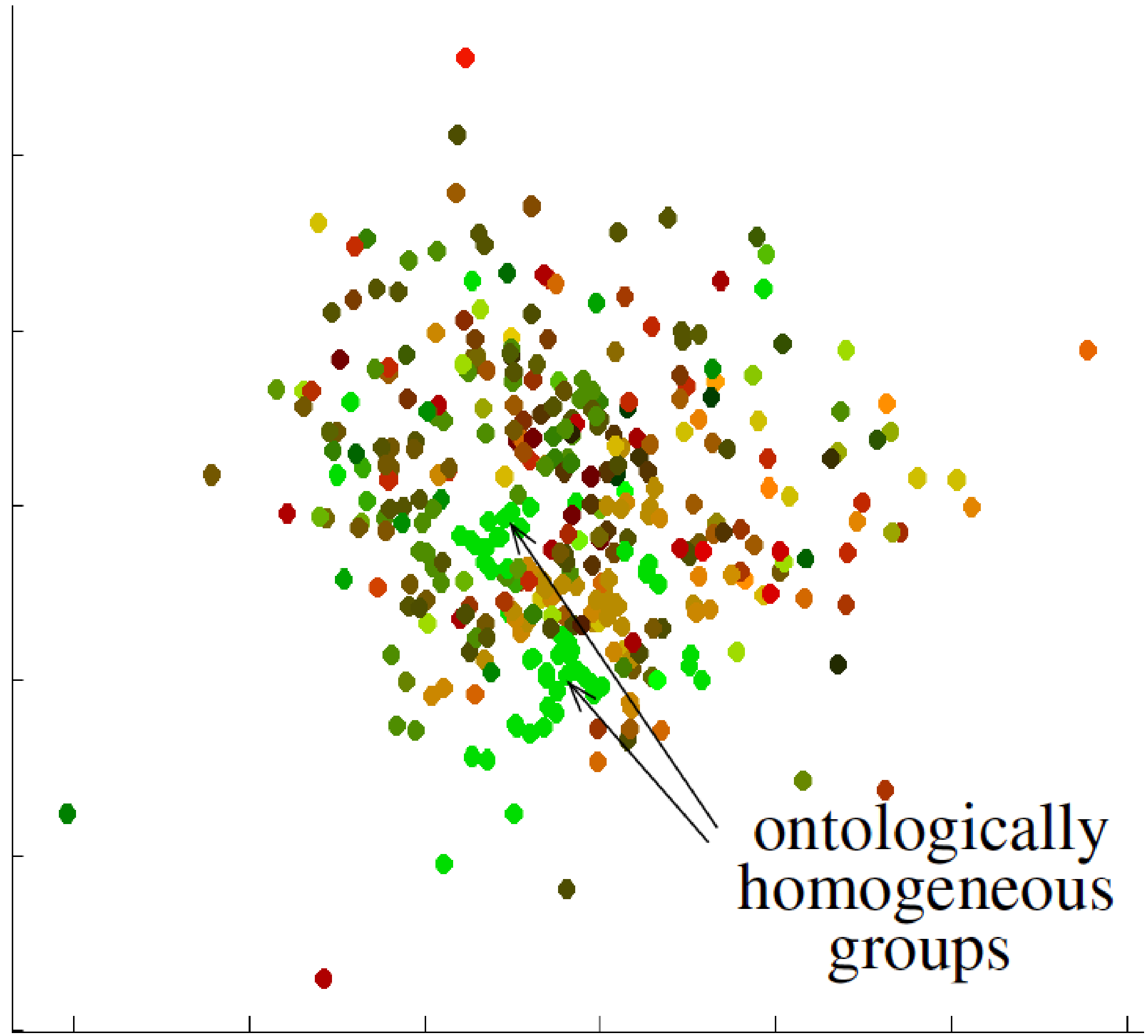


Case 1: Visualization Supervised by Ontologies

To visualize correspondences of gene expression similarity and ontology similarity, give the distances of gene expression profiles as inputs to linear NeRV, and give ontology distances as targets --->

Finds a subspace of expression profiles,

so that neighbors in the subspace best match neighbors in the ontology.

