Visualization of High Dimensional Scientific Data

Roberto Tagliaferri and Antonino Staiano

Department of Mathematics and Computer Science,

University of Salerno, Italy

{robtag,astaiano}@unisa.it

Outline

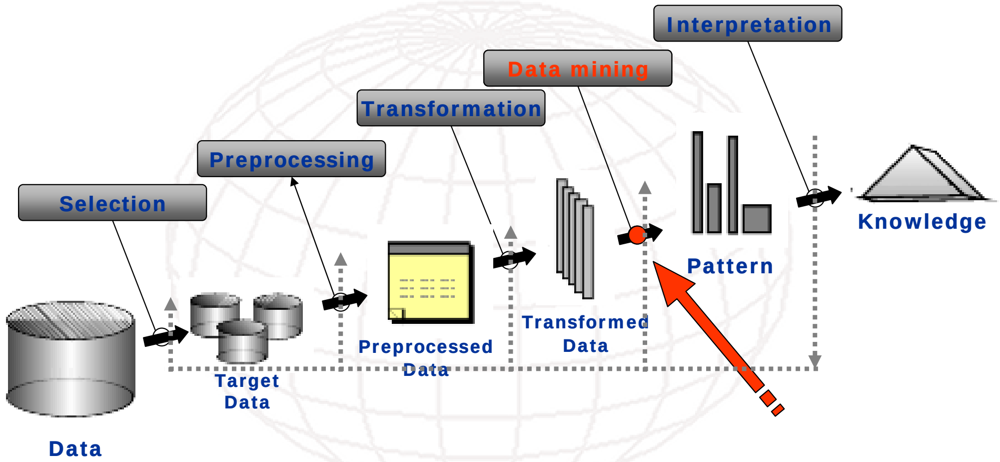
- **Introduction**
  - **Knowledge Discovery in Databases**
  - **Data Mining**
  - **Data Visualization**
  - **Sample dataset**
- **Traditional Visualization Methods**
  - **Scatter Plots**
  - **Principal Component Analysis**
  - **Multidimensional Scaling (MDS) and others (not in this talk)**
- **Latent variable models**
  - **Linear models**
    - ✓ Probabilistic PCA
    - ✓ Mixture of Probabilistic PCA
- **Global nonlinear models**
  - **Self Organizing Maps**
  - **Nonlinear latent variable models**
    - ✓ Generative Topographic Mapping
    - ✓ Probabilistic Principal Surfaces
      - ❖ Spherical PPS
      - ❖ An easy-to-use graphical user interface
- **Hierarchical latent variable models: overview**
  - **Hierarchical agglomeration of PPS: the Neg Entropy Clustering algorithm**
- **Case Study: Yeast Gene Microarray Analysis**
- **Conclusions**

IJCNN 2005 Tutorial, Montréal, August 2 **2**

Abstract: the recent technological advances are producing huge data sets in almost all fields of scientific research, from astronomy to genetics. Although each research field often requires ad-hoc, fine tuned, procedures to properly exploit all the available information inherently present in the data, there is an urgent need for a new generation of general computational theories and tools capable to boost most human activities of data analysis. Traditional data analysis methods, in fact, are inadequate to cope with such exponential growth in the data volume and especially in the data complexity (ten or hundreds of dimensions of the parameter space). Among the data mining methodologies, visualization plays a key role in developing good models for data especially when the quantity of data is large. For a scientist, i.e. the expert in a specific domain, is essential the need for a visual environment that facilitates exploring high-dimensional data dependent on many parameters. Data visualization is an important means of extracting useful information from large quantities of raw data. The human eye and brain together make a formidable pattern detection tool, but for them to work the data must be represented in a low-dimensional space, usually two or three dimensions. Even quite simple relationship can seem very obscure when the data is presented in tabular form, but are often very easy to see by visual inspection. Many algorithms for data visualization have been proposed by both neural computing and statistics communities, most of which are based on a projection of the data onto a two or three dimensional visualization space. This tutorial embraces a number of these visualization techniques both linear and nonlinear: Principal Component Analysis (PCA), Probabilistic PCA (PPCA), Mixture of PPCA. PCA, PPCA and mixture of PPCA are appropriate when the data is linear or approximately piece-wise linear. An alternative approach is to use global nonlinear methods such as Self Organizing Maps (SOM). However, SOM does not define any density model and suffers of other drawbacks which can be overcame employing nonlinear latent variable models: Generative Topographic Mapping (GTM) and Probabilistic Principal Surfaces (PPS). Finally, the tutorial reviews hierarchical linear (based on mixture of PPCA) and nonlinear (based on GTM) latent variable models and concludes by illustrating a new proposed hierarchical model based on PPS.

Intro: Knowledge Discovery in Databases (KDD)

- KDD Main Steps



Process involved in whatever data-rich field aimed to extract meaningful information from data

Intro: KDD and Data Mining

- *Data Mining is a key step in KDD process aimed to find meaningful patterns in the data.*
- Data Mining Methods
  - Regression
  - Classification
  - Clustering
  - Data Visualization

Intro: Data Visualization

Visualization plays a key role in developing good models for data, especially when the quantity of data is large.

- It allows the user to interact with and query the data more effectively.
- It is an important aid in feature selection, gives information about local deviations in performance and provides a useful `sanity check' for objective quantitative measures (such as generalization performance).
- It plays an important role in the search for clusters of similar data points, which are most easily determined by eye.
- The quantity and complexity of many datasets means that simple visualization methods, such as Principal Component Analysis, are not very effective.

Intro: sample data set

- GOODS Catalog (7 optical bands: *U,B,V,R,I,J,K*)
  - 28405 sources (WFI+SOFI)
  - 21 parameters (Magnitude, Kron Radius, Flux, for each band)
  - 24872 “drop outs”
- Sources labeled as *Star*, *Galaxy*, *DStar* (Dropped Star) and *DGalaxy* (Dropped Galaxy)

IJCNN 2005 Tutorial, Montréal, August 2 6

The Great Observatories Origins Deep Survey (or GOODS) is an international project which joins together NASA, ESA (European Space Agency) and some of the most powerful ground-based facilities, to survey the distant universe to the faintest flux limits across the broadest range of wavelengths. At the end of the project, GOODS will survey a total of roughly 320 square arcminutes in two fields centered on the Hubble[1] Deep Field North and the Chandra[2] Deep Field South, respectively. The GOODS catalogue used in this tutorial is composed by 28405 objects. Each object has been measured in 7 optical bands, namely U,B,V,R,I,J,K bands. For each band 3 different parameters, geometric (Kron radius) and photometric (Flux and Magnitudes) were measured, adding up to 21 parameters for each object in the catalogue. Objects are classified as angularly resolved (or galaxies, in the astronomical jargon) and non resolved (stars). Moreover, GOODS (and more in general astronomical surveys) data present a further peculiarity: the majority of the objects are "drop outs", id est they are detected only in some bands and not detected in the others due to either instrumental (different detection limits) or intrinsic (different spectral properties) reasons. Without entering into details we must stress that the characterization of an object as a "dropout" (id est as an object with a strong relative flux difference between two or more spectral regions) is very important from the astronomical point of view since it allows to discriminate among different classes of celestial objects. From our statistical clustering point of view, therefore, the data set contains four classes of objects, namely stars, galaxies, stars which are drop outs and galaxies which are drop outs (at this stage, we do not take into account the number of bands for which an object is a drop out).

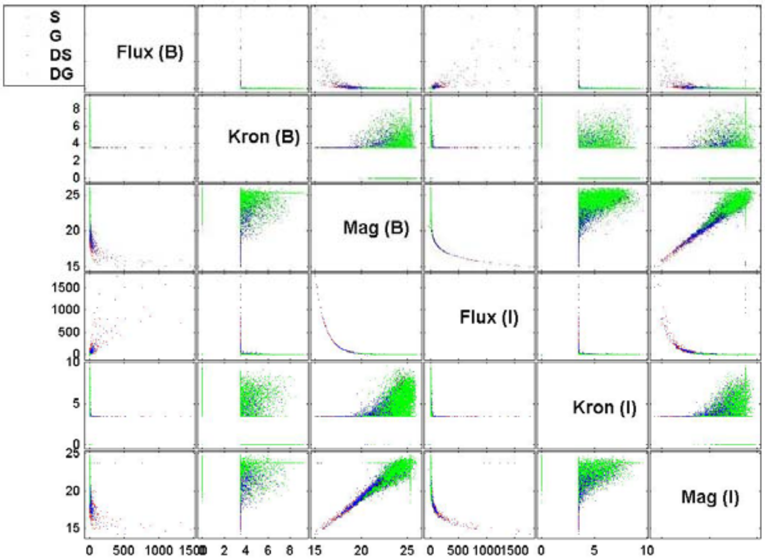
[1] Hubble Space Telescope

[2] Satellite for X-ray Surveys

Traditional Visualization Methods
Scatter Plots

- *Scatter Plot*: simple plot of one variable against another.
- *Scatter Plot Matrix*: matrix of scatter plots showing the relationship between several pairs of variables.
- Useful for determining whether the values of two variables or the relationship between those variables is the same.

Scatter plot: *example*



Traditional Visualization Methods
Scatter Plots

- Scatter plots results less useful:
  - for very high dimensional data
  - the relations between variables are very complex and hard to interpret
  - Relations only between pairs of features

Traditional Visualization Methods

Principal Component Analysis (PCA)

- A classical linear projection method that preserves as much data variance as possible. Fast and easy to compute.

- Suppose that we are trying to map a dataset of vectors $\boldsymbol{x}^n$ for $n = 1,\ldots, N$ in $V = \boldsymbol{R}^D$ to vectors $\boldsymbol{z}^n$ in $U = \boldsymbol{R}^Q$, a subspace of V.
- The quality of the approximation is measured by the residual sum-of-squares error

$$E = \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{t}^n - \mathbf{x}^n\right\|^2 = \frac{1}{2}\sum_{i=Q+1}^{D}\mathbf{u}_i^T\boldsymbol{\Sigma}\mathbf{u}_i$$

where Σ is the covariance matrix of the data.

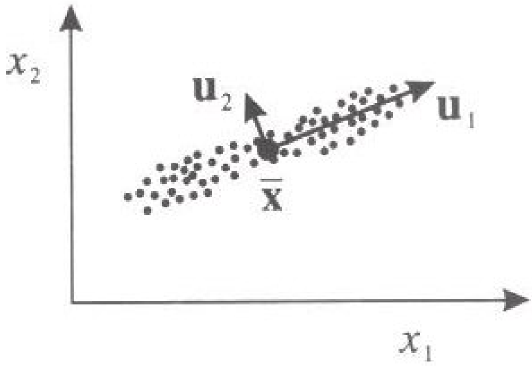
- The minimal error is achieved by projecting the data into the space spanned by the eigenvectors corresponding to the largest Q eigenvalues.

For a comprehensive review please refer to:

Bishop, C. M., **Neural Networks for Pattern Recognition,** Oxford: Clarendon Press, 1995

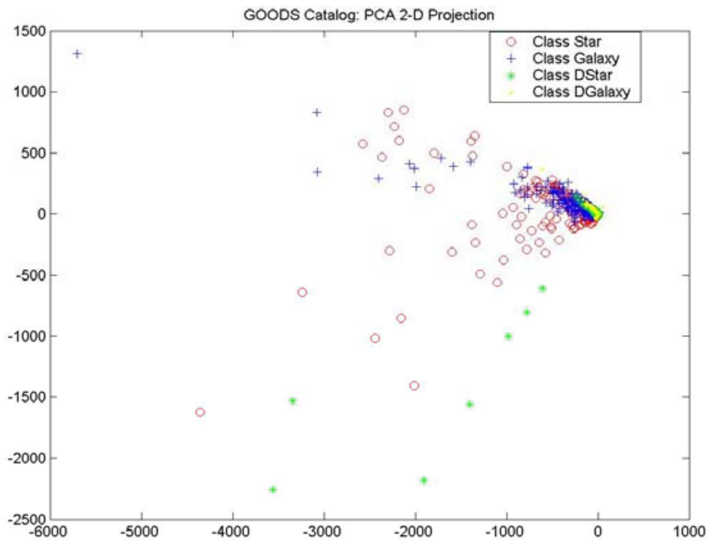
Traditional Visualization Methods

PCA

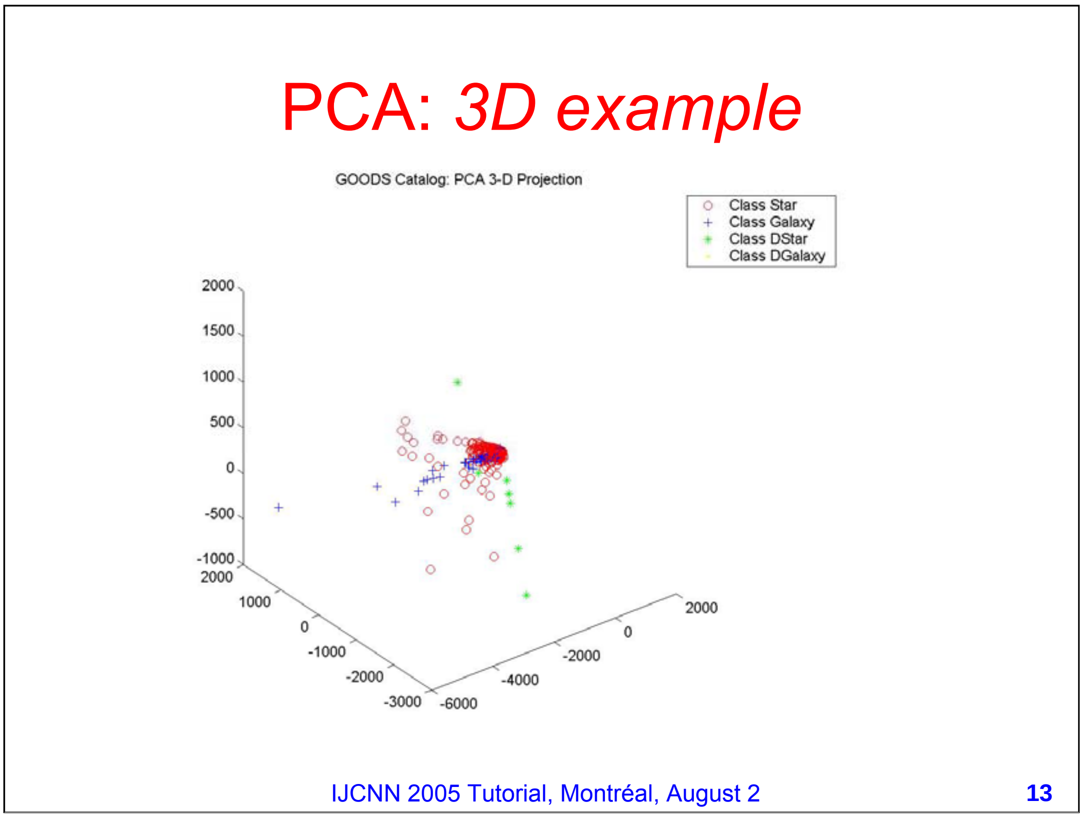


PCA 2D illustration: *In a linear projection down to one dimension, the optimum choice of projection, in sense of minimizing the sum of squares error, is obtained by first subtracting off the mean* $\bar{\mathbf{x}}$ *of the data set, and then projecting the data into the first eigenvector* $\mathbf{u}_1$ *of the covariance matrix.*

PCA: *2D example*



As the figure suggests, high nonlinear complex data can not be effectively characterized by linear PCA and …



… the 3D representation can not help us more than the 2D plots!!!

