Machine Learning
Unsupervised Methods
Part 1

Sepp Hochreiter

Institute of Bioinformatics
Johannes Kepler University, Linz, Austria

Course

3 ECTS 2 SWS VO (class) 1.5 ECTS 1 SWS UE (exercise)
Basic Course of Master Bioinformatics
Basic Course of Master Computer Science: Computational Engineering / Int. Syst.

Class: Mo 15:30-17:00 (S3 055); Exercise: Mo 14:30-15:15 (S2 059)

VO: final exam (oral if few students subscribe)
UE: weekly homework (evaluated)

	Lecture		Lecturer	
365,077	Machine Learning: Unsupervised Techniques	VL	Hochreiter	Mon 15:30-17:00/S3 055
365,078	Machine Learning: Unsupervised Techniques	UE	Mitterecker	Mon 14:30-15:15/S2 059
365,041	Theoretical Concepts of Machine Learning	VL	Hochreiter	Thu 15:30-17:00/S3 055
365,042	Theoretical Concepts of Machine Learning	UE	Povysil	Mon 13:45-14:30/S2 059
365,081	Genome Analysis & Transcriptomics	KV	Regl	Tue 8:30-11:00/S3 055
365,082	Structural Bioinformatics	KV	Regl	Tue 8:30-11:00/S3 055
365,036	Special Topics on Bioinformatics - Population genetics	KV	Povysil	Tue 11:00-12:45/S3 318
365,079	Introduction to R	KV	Bodenhofer	Wed 15:30-17:00/S3 055
365,067	Master's Seminar	SE	Hochreiter	Mon 10:15-11:45/S3 318
365,080	Master's Thesis Seminar SS	SE	Hochreiter	Mon 10:15-11:45/S3 318
365,019	Dissertantenseminar 3	SE	Hochreiter	Mon 10:15-11:45/S3 318
347,337	Bachelor Seminar Biological Chemistry (incl. Bachelor Thesis)		Hochreiter	-

Computational Engineering (CE) Colloquium

The CE Colloquium provides a forum for students and researchers from JKU and industry to exchange ideas and experiences related to computational engineering.

Summer Semester 2014

The colloquium is scheduled for **Thursdays at 5:15 pm** in room MT 226/1 in Science Park 1 (Mechatronics Building).

Mar 20	Christoph Steindl, Catalysts	HPC for Industry and Space
May 8	Christian Eitzinger, PROFACTOR	Computational Engineering and Machine Learning for Robotic Inspection Systems
May 15	Peter Stadelmeyer, RISC Software	Multidisziplinäre Strukturoptimierung aus Softwarearchitektursicht - Algorithmen und Technologielebenszyklen
May 22	Robert Raschhofer, Fujitsu Austria	Die Entwicklung von Automotive Grade grafische User Interfaces
Jun 5	Alexander Haas, Magna	The free lunch is really over: über (unerwartete) Probleme bei SW-Parallelisierungsprojekten in der Praxis

Outline

Outline

Outline

Outline

Outline

Literature

- ML: Duda, Hart, Stork; Pattern Classification; Wiley & Sons, 2001
- ML: C. M. Bishop; Neural Networks for Pattern Recognition, Oxford Univ. Press, 1995
- ML: T. M. Mitchell; Machine Learning, Mc Graw Hill, 1997
- Statistics: S. M. Kay; Fundamentals of Statistical Signal Processing, Prent. Hall, 1993
- Belief Nets: M. I. Jordan; Learning in Graphical Models, MIT Press, 1998
- Data Analysis: R. Peck, C. Olsen and J. L. Devore; Introduction to Statistics and Data Analysis, 3rd edition, ISBN: 9780495118732, Brooks/Cole, Belmont, USA, 2009
- Statistical Data Analysis: B. Shahbaba; Biostatistics with R: An Introduction to Statistics Through Biological Data; Springer, series UseR!, New York, 2012
- Statistical Data Analysis: C. T. Ekstrom and H. Sorensen; Introduction to Statistical Data Analysis for the Life Sciences; CRC Press, Taylor & Francis Group, USA, 2011
- Clustering: L. Kaufman and P. J. Rousseeuw; Finding Groups in Data. An Introduction to Cluster Analysis, Wiley, 1990

Chapter 1

Introduction

Introduction

- part of curriculum “master of science in bioinformatics”
- part of curriculum “computer science” (major CE, major int. sys.)