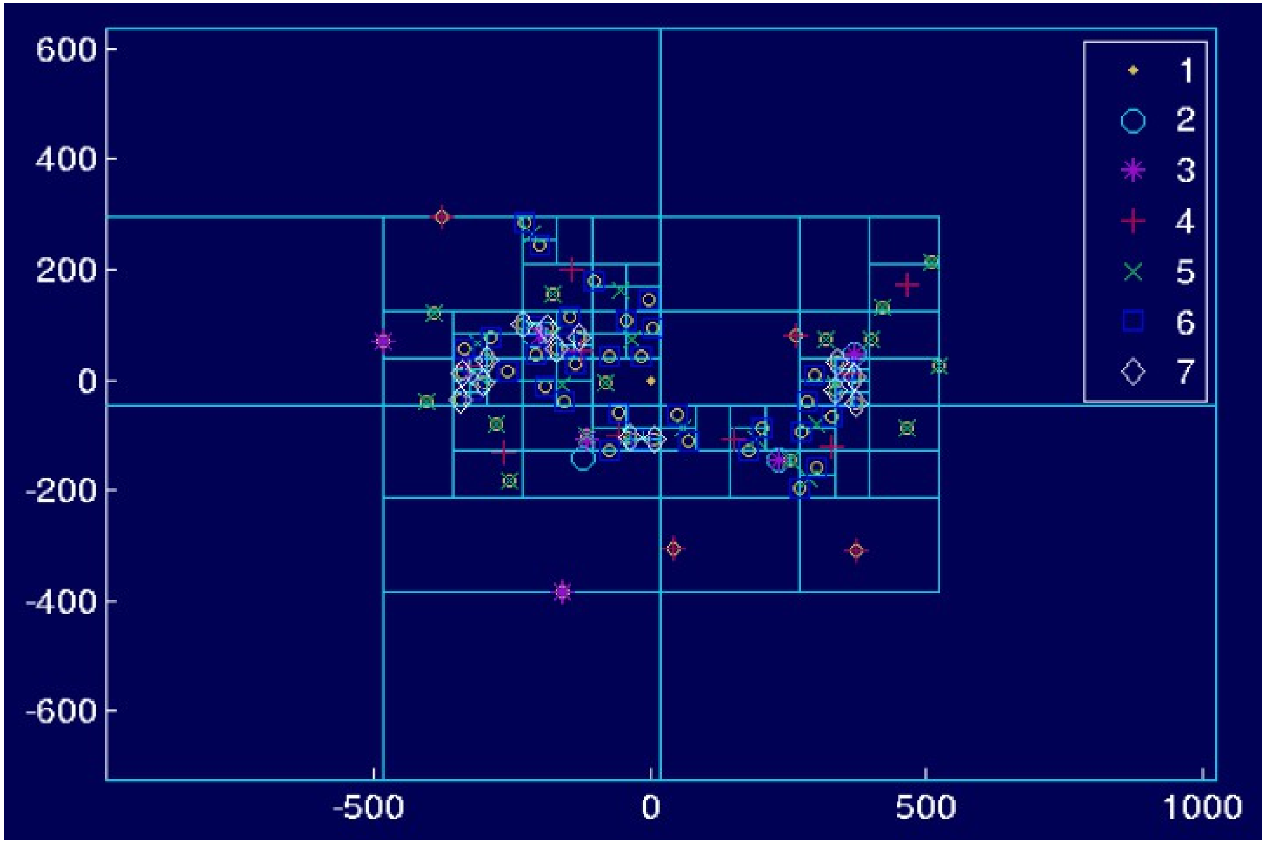


Extension 5: Fast scalable visualisation

(Yang, Peltonen and Kaski, ICML 2013)

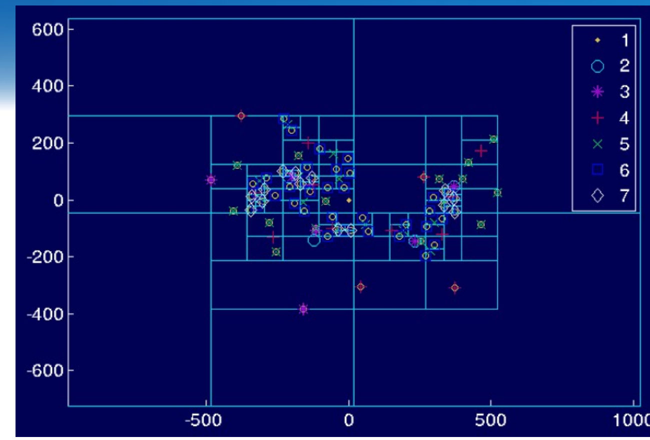
Neighbor embedding is state of the art but takes quadratic time.
New O(N logN) methods based on Barnes-Hut approximation:

1. Original neighborhoods are likely to be sparse.
2. quickly build hierarchical **quadtree**
3. approximate pairwise interactions to far-off points by **interaction with quadtree cluster centroid (the further off, the simpler the hierarchy level needed)**.



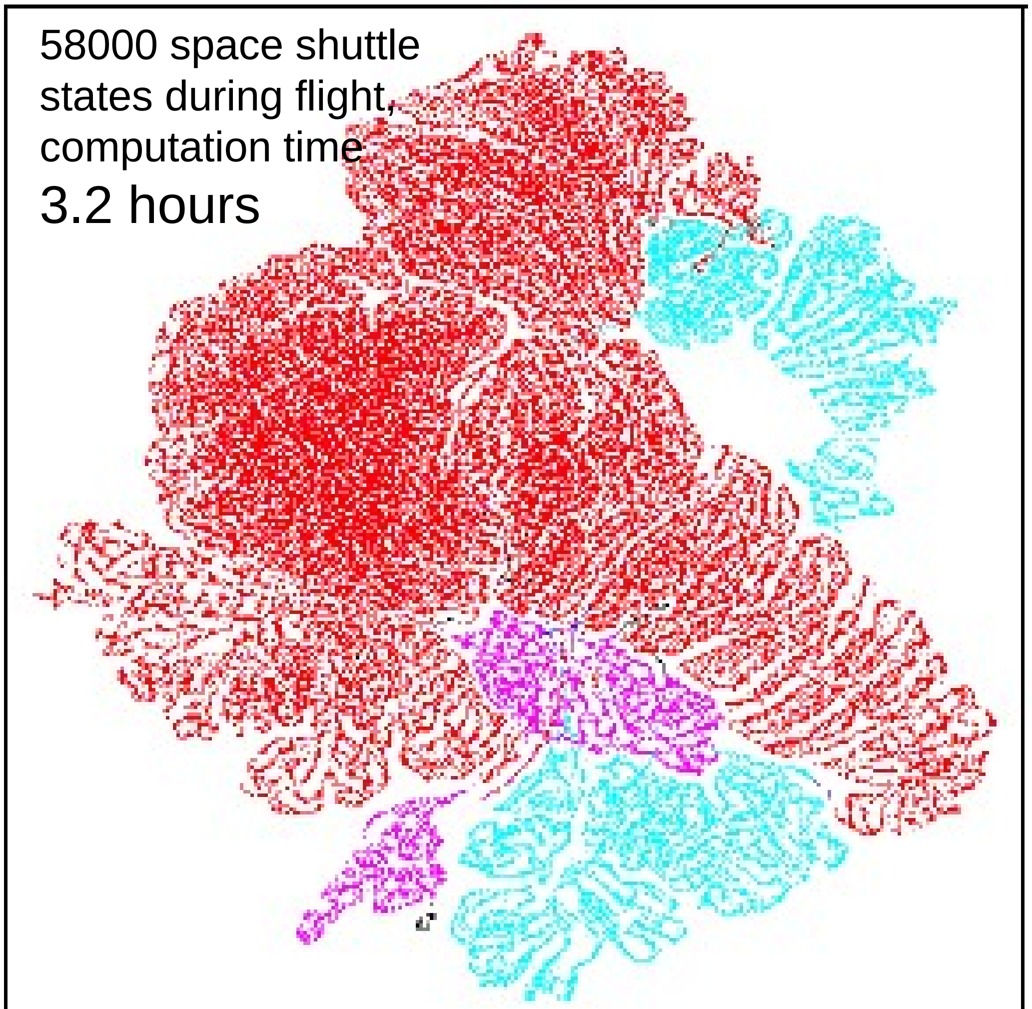
Extension 5: Fast scalable visualisation

Neighbor embedding is state of the art but takes quadratic time. New O(N logN) methods based on Barnes-Hut approximation.