Traditional Visualization Methods
PCA

- ❑Unable to capture the nonlinear nature of data.
- ❑Inadequate to characterize strong overlapping data.
- ❑Not effective for complex data visualization.

Latent Variable Models (1)

- **Goal**: to express the distribution $p(\boldsymbol{t})$ of the variable $\boldsymbol{t}=(t_1,\ldots,t_D)$ in terms of a smaller number of latent variables $\boldsymbol{x}=(x_1,\ldots,x_Q)$, $Q<D$.
- **How**: by expressing the joint distribution

$$p(\mathbf{t},\mathbf{x}) = p(\mathbf{x})p(\mathbf{t}\mid\mathbf{x}) = p(\mathbf{x})\prod_{d=1}^{D} p(t_d \mid \mathbf{x}) \qquad (1)$$

$p(\boldsymbol{x})$ ≡ marginal distribution of the latent variables

$p(\boldsymbol{t}|\boldsymbol{x})$ ≡ conditional distribution of the data variables given the latent variables

The idea behind latent variable models is to have a sound probabilistic model describing the generative process underlying a set of user data points. This model is expressed in terms of two spaces: the original data space and an auxiliary space, called latent space, which needs to be of lower dimension. This latter issue can be useful exploited for visualization purpose if one chooses a latent space of 2 or at most 3 dimensions. Here we provide a theoretical review of latent variables defining the way the model can be probabilistically defined and giving details about the link between the latent space and the original data space.

A complete review of latent variable models can be found in:

Bishop, C. M., **Latent variable models.** In M. I. Jordan (Ed.), Learning in Graphical Models, pp. 371–403. MIT Press, 1999.

Latent Variable Models (2)

- *p(**t**|**x**)* is expressed in terms of a mapping from latent variables to data variables, so that

$$\mathbf{t} = y(\mathbf{x}, \mathbf{W}) + \mathbf{u}$$

  *y(**x**,**W**)* is a function of the latent variable ***x*** with parameters ***W***; ***u*** is an ***x***-independent noise process.

- If the components of ***u*** are uncorrelated, the conditional distribution for t will factorize as in (1).

Latent Variable Models (3)

- The definition of the model is completed by specifying *y(**x**,**W**)* and *p(**x**)*. *y(**x**,**W**)* determines the type of the latent variable model:
  - A linear **y** implies a linear latent variable model;
  - A nonlinear **y** implies a nonlinear latent variable model.

- By margilizing over the latent variables, we obtain

$$p(\mathbf{t}) = \int p(\mathbf{t} / \mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- The integral is analytically intractable, except for specific forms of the distributions *p(**t**|**x**)* and *p(**x**)*.

Linear latent variable models
Probabilistic PCA (PPCA)

- Classical PCA is made into a density model by using a latent variable approach, derived from factor analysis, in which the data t is generated by a linear combination of a number of hidden variables x:

$$\mathbf{t}=\mathbf{W}\mathbf{x}+\mu+\mathbf{u}$$

where **x** has a zero mean, unit isotropic variance, Gaussain distribution N(0,I), μ is constant and **u** is a **t**-independent noise process

Refer to:

M. E. Tipping, C. M. Bishop, **Probabilistic principal component analysis,** Journal of the Royal Statistical Society, Series B **21**(3), 611–622 , 1999.

Linear latent variable models
PPCA

- The probability model for PPCA is written as a combination of the conditional distribution

$$p(\mathbf{t}/\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left\{ -\frac{\left\| \mathbf{t} - \mathbf{W}\mathbf{x} - \boldsymbol{\mu} \right\|^2}{2\sigma^2} \right\}$$

and the latent variable distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{Q/2}} \exp\left\{ -\frac{\mathbf{x}^T \mathbf{x}}{2} \right\}$$

- By integrating out the latent variable **x**, we obtain the marginal distribution of the input data points, which is also Gaussian: $\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{C})$, with $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}$.

- This model represents the data as consisting of a lower dimensional linear subspace surrounded by equal noise in all directions.

- The parameters of the distribution, **W** and σ can be computed by an iterative maximization of the log-likelihood function through the EM algorithm.

Linear latent variable models
PPCA

- The input data points are plotted, in the latent space, by using the posterior distribution of the latent variable **x** given the observed data **t.** By using the Bayes' theorem, we obtain the distribution

$$p(\mathbf{x}/\mathbf{t}) \sim N(\mathbf{M}^{-1}\mathbf{W}^{T}(\mathbf{t}-\boldsymbol{\mu}), \sigma^{2}\mathbf{M}^{-1}),$$

where $M=\mathbf{W}^{T}\mathbf{W}+\sigma^{2}\mathbf{I}$ (whose dimensions are QxQ).

- In order to map **t** to a single point in the latent space, the mean of the posterior distribution $\mathbf{M}^{-1}\mathbf{W}^{T}(\mathbf{t}-\boldsymbol{\mu})$ is computed.

Linear latent variable models

Mixture of PPCA

- PCA is a rather limited technique since it only defines a linear projection of data.
- An alternative approach is to model a complex nonlinear structure by a collection of local linear models.
- A major advantage of developing a probabilistic formulation of PCA is that we can formalize the idea of a collection of models as a mixture of PPCA:

$$p(\mathbf{t}) = \sum_{i=1}^{M_0} \pi p(\mathbf{t}/i)$$

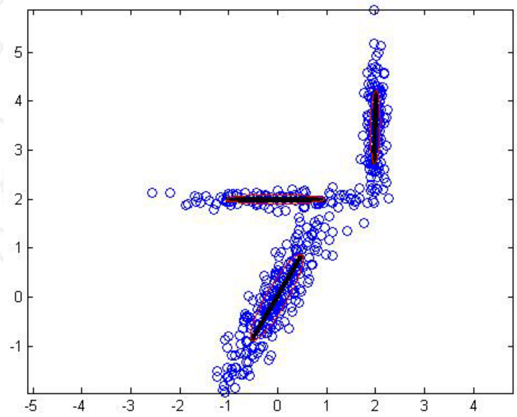
- It is straightforward to obtain an EM algorithm to determine the parameters of the mixture.

Refer to:

M. E. Tipping, C. M. Bishop, **Mixtures of probabilistic principal component analyzers,** Neural Computation **11**(2), 443–482, 1999.

Linear latent variable models
Mixture of PPCA

- However, the mixture of PPCA model is appropriate when the data is approximately piece-wise linear.



Global nonlinear models

Self-Organizing Maps (SOM)

SOM is based on an unsupervised competitive learning (training is entirely data-driven and the neurons compete with each other).

- ❑ A SOM is composed by neurons located on a regular 1 or 2-dimensional grid.
- ❑ Each neuron *i* of the SOM is represented by a n-dimensional weight or reference vector :

$$m_i = [m_{i1}, \dots, m_{in}]^T \qquad n = \textit{dimension} \text{ of input vectors}$$

- ❑ Neurons are connected to adjacent ones by a neighbourhood relation dictating the topology of the map.

IJCNN 2005 Tutorial, Montréal, August 2 **23**

PCA, PPCA and mixture of PPCA are appropriate when the data is linear or approximately piece-wise linear. An alternative approach is to use global nonlinear methods: Self Organizing Maps (SOM), a neural network algorithm based on a competitive learning which summarizes a set of data vectors in a high-dimensional space by a set of reference vectors organized on a lower dimensional sheet (usually two dimensional). SOM has been used for a wide variety of applications thanks to its simplicity and for its several plotting options.

For theoretical details refer to:

S. Kaski, **Data Exploration Using Self Organizing Maps**, PhD Thesis, Helsinki Institute of Technology, 1997.

T. Kohonen, **Self Organizing Maps**, Springer, Berlin, Heidelberg, 1995.

J. Vesanto, **SOM-Based Data Visualization Methods,** Intelligent Data Analysis Journal, 1999.

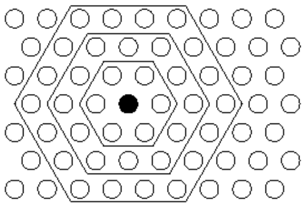
For details concerning with application to astrophysical data, refer to:

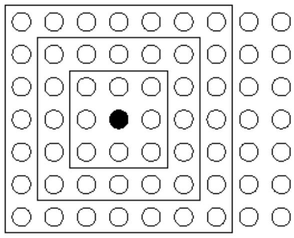
R. Tagliaferri R., G. Longo, A. Staiano A. et al., **Neural Networks in Astronomy**, in Neural Networks. Special Issue on Neural networks for analysis of complex scientific data: Astronomy and Geosciences, R. Tagliaferri, G. Longo, D'Argenio B. (Eds.), vol. 16 (3- 4), 2003.

R. Tagliaferri R., G. Longo, A. Staiano et al., **Applications of Neural Networks in Astronomy and Astroparticle Physics**, invited review on "Recent Research developments in Astronomy and Astrophysics", 2 (2005), pp.27-58, by Research Signpost.

Global nonlinear models

SOM





in the 2D dimensional case, the neurons of the map can be arranged either on a rectangular or a hexagonal lattice.

Training: based on Competitive Learning

(not only the most similar prototype vector, but also its neighbors on the map are moved towards the data vector).

In each training step, one sample vector **t**, from the input data set, is chosen and a similarity measure is calculated between it and all the weight vectors of the map.

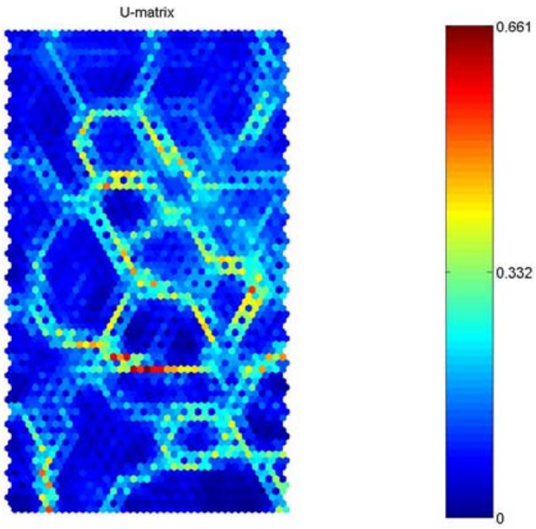
The Best-Matching Unit (BMU) is the unit whose weight vector has the greatest similarity with the input sample **t**.

Global nonlinear models
SOM

U-Matrix (Unified distance matrix)

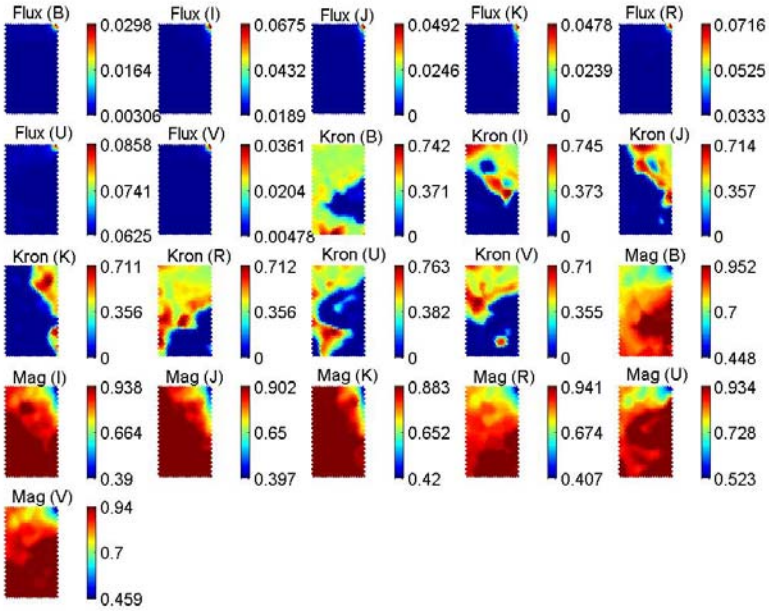
Visualizes the clustering structures of the SOM as distances (in the assumed metric) between neighboring map units, thus high values of the U-matrix indicate a cluster border, uniform areas of low values indicate clusters themselves.

