MTTTS17
Dimensionality Reduction and Visualization

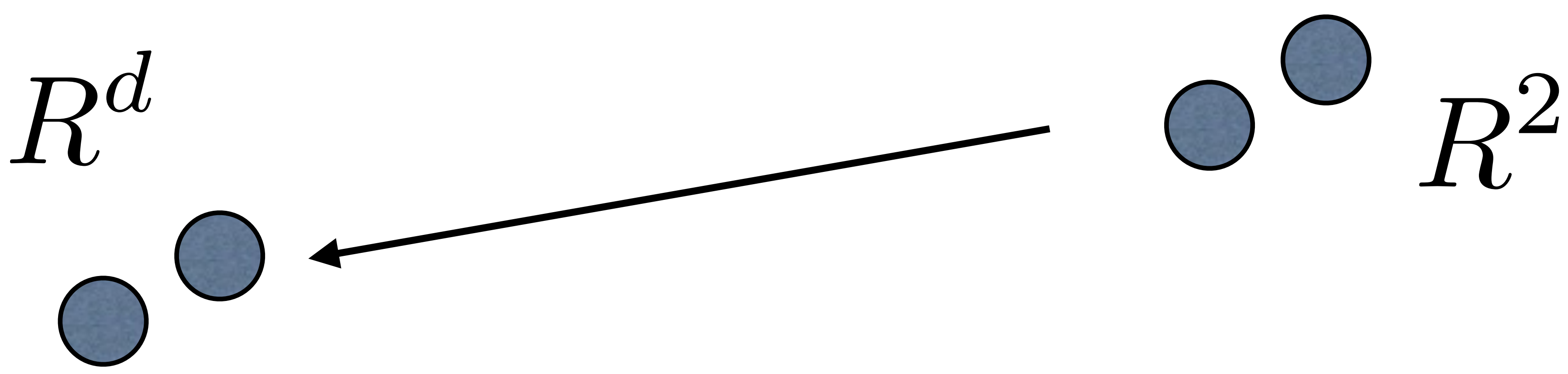
Spring 2020
Jaakko Peltonen

Lecture 7: Nonlinear dimensionality reduction, part 2

Two measures of faithfulness - precision and recall

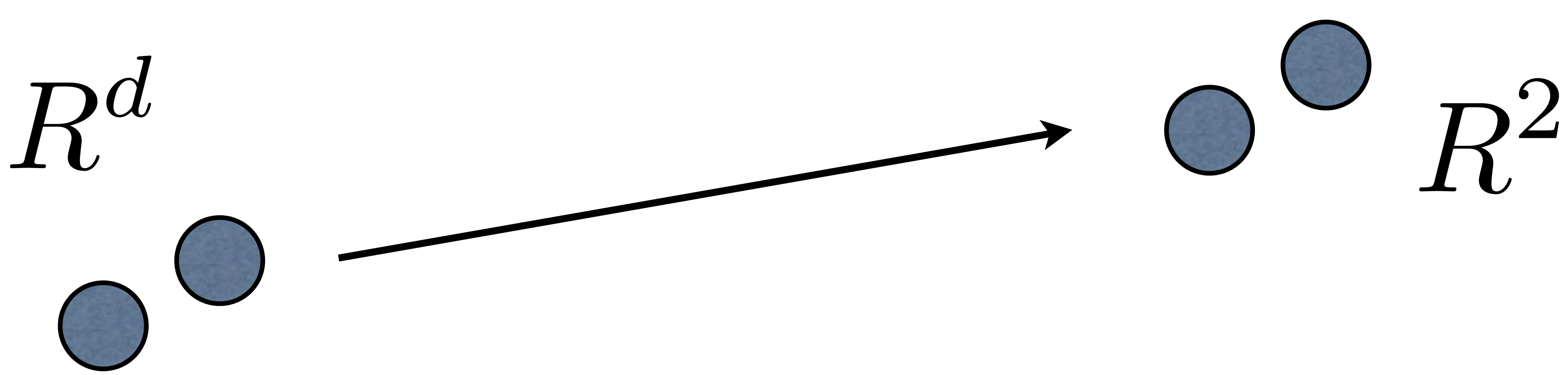
Faithfully?

- Good Precision: Points that are close in the "reduced" space are close in the original space
- Good Recall: Points that are close in the original space are close in the "reduced" space



Faithfully?

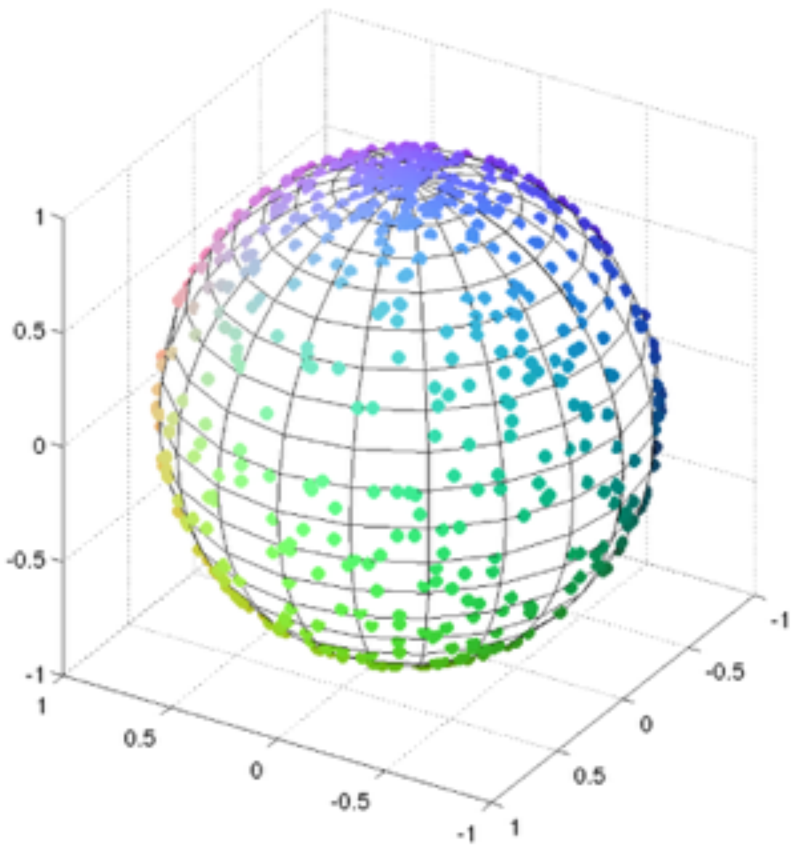
- Good Precision: Points that are close in the "reduced" space are close in the original space
- Good Recall: Points that are close in the original space are close in the "reduced" space



Faithfully?

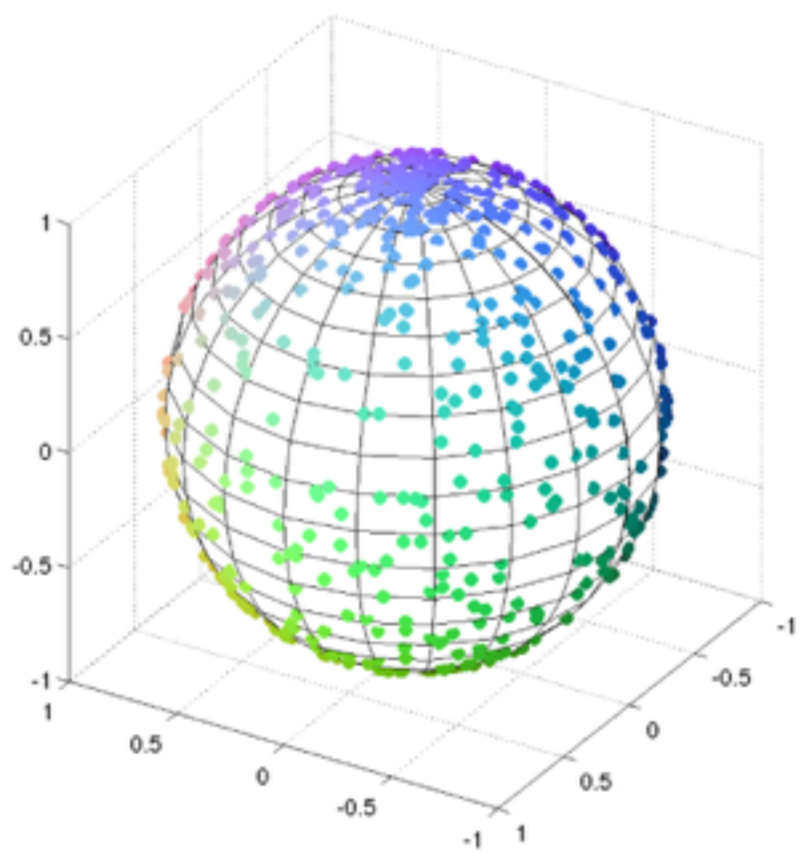
- Good Precision: Points that are close in the "reduced" space are close in the original space
- Good Recall: Points that are close in the original space are close in the "reduced" space
- In general, impossible to get both

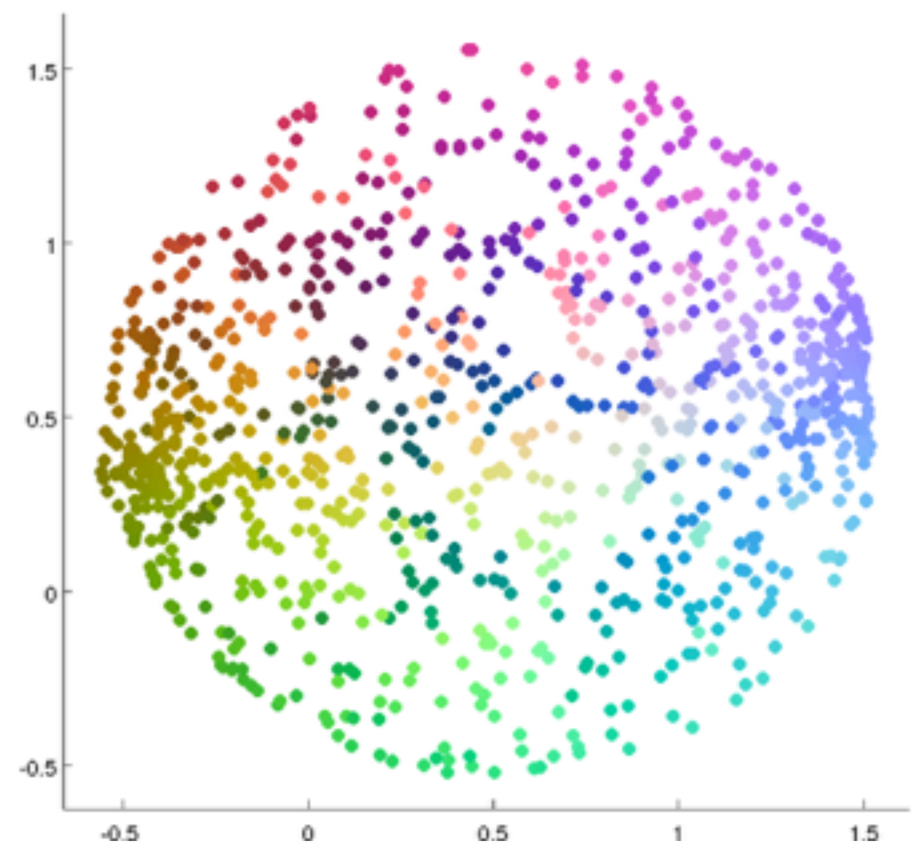
May be impossible to get both good recall and good precision



from Venna, Peltonen, Nybo, Aidos, and Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization", Journal of Machine Learning Research, 2010