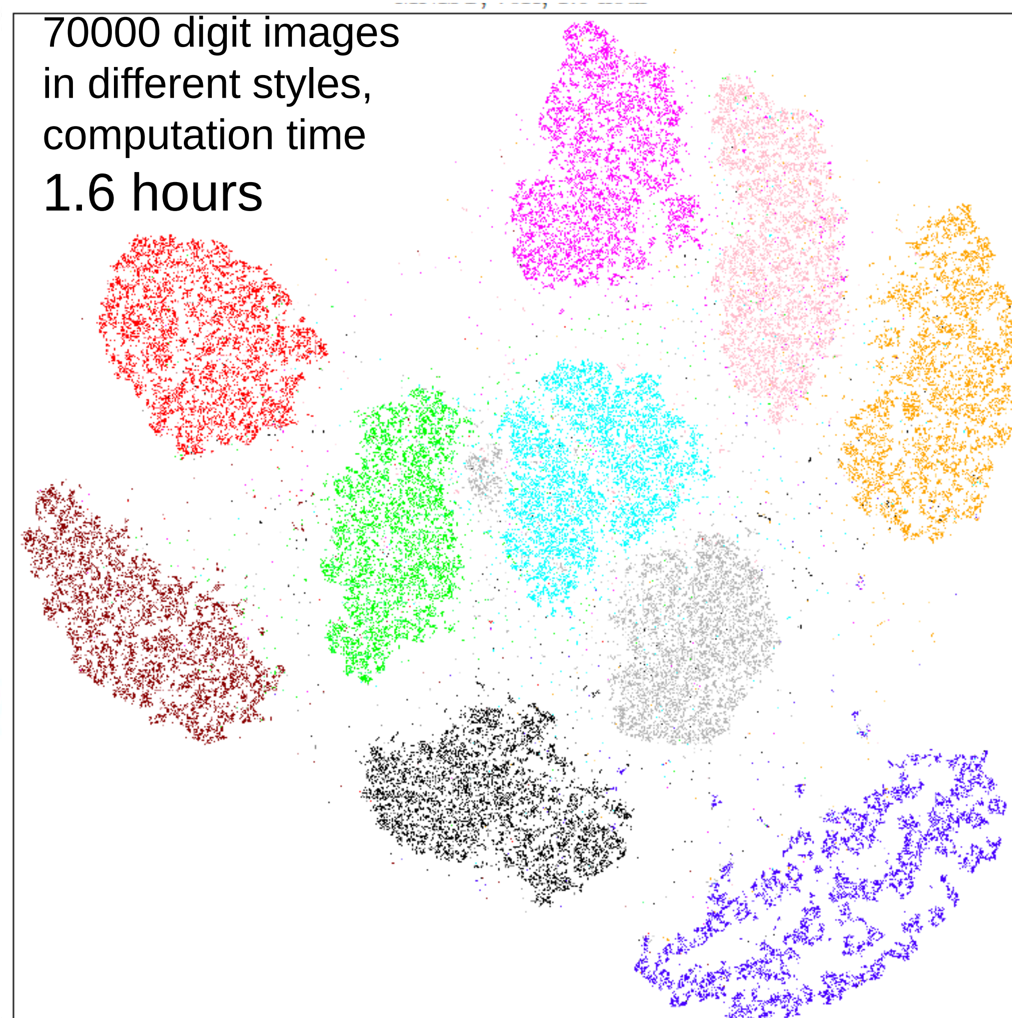


(Yang, Peltonen and Kaski, ICML 2013)

58000 space shuttle states during flight, computation time
3.2 hours



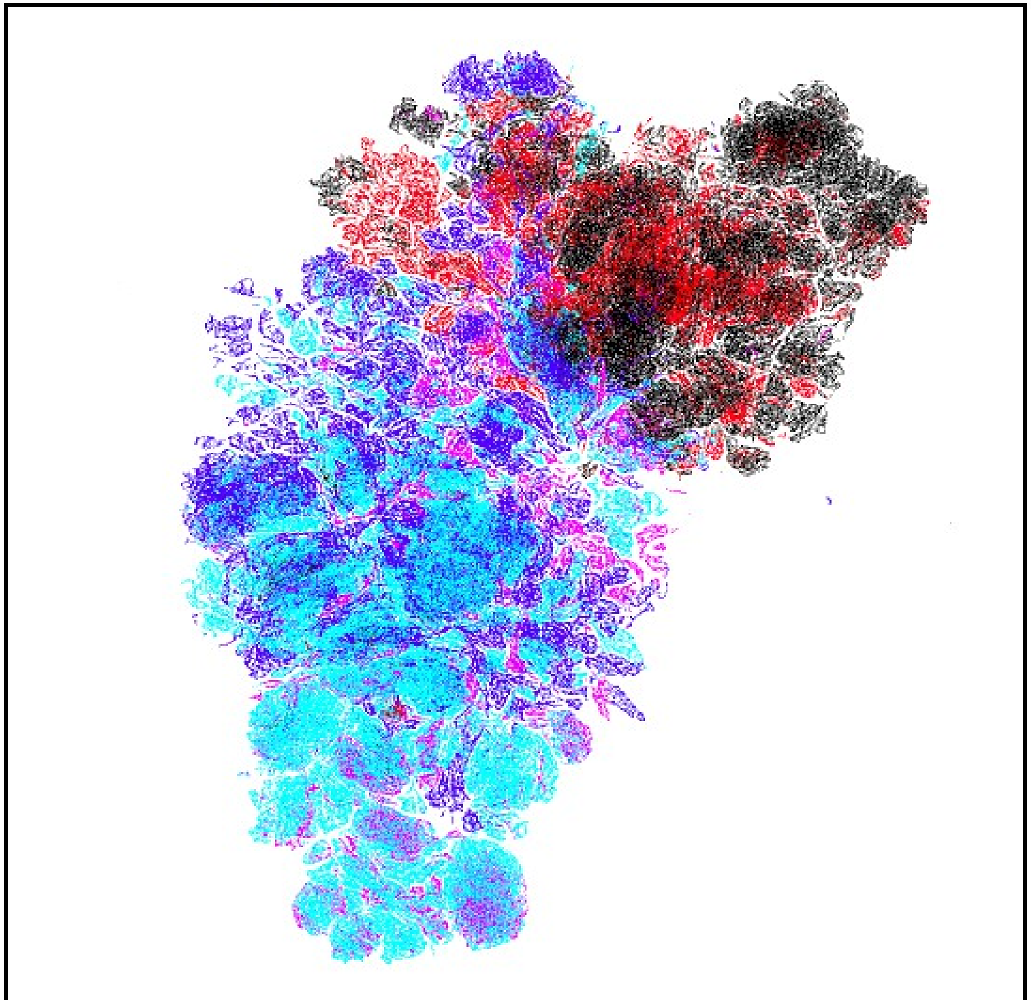
70000 digit images in different styles, computation time
1.6 hours