SOM: U-Matrix



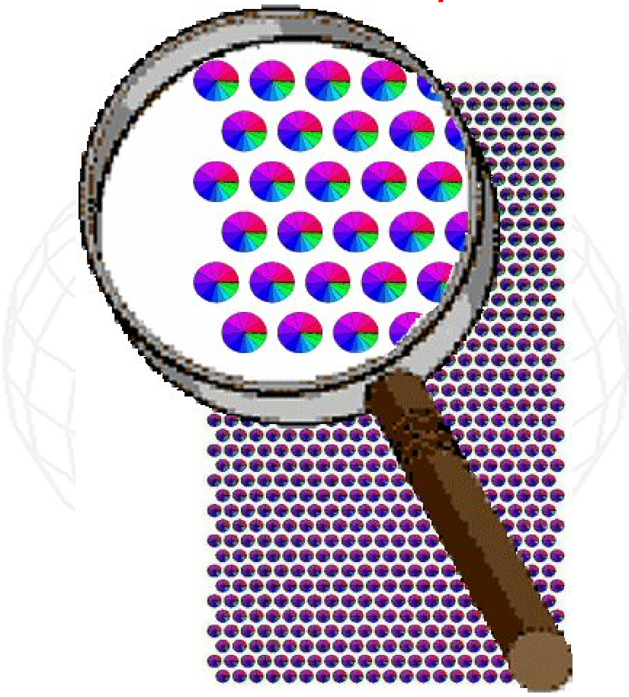
- Regions of low values (blue color) represent clusters themselves
- Regions of high values (red color) represent cluster borders

SOM: Parameter Analysis



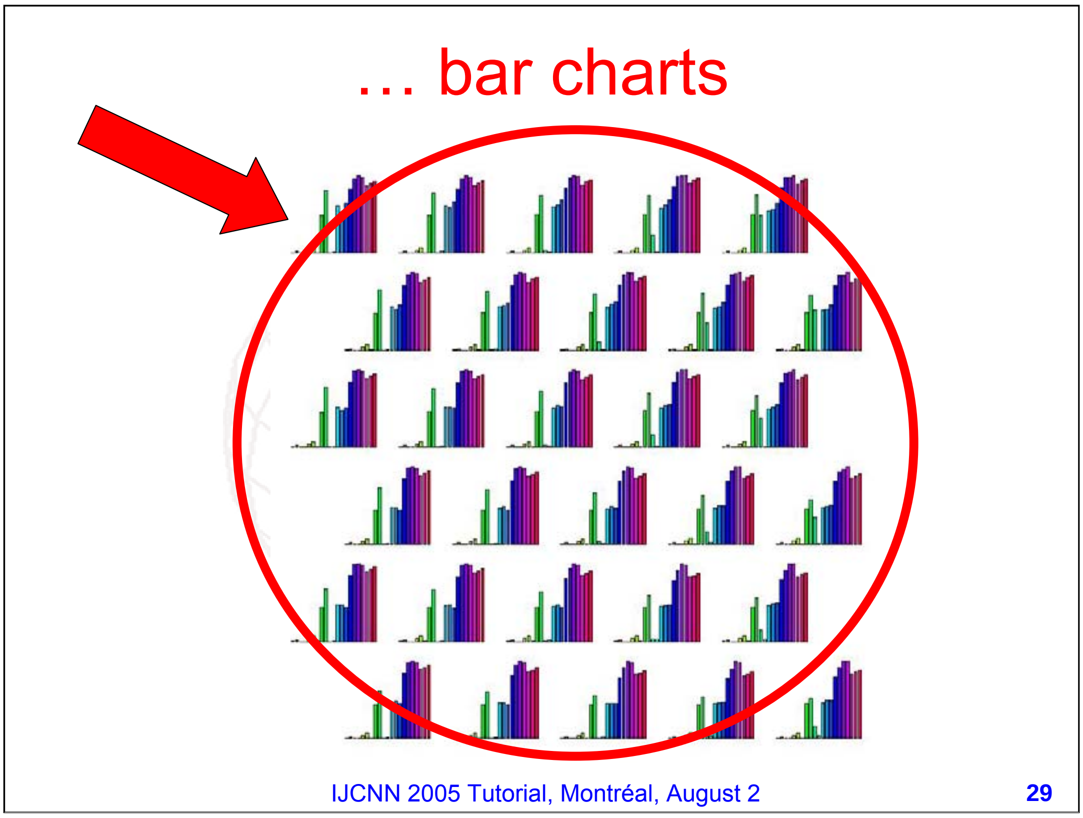
For each input parameter the corresponding map structure is computed. In this way the relation between the input parameters can be analyzed. If one or more input parameters lead to the same map structure this could mean that the parameters are redundant and so some of them could be removed.

SOM *parameter influence: pie charts and …*



IJCNN 2005 Tutorial, Montréal, August 2 **28**

This graphical representation allows to derive the influence of each input parameter on each neuron of the map…



…the same kind of graphical representation in another fashion. These kind of visualizations allow to derive the importance of each parameter in order to characterize the input data points. Eventually one could exploits this knowledge for parameter selection.

Global nonlinear models
SOM

Advantages

- The SOM algorithm is quick in convergence.
- It is good in pre-analysis.
- In many problems it is good enough.

Limitations

- The SOM algorithm is not derived by optimizing an objective function.
- SOM does not define a density model.
- Neighbourhood preservation is not guaranteed by the SOM procedure.

Although SOM provides easy of computation and powerful visualizations it, indeed, does not define any density model and suffers of other drawbacks which can be overcame employing nonlinear latent variable models…

Nonlinear latent variable models
Generative Topographic Mapping (GTM)

- GTM is a latent variable model with a non-linear function *y*, mapping a (usually two dimensional) latent space Q to the data space *D*. This is a generative probabilistic model.
- For the purpose of data visualization, the Bayes' theorem is used to invert the transformation *y*.
- This model assumes that the data lies close to a two dimensional manifold; however, this is likely to be a too simple model for interesting data.

Refer to:

C. M. Bishop, M. Svensèn, C. K. I. Williams, **GTM: the Generative Topographic Mapping,** Neural Computation **10**(1), 215–234, 1998.

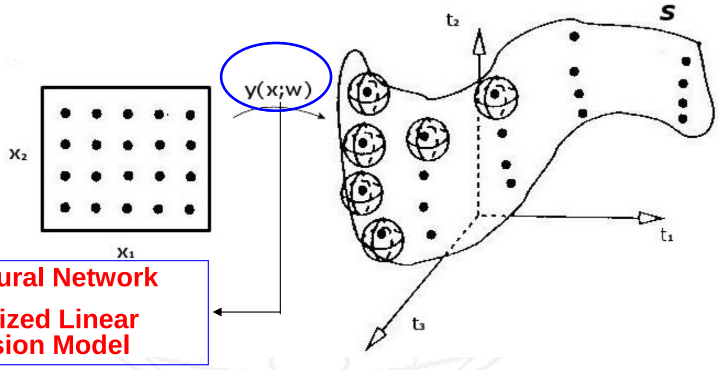
For details concerning with application to astrophysical data, refer to:

R. Tagliaferri R., G. Longo, A. Staiano A. et al., **Neural Networks in Astronomy**, in Neural Networks. Special Issue on Neural networks for analysis of complex scientific data: Astronomy and Geosciences, R. Tagliaferri, G. Longo, D'Argenio B. (Eds.), vol. 16 (3- 4), 2003.

R. Tagliaferri R., G. Longo, A. Staiano et al., **Applications of Neural Networks in Astronomy and Astroparticle Physics**, invited review on "Recent Research developments in Astronomy and Astrophysics", 2 (2005), pp.27-58, by Research Signpost.

Nonlinear latent variable models
GTM

Goal: to express the distribution $p(\mathbf{t})$ of the variable $\mathbf{t}=(t1,\ldots,tD)$, in terms of a smaller number of latent variables $\mathbf{x}=(x1,\ldots,xQ)$, $Q<D$. The link between the latent and data spaces is obtained by the nonlinear function $\mathbf{y}(\mathbf{x},\mathbf{w})$.



- **RBF Neural Network**
- **Generalized Linear Regression Model**

The data is modeled as a constrained mixture of Gaussians with **unoriented COVARIANCE**. The latent variable model can be trained using an EM algorithm that is a generalization of the standard EM for (unconstrained) Gaussian mixtures.

Nonlinear latent variable models
GTM

- Defining a probability distribution over the latent space, *p(**x**)*, will induce a corresponding probability distribution in the data space:

$$p(\mathbf{t}\,|\,\mathbf{x},\mathbf{W},\beta)=\left(\frac{\beta}{2\pi}\right)^{D/2}\exp\left\{-\frac{\beta}{2}\sum_{d=1}^{D}\left(t_d-y_d(\mathbf{x},\mathbf{W})\right)^2\right\}$$

  - **t** point in data space
  - β^{-1} noise variance

Nonlinear latent variable models
GTM

- By integrating out the latent variable, we get

$$p(\mathbf{t} \mid \mathbf{W}, \beta) = \int p(\mathbf{t} \mid \mathbf{x}, \mathbf{W}, \beta) p(\mathbf{x}) d\mathbf{x},$$

which is intractable, but choosing *p(**x**)* as a set of M equally weighted delta functions on a regular grid, i.e.

$$p(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \delta(\mathbf{x} - \mathbf{x}_m),$$

the integral turns into a sum

$$p(\mathbf{t} \mid \mathbf{W}, \beta) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{t} \mid \mathbf{x}_m, \mathbf{W}, \beta). \qquad (2)$$

Nonlinear latent variable models
GTM

- Equation (2) defines a constrained mixture of Gaussians in which:
  - the centers of the mixture components can not move independently of each other;
  - depends on the mapping $y(\mathbf{x};\mathbf{W})$;
  - all components of the mixture share the same variance, and the mixing coefficients are all fixed to $1/M$.

Nonlinear latent variable models
GTM

GTM: topographic ordering

- Provided the mapping function $y(\mathbf{x};\mathbf{W})$ is smooth and continuous, any two points $\mathbf{x}_A$ and $\mathbf{x}_B$, which are close in the latent space, will map to points $y(\mathbf{x}_A;\mathbf{W})$ and $y(\mathbf{x}_B;\mathbf{W})$ which are close in the data space.

Nonlinear latent variable models
GTM

GTM visualization (1)

- A trained GTM defines a probability distribution $p(\mathbf{t}|\mathbf{x}_m)$, $m=1,\ldots,M$.
- We can compute the corresponding posterior distribution in latent space for any given point in data space **t**, as

$$p(\mathbf{x}_m|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{x}_m,\mathbf{W},\beta)\,p(\mathbf{x}_m)}{\sum_{m'=1}^{M} p(\mathbf{t}|\mathbf{x}_{m'},\mathbf{W},\beta)\,p(\mathbf{x}_{m'})}$$

Nonlinear latent variable models
GTM

GTM visualization (2)

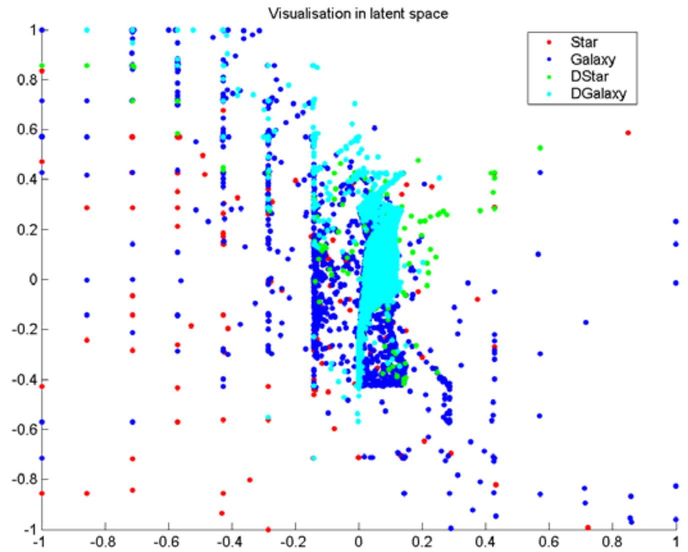
- To visualize whole sets of data, two possibilities are, for each data point $\mathbf{t}_n$, to plot:
  - The **mode** of the posterior distribution in latent space,

$$\mathbf{x}_n^{md} = \arg\max_{\mathbf{x}_m} p(\mathbf{x}_m \mid \mathbf{t}_n),$$ *posterior-mode projection*

  - The **mean** of the posterior distribution

$$\mathbf{x}_n^{mean} = \sum_{m=1}^{M} \mathbf{x}_m p(\mathbf{x}_m \mid \mathbf{t}_n)$$ *posterior-mean projection*

GTM: latent space visualization



IJCNN 2005 Tutorial, Montréal, August 2 **39**

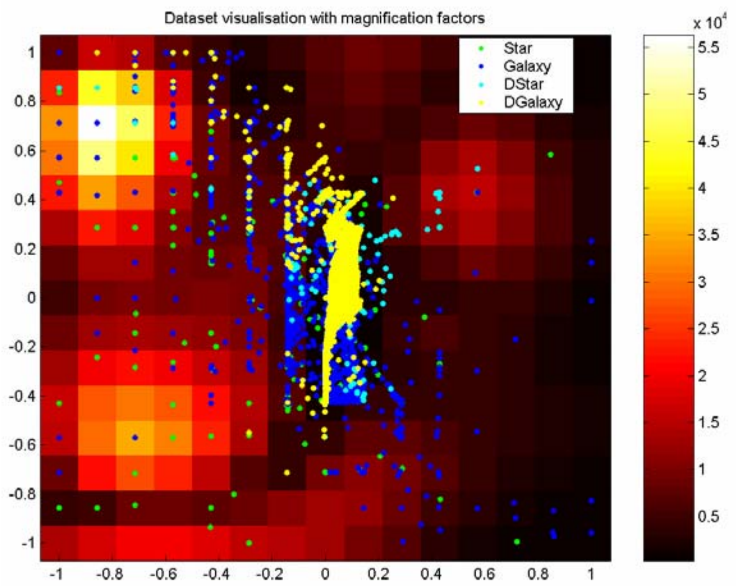
Two-dimension latent space with input data point projections.

Nonlinear latent variable models
GTM

Magnification Factors

- We can measure the stretch in the manifold using magnification factors, and this can be used to detect the gaps between data clusters.
- More stretched areas indicate gaps between clusters, conversely less stretched areas correspond to regions of high density (clusters).

GTM: Magnification Factors



High stretch

Low stretch

Nonlinear latent variable models

Probabilistic Principal Surfaces (PPS)

- PPS=GTM+oriented covariance

$$\Sigma(\mathbf{x}) = \frac{\alpha}{\beta}\sum_{q=1}^{Q}\mathbf{e}_q(\mathbf{x})\mathbf{e}_q^T(\mathbf{x}) + \frac{(D-\alpha Q)}{\beta(D-Q)}\sum_{d=Q+1}^{D}\mathbf{e}_d(\mathbf{x})\mathbf{e}_d^T(\mathbf{x})$$

$$0<\alpha<D/Q$$

- $\{\mathbf{e}_q(\mathbf{x})\}_{q=1,\ldots,Q}$ set of orthonormal vectors tangential to the manifold at $y(\mathbf{x};\mathbf{W})$
- $\{\mathbf{e}_d(\mathbf{x})\}_{d=Q+1,\ldots,D}$ set of orthonormal vectors orthogonal to the manifold at $y(\mathbf{x};\mathbf{W})$

IJCNN 2005 Tutorial, Montréal, August 2 42

Probabilistic Principal Surfaces are a non linear latent variable model with very powerful visualization and classification capabilities which seem capable to overcome most of the shortcomings of other neural tools such as SOM, GTM, etc. PPS generalizes the GTM model by building a unified model and shares the same formulation as the GTM, except for an oriented covariance structure for the Gaussian mixture in the data space. This means that data points projecting near a principal surface node (i.e., a Gaussian center of the mixture) have higher influences on that node than points projecting far away from it. Particularly interesting is the case in which the latent space is 3 dimensional which allows to project the patters on a spherical manifold (of unit radius) which turns out to be optimal when dealing with sparse data.