- Machine learning major research topic: Google, Microsoft, Amazon, Facebook, AltaVista, Zalando, and many more

- Applications: computer vision (image recognition), speech recognition, recommender systems, analysis of Big Data, information retrieval

- Mining the web: search engines, social networks, videos, music

- Machine learning applications in biology and medicine:
  - microarrays, sequencing
  - alternative splicing, nucleosome positions, gene regulation
  - single nucleotide polymorphisms / variants (SNPs, SNVs)
  - copy number variations (CNVs)
  - diseases: Alzheimer, Parkinson, cancer, multiples sclerosis, schizophrenia or alcohol dependence

Introduction

This course introduces **unsupervised** machine learning methods:

- output is not given
- objective: cumulative output on all samples

Objectives:

- information content,
- orthogonal
- statistical independence
- variation explained
- entropy
- likelihood: probability that model produces observed data
- distances between and within clusters

Used for analyze data:

- explore
- find structure
- visualize
- compress

Understand and explore the data and generate new knowledge

Introduction

concepts of unsupervised learning:

- maximum likelihood
- maximum a posteriori
- maximum entropy
- expectation maximization
- maximal variance
- independence
- non-Gaussianity
- sub- and super-Gaussian distributions
- sparse and population codes

Introduction

- Goal: select model
  with highest generalization performance, that is
  with the best performance on future data,
  from the model class

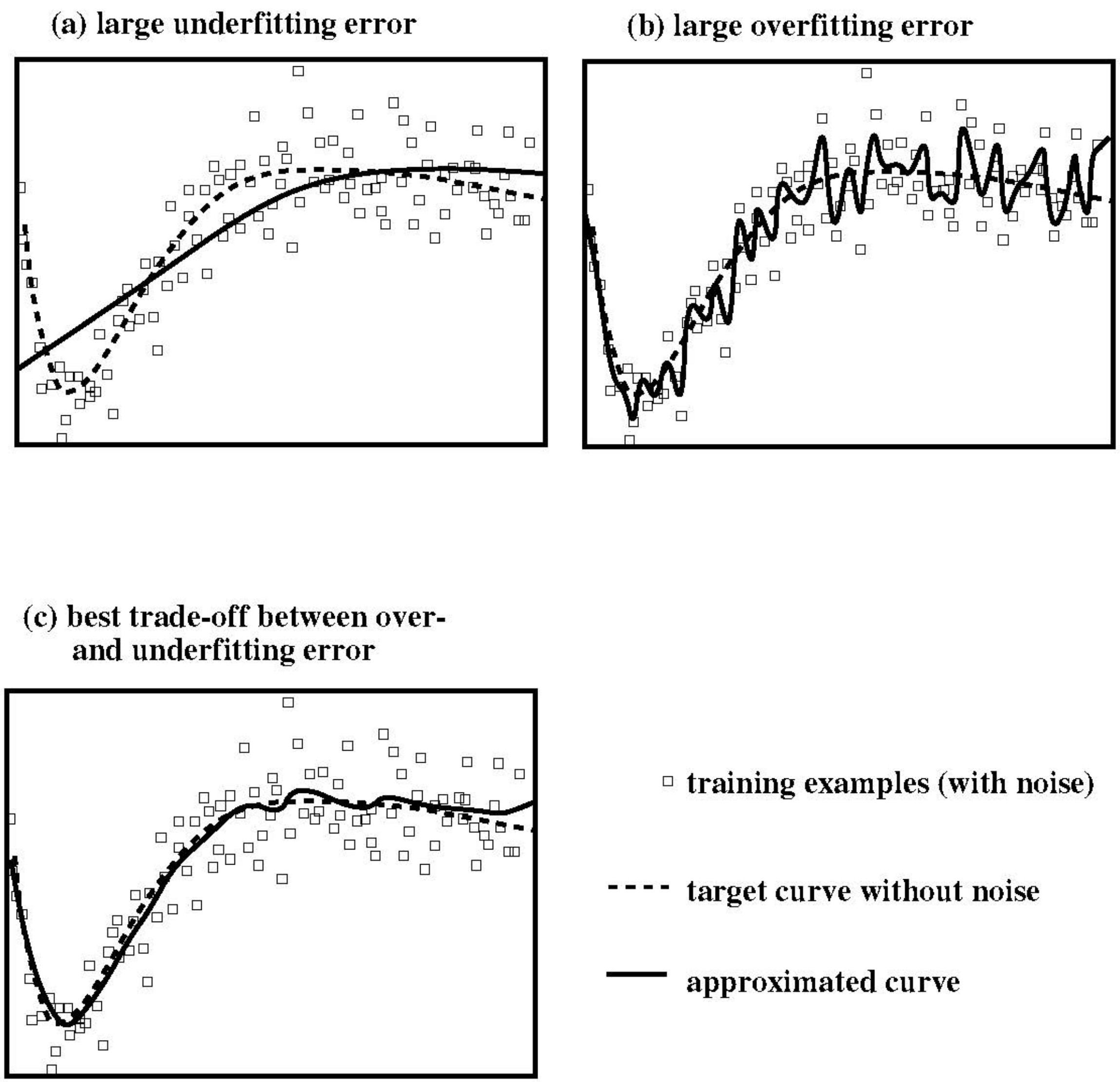
- model selection is training is learning