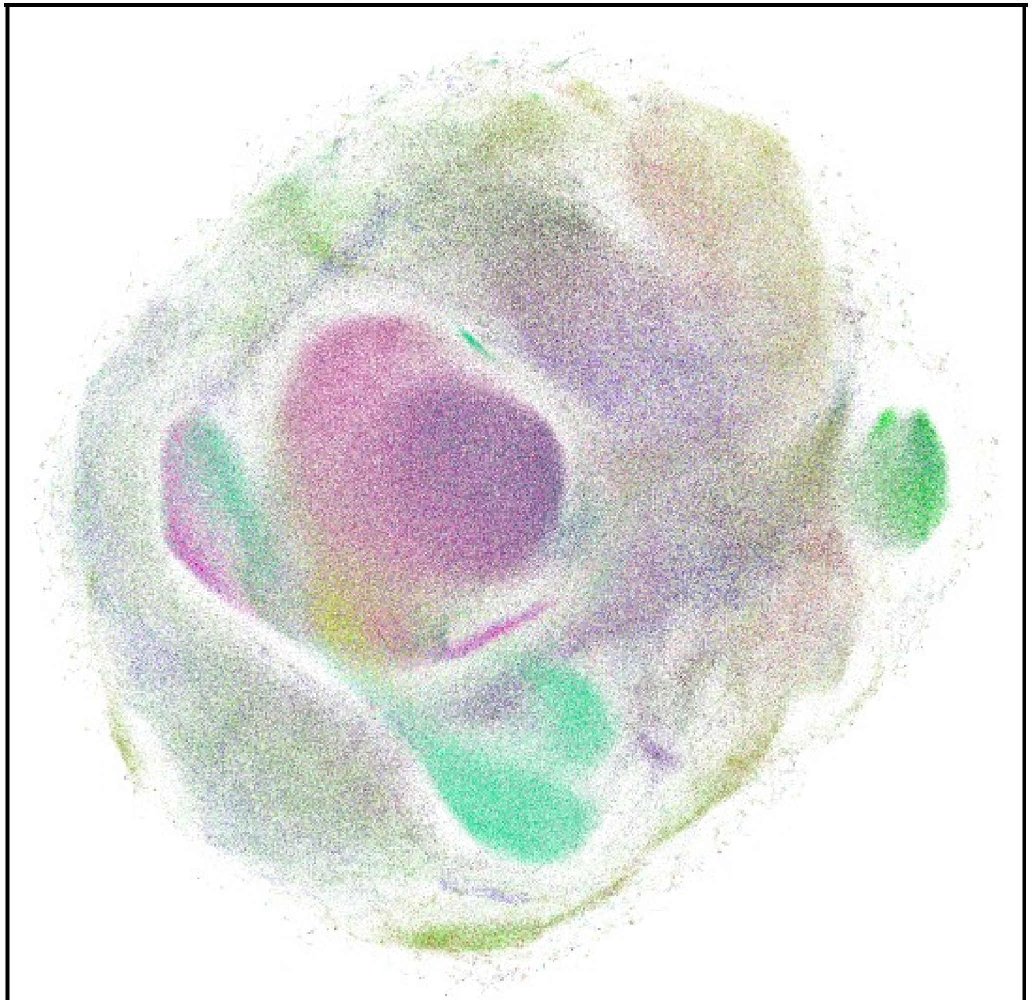
Extension 5: Fast scalable visualisation

(Yang, Peltonen and Kaski, ICML 2013)

Soil samples: 581000 samples of different soil types, visualization computed in 46 hours.
Colors = known soil types



1.3 million phoneme audio samples, visualization computed in 33 hours. Colors = known phoneme types



Extension 6: Optimal Visualization

(Bunte, Järvisalo, Berg, Myllymäki, Peltonen and Kaski, AAAI 2014)

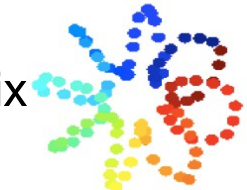
Rigorous information retrieval cost of misses&false neighbors encoded as a maximum satisfiability task on a grid.

$$(\mathrm{RN}^{xy} \vee \mathrm{CN}^{xy} \vee \mathrm{DN}^{xy})$$

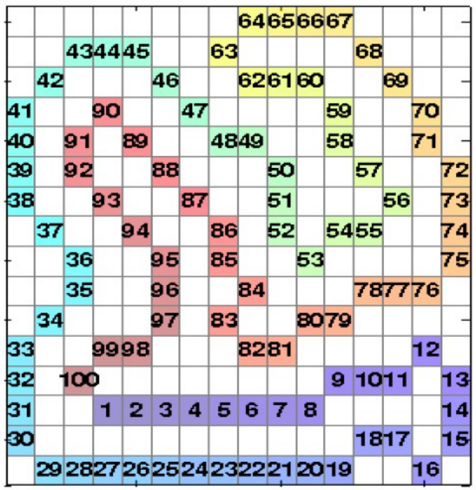
"recall: a true neighbor should be in an adjacent row, column, or diagonal"

Solvers yield globally optimal visualizations!