For theoretical details refer to:

K. Chang, **Nonlinear Dimensionality Reduction Using Probabilistic Principal Surfaces,** PhD thesis, The University of Texas at Austin, USA, 2000

K. Chang, J. Ghosh, **A unified model for probabilistic principal surfaces**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23, (1), 2001

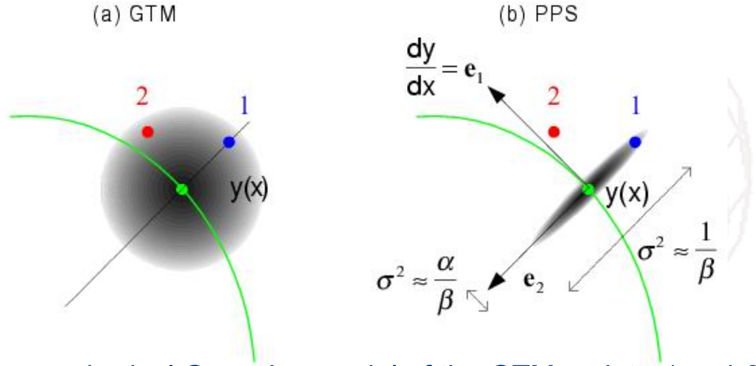
For details concerning application to astrophysics and both visualization enhancement and classification refer to:

A. Staiano, **Unsupervised Neural Networks for the Extraction of Scientific Information from Astronomical Data**, PhD thesis, Università di Salerno, Italy, 2003.

A. Staiano, R. Tagliaferri, G. Longo, P. Benvenuti, **Committee of Spherical Probabilistic Principal Surfaces**, Proceedings of IJCNN 2004.

Nonlinear latent variable models
PPS

Why oriented covariance?



Under a spherical Gaussian model of the GTM, points *1* and *2* have equal influence on the center node *y(**x**)* (a) PPS have an oriented covariance matrix so point *1* is probabilistically closer to the center node *y(**x**)* than point *2* (b)

The figure is taken from K. Chang, **Nonlinear Dimensionality Reduction Using Probabilistic Principal Surfaces,** PhD thesis, The University of Texas at Austin, USA, 2000.

Nonlinear latent variable models
PPS

Unified model

- The unified PPS model reduces to GTM for α=1 and to the manifold –aligned GTM for α>1

$$\Sigma(\mathbf{x}) = \begin{cases} 0<\alpha<1 & \perp \text{ to the manifold} \\ \alpha=1 & I_D \text{ or spherical} \\ 1<\alpha<D/Q & \parallel \text{ to the manifold} \end{cases}$$

Nonlinear latent variable models
PPS

Training algorithm

- Based on a generalized EM for parameters **W**, α, β,
- Computationally more complex than GTM, but ...
- Faster convergence!

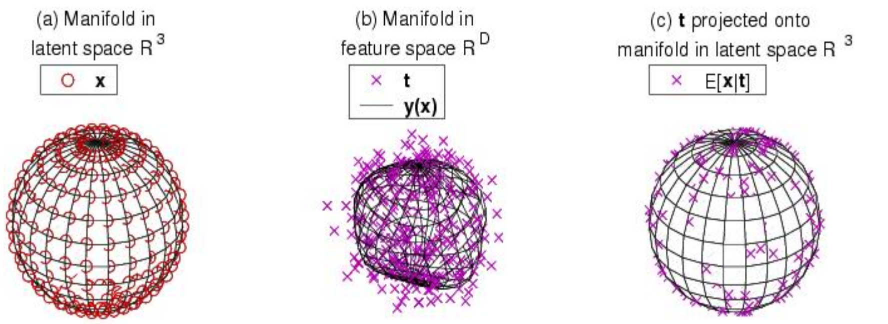
Nonlinear latent variable models
PPS

Spherical PPS

- Manifold composed by nodes regularly arranged on the surface of a sphere in *3D* space (*Q=3*)
- Use manifold as a classification reference template
- Use projections for visualizations

Nonlinear latent variable models
PPS

Spherical PPS: example



(a) The spherical manifold in R^3 latent space.
(b) The spherical manifold in R^3 data space.
(c) Projection of data point **t** onto the latent spherical manifold.

The figure is taken from K. Chang, **Nonlinear Dimensionality Reduction Using Probabilistic Principal Surfaces,** PhD thesis, The University of Texas at Austin, USA, 2000.

Nonlinear latent variable models
PPS

Spherical PPS visualization (1)

- ❑ A spherical manifold is first fitted to the data.
- ❑ The data is projected into the manifold in R^3.
- ❑ The projected locations are plotted into R^3 as points on a sphere.

Nonlinear latent variable models
PPS

Spherical PPS visualization (2)

- *Probabilistic Projection*: the projected latent coordinate is computed as a linear combination of all latent nodes weighted by the responsibility matrix,

$$\mathbf{x}_n^{proj} \equiv \langle \mathbf{x} \mid \mathbf{t}_n \rangle = \int \mathbf{x} p(\mathbf{x} \mid \mathbf{t}_n) dx = \sum_{m=1}^{M} r_{mn} \mathbf{x}_m$$

- Since $||\mathbf{x}_m||=1$ for $m=1,\ldots,M$ and $\Sigma_m r_{mn}=1$ for $n=1,\ldots,N$, all projections lie within the sphere, i.e. $||\mathbf{x}_m|| \leq 1$ *and*

- r_{mn} is the responsibility of latent variable $\boldsymbol{x}_m$ with respect to data point $\boldsymbol{t}_n$

$$p(\mathbf{x}_m \mid \mathbf{t}_n) = \frac{p(\mathbf{t}_n \mid \mathbf{x}_m, \mathbf{W}, \beta)\, p(\mathbf{x}_m)}{\sum_{m'=1}^{M} p(\mathbf{t}_n \mid \mathbf{x}_{m'}, \mathbf{W}, \beta) p(\mathbf{x}_{m'})}$$

Nonlinear latent variable models
PPS

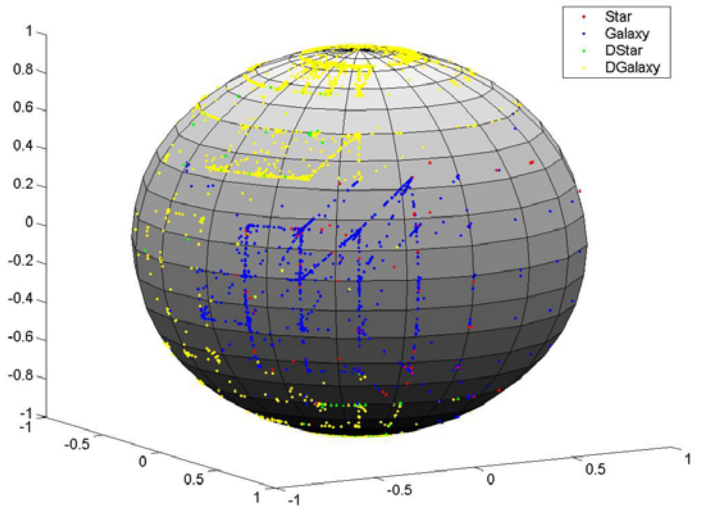
Spherical PPS: graphical user interface

- We built a graphical user interface which extends the visualization possibilities offered by PPS:
  - Visualization on the sphere surface;
  - Possibility to interact with points on the sphere;
  - Visualization of the data probability density function on the sphere;
  - Cluster determination and visualization.

Refer to:

A. Staiano, **Unsupervised Neural Networks for the Extraction of Scientific Information from Astronomical Data**, PhD thesis, Università di Salerno, Italy, 2003.

PPS
Latent Projections



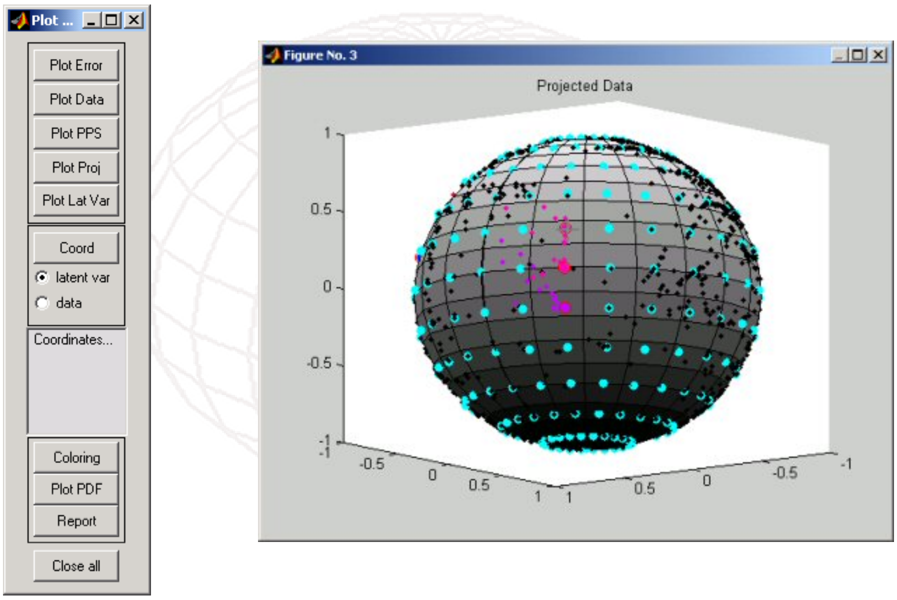
IJCNN 2005 Tutorial, Montréal, August 2 **51**

A latent spherical manifold with data points probabilistic projections.

PPS
GUI: User-Data Interaction

- The user is allowed to:
  - Visualize latent variables on the sphere;
  - Select a chosen latent variable and color that variable and all the data points for which it is responsible and vice versa;
  - For each data point compute its coordinates, confidence level and the index of the corresponding source in the catalog space;
  - Create a report of all the information deriving from the previous operations.

PPS
GUI: User-Data Interaction



IJCNN 2005 Tutorial, Montréal, August 2 **53**

Latent spherical manifold with data points projections (black dots), and latent variables (cyan bigger dots) superimposed. The user is allowed to:

1)select a data point and color the latent variable which is responsible for it and the remaining points for which the same latent variable is responsible.

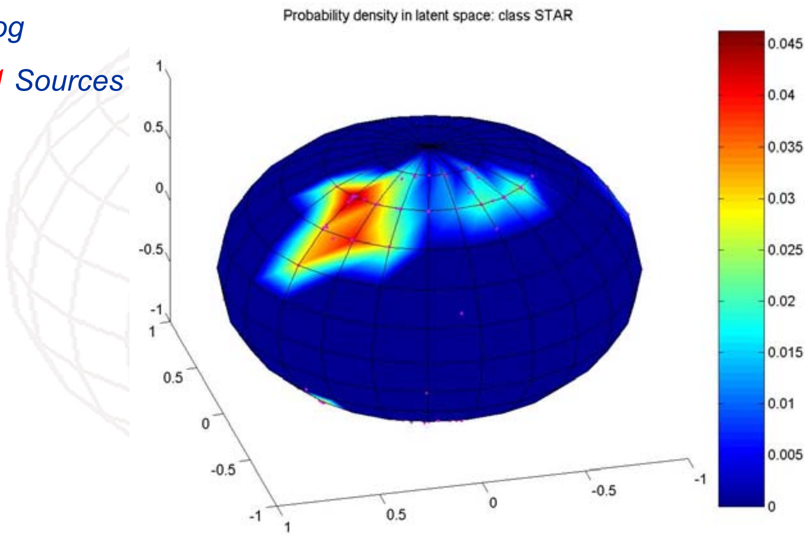
2)select a latent variable and color the latent variable and all the points for which it is responsible.

All the points belonging to the same latent variable share some similarity property.

PPS
Density in latent space

GOODS Catalog

Class Star: 421 Sources

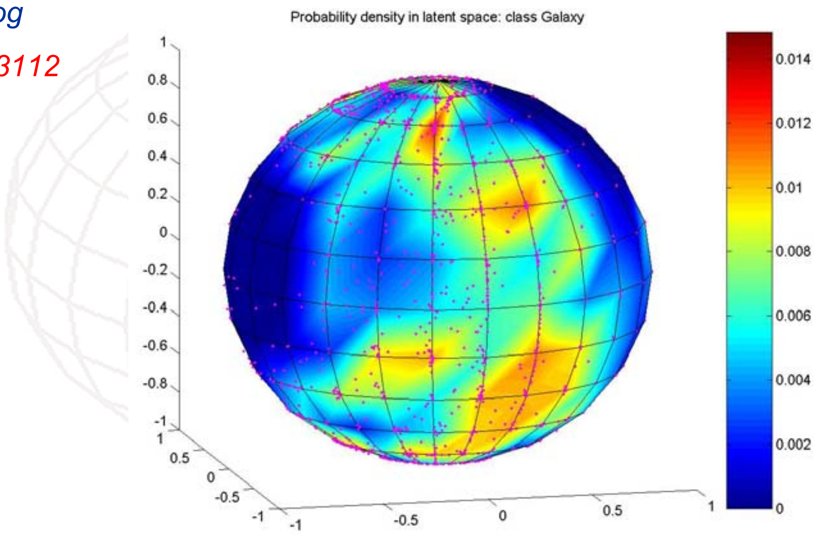


Latent spherical manifold with probability density function superimposed. The red areas are zones with higher probabilities.