- model which best explains or approximates the training set

- remember: salmon vs. sea bass → the model which perfectly explains the training data had low generalization performance

- "overfitting": model is fitted (adapted) to special training characteristics
  - noisy measurements
  - outliers
  - labeling errors

Introduction



Introduction

Unsupervised Methods:

- principal component analysis
- independent component analysis
- factor analysis
- projection pursuit
- k-means clustering
- hierarchical clustering
- mixture models: Gaussian mixtures
- self-organizing maps
- kernel density estimation
- hidden Markov models
- Markov networks (Markov random fields)
- restricted Boltzmann machines
- neural network: auto-associators, unsupervised deep nets

Introduction

Projection methods:

- new representation of objects
- down-projection into lower-dimensional space: keeps the neighborhoods
- finding structure in the data

Generative models:

- build a model of the observed data
- match the observed data density

Introduction

- projection: representation of objects,
  down-project feature vectors ,
  PCA: orthogonal maximal data variation components,
  ICA: statistically mutual independent components,
  factor analysis: PCA with noise

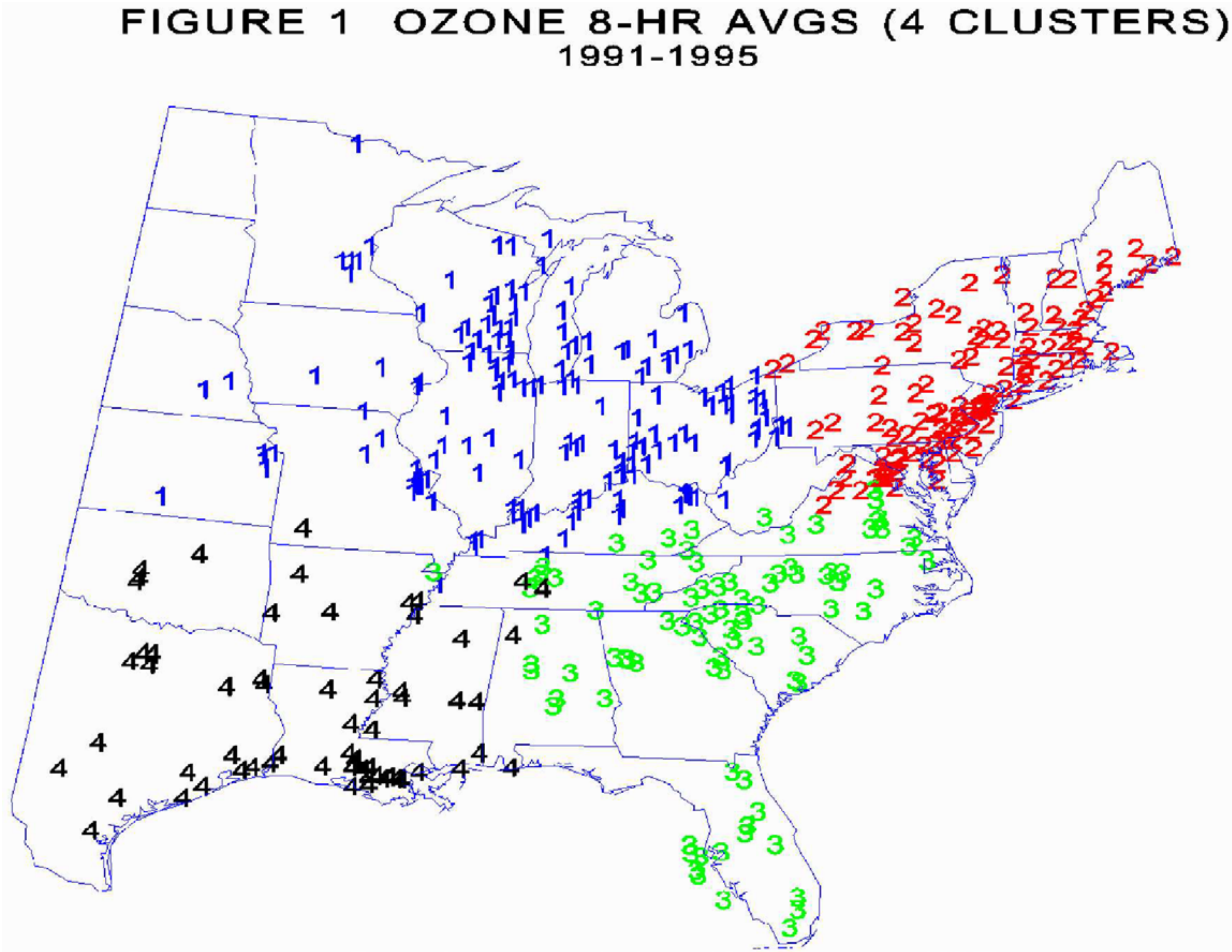
- density estimation: density model of observed data