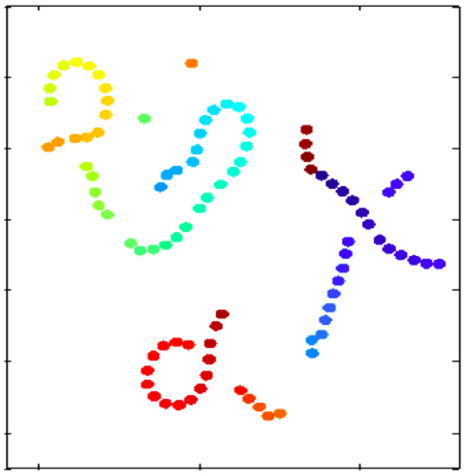
3d helix

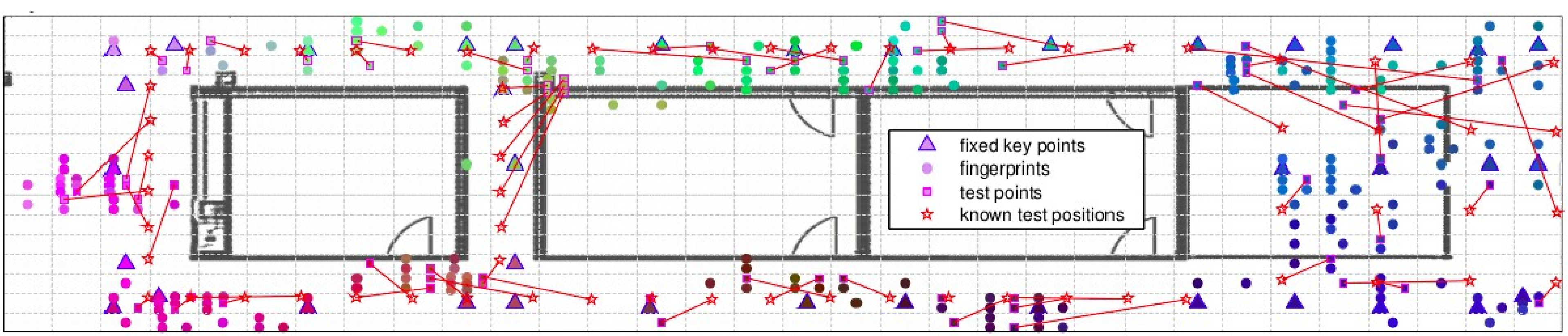


MaxSAT



t-SNE





Extension 7: Interactive Visualization

(Peltonen, Sandholm, and Kaski, Eurovis 2013)

A metric for visualizing data can be learned interactively while inspecting the visualization, in a rigorous neighbor retrieval task.

Visualization is optimized to preserve expected desired neighborhoods of the user (as estimated from feedback).

Extension 7: Interactive Visualization

(Peltonen, Sandholm, and Kaski, Eurovis 2013)

1. Show an initial visualization.

2. User points out pairs of data on the visualization, tells if they should be considered similar or dissimilar (neighbors vs nonneighbors).

3. Probabilistic model of the observed feedback: assume the desired metric is parametric, e.g. a linear Mahalanobis metric.

Data pair is likely to be labeled dissimilar if the two points are far away in the desired metric (logistic function of distance). Infer the posterior distribution of the metric parameters.

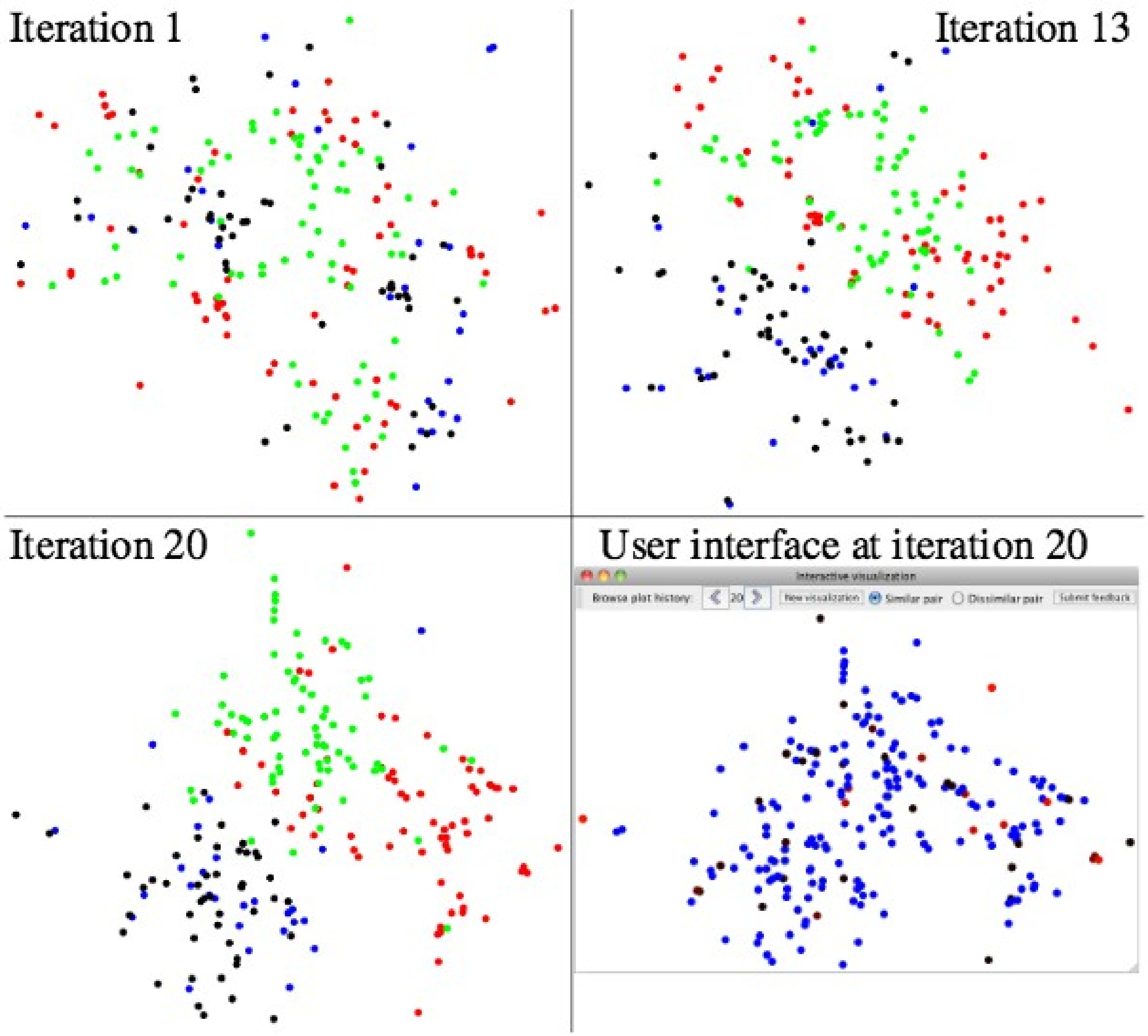
4. Compute expected distances according to the posterior, visualize the data.