PPS
Density in latent space

GOODS Catalog

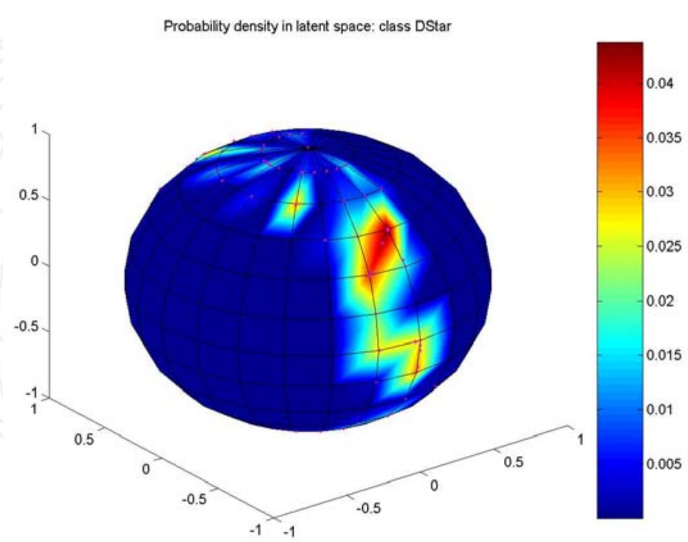
Class Galaxy: 3112 Sources



PPS
Density in latent space

GOODS Catalog

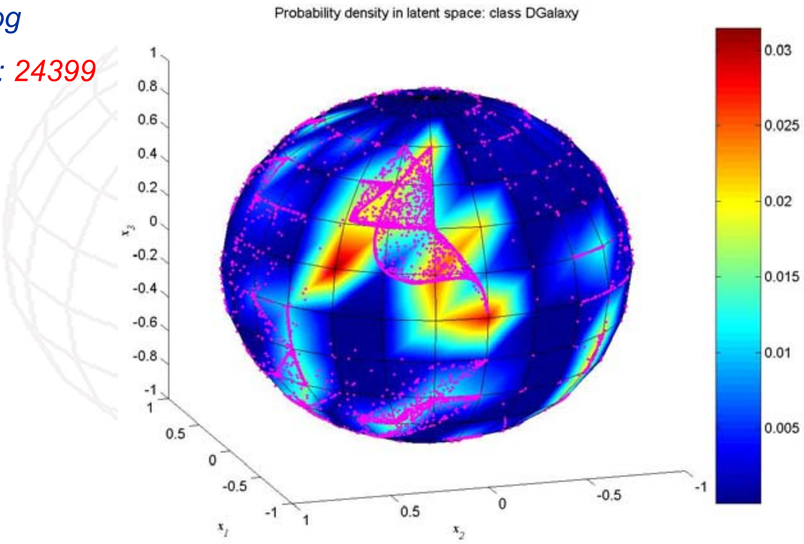
Class DStar: 473 Sources



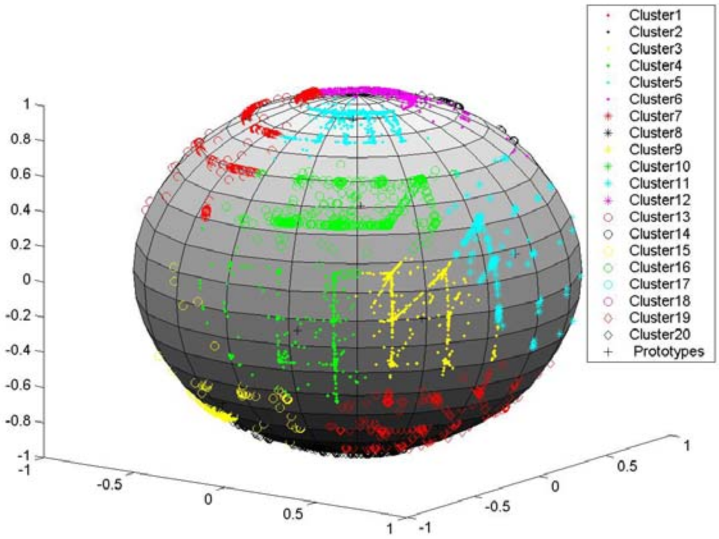
PPS
Density in latent space

GOODS Catalog

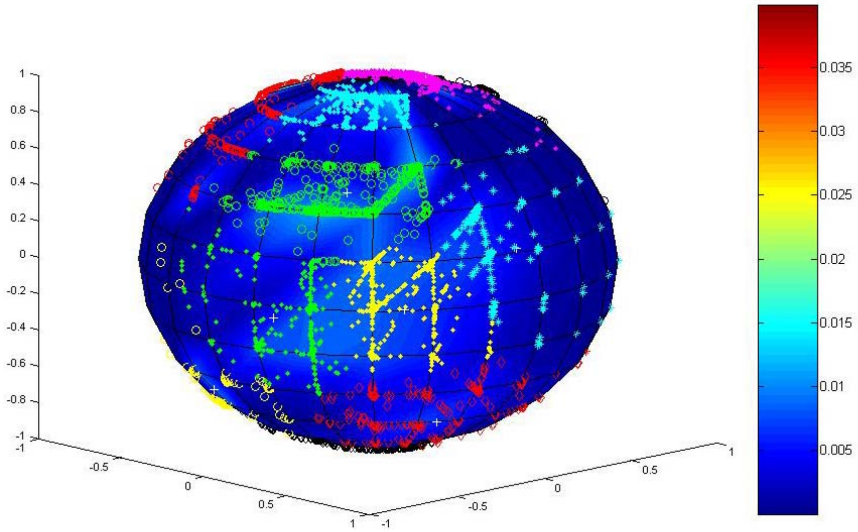
Class DGalaxy: 24399 S.



PPS
Clusters computation and visualization



PPS
Clusters computation and visualization



Hierarchies of latent variable models
Overview

- Most of the visualization algorithms described so far, project the data onto a two-dimensional visualization space...
- But a single two-dimensional projection, even if nonlinear, may not be sufficient to capture all of the interesting aspects of the data.
- This intuition is behind the hierarchical development of a linear latent variable model, namely mixture of PPCA, and the nonlinear counterpart based on the GTM.

Hierarchies of latent variable models

- When exploring a data set through low-dimensional projections in a hierarchical way, one first constructs a top-level plot and then focuses the attention on local region of interest by recursively building the corresponding sub-projections.

- The regions of interest are chosen interactively by the user which clicks on those areas considered as particularly complex and thus hiding potential substructure not visible at a first glance.

- All the models in the hierarchy are organized in a tree and need to be a consistent probabilistic model of the data.

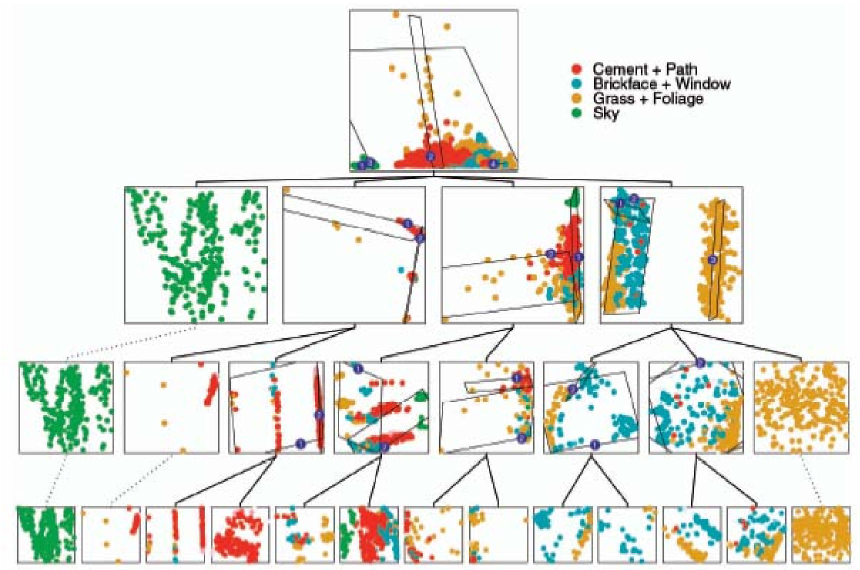
Hierarchies of latent variable models

- From a technical point of view, it is necessary to derive the hierarchical version of the EM algorithm in order to make the hierarchy of sub-models a consistent probabilistic model as a whole.

- A further appealing possibility offered by the hierarchical versions of the latent variable models, is that if the base model provides special kind of plots then all the visualization power of these plots can be exploited at any level of the hierarchy.

Hierarchies of latent variable models

Example: hierarchical linear model (PPCA)

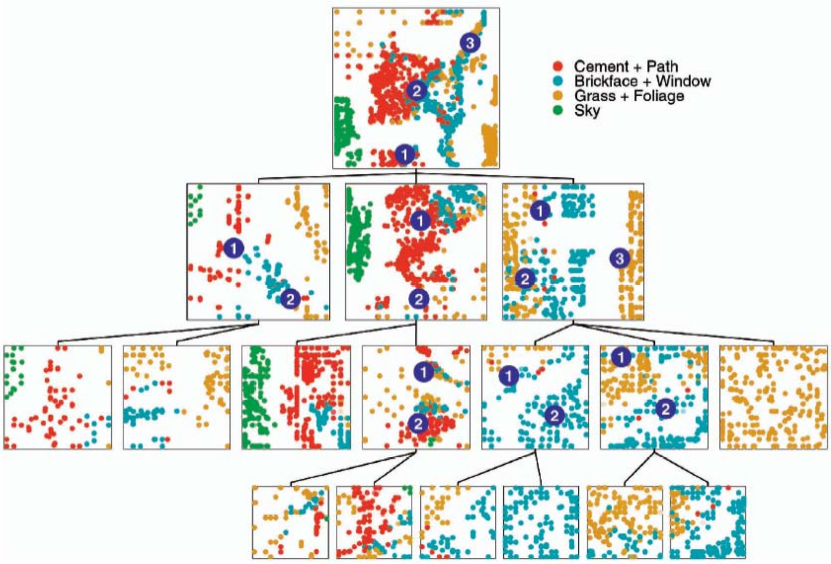


Refer to:

C. M. Bishop, M. E. Tipping, **A hierarchical latent variable model for data visualization**, IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3), 281–293, 1998.

Hierarchies of latent variable models

Example: hierarchical nonlinear model (HGTM)



Refer to:

P.Tino, I. Nabney, **Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way,** Pattern Analysis and Machine Intelligence, IEEE Transactions on ,Volume: 24 , Issue: 5 , May 2002, Pages:639 - 656

Hierarchical models: linear vs nonlinear

- Allowing for non linearity in the projection manifolds lead to create more detailed and parsimonious visualization plots.
- While PCA can introduce, in the visualization plot, only global stretching along the principal axes, the nonlinear projection manifold of GTM can locally stretch and fold in the data space.
- This gives the possibility to the hierarchical GTM to make full use of the latent space when describing the local distributions of points.
- On the contrary, the PPCA-based linear hierarchy, provides plots often characterized by dense isolated clusters.

Hierarchies of latent variable models
PPS

- Obviously the hierarchical extension may be applied to PPS as well.
- The power and the variety of PPS visualization plots can be fully exploited by developing a hierarchical PPS model in the HGTM fashion.
- We are currently implementing the HPPS model so the work is still in progress…
- …however, we provided a second hierarchical view of PPS…

PPS: hierarchical agglomeration

- PPS can be used in conjunction with a type of hierarchical agglomerative clustering for the construction of a powerful visualization-clustering tool.

- The idea is to start with the probability density function computed by PPS and then applying a hierarchical clustering which merges the Gaussian components of the mixture model.

- This task could be accomplished by any clustering algorithm (eventually even not hierarchical as k-means), but…

  … we developed a special kind of clustering algorithm mainly able to find autonomously the correct number of clusters.

Neg-entropy based Clustering (NEC)

- Starting from the PPS density function its Gaussian components can be clustered using information based on entropy.
- Several approaches have been introduced based on the *hypothesis test* or *Kullback-Leibler* divergence.
- We introduced an approach based on the *Neg-entropy.*
- The algorithm permits to agglomerate automatically the clusters using non-Gaussianity information.

IJCNN 2005 Tutorial, Montréal, August 2 **68**

Refer to:

A. Ciaramella, A. Staiano, R. Tagliaferri, G. Longo, **NEC: an Hierarchical Agglomerative Clustering based on Fischer and Negentropy Information**, Proceedings of WIRN 2005, (LNCS Springer volume, to appear)

NEC

- Neg-entropy is based on the information-theoretic quantity of differential entropy.

- It is used to obtain a measure of non-Gaussianity that is zero for a Gaussian variable:

$$J(\mathbf{t}) = H(\mathbf{t}_{\mathbf{Gauss}}) - H(\mathbf{t})$$

  where $\mathbf{t}_{\mathbf{Gauss}}$ is a Gaussian random variable of the same correlation (and covariance) matrix as $\mathbf{t}$.

- Neg-entropy is always non-negative and it is zero if and only if $\mathbf{t}$ has a Gaussian distribution

NEC: approximate neg-entropy

- The classical method to approximate neg-entropy is using high-order cumulants

$$J(\mathbf{t}) \approx \frac{1}{12} E\{\mathbf{t}^3\}^2 + \frac{1}{4}\mathrm{kurt}(\mathbf{t})^2$$

where **kurt** is the kurtosis.

- A different and more robust approximation of the neg-entropy is

$$J(\mathbf{t}) \propto \left[E\{G(\mathbf{t})\} - E\{G(\upsilon)\}\right]^2$$

where υ is a standardized Gaussian variable and **t** has zero mean and unit variance.

- Choosing a *G* that does not grow fast, one obtains more robust estimators. The following choices of *G* have proved very useful:

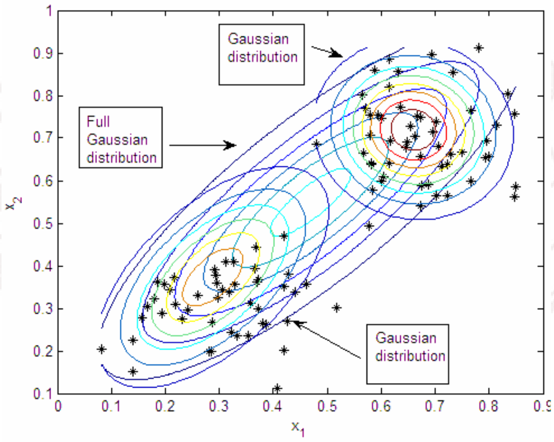
$$G^1 = \frac{1}{a}\log\cosh(a\mathbf{t}) \quad - \quad G^2 = \frac{1}{4}\mathbf{t}^4 \quad - \quad G^3 = -\frac{1}{a}e^{-a\frac{\mathbf{t}^2}{2}}$$

NEC: algorithm

- Starts from M clusters (one for each PPS mixture component);
- Agglomerates two components, i and j:
  - if the new cluster candidate Neg-entropy value is less of a fixed threshold
    - then i U j replaces clusters i and j. i U j becomes cluster i and j=j+1;
    - else j=j+1
  - the steps are repeated until all the components are processed
- Ends with the final number of clusters.