- clustering: extract clusters – regions data accumulation (typical data)

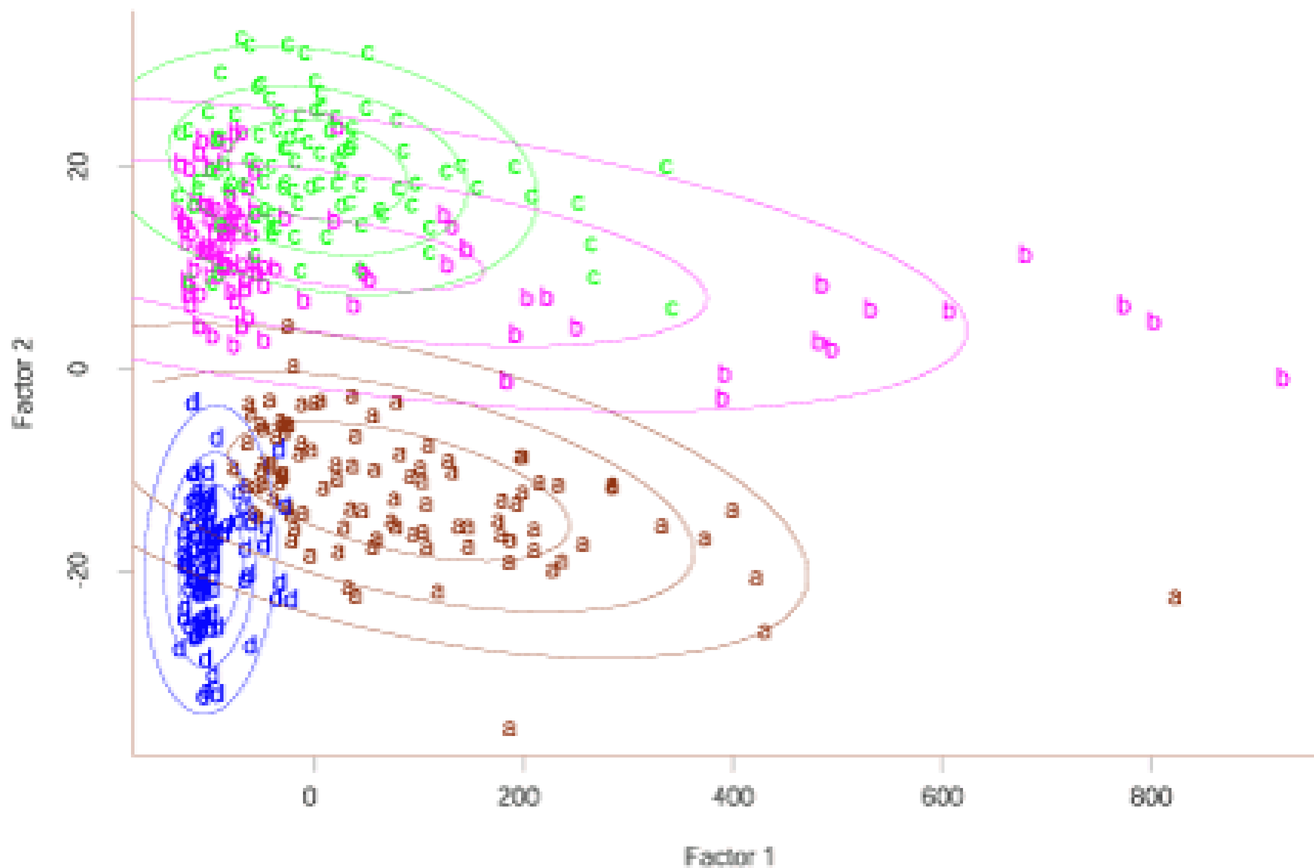
Goals of this course:

- how to chose appropriate methods from a given pool
- understand and evaluate the different approaches
- where to obtain and how to use them
- adapt and modify standard algorithms

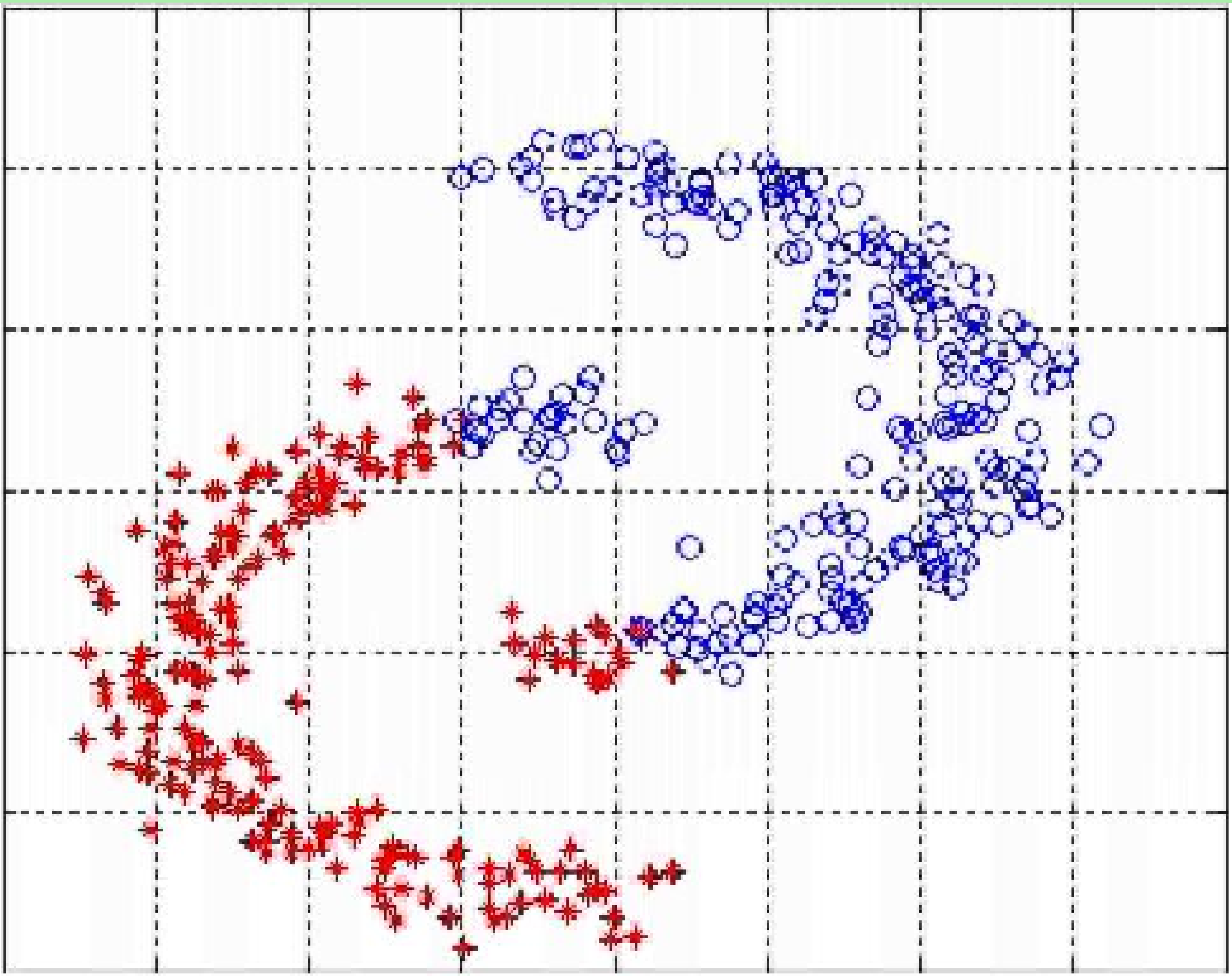
Introduction



Introduction

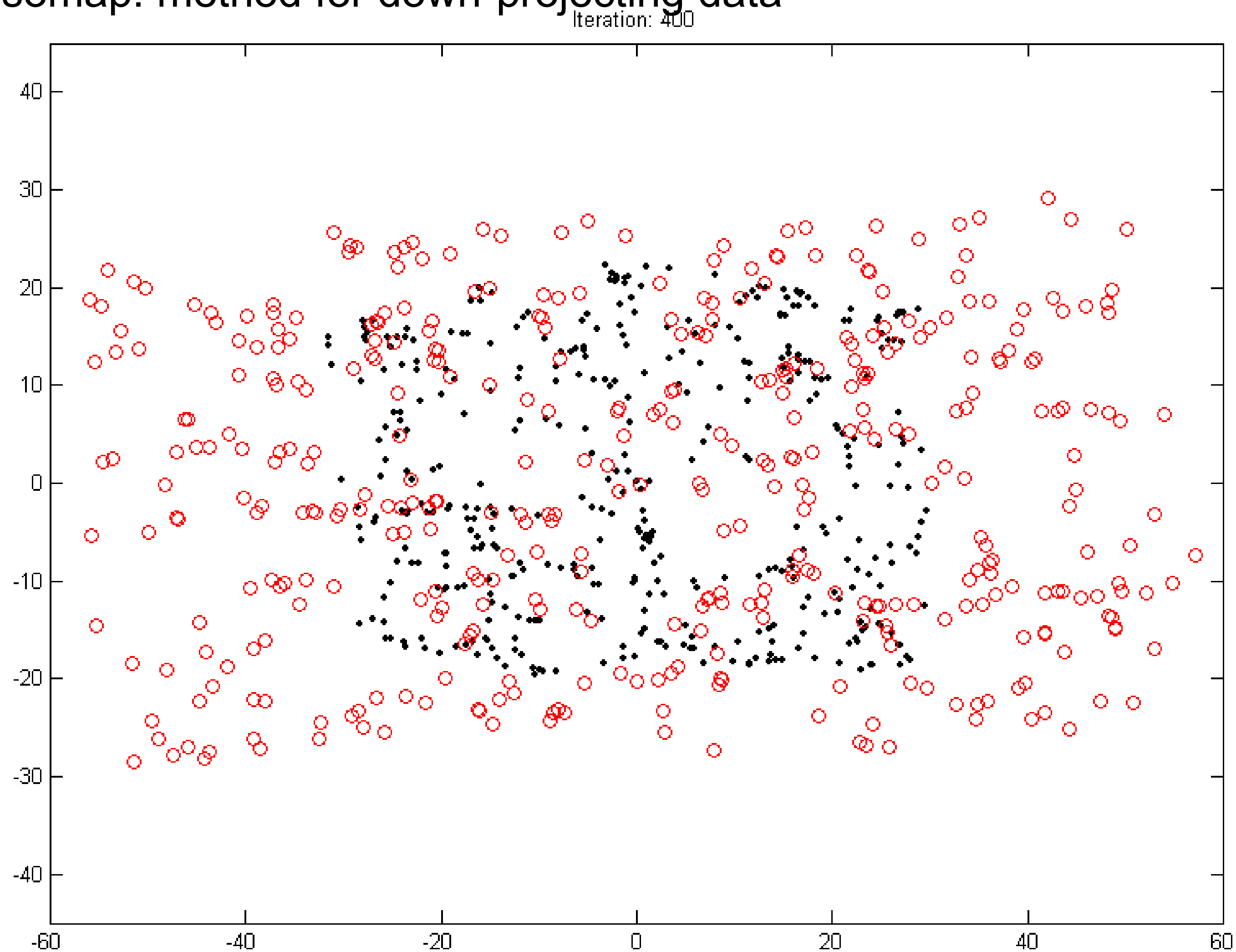