Extension 7: Interactive Visualization

(Peltonen, Sandholm, and Kaski, Eurovis 2013)

Publications of Helsinki Institute for Information Technology HIIT

Data become ordganized according to underlying ground-truth.

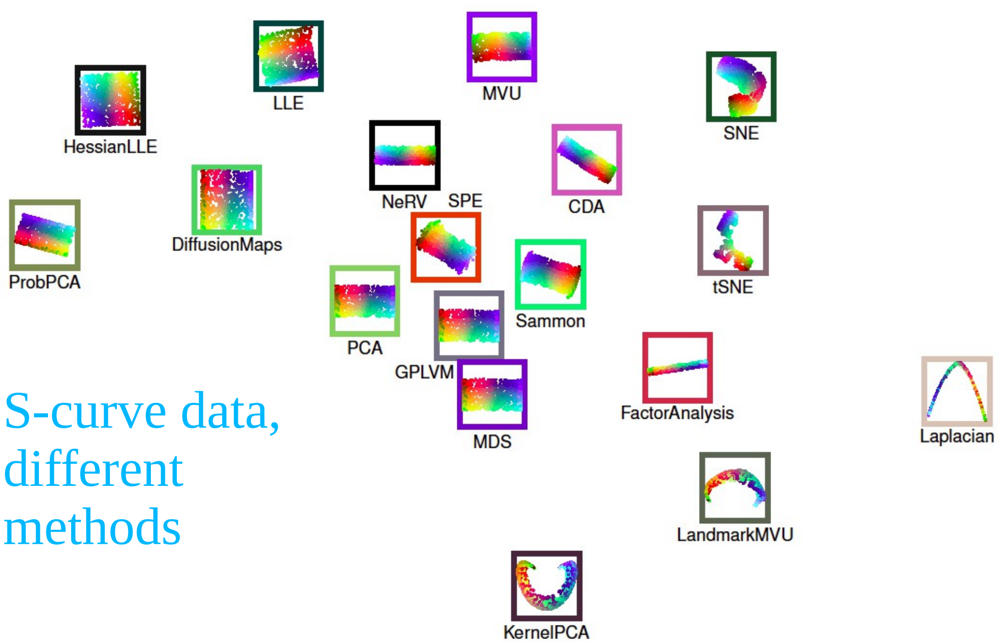


Extension 8: Meta-visualization

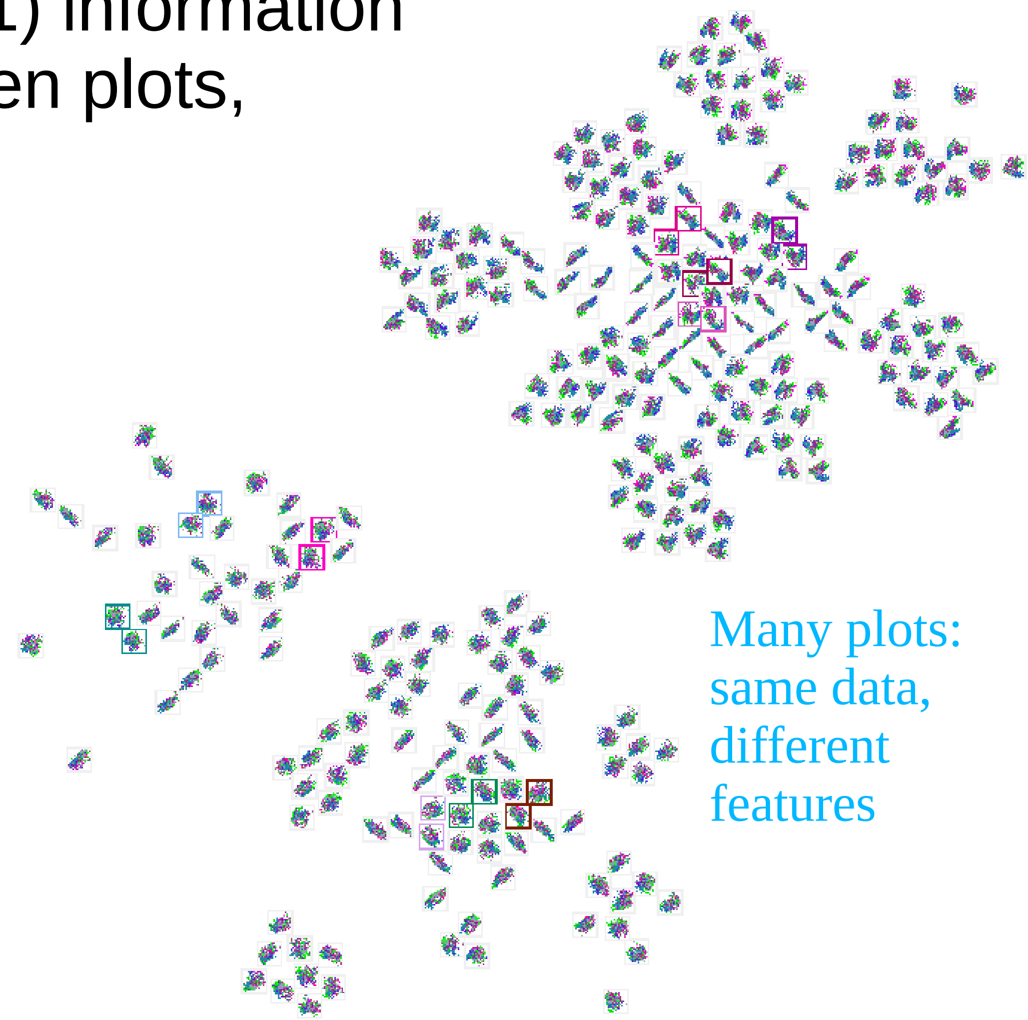
(Peltonen and Lin, Machine Learning, 2012)

No single plot suffices to analyze high-dimensional data. Many plots needed; analyzing unorganized plots is hard.