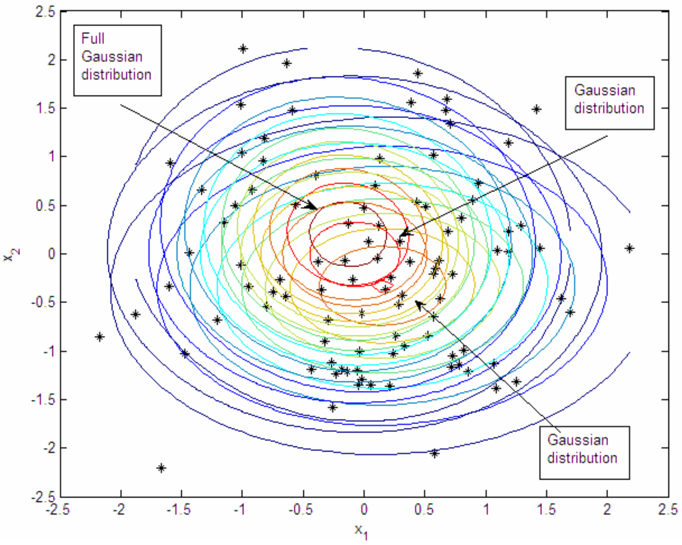
NEC: Gaussians not merged by the algorithm

NegE=750



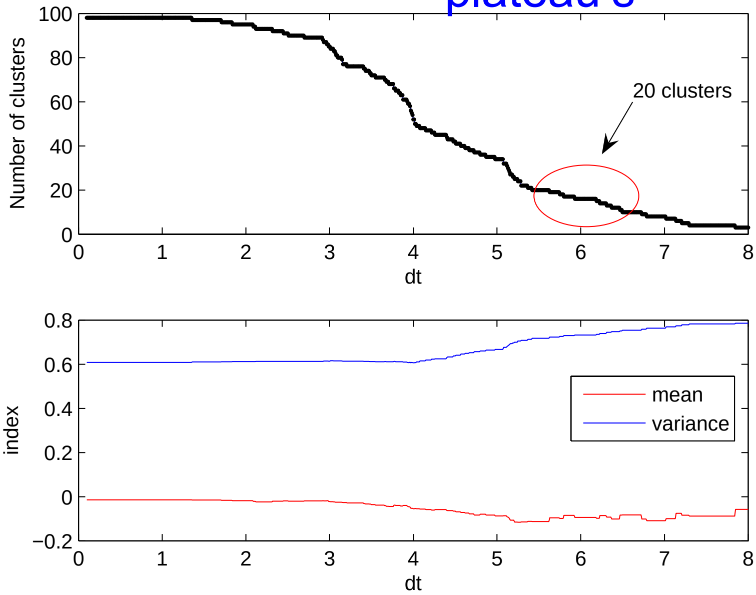
NEC: two merged Gaussian distributions

NegE=4



NEC: search for structures in the data

plateau's



Using the threshold we can study the structure of the data focusing our attention on the stable points (plateau's)

Number of clusters varying the threshold dt; clusters variance and mean varying the threshold dt.

Obviously, the threshold used in the algorithm determines the clustering results one obtains. An interesting approach we can use here, however, is to exploit an interval of values for the threshold in order to study the substructures hidden in the data. The idea is to have a plot of the threshold values vs the number of corresponding clusters that the algorithm returns and to focus the attention on those threshold values which correspond to plateau's in the plot: these, in fact, reveal a substructure which is a stable configuration of the clustering structure. These approach is especially useful when the user has no a priori information at all about the data under investigation.

Case Study
Yeast Gene Microarray Data

- P. T. Spellman et al., **Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization,** Molecular Biology of the Cell, Vol. 9, 3273-3297, December, 1998
- 6178 genes each one subject to 6 experiments:
  - cln3
  - clb2
  - alpha factor arrest
  - cdc15 temperature-sensitive mutant
  - cdc28
  - elutriation
- 73 features associate to each gene. After a preprocessing phase the features were reduced to 32.

IJCNN 2005 Tutorial, Montréal, August 2 **75**

Refer to:

R. Amato, A. Ciaramella, A. Staiano, R. Tagliaferri, G. Longo, et al., **NEC for Gene Expression Analysis**, Second International Meeting on Computational Intelligence Methods For Bioinformatics and Biostatistics, Crema, Italy, 2005

A. Staiano, A. Ciaramella, G. Raiconi, R. Tagliaferri et al., **Data Visualization Methodologies for Data Mining Systems in Bioinformatics**, Proceedings of IJCNN 2005, special session on Neural Networks Applications in Bioinformatics, Montreal (Canada), 2005

A. Staiano, L. De Vinco, R. Tagliaferri, G. Longo et al., **Probabilistic Principal Surfaces for Yeast Gene Microarray Data Mining**, Proceedings of the Fourth IEEE International Conference on Data Mining: ICDM 2004, pp. 202-209, Brighton, UK, 2004

A. Staiano, R. Tagliaferri, G. Longo et al., **Novel Techniques for Microarray Data Analysis: Probabilistic Principal Surfaces and Competitive Evolution on Data**, Journal of Computational and Theoretical Nanoscience, Special Issue on Computational Intelligence for Molecular Biology and Bioinformatics, in print.

Case Study

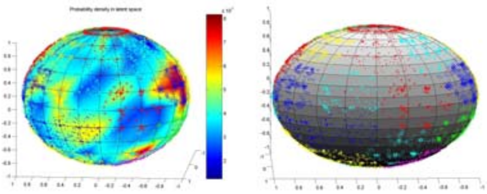
Computational Steps

1. PREPROCESSING: **Noise Estimation Method** and ***Nonlinear PCA***

		features (73)			
	experiments	alpha	cdc15	cdc28	elu
	time points	1 ... 18	1 ... 24	1 ... 17	1 ... 14
Systematic Name (6178)	YAL001C				
	YAL002W				
	...				
	YPR203W				
	YPR204W				

		features(32)			
	experiments	alpha	cdc15	cdc28	elu
	time points	1 .. 8	1 ... 8	1 ... 8	1 .. 8
Systematic Name (6178)	YAL001C				
	YAL002W				
	...				
	YPR203W				
	YPR204W				

2. DATA MINING: ***3D Spherical PPS*** *and* ***Clustering***



Case Study
Gene Noise Estimation Method

- The genes behaviour is periodic. The period is the cell cycle.
- This implies that a gene behaviour, sampled for two cell cycles, can be considered as two measurements of the same thing.
- This can be used to obtain an estimation for the uncertainty of the measurement.

Case Study
Gene Noise Estimation Method

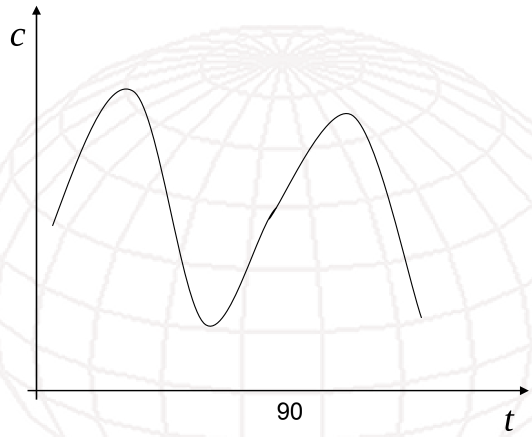
- Cell cycle duration, i.e. period, depends on some parameters such as temperature, nutrient source, density of cells and so on (for our experiments, periods were in the limits 90 ± 11 min).

- To find the exact period length of each experiment we divided the gene time series in two parts and searched for (moving the cutting point in the interval 90 ± 11) the point of best correlation between the two parts.

Case Study
Gene Noise Estimation Method

- Once obtained the period length, we have computed the noise/signal ratio of each gene, considering:
  - the difference between the two periods of each gene as an estimation of its noise;
  - the mean of the two periods as the “real” signal of the gene.
- This value was used to exclude too noisy genes.
- This estimation is accomplished independently for each experiment.

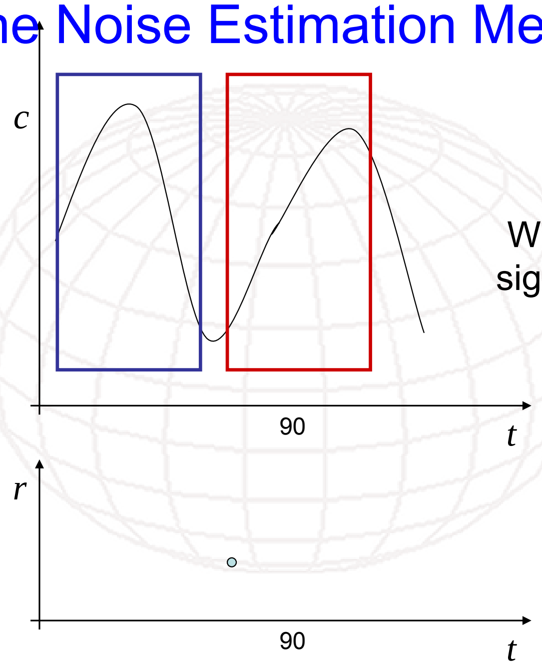
Case Study
Gene Noise Estimation Method



Consider a generic gene signal over an experiment

Gene expression signal vs time

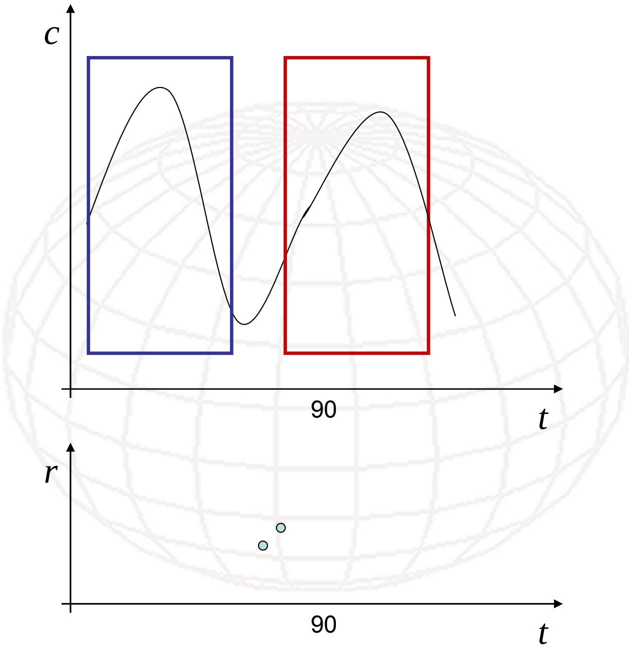
Case Study
Gene Noise Estimation Method



We estimate the signal time period

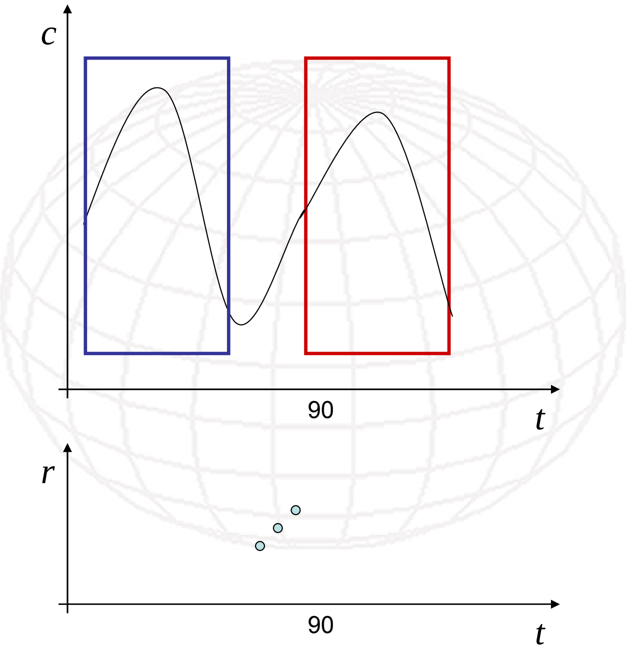
A time window (about 90 min) runs over the signal and the correlation coefficient between the two curve pieces is computed.