May be impossible to get both good recall and good precision

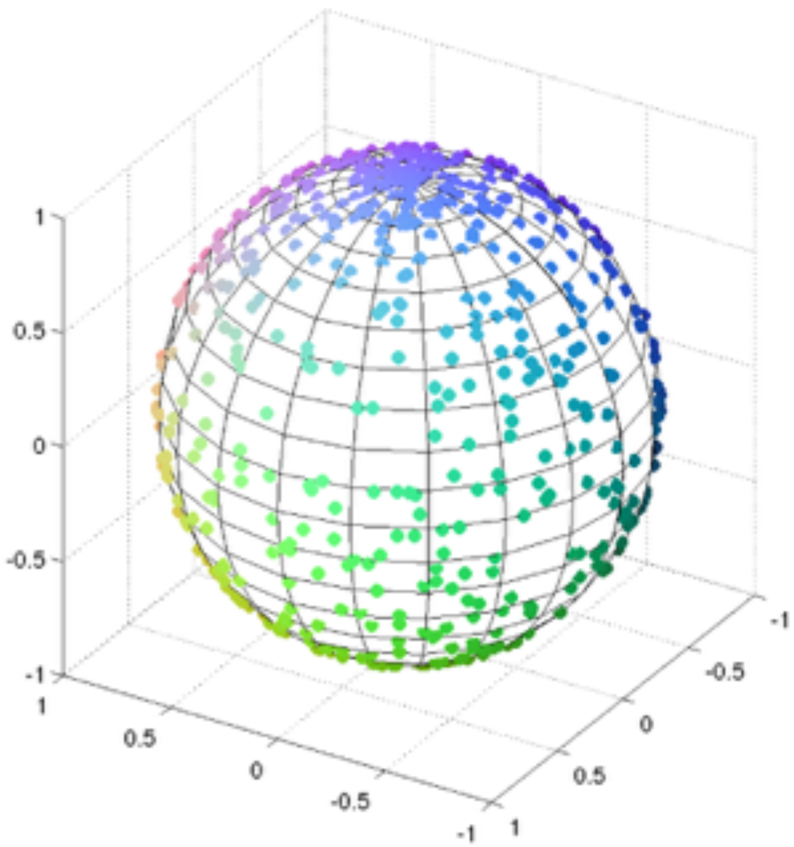


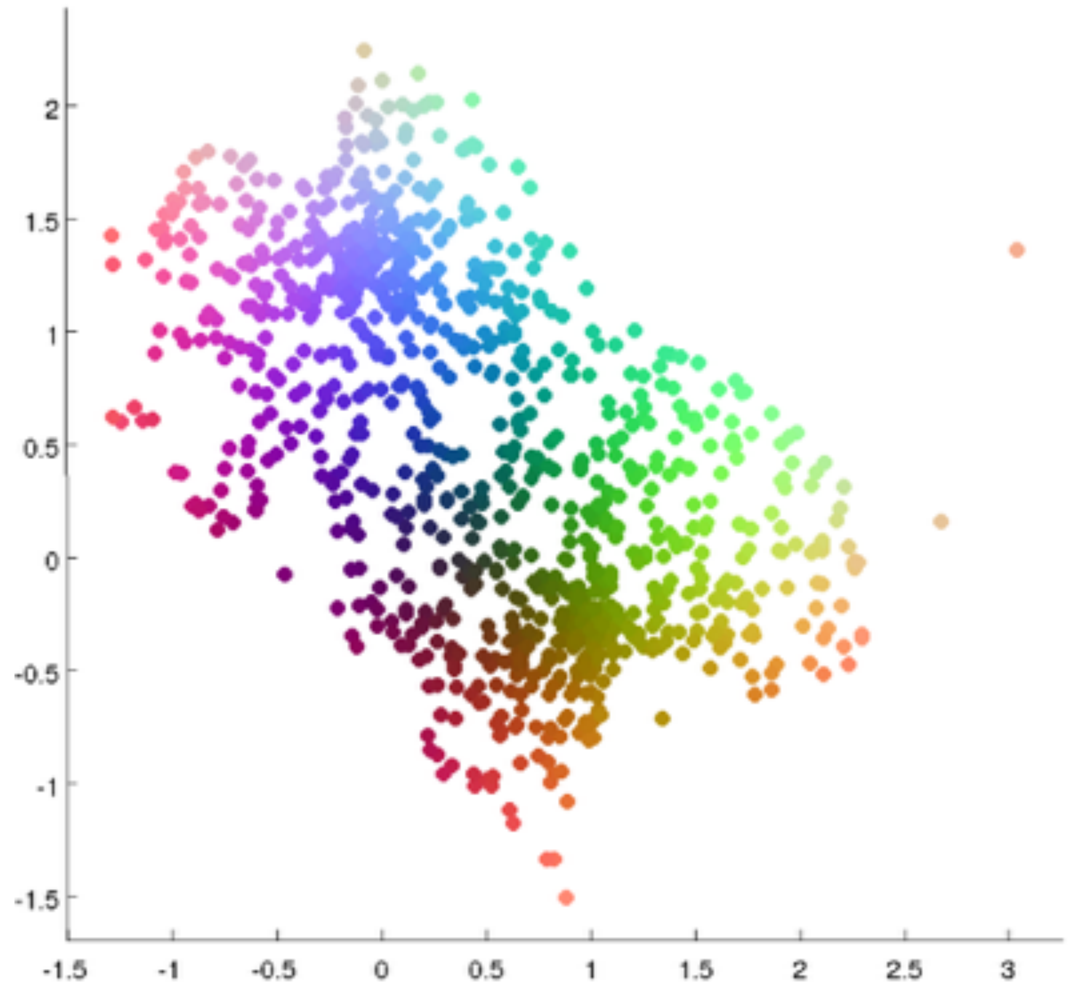


“Squashed flat sphere surface”
yields good recall but bad precision

from Venna, Peltonen, Nybo, Aidos, and Kaski, “Information retrieval perspective to nonlinear dimensionality reduction for data visualization”, Journal of Machine Learning Research, 2010

May be impossible to get both good recall and good precision





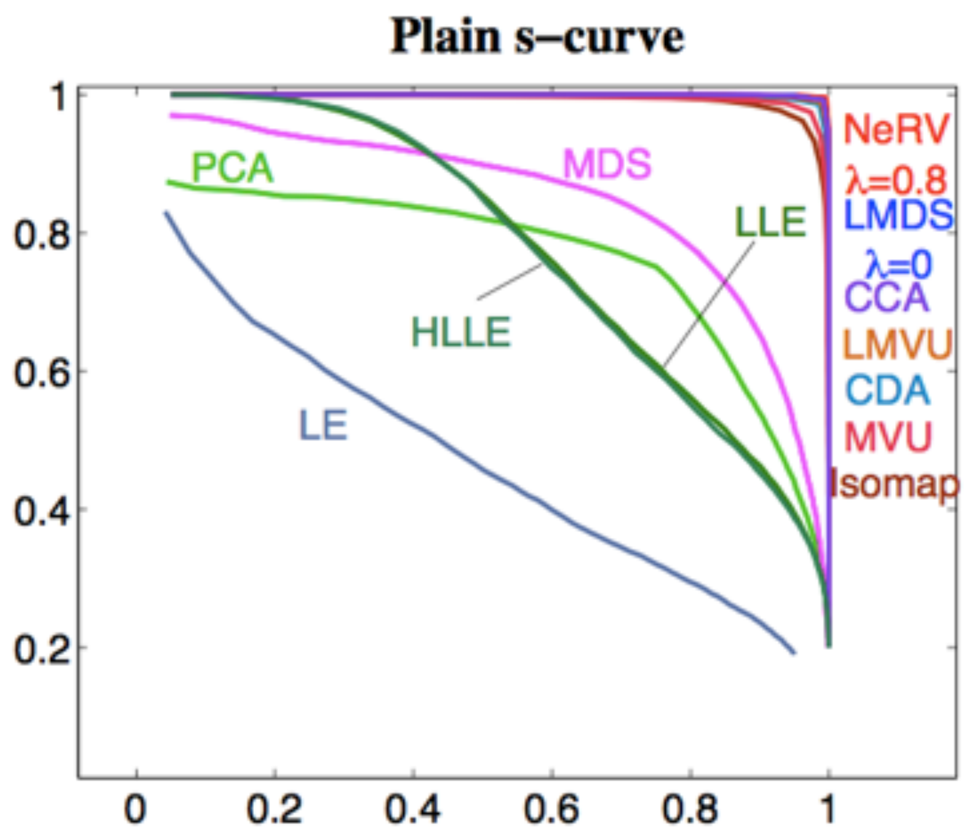
"Cut-open sphere surface"
yields good precision but less good recall

from Venna, Peltonen, Nybo, Aidos, and Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization", Journal of Machine Learning Research, 2010

Performance of MDS

- MDS is tries to preserve the large distances at the expense of small ones, hence, it can "collapse" some small distances on the expense of preserving large distances
- A projection is *trustworthy (precision)* if *k* closest neighbors of a sample on the projection are also close by in the original space. A projection *preserves the original neighborhoods* (*recall*) if all *k* closest neighbors of a sample in the original space are also close by in the projection.

Precision



Recall

Precision-recall curve, as a function of the neighborhood size *k* for an s-curve data set. k not shown - each k yields some (precision, recall) value pair onto the curve. 20 closest original-space neighbours are considered "true matches". Curves close to top right are better.

Figures are from Venna et al., Information retrieval perspective to nonlinear dimensionality reduction for data visualization, JMLR 2010.

More methods:
Self-Organizing Map,
Laplacian Eigenmap,
Curvilinear Component Analysis

Self-Organizing Maps

Teuvo Kohonen

Dr. Eng., Emeritus Professor of the Academy of Finland; Academician

His research areas are the theory of self-organization, associative memories, neural networks, and pattern recognition, in which he has published over 300 research papers and four monography books. His fifth book is on digital computers. His more recent work is expounded in the ***third, extended edition (2001)*** of his book **Self-Organizing Maps**.

Self-Organizing Maps

Google scholar | kohonen | Search | Advanced Scholar Search

Scholar | Articles and patents | anytime | include citations | Create email alert

Redo the above query as: Quoted author name ○ Word matching ◉

Simple Interface. Go to the Advanced interface from here.

Impact indices:
(Plain values)
Citations selected: 42688 **h-index:** 69 **g-index:** >100 **e-index:** 190 **delta-h:** 1 **delta-g:** -. Data in this page might be insufficient for computing g-index.
(Normalized per co-authorship)
Citations selected: 33606.2 **h-index:** 55.0 **g-index:** >100 **e-index:** 170.0 **delta-h:** 1.5 **delta-g:** -. Data in this page might be insufficient for computing g-index.

Insufficient data in this page. Try to ask Scholar for 100 results by clicking here.

[PDF] Springer Series in Information Sciences — [PDF] from preterhuman.net
T **Kohonen**, SO Maps - New York, New York, 2001 - preterhuman.net
Professor B. Roy Frieden, Ph.D. Optical Sciences Center, The University of Arizona Tucson, AZ 85721, USA } Series Editors: Professor Thomas S. Huang Department of Electrical Engineering and Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, USA ...
Cited by 8898 - Related articles - Library Search - All 20 versions

The self-organizing map — [PDF] from psu.edu
T **Kohonen** - Proceedings of the IEEE, 2002 - ieeexplore.ieee.org
Among the architectures and algorithms suggested for artificial neural networks, the Self-Organizing Map has the special property of effectively creating spatially organized "internal representa-tions" of various features of input signals and their abstractions. One novel result is that ...
Cited by 13537 - Related articles - Library Search - BL Direct - All 62 versions

Self-organized formation of topologically correct feature maps
T **Kohonen** - Biological cybernetics, 1982 - Springer
Abstract. This work contains a theoretical study and computer simulations of a new self-organizing process. The principal discovery is that in a simple network of adaptive physical elements which receives signals from a primary event space, the signal representations are automatically ...
Cited by 4356 - Related articles - All 6 versions

today: about 93900 Google Scholar results for "Kohonen".

Self-Organizing Maps

Google

self-organizing map

Search

About 4,460,000 results

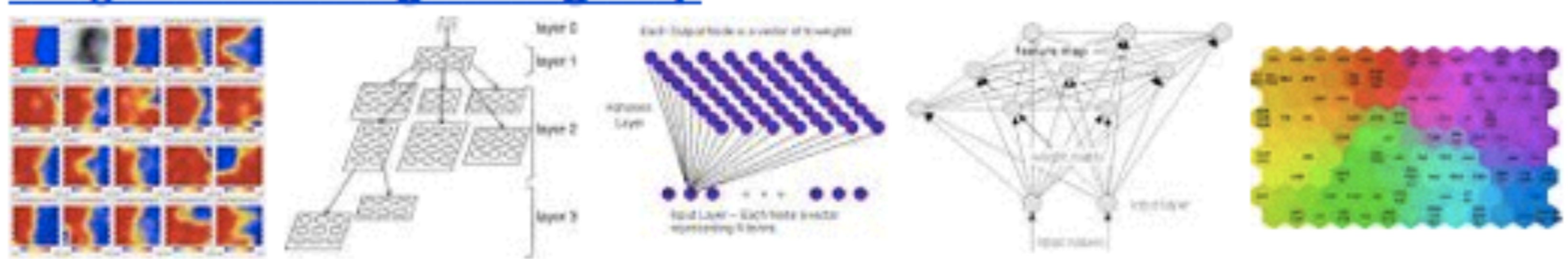
Web
Images
Maps
Books
More

Any time
Past hour
Past 24 hours
Past week
Past month
Past year

All results
Verbatim

[**Self-organizing map** - Wikipedia, the free encyclopedia]
en.wikipedia.org/wiki/**Self-organizing_map** - Cached - Similar
A **self-organizing map** (**SOM**) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to ...
Learning algorithm - Interpretation - Alternatives - See also