Rigorous meta-visualization: (1) information retrieval based distance between plots, (2) information retrieval based optimization of plot locations



S-curve data, different methods



Many plots: same data, different features

Extension 8: Meta-visualization

(Peltonen and Lin,
Machine Learning, 2012)

Task-based similarity of visualizations: for an analyst studying data neighborhoods, two visualizations are similar if an analyst would retrieve the same neighborhood relationships by looking at either one of them!

Neighborhood preservation for visualizations preserves “neighboring plots”: if two plots show similar data, they should be placed close-by in meta-visualization.

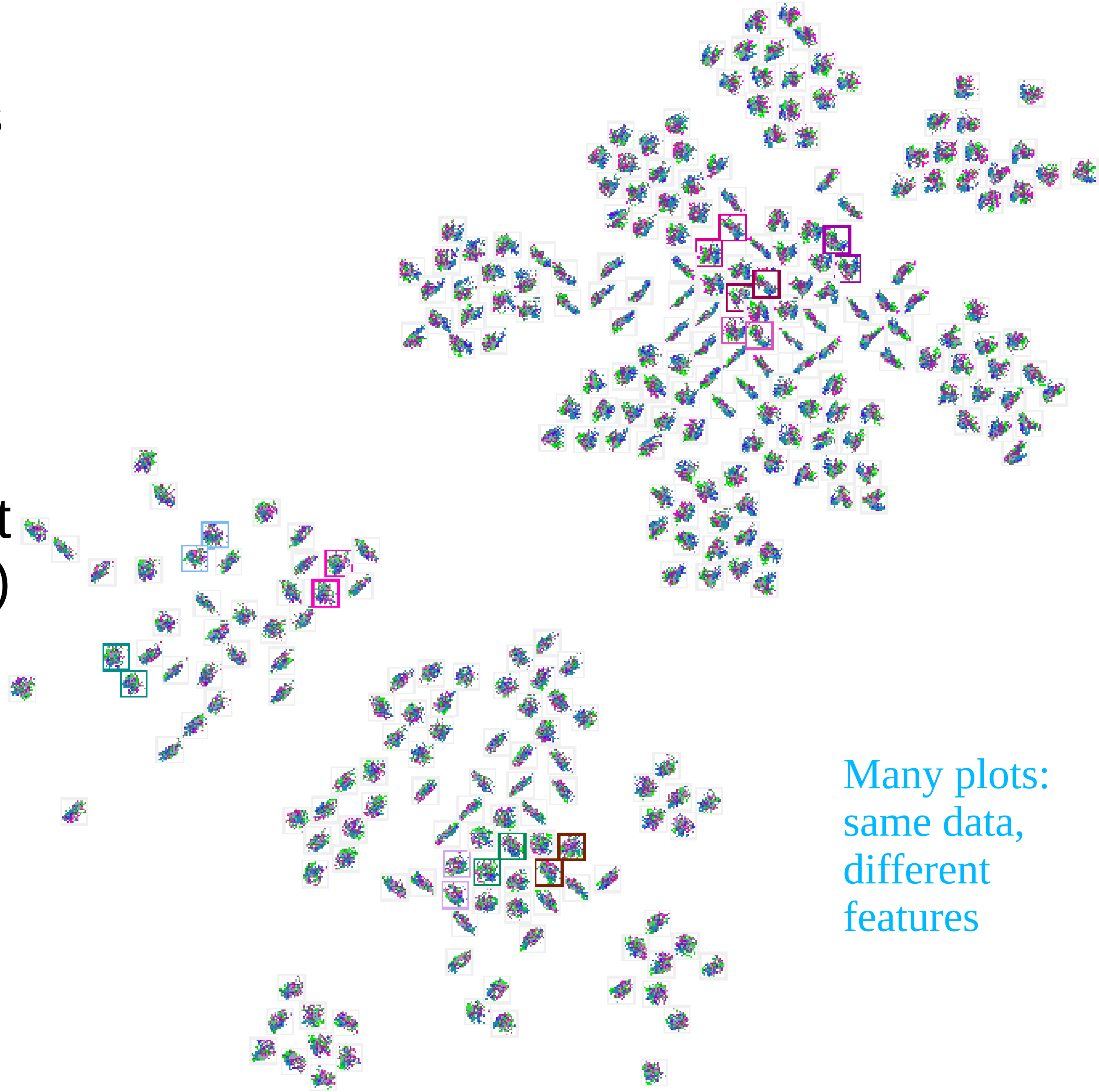
Extension 8: Meta-visualization

(Peltonen and Lin,
Machine Learning, 2012)

Visualize differences among feature pairs for the same data

(unlike a scatterplot

matrix, this is data driven: able to detect nontrivial similarities)