Case Study: Gene Noise Estimation Method



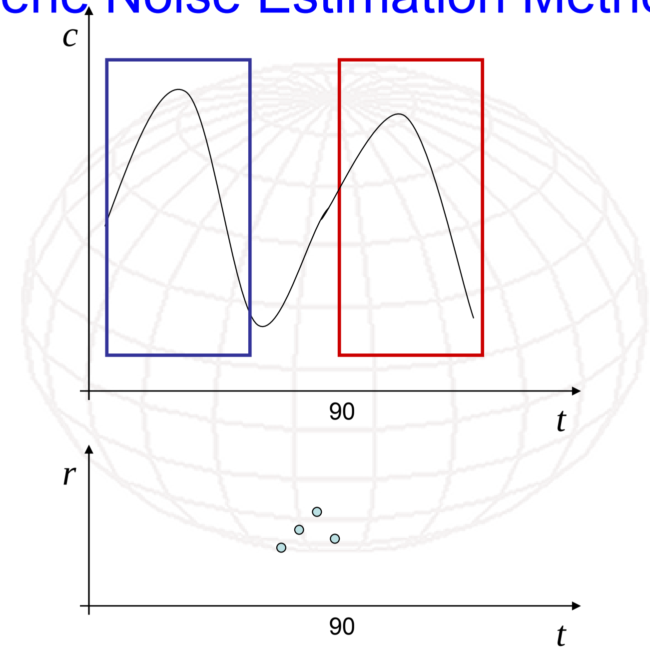
Case Study

Gene Noise Estimation Method

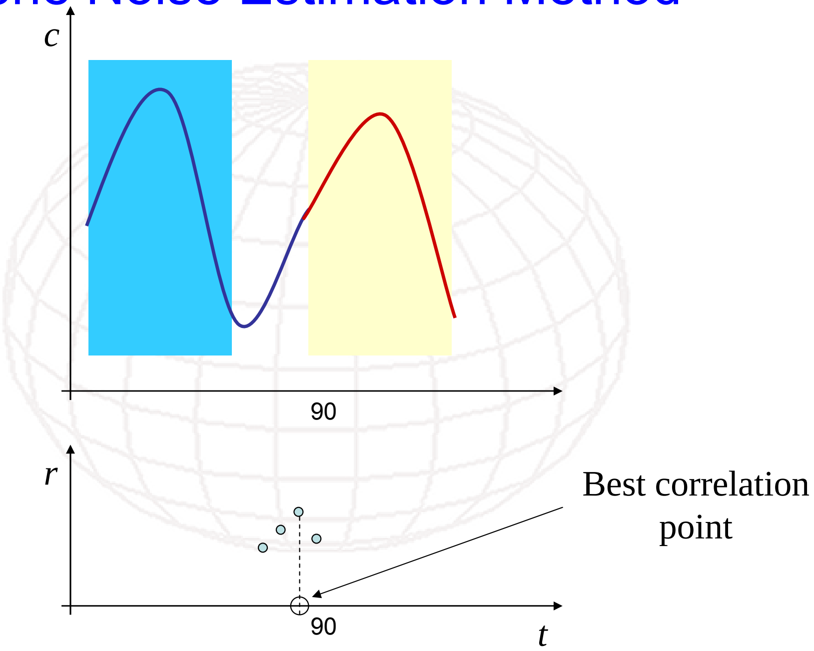


Case Study

Gene Noise Estimation Method

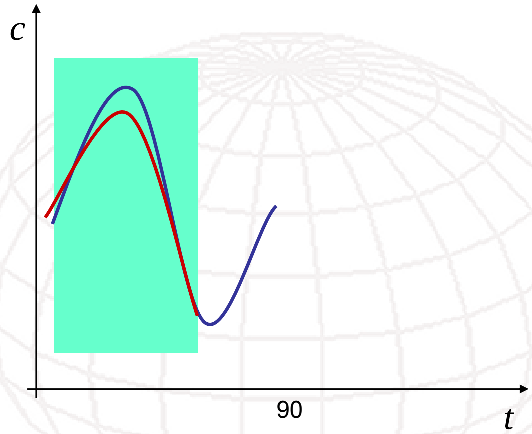


Case Study
Gene Noise Estimation Method



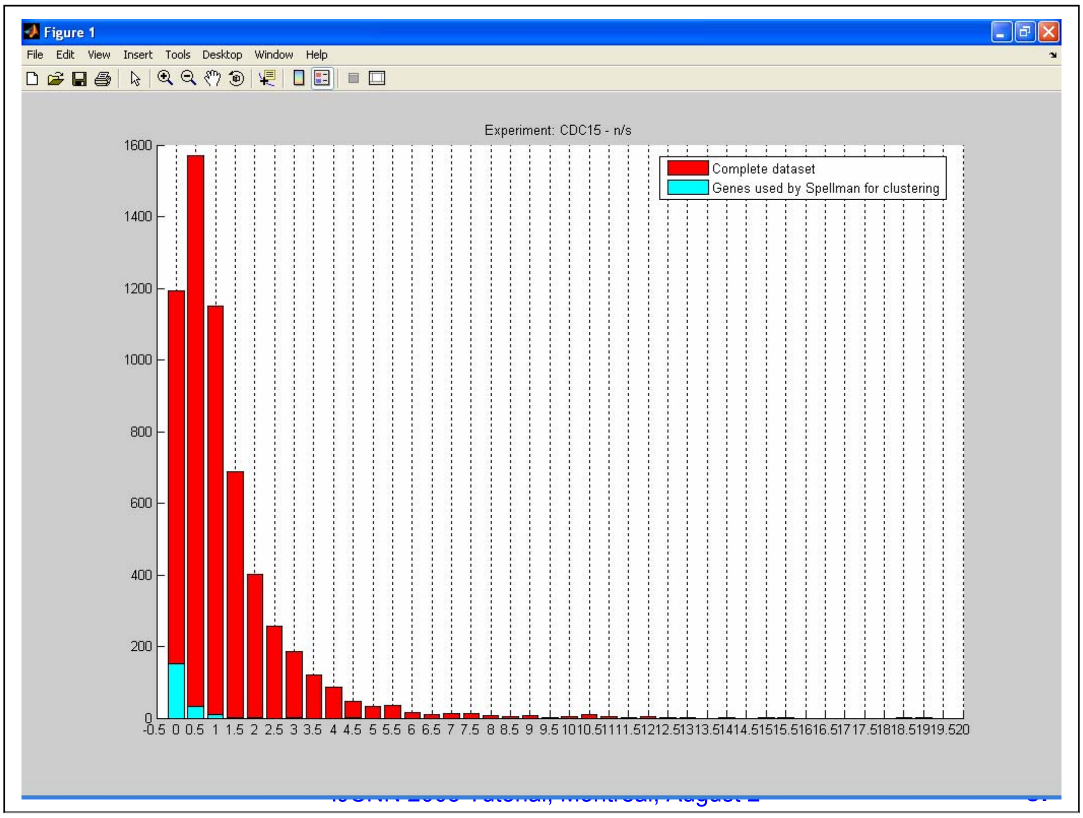
The best correlation point is set as time period.

Case Study
Gene Noise Estimation Method



The signals (before and after the cutting point) are superimposed: the average between them it's the "true" signal. The difference is our estimate of the noise

… the two curve pieces are overlapped. Afterwards, their semi difference represents the noise amplitude.



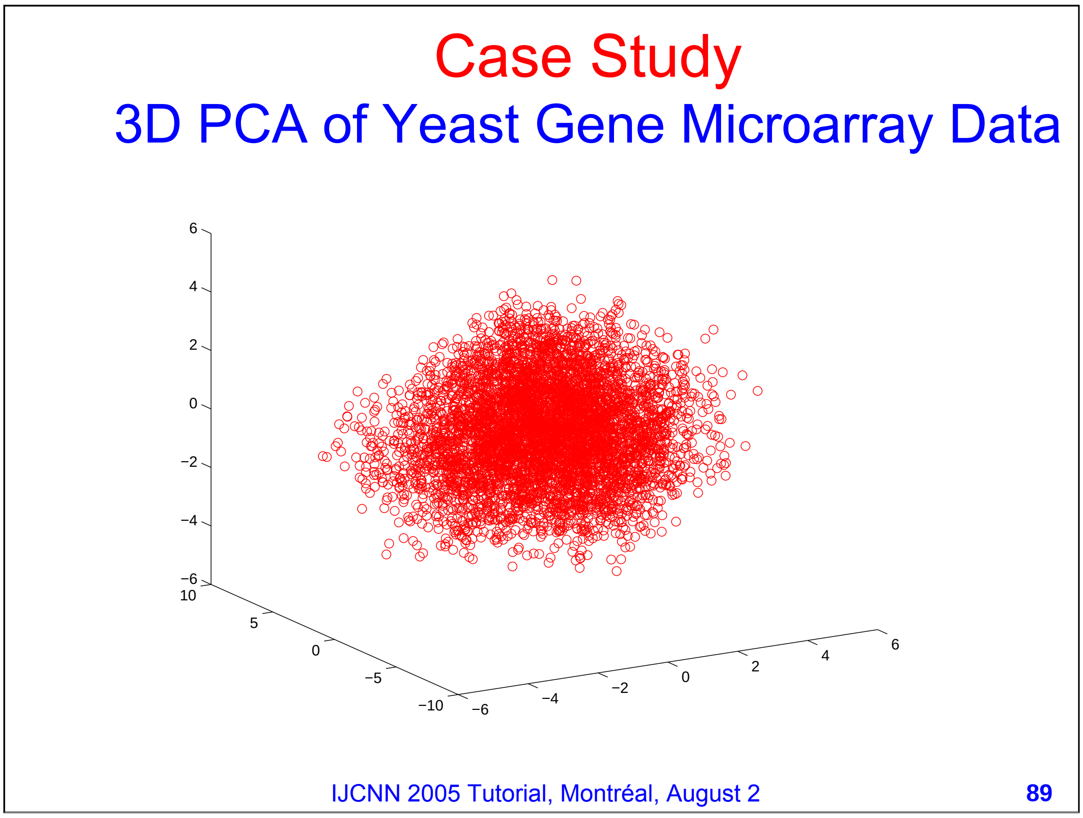
Noise to signal plot in the experiment CDC15. In red are represented the genes of the whole data set while in cyan are the genes used by Spellman et al. This preprocessing step is consistent with the results obtained by Spellman.

Case Study
Preprocessing (nonlinear PCA)

- The data of the experiments are unevenly sampled;
- To extract the features from the experiments we apply a non-linear Principal Component Analysis;
- In details, we apply for each experiment the non-linear PCA to extract the components (1 in our case) to obtain the features.

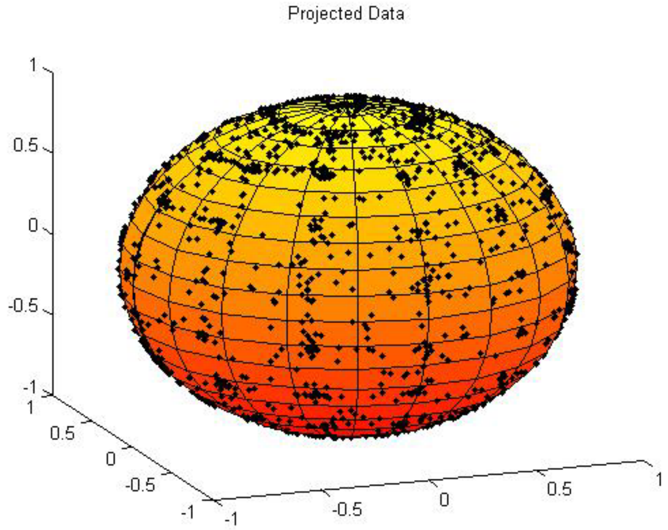
Refer to:

Tagliaferri R., Ciaramella A., Milano L., Barone F., Longo G., **Spectral analysis of stellar light curves by means of neural networks**, Astronomy and Astrophysics Supplement Series, 137:391--405, 1999.

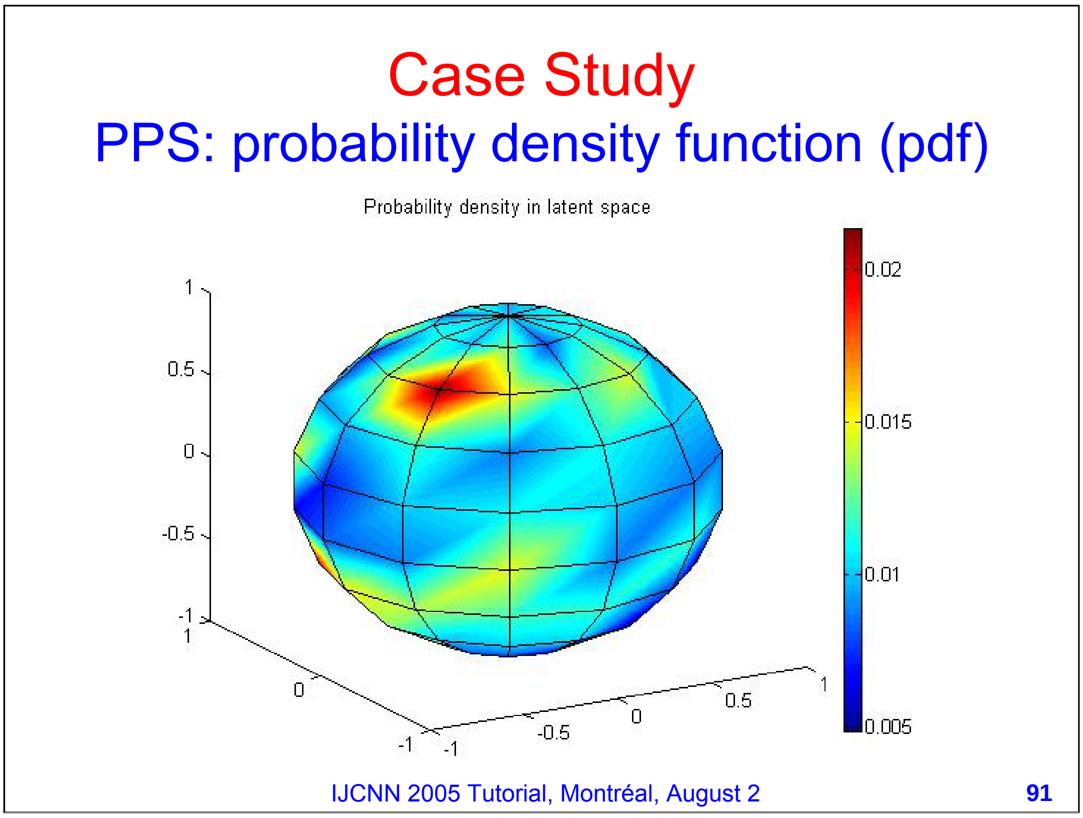


It is clear that a method based on PCA gives no visual information at all since the high nonlinearity of the genetic data, therefore…

Case Study
PPS: data point projections

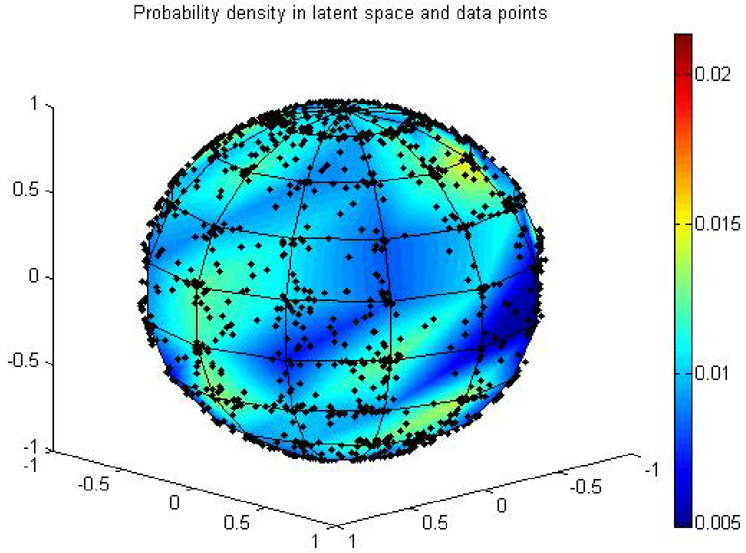


…we recall to spherical PPS. As it is clear from the figure here the data points become more sparse and several little groups are visible bye eye.

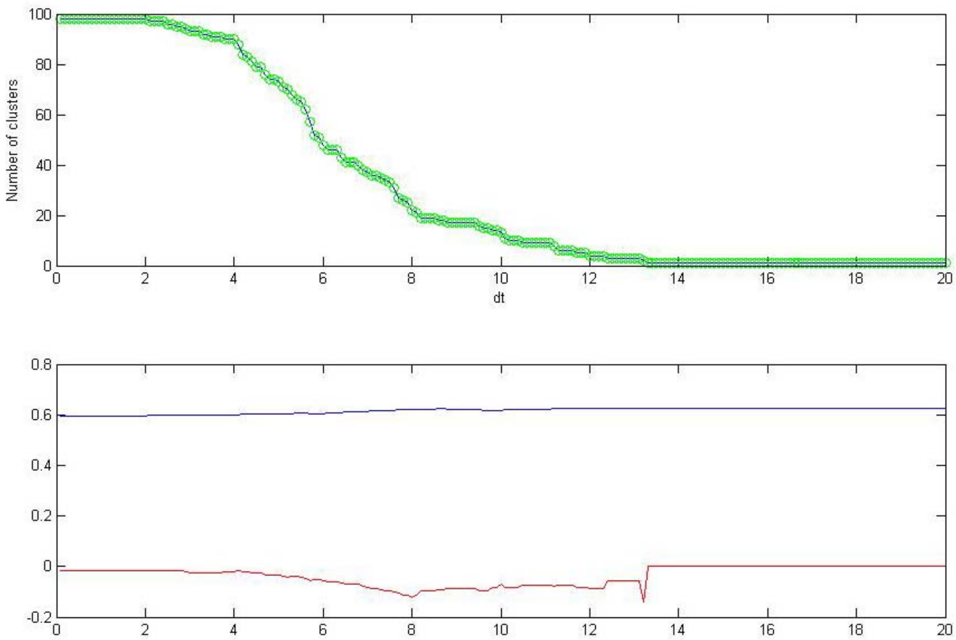


Further studies concerning with the probability density function reveal the presence of several groupings which need to be detailed.