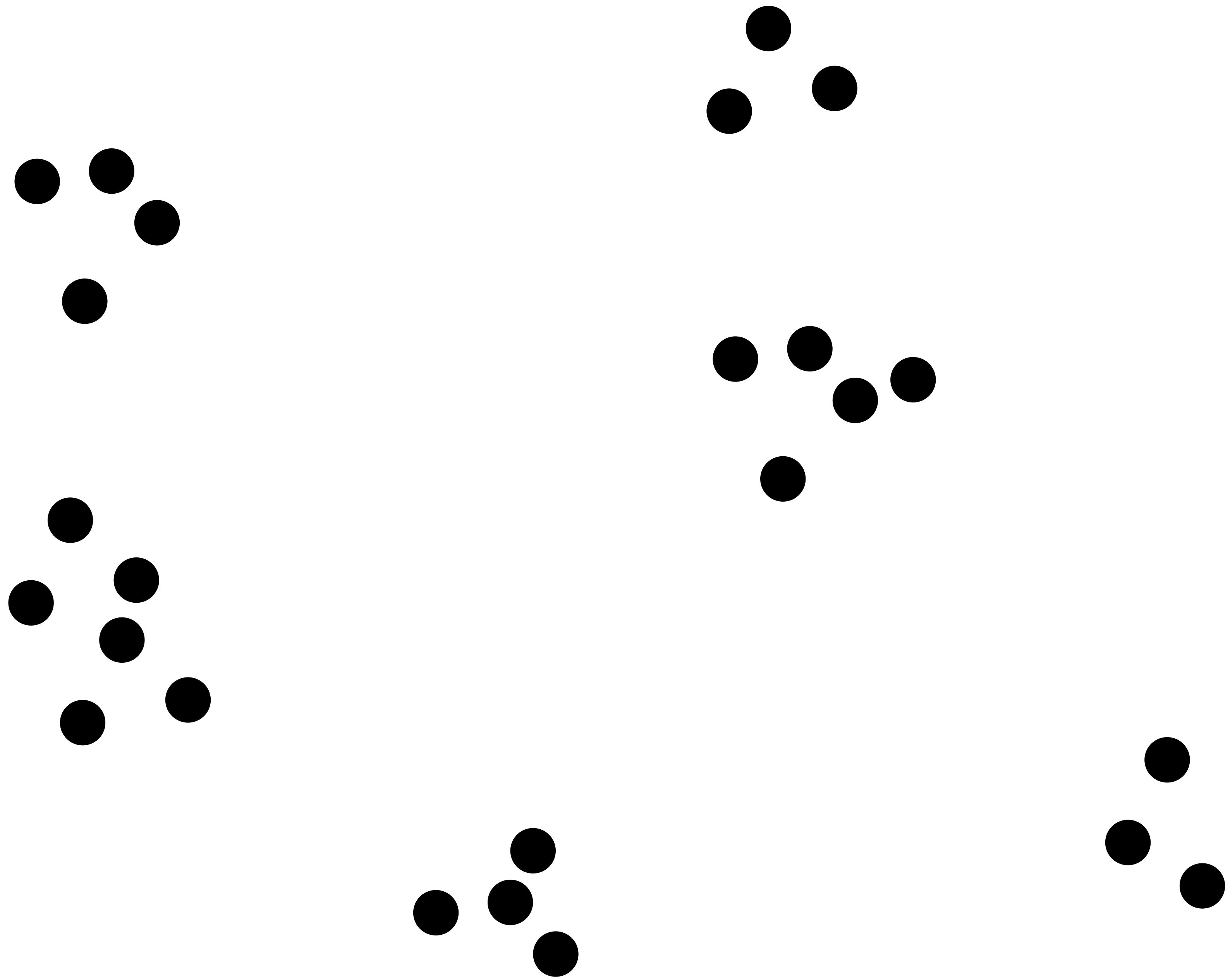
Images for **self-organizing map**



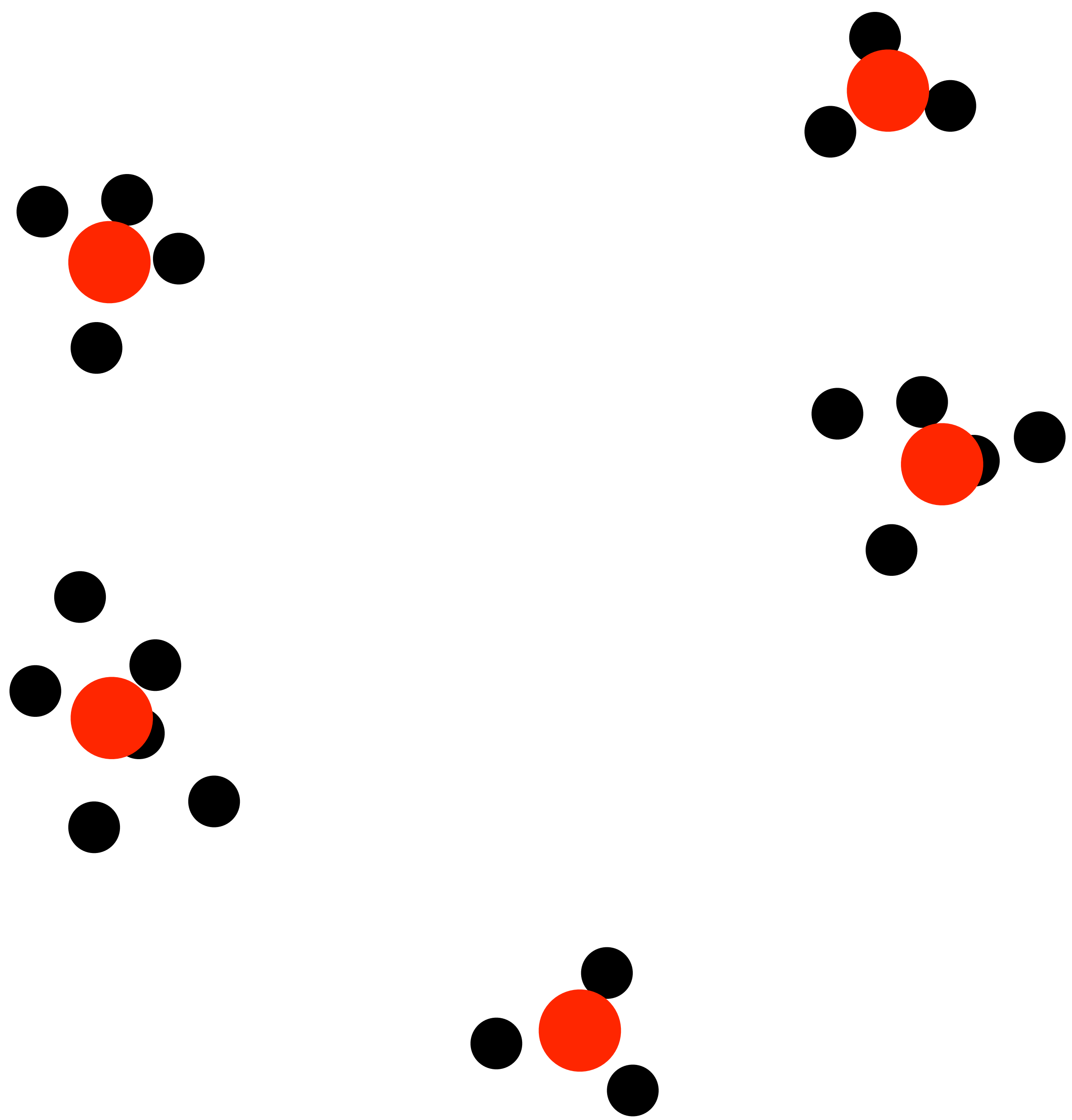
[**Self-Organizing Maps**]
davis.wpi.edu/~matt/courses/soms/ - Cached - Similar
23 Mar 1999 ... **Self-organizing maps** (SOMs) are a data visualization technique invented by Professor Teuvo Kohonen which reduce the dimensions of data ...

[**Self-Organizing Map** (SOM)]
users.ics.aalto.fi/jhollmen/dippa/node9.html - Cached
The **Self-Organizing Map** is one of the most popular neural network models. It belongs to the category of competitive learning networks. The **Self-Organizing Map** ...

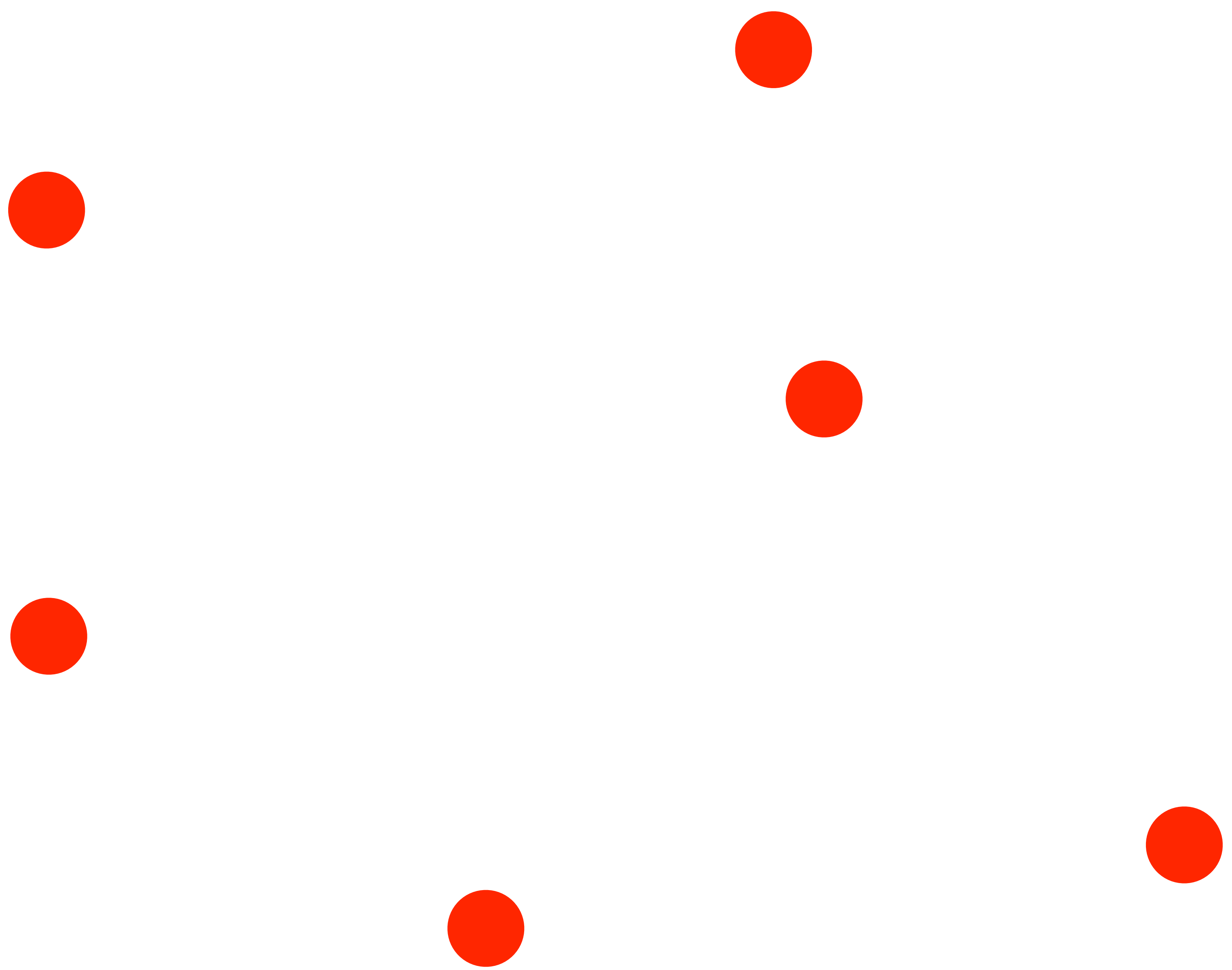
Vector Quantization



Vector Quantization



Vector Quantization



Vector Quantization

- minimizes the distortion

$$E = \frac{1}{NV} \sum_{i=1}^{N} \|x_i - v(x_i)\|^2$$

- with N the number of points (samples), V the number of centroids (units), x the samples, v the centroids

Vector Quantization

- One of the 1000 algorithms:
  - random initialization among the samples
  - iterations

$$v(x(t)) = v(x(t)) + \alpha(t)(x(t) - v(x(t)))$$

with v(x(t)) the closest centroid of x(t)

and Robbins-Monro conditions

$$\int_0^\infty \alpha(t)dt = \infty \qquad \int_0^\infty \alpha(t)^2 dt < \infty$$