Introduction

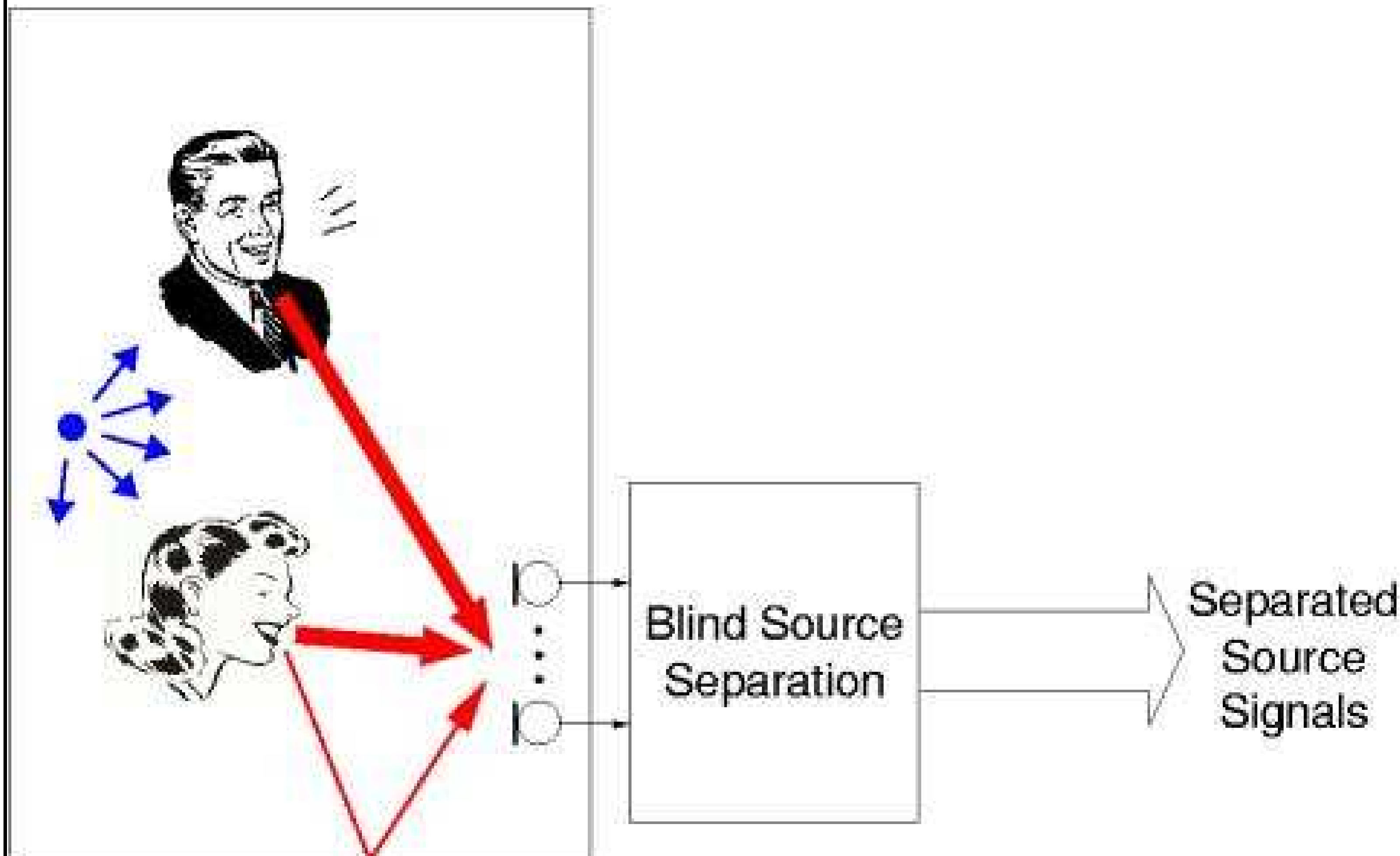


Introduction

Isomap: method for down-projecting data



Introduction