Case Study
PPS: pdf and data point projections



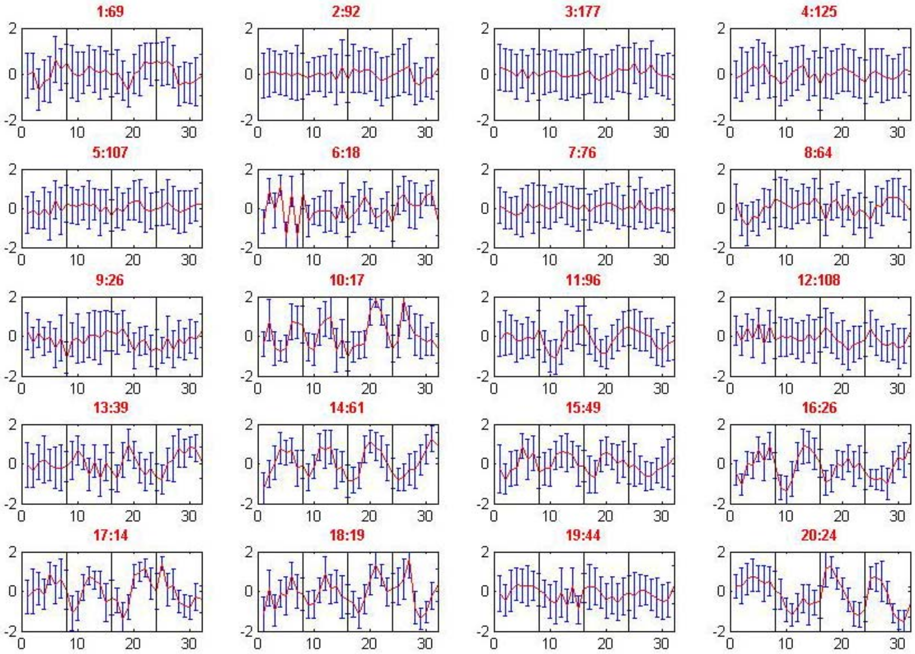
Substructures in the Yeast Gene Data



IJCNN 2005 Tutorial, Montréal, August 2 **93**

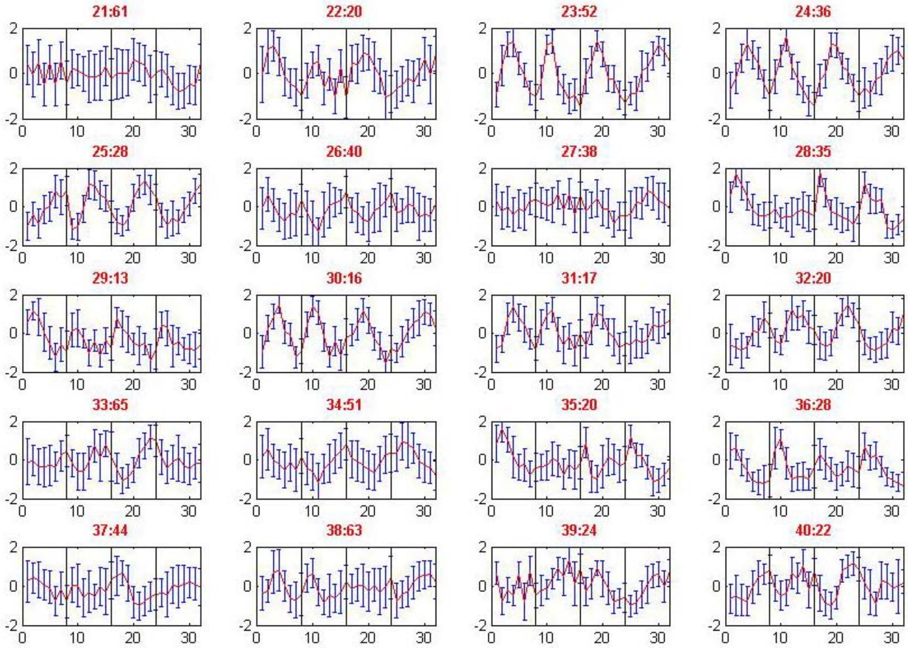
Initializing the NEC algorithm with the PPS previously trained, we studied the threshold values in the interval [0,20]. Zooming on the upper subfigure some little plateaus appear: we decided to investigate on the plateau corresponding to 56 clusters.

Case Study: PPS+NEC results

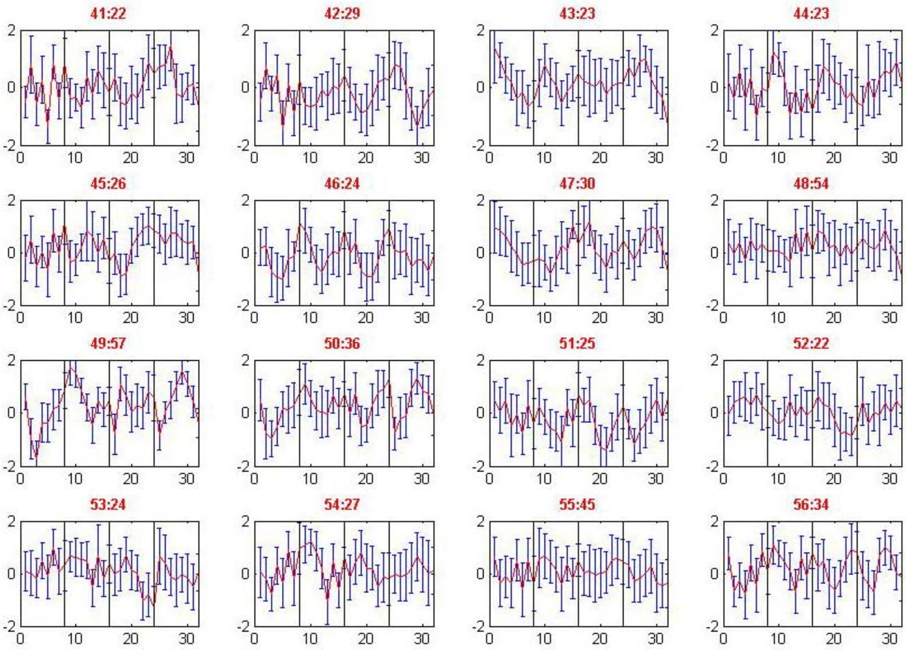


For each cluster the prototype behavior is computed and plotted with the corresponding error bars. In each sub plot the behavior of each of the 4 experiment is shown (each experiment is identified by the vertical lines). Furthermore, the numbers on the top of each plot represent the cluster number and the number of its elements, respectively.

Case Study: PPS+NEC results



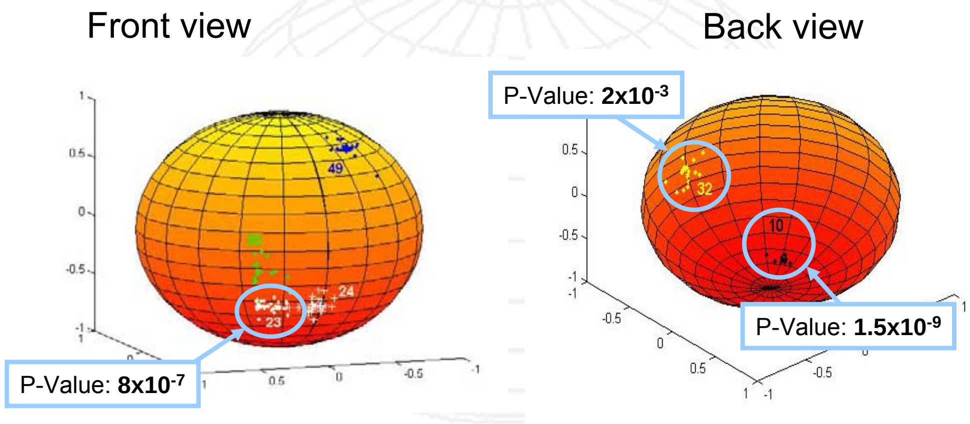
Case Study: PPS+NEC results



IJCNN 2005 Tutorial, Montréal, August 2 **96**

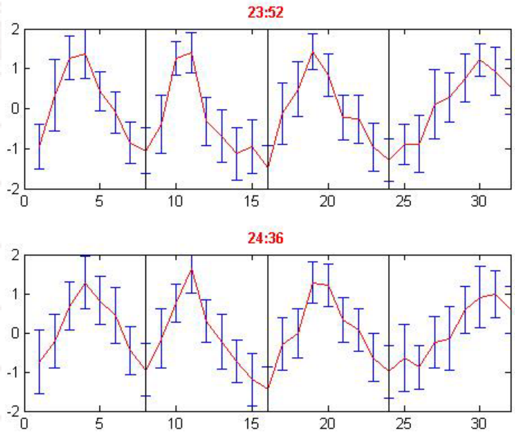
Looking at the prototypes it is possible to discriminate between meaningful clusters (the ones with a regular periodic behavior) from the "noisy" ones (the ones with a constant behavior).

Case Study
PPS+NEC Results



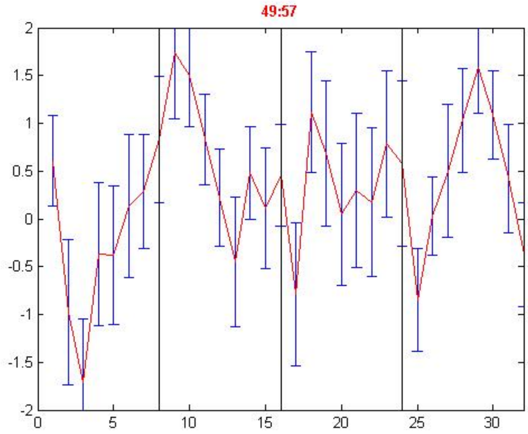
So, let's take a look on some significant clusters: they are very well separated and the corresponding points are not very spread on the sphere surface. The p-value computation confirms the importance of the discovered clusters.

	1 (32)	2 (49)	3 (9)	4 (10)	5 (36)	6 (17)	7 (26)	8 (30)
1	8	0	0	0	0	0	0	0
2	0	9	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	2	3	***7***	0	0	0	0
5	0	0	0	0	8	1	0	0
6	0	0	0	0	0	6	0	0
7	0	0	0	0	0	2	4	0
8	0	0	0	0	0	0	0	5
9	0	0	0	0	0	0	1	0
10	0	0	0	0	0	***7***	0	0
11	5	0	0	0	6	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	1	2	0	0	1	0	0
15	0	0	***4***	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	1	0	0	0	0
23	0	***24***	0	0	0	0	0	0
24	0	6	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0
26	0	0	0	0	***12***	0	1	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0
30	0	3	0	0	0	0	0	0
31	0	2	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	***18***	0	0	0	0	0	0	0
34	0	0	0	0	3	0	0	0
35	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	***16***	0
37	0	0	0	0	1	0	3	0
38	0	2	0	0	2	0	0	***25***
39	0	0	0	0	0	0	0	0
40	1	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	1	0
43	0	0	0	1	0	0	0	0
44	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0
46	0	0	0	0	4	0	0	0
47	0	0	0	1	0	0	0	0
48	0	0	0	0	0	0	0	0
49	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
50-56	0	0	0	0	0	0	0	0



The table illustrates a comparison between the 8 clusters computed by Spellman et al. and the 56 clusters found by PPS+NEC. While some clusters share some genes and is evident that some Spellman clusters are divided in two (see, as an example, PPS+NEC clusters 23 and 24 which contains Spellman cluster 2 and that are very similar) or more PPS+NEC clusters, there are other PPS+NEC meaningful clusters (high p-value) which do not contain any Spellman genes. As an example look at PPS+NEC cluster 49…

Case Study
Cluster 49…



P-Value: 1.6 x 10^{-21}

Case Study
Cluster 23

- 29 genes;
- p-value = 8x10-7;
- 48,98% intersection with Spellman CLN2 cluster;
  - Most of these genes are strongly cell-cycle regulated, peak expression occurs in mid–G1 phase;
  - strongly induced by GAL-CLN3 but are strongly repressed by GAL-CLB2;
  - All these genes are involved in DNA replication;
- The rest of cluster contains some genes with unknown functions.

Here are some biological motivation of cluster 23. The same interesting studies have been done on other meaningful clusters (such as cluster 49).

Conclusions

- Visualization is an important tool in data mining applications for all types of user.
- The domain expert must be involved in the process.
- Interaction with the plots allows the user to query the data more effectively.

Conclusions (2)

- Spherical PPS exhibits a number of attractive abilities for classification (not treated here) and visualization of high-D data.
- The spherical manifold is able to better characterize and represent the periphery and the sparsity of high-D data due to the *curse of dimensionality*.
- Overcome border effects as in rectangular manifold (GTM) and grid (SOM).

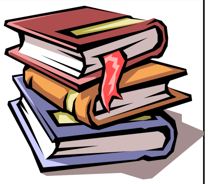
Conclusions (3)

- We built a graphical user interface which allows to interact with the data projected on a unit sphere surface.

- A user is allowed to
  - Interact with data by selecting points on the latent manifold retrieving the corresponding source in the original catalog.
  - The user is able to localize clusters of data on the sphere which correspond to clusters of similar data in the input space.

- Useful for data mining in whatever data rich field.

Bibliography …



- ✓ K. Chang, J. Ghosh, *A Unified Model for Probabilistic Principal Surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 1, 2001.
- ✓ T. Kohonen, *Self-organized formation of topologically correct feature maps*, Biological Cybernetics, 43, 1982.
- ✓ C. M. Bishop, M. Svensen, C.K.I. Williams, *GTM: The Generative Topographic Mapping*, Neural Computation, 10(1), 1998.
- ✓ J. Vesanto, *Data Mining Techniques based on the Self-Organizing Maps*, PhD Thesis, Helsinki University of Technology, 1997.
- ✓ A. Staiano, *Unsupervised Neural Networks for the Extraction of Scientific Information from Astronomical Data*, PhD Thesis, University of Salerno, 2003.

Bibliography

- ✓ A. Staiano, R. Tagliaferri at al., *Probabilistic Principal Surfaces for Yeast Gene Microarray Data Mining,* Proceedings of the IEEE Conference on Data Mining, Brighton (UK), pp. 202-209, 2004.
- ✓ P. Tino, I. Nabney, *Hierarchical GTM: Constructing Localized Nonlinear Projection Manifolds in a Principled Way,* IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol24, N. 6, 2002.
- ✓ C.M. Bishop, M.E. Tipping, *A Hierarchical Latent Variable Model for Data Visualization,* IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20,N. 3, 1998.
- ✓ http://www.statsoft.com/textbook/stmulsca.html