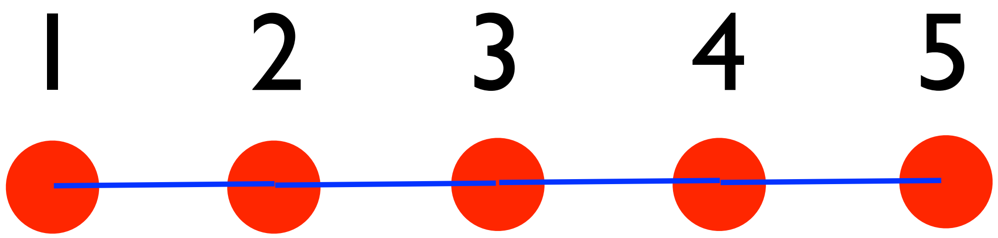
Vector Quantization

- Matlab examples

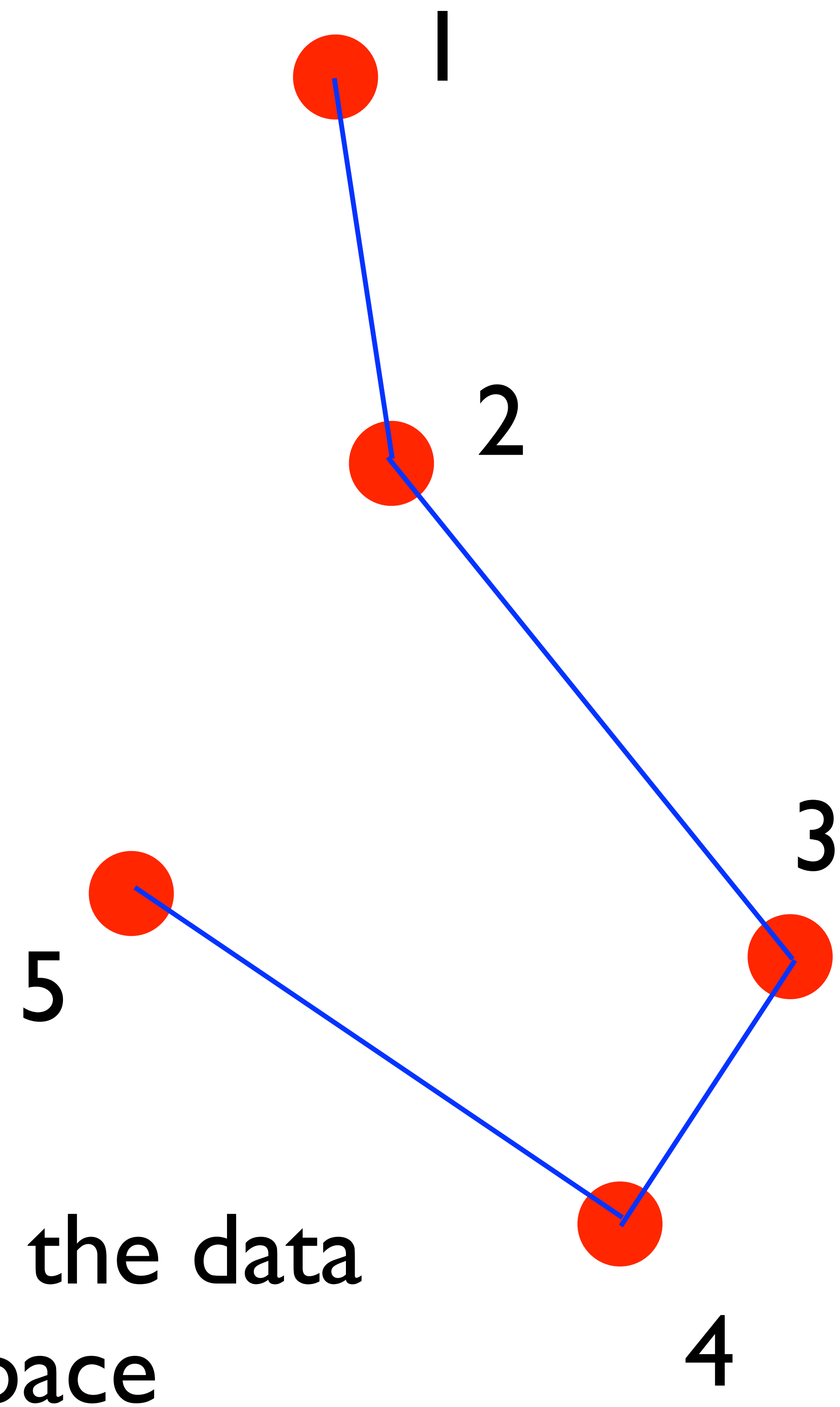
Self-Organizing Maps

- like stretching a sheet of plastic by pulling on certain points on it

SOM



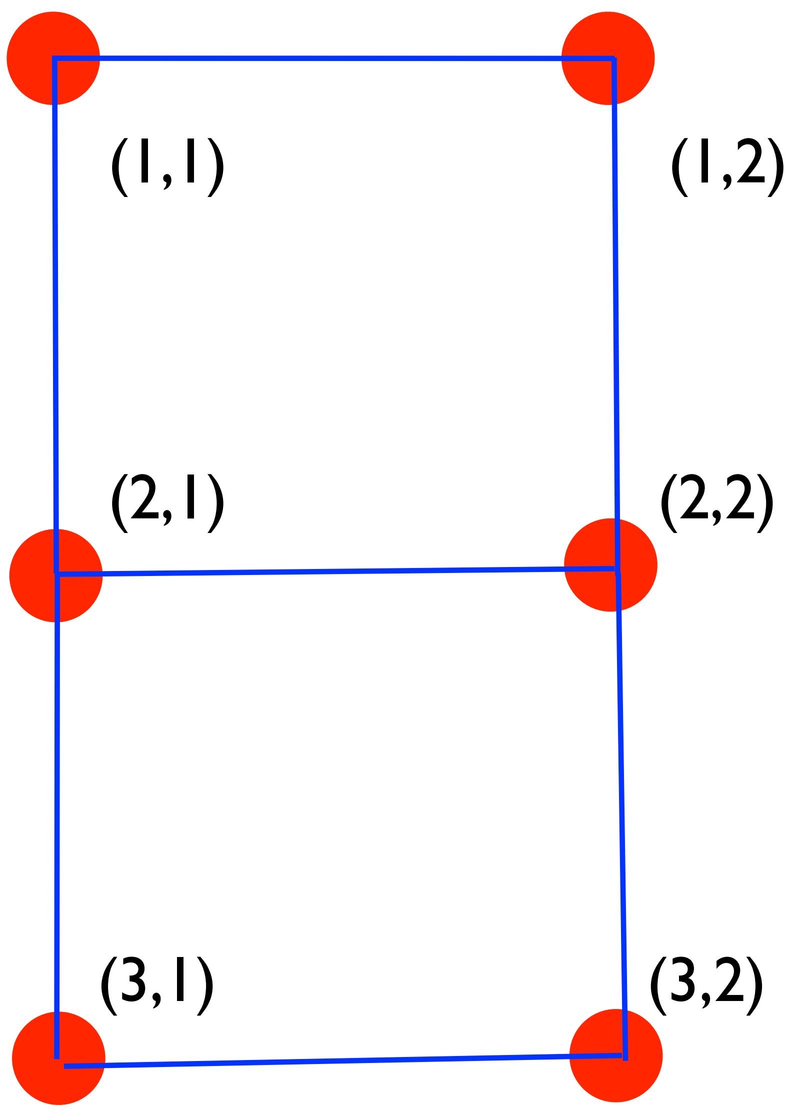
Structure of the map (definition)



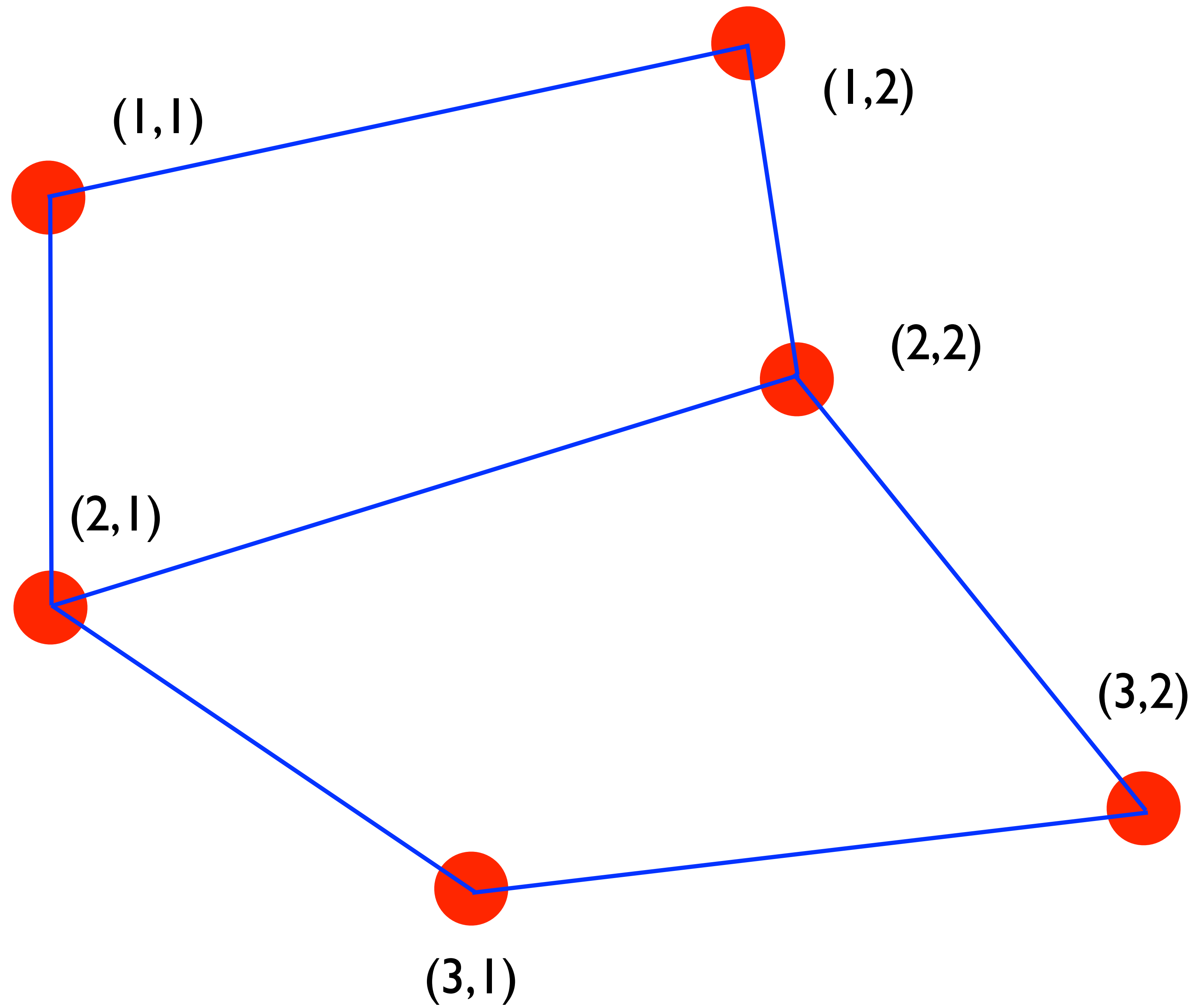
Map in the data space

SOM

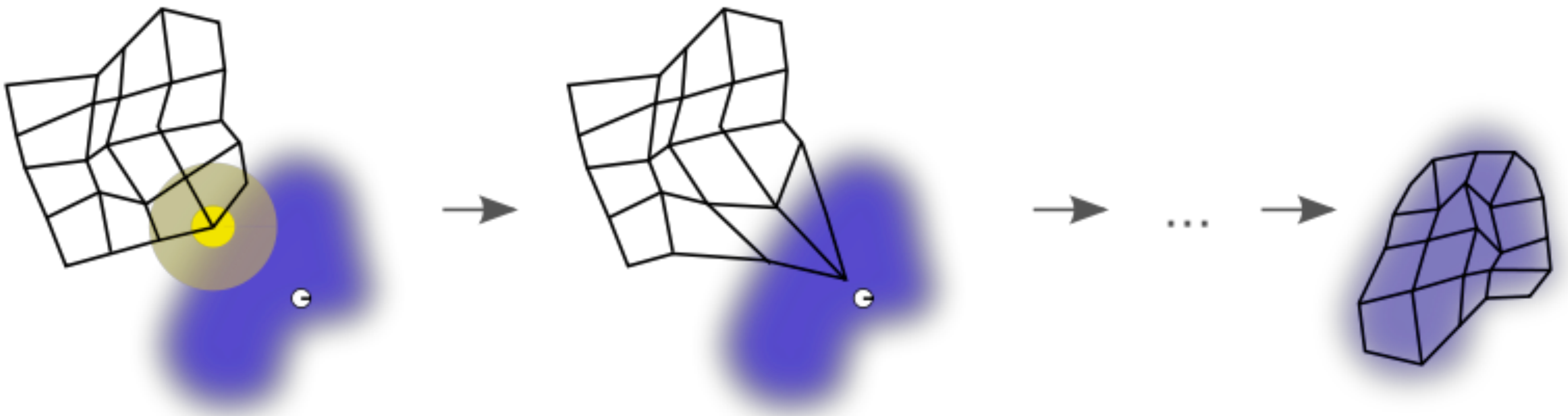
Structure of the map (definition)



Map in the data space



SOM



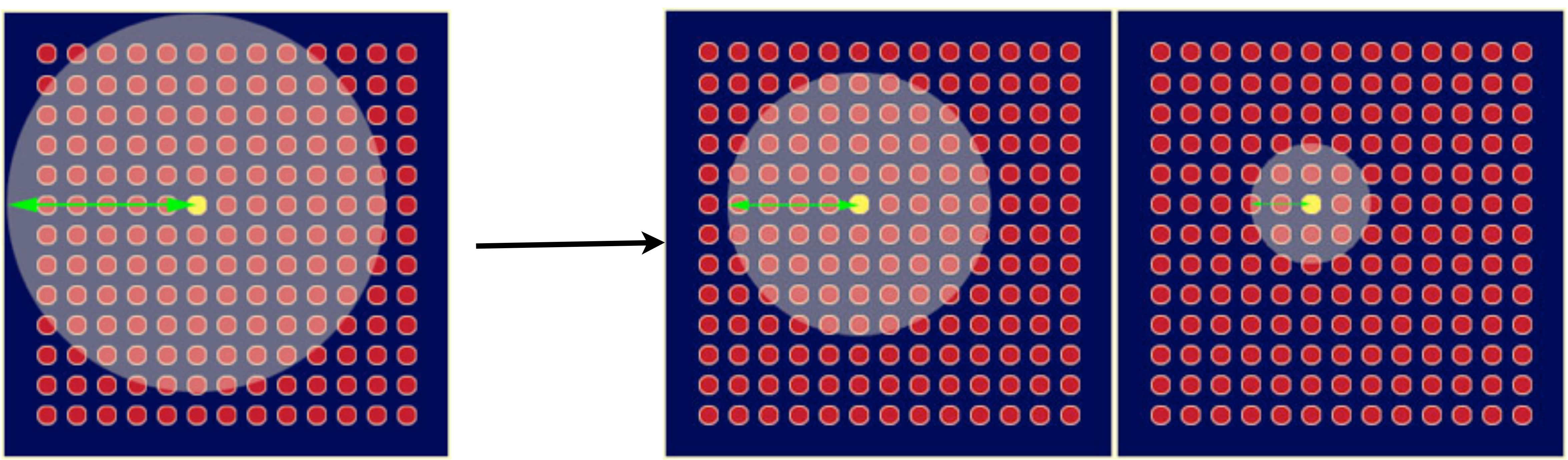
SOM

- Random initialization or PCA initialization
- Iterations

$$u(t+1) = u(t) + h_{u,v}(t)\alpha(t)(x(t)-u(t))$$

$$h_{u,v}(t)=\exp\left(\frac{-d_{grid}(u,v)}{\sigma(t)}\right) \quad \sigma(t)=\sigma_0\exp\left(\frac{-t}{\lambda}\right)$$

- Example:

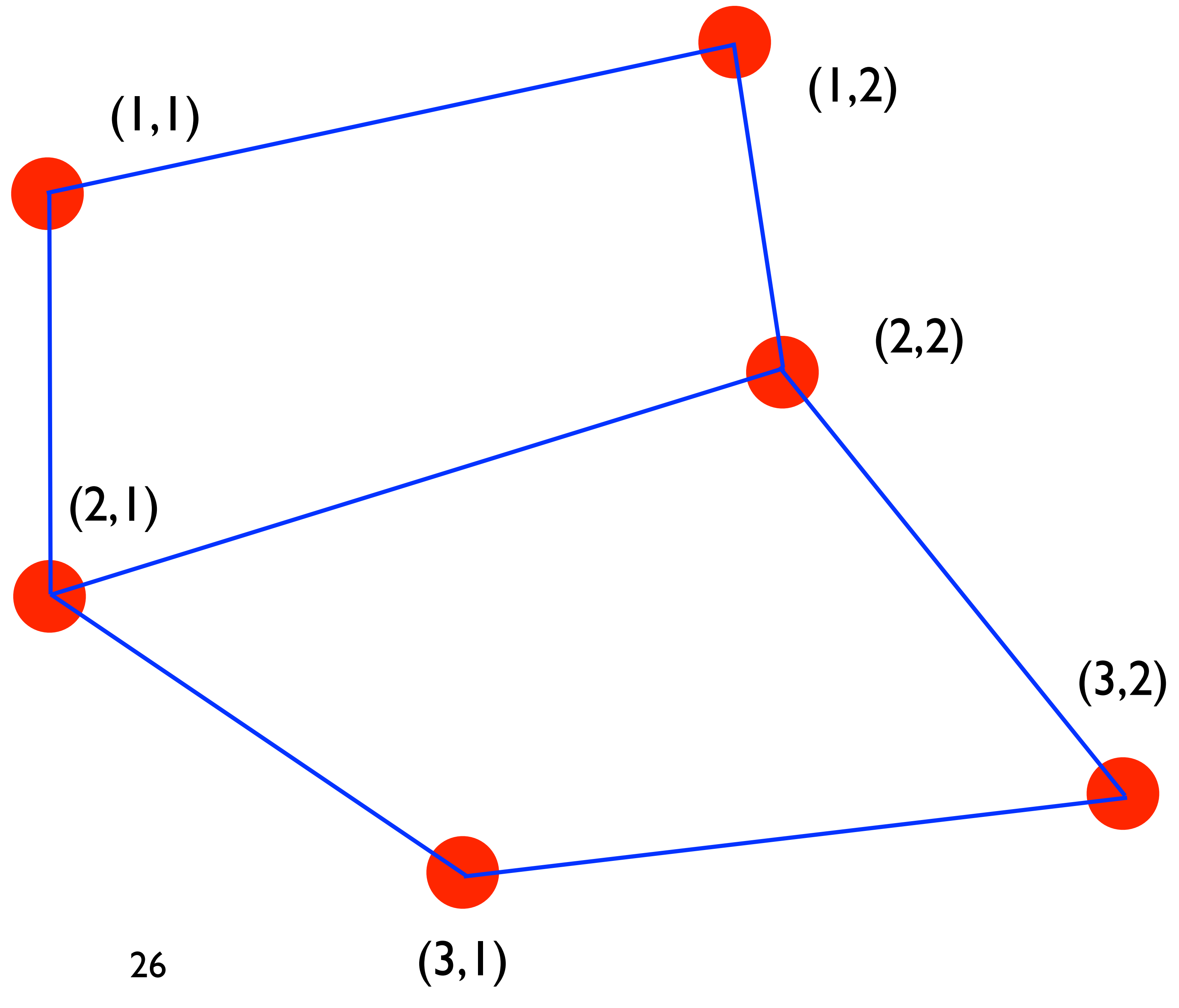


SOM

- Matlab Examples

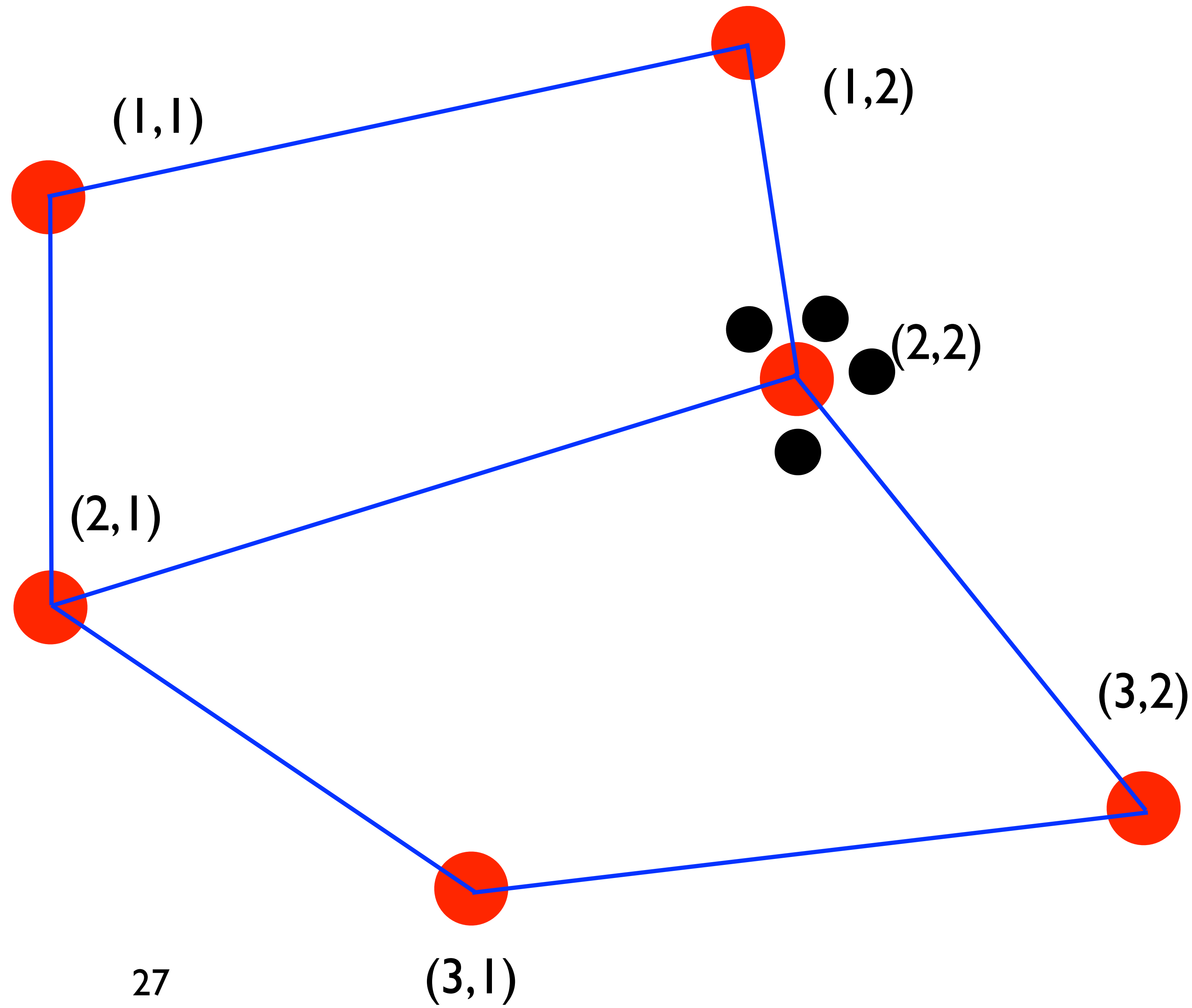
SOM

- And then? What about visualization?



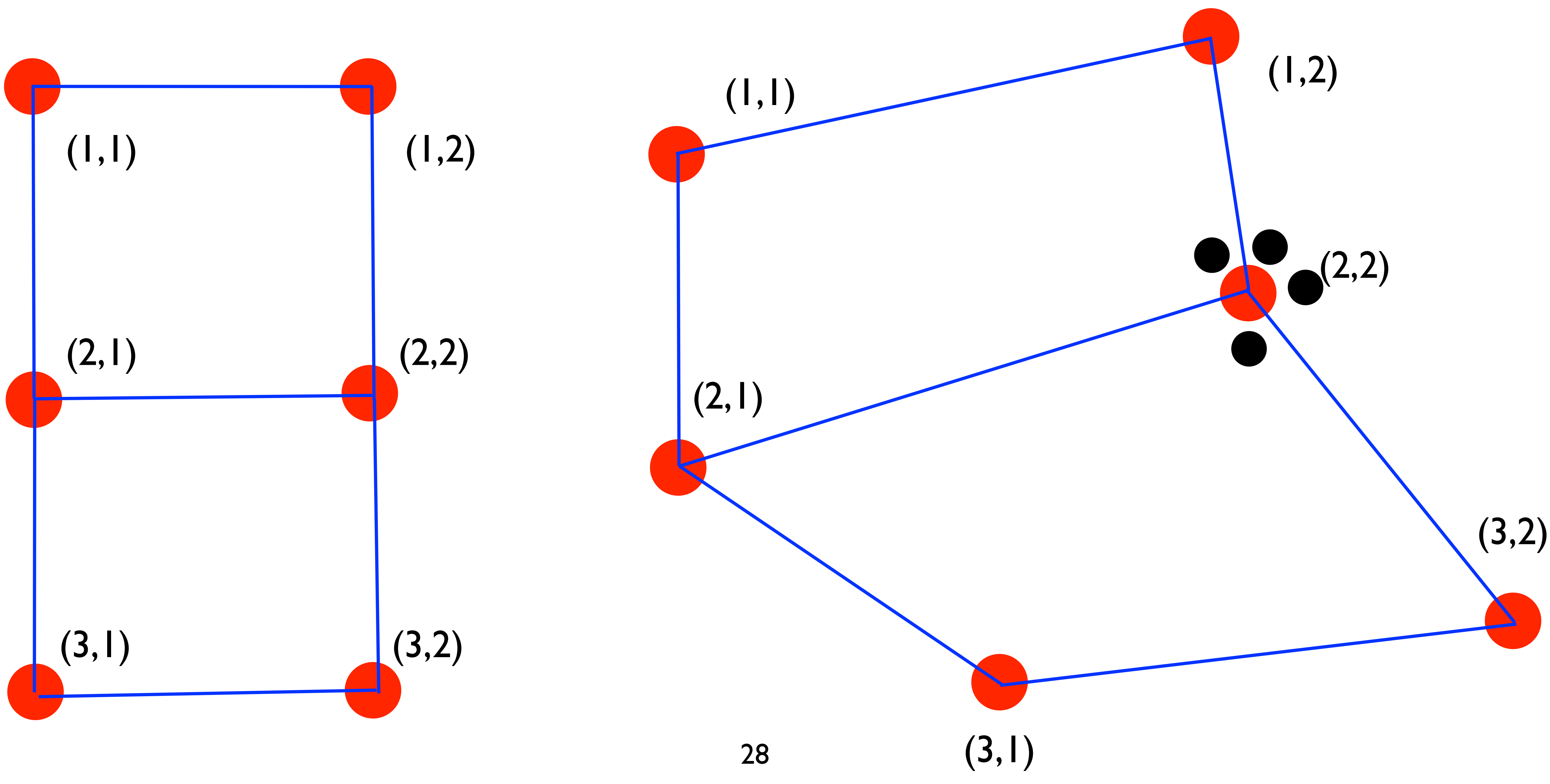
SOM

- And then? What about visualization?



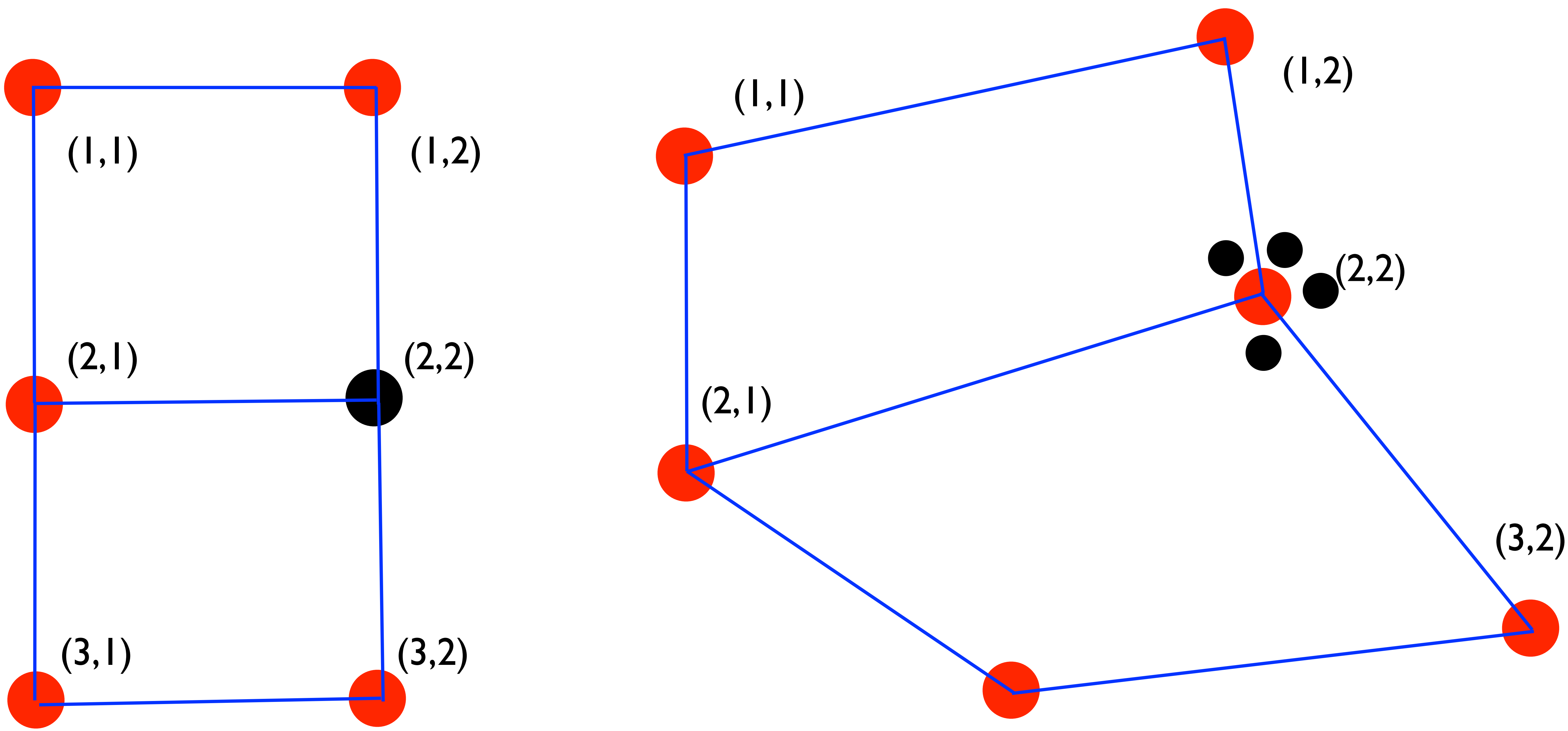
SOM

- And then? What about visualization?



SOM

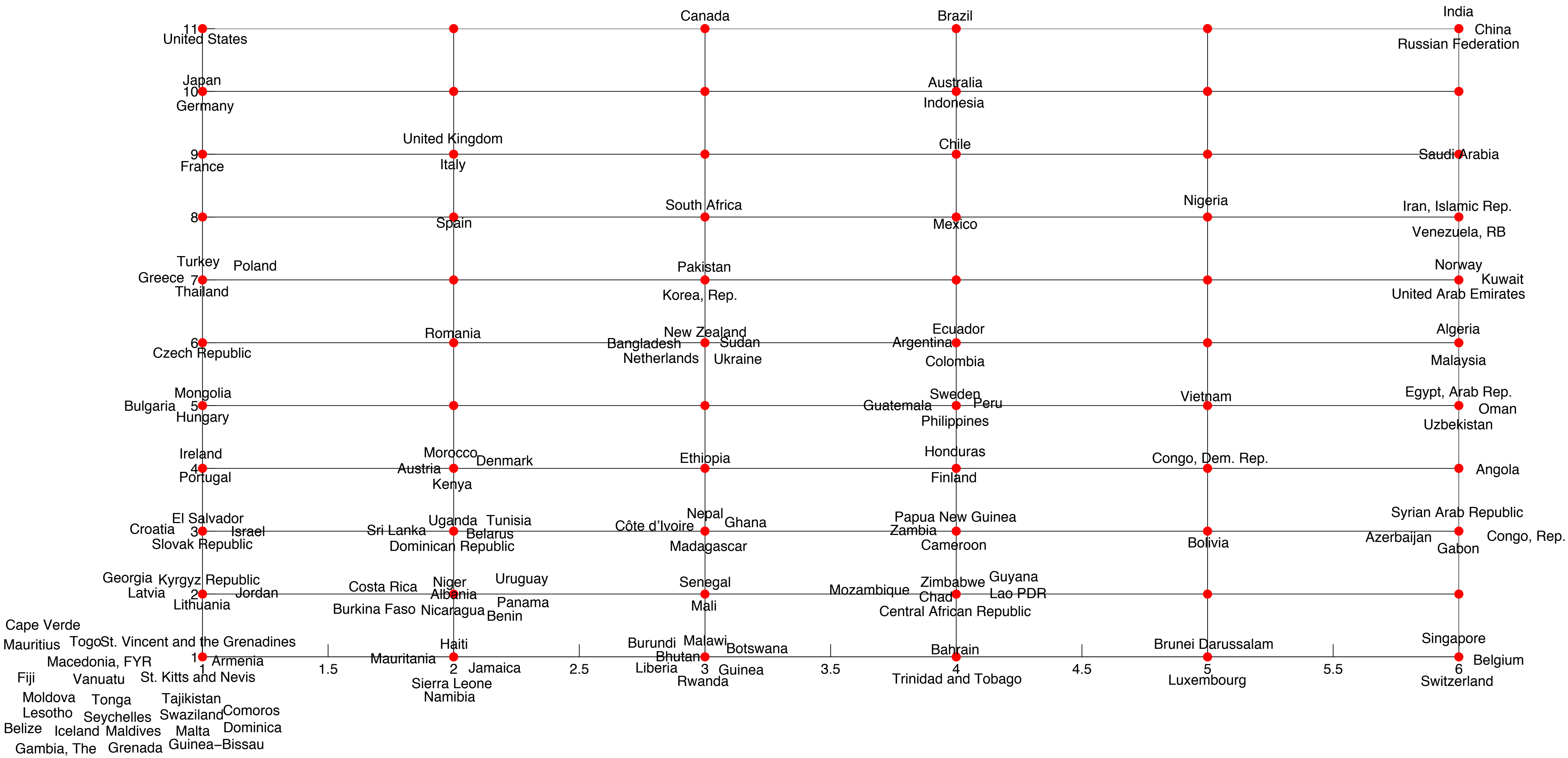
- And then? What about visualization?



SOM

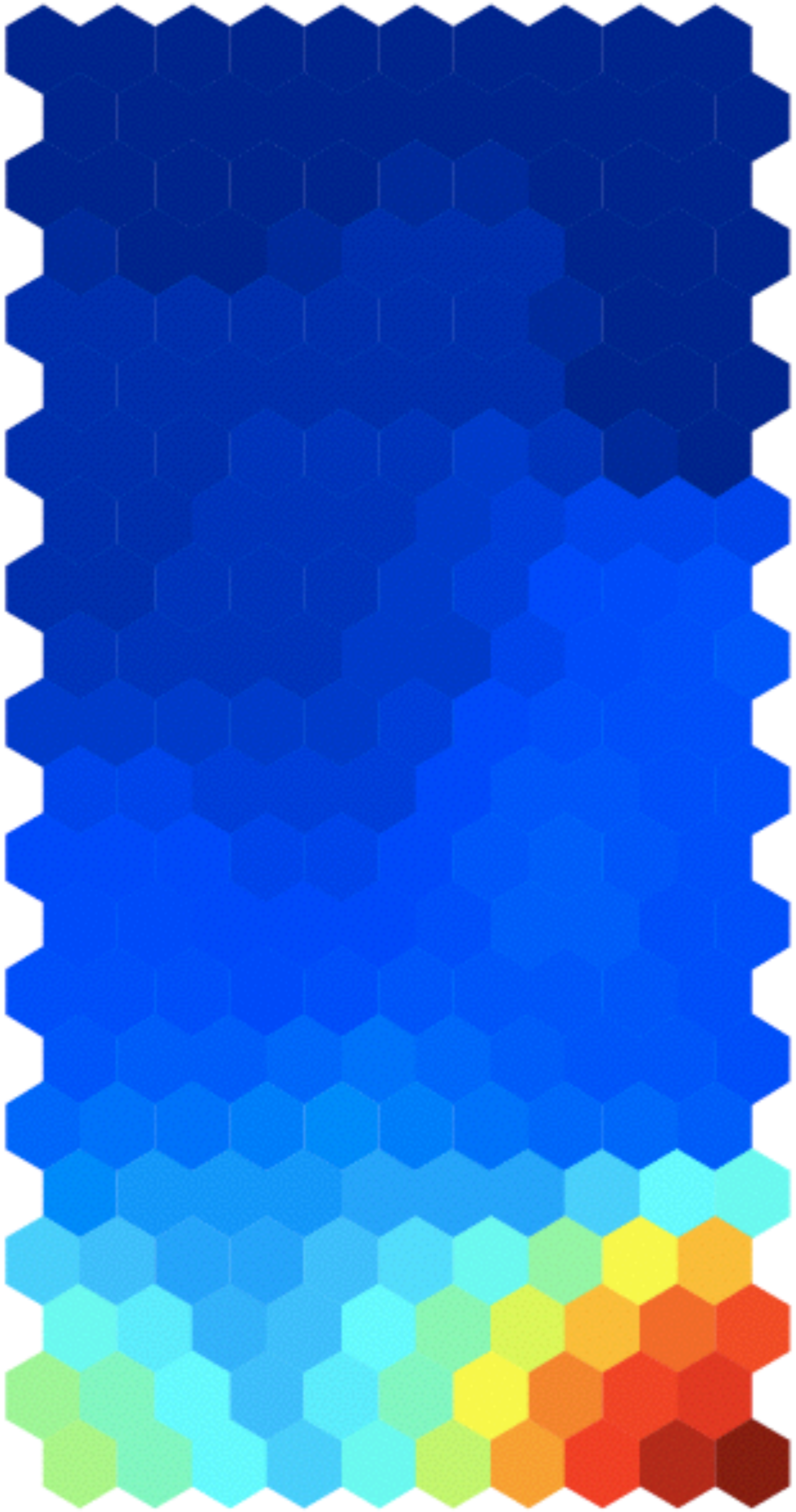
- Wealth Data Example: Excel + Matlab

SOM

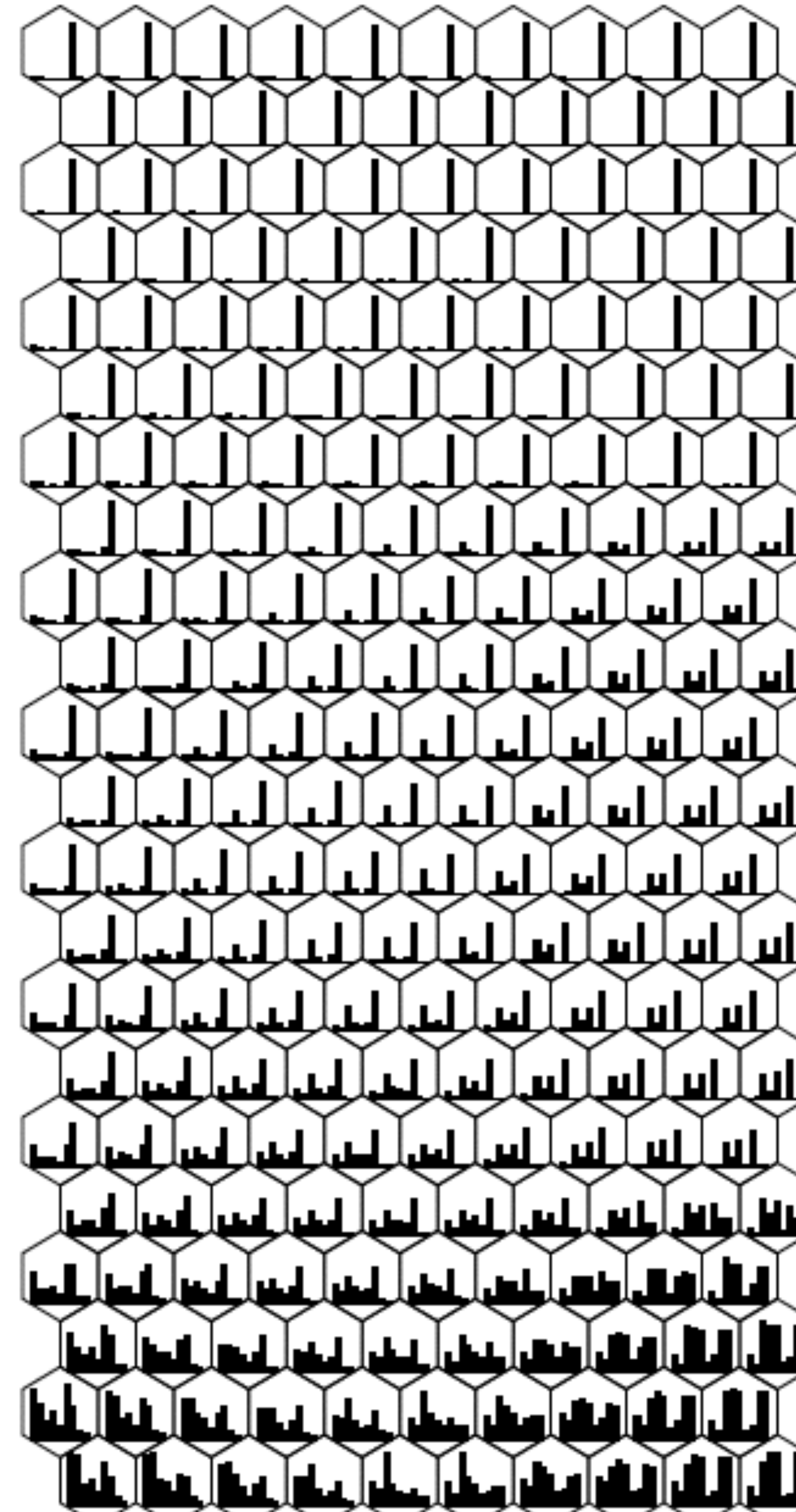


India
China
Russian Federation
Canada
Brazil
United States
Japan
Germany
Australia
Indonesia
United Kingdom
Italy
Chile
France
Saudi Arabia
South Africa
Nigeria
Iran, Islamic Rep.
Spain
Mexico
Venezuela, RB
Turkey
Poland
Greece
Thailand
Pakistan
Korea, Rep.
Norway
Kuwait
United Arab Emirates
Romania
Czech Republic
Bangladesh
Netherlands
New Zealand
Sudan
Ukraine
Ecuador
Argentina
Colombia
Algeria
Malaysia
Mongolia
Bulgaria
Hungary
Sweden
Guatemala
Peru
Philippines
Vietnam
Egypt, Arab Rep.
Oman
Uzbekistan
Ireland
Portugal
Morocco
Austria
Denmark
Kenya
Ethiopia
Honduras
Finland
Congo, Dem. Rep.
Angola
El Salvador
Croatia
Israel
Slovak Republic
Sri Lanka
Uganda
Tunisia
Belarus
Dominican Republic
Côte d'Ivoire
Nepal
Ghana
Madagascar
Papua New Guinea
Zambia
Cameroon
Bolivia
Syrian Arab Republic
Azerbaijan
Gabon
Congo, Rep.
Georgia
Latvia
Kyrgyz Republic
Jordan
Lithuania
Costa Rica
Burkina Faso
Niger
Albania
Nicaragua
Uruguay
Panama
Benin
Senegal
Mali
Mozambique
Zimbabwe
Chad
Guyana
Lao PDR
Central African Republic
Cape Verde
Mauritius
Togo
St. Vincent and the Grenadines
Macedonia, FYR
Armenia
Fiji
Vanuatu
St. Kitts and Nevis
Moldova
Lesotho
Tonga
Seychelles
Tajikistan
Swaziland
Comoros
Belize
Iceland
Maldives
Malta
Dominica
Gambia, The
Grenada
Guinea−Bissau
Haiti
Mauritania
Jamaica
Sierra Leone
Namibia
Burundi
Malawi
Bhutan
Liberia
Botswana
Guinea
Rwanda
Bahrain
Trinidad and Tobago
Brunei Darussalam
Luxembourg
Singapore
Belgium
Switzerland
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
1
2
3
4
5
6
7
8
9
10
11

SOM



Values of the prototype vectors shown on the SOM grid.
Left: one of the feature values.
Right: many of the feature values.



From Juha Vesanto's doctoral thesis, 2002

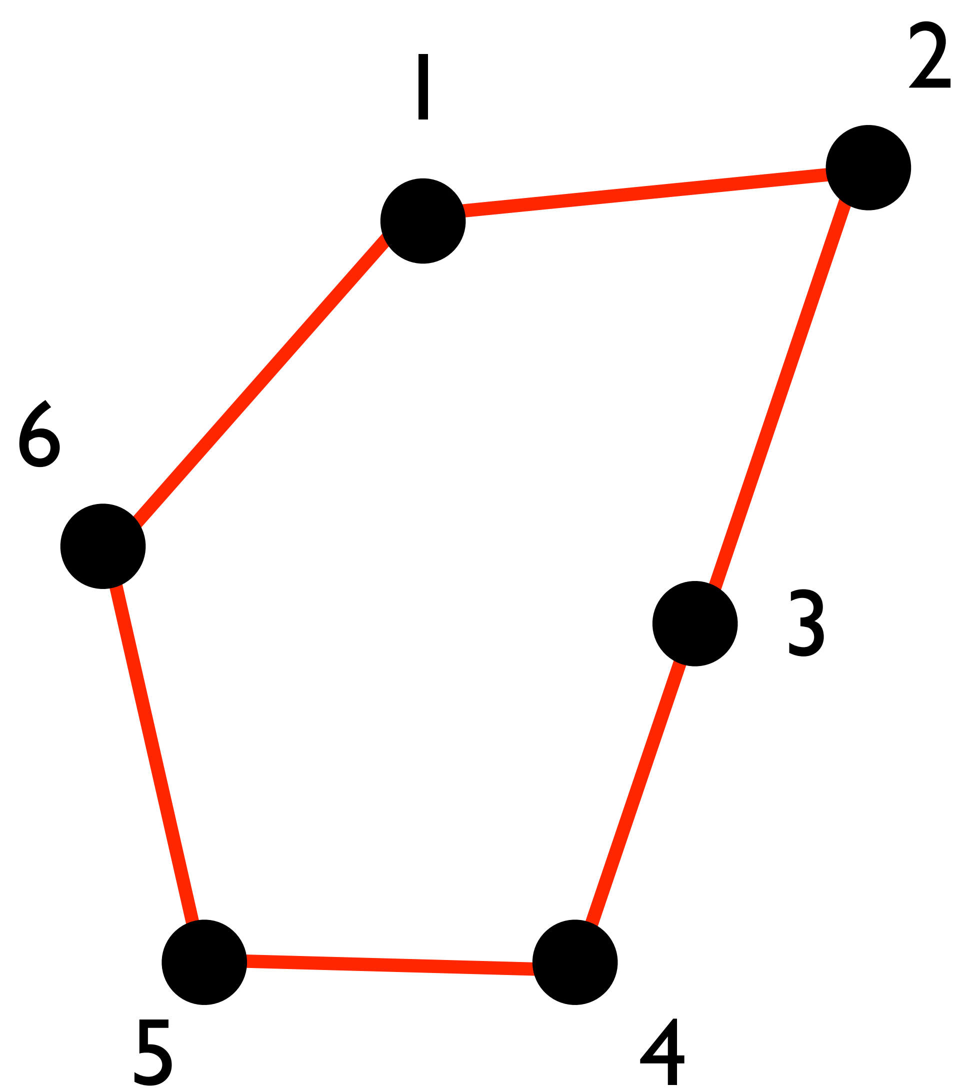
Laplacian eigenmap

- Laplacian eigenmap is a spectral method, like PCA.
- construct *k*-nearest neighbors graph. Assign $W_{ij}=1$, if i and j are neighbors, otherwise assign $W_{ij}=0$. Define diagonal matrix D, $D_{ii}=\sum_j W_{ij}$, and graph Laplacian, $L=D-W$.
- The embedding of data points is given by the eigenvectors of L, corresponding to the d smallest non-zero eigenvalues.

Laplacian eigenmap

N=6

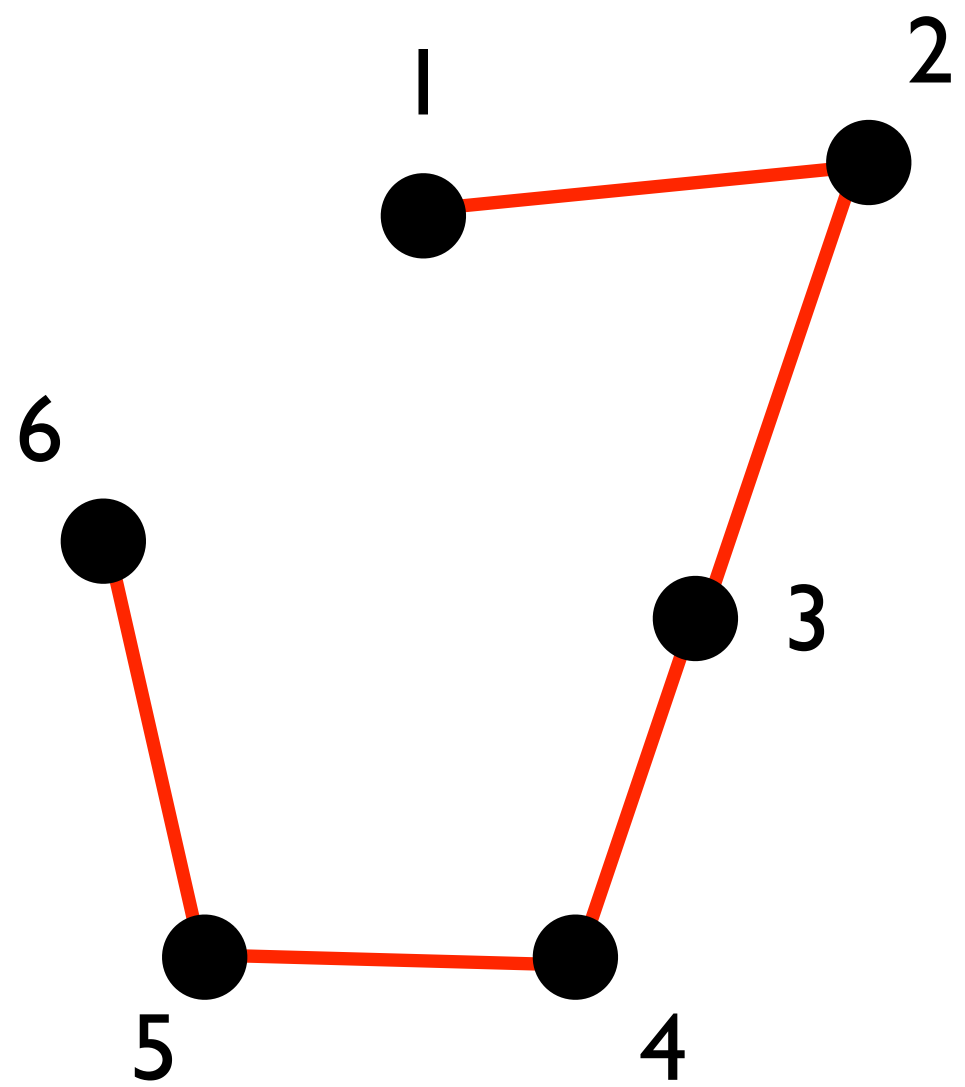
k=2




```
octave:20> L
L =

  -2   1   0   0   0   1
   1  -2   1   0   0   0
   0   1  -2   1   0   0
   0   0   1  -2   1   0
   0   0   0   1  -2   1
   1   0   0   0   1  -2

octave:21> [V,D] = eig(L)
V =

  -0.408248   0.458032  -0.351483   0.551189  -0.171826   0.408248
   0.408248  -0.533409  -0.220926   0.424400   0.391430   0.408248
  -0.408248   0.075377   0.572409  -0.126788   0.563257   0.408248
   0.408248   0.458032  -0.351483  -0.551189   0.171826   0.408248
  -0.408248  -0.533409  -0.220926  -0.424400  -0.391430   0.408248
   0.408248   0.075377   0.572409   0.126788  -0.563257   0.408248

D =

  -4.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   0.00000  -3.00000   0.00000   0.00000   0.00000   0.00000
   0.00000   0.00000  -3.00000   0.00000   0.00000   0.00000
   0.00000   0.00000   0.00000  -1.00000   0.00000   0.00000
   0.00000   0.00000   0.00000   0.00000  -1.00000   0.00000
   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
```

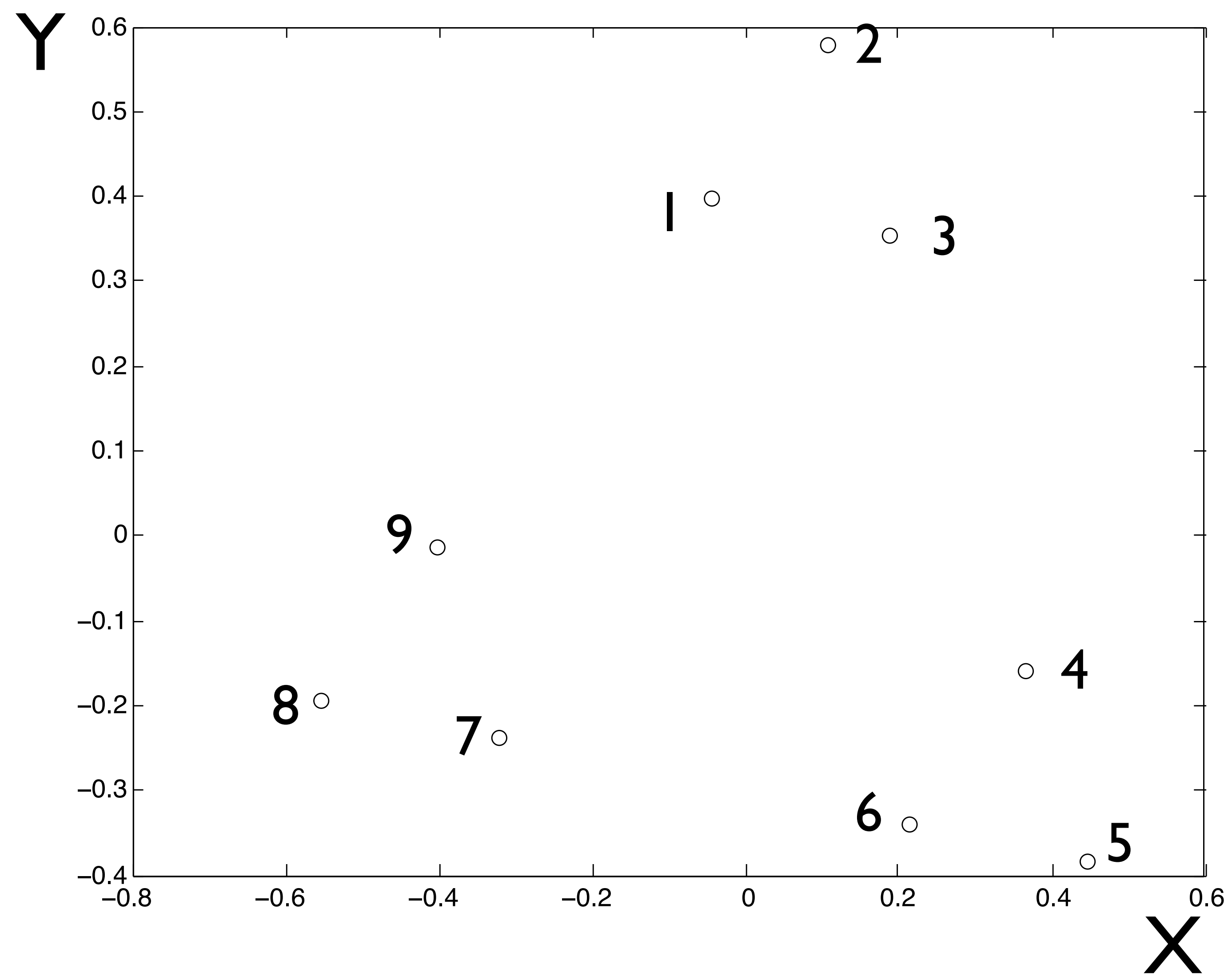
Laplacian eigenmap

```
>> L

L =

    -1     1     0     0     0     0
     1    -2     1     0     0     0
     0     1    -2     1     0     0
     0     0     1    -2     1     0
     0     0     0     1    -2     1
     0     0     0     0     1    -1

>> [V,D]=eig(L)

V =

    0.1494    0.2887    0.4082   -0.5000    0.5577    0.4082
   -0.4082   -0.5774   -0.4082    0.0000    0.4082    0.4082
    0.5577    0.2887   -0.4082    0.5000    0.1494    0.4082
   -0.5577    0.2887    0.4082    0.5000   -0.1494    0.4082
    0.4082   -0.5774    0.4082    0.0000   -0.4082    0.4082
   -0.1494    0.2887   -0.4082   -0.5000   -0.5577    0.4082


D =

   -3.7321         0         0         0         0         0
         0   -3.0000         0         0         0         0
         0         0   -2.0000         0         0         0
         0         0         0   -1.0000         0         0
         0         0         0         0   -0.2679         0
         0         0         0         0         0   -0.0000
```



Laplacian eigenmap

```
>> L

L =

    -3     1     1     0     0     0     0     0     1
     1    -2     1     0     0     0     0     0     0
     1     1    -3     1     0     0     0     0     0
     0     0     1    -3     1     1     0     0     0
     0     0     0     1    -2     1     0     0     0
     0     0     0     1     1    -3     1     0     0
     0     0     0     0     0     1    -3     1     1
     0     0     0     0     0     0     1    -2     1
     1     0     0     0     0     0     1     1    -3

>> [V,D]=eig(L)

V =                                                         Y         X

   -0.4082   0.5014  -0.1803  -0.3099   0.3433  -0.2527   0.3982  -0.0451   0.3333
   -0.0000  -0.1232   0.2895   0.5573   0.1121   0.3482   0.5775   0.1093   0.3333
    0.4082  -0.2177  -0.4863  -0.2475  -0.4554  -0.0955   0.3542   0.1874   0.3333
   -0.4082  -0.0946   0.5243  -0.2475  -0.4554  -0.0955  -0.1601   0.3674   0.3333
   -0.0000  -0.1891  -0.2514   0.3798   0.3669  -0.4069  -0.3834   0.4455   0.3333
    0.4082   0.5300   0.0546  -0.1323   0.0885   0.5024  -0.3394   0.2130   0.3333
   -0.4082  -0.4068  -0.3441  -0.1323   0.0885   0.5024  -0.2381  -0.3223   0.3333
    0.0000   0.3123  -0.0380   0.4422  -0.4319  -0.2497  -0.1941  -0.5548   0.3333
    0.4082  -0.3123   0.4317  -0.3099   0.3433  -0.2527  -0.0148  -0.4005   0.3333


D =

   -5.0000        0        0        0        0        0        0        0        0
         0  -4.3028        0        0        0        0        0        0        0
         0        0  -4.3028        0        0        0        0        0        0
         0        0        0  -3.0000        0        0        0        0        0
         0        0        0        0  -3.0000        0        0        0        0
         0        0        0        0        0  -3.0000        0        0        0
         0        0        0        0        0        0  -0.6972        0        0
         0        0        0        0        0        0        0  -0.6972        0
         0        0        0        0        0        0        0        0  -0.0000
```



Curvilinear component analysis (CCA)

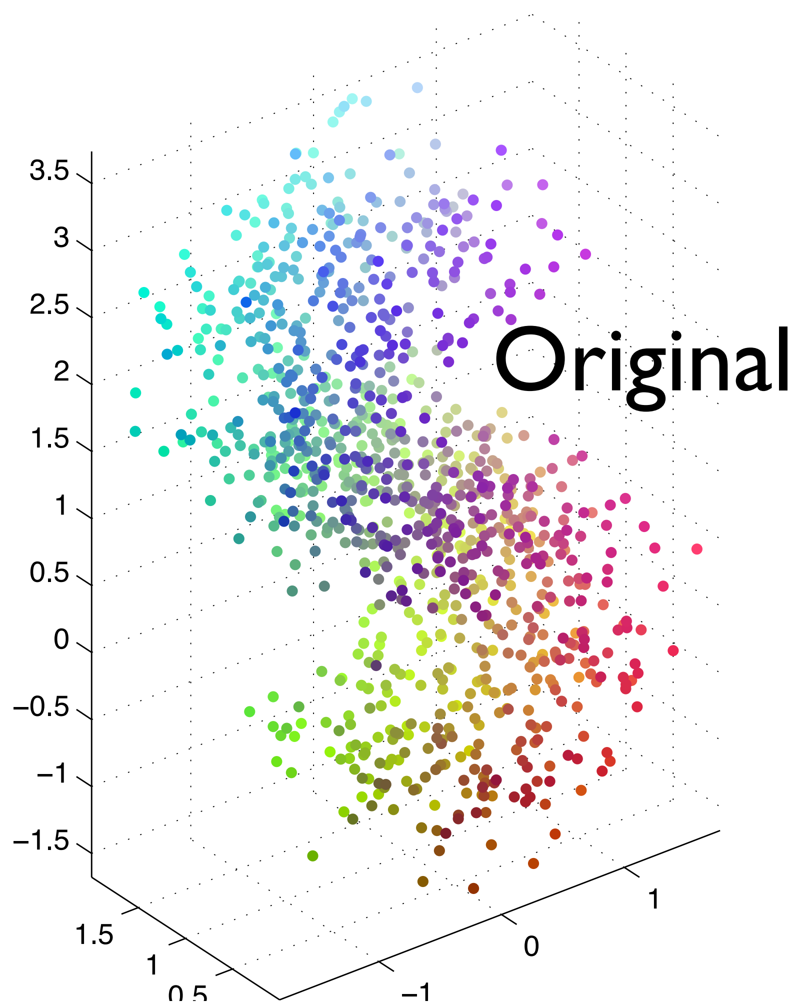
- Demartines, Hérault, 1997.
- *Curvilinear component analysis* (CCA) is like (absolute) MDS, except that only short distances are taken into account.
- More formally, the cost function reads

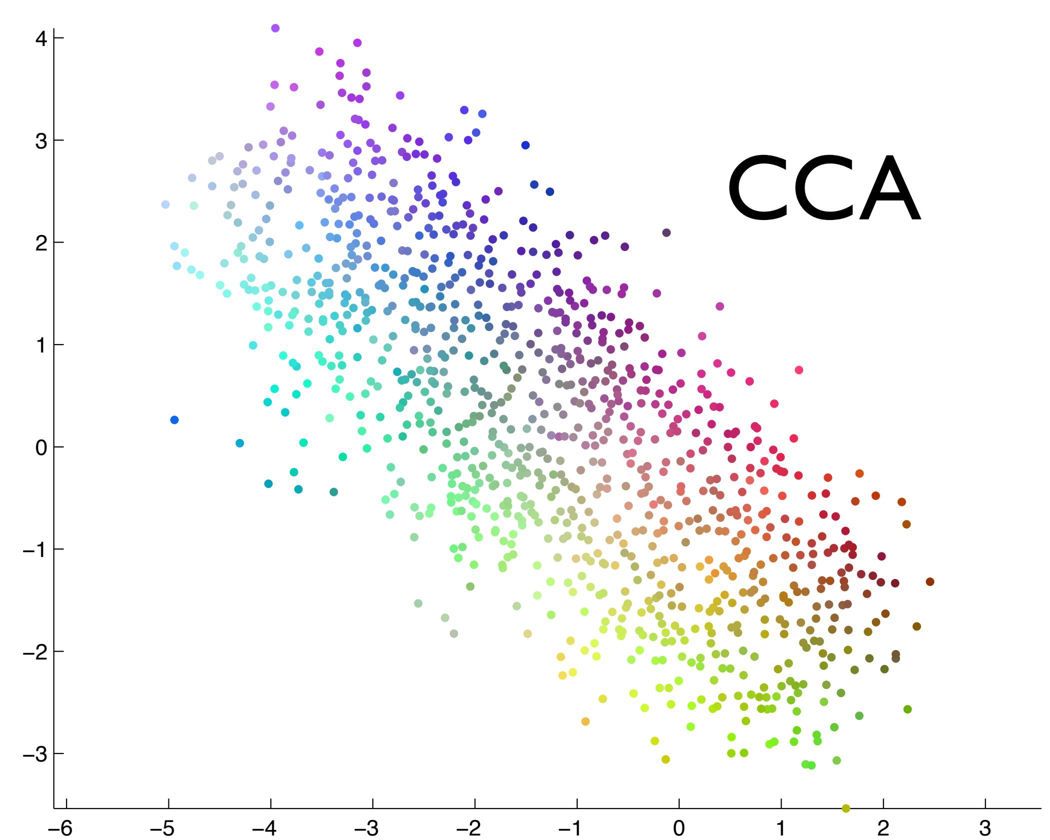
$$\sigma_r = \sum_{i<j} (d(x_i, x_j) - d(y_i, y_j))^2 \, F(d(y_i, y_j), \lambda_y)$$

where $F(d,\lambda_y)$ equals unity, if $d<\lambda_y$, and zero otherwise; and d denotes the Euclidean distance of points in the original space (x) and in the projection (y), respectively. (Actually, $F(d,\lambda_y)$, could be any monotonically decreasing function in d.)

Curvilinear component analysis (CCA)

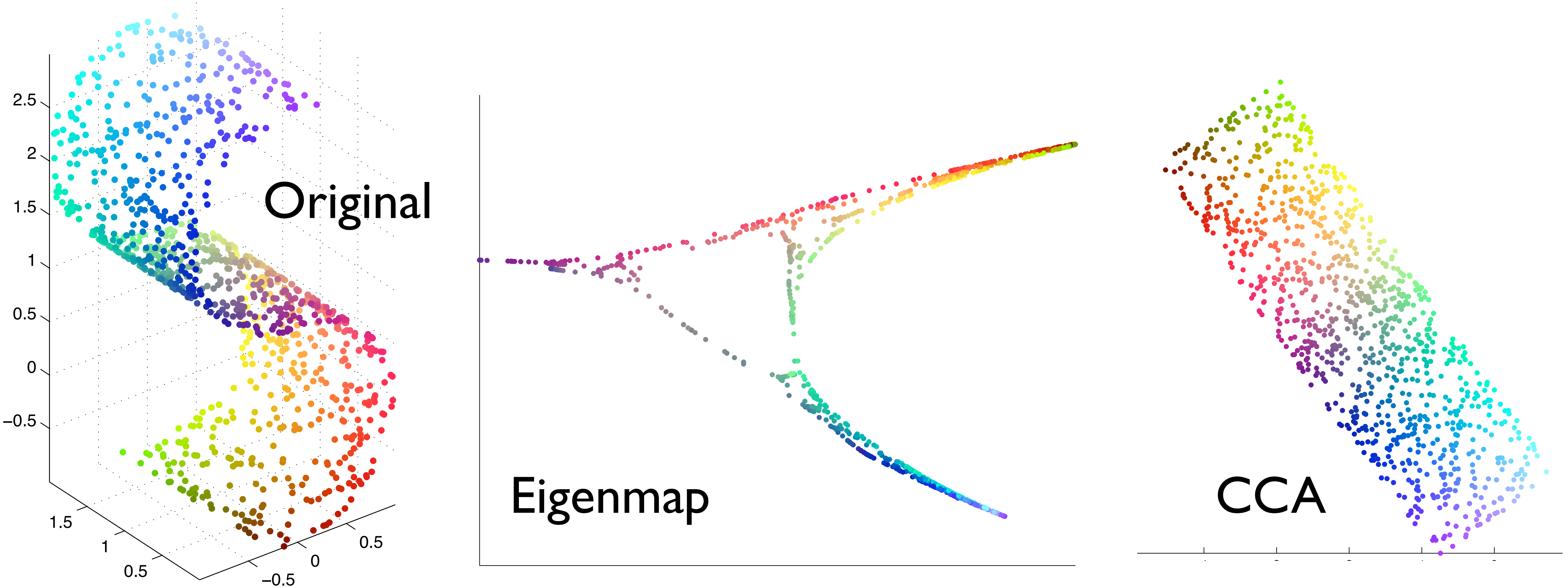
- CCA performs generally well in terms of precision; it appears to be quite robust.
- Notice outliers at right: they are result of small neighborhood.





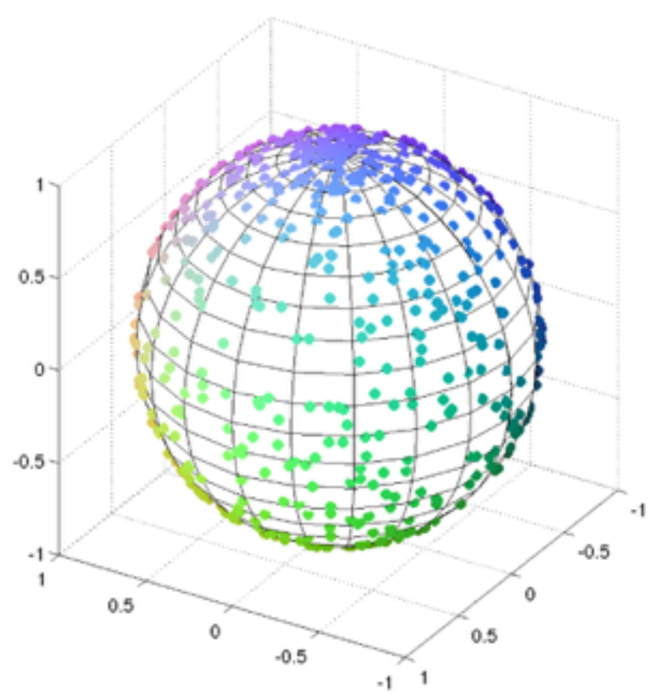
Figures by Jarkko Venna

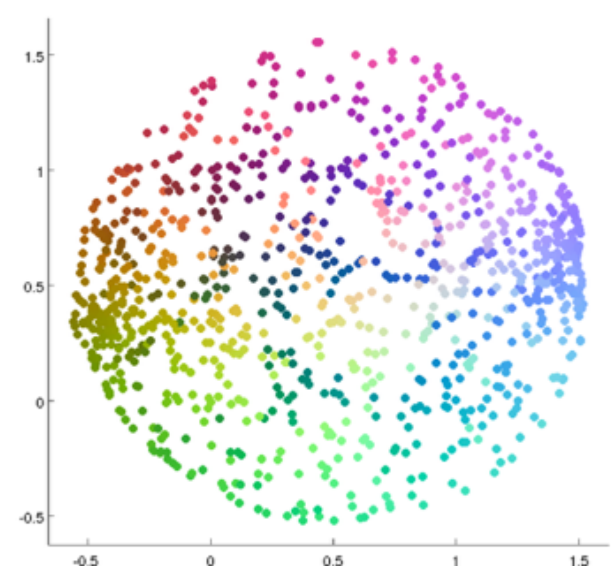
Laplacian eigenmap



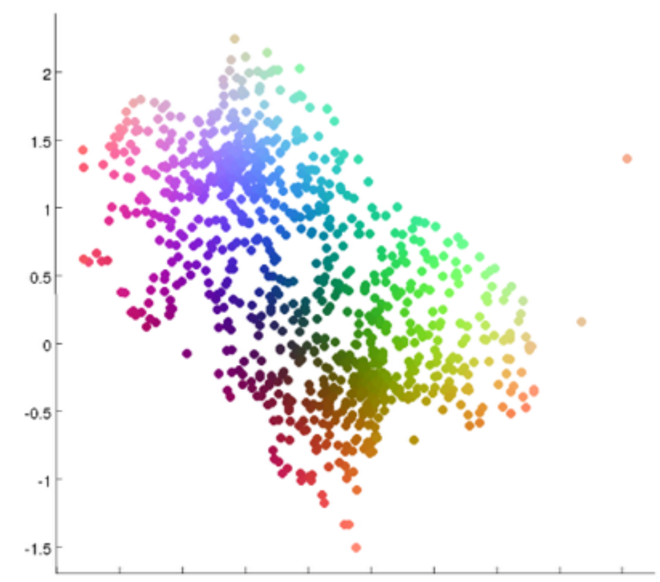
Methods that directly use precision and recall?

- We interpreted some methods by precision and recall. Most methods don't optimize these, they use various cost functions.
- The Neighbor Retrieval Visualizer (Venna & Kaski 2007; Venna, Peltonen, Nybo, Aidos, Kaski 2010) directly maximizes a user-specified tradeoff of precision & recall





NeRV, result for max recall



NeRV, result for max precision

- Stochastic Neighbor Embedding (Hinton&Roweis 2002) can also be interpreted like this: maximizes recall
- Details later on the Neighbor Embedding lectures

Next lecture:

- general mathematical framework for nonlinear dimensionality reduction
- more methods: Isomap, Locally Linear Embedding, Maximum Variance Unfolding
- Supervised nonlinear dimensionality reduction