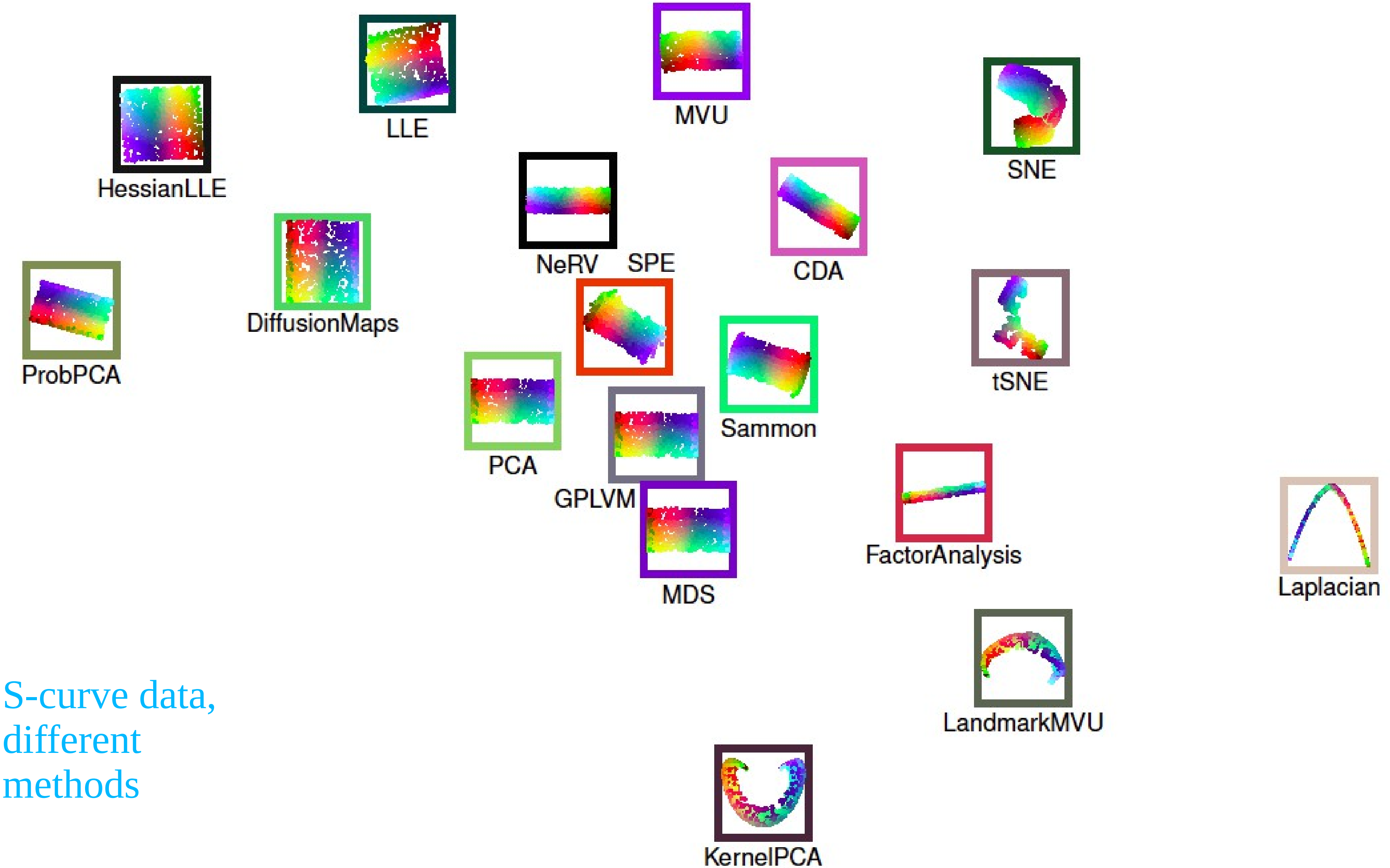


Many plots:
same data,
different
features

Extension 8: Meta-visualization

(Peltonen and Lin, Machine Learning, 2012)

Visualize differences among solutions of several embeddings for the same data



S-curve data, different methods

Summary:

- Formalized the task of visualization
- Rigorous tradeoff, modeled as information retrieval and as generative modeling
- Very well performing visualizer
- Rigorous data similarities through modeling
- Full suite of extensions: parametric (linear) visualization, supervised visualization, interactive visualization, meta-visualization...

Free software!

http://tinyurl.com/vismethods

“dredviz”, “NE”, and “satnerv” packages

References

Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. Journal of Machine Learning Research, 11:451-490, 2010.

Jaakko Peltonen. Visualization by Linear Projections as Information Retrieval. In José Príncipe and Risto Miikkulainen, editors, Advances in Self-Organizing Maps (proceedings of WSOM 2009), pages 237-245. Springer, Berlin Heidelberg, 2009.

Jaakko Peltonen and Samuel Kaski. Generative Modeling for Maximizing Precision and Recall in Information Visualization. In Geoffrey Gordon, David Dunson, and Miroslav Dudik, eds., Proceedings of AISTATS 2011, the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP, vol. 15, 2011.

Jaakko Peltonen, Helena Aidos, and Samuel Kaski. Supervised Nonlinear Dimensionality Reduction by Neighbor Retrieval. In Proceedings of the IEEE 2009 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), pages 1809-1812. IEEE, 2009.

Jaakko Peltonen, Helena Aidos, Nils Gehlenborg, Alvis Brazma, and Samuel Kaski. An information retrieval perspective on visualization of gene expression data with ontological annotation. In Proceedings of ICASSP 2010, pages 2178-2181. IEEE, 2010.

Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Scalable Optimization of Neighbor Embedding for Visualization. In Proceedings of ICML 2013, International Conference on Machine Learning, JMLR W&CP 28(2):127-135, 2013.

See also: Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Optimization Equivalence of Divergences Improves Neighbor Embedding. In Proceedings of ICML 2014, The 31st International Conference on Machine Learning, 2014.

See also: Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Majorization-Minimization for Manifold Embedding. In Proceedings of AISTATS'15, The 18th International Conference on Artificial Intellgence and Statistics, JMLR W&CP, pp. 1088-1097, 2015.

Kerstin Bunte, Matti Järvisalo, Jeremias Berg, Petri Myllymäki, Jaakko Peltonen and Samuel Kaski. Optimal Neighborhood Preserving Visualization by Maximum Satisfiability. In Proceedings of AAAI-14, The Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

Jaakko Peltonen, Max Sandholm, and Samuel Kaski. Information Retrieval Perspective to Interactive Data Visualization. In Proceedings of Eurovis 2013, the Eurographics Conference on Visualization - short papers, pages 49-53. European Association for Computer Graphics, 2013.

Jaakko Peltonen and Ziyuan Lin. Information Retrieval Approach to Meta-visualization. Machine Learning, 99(2):189-229, 2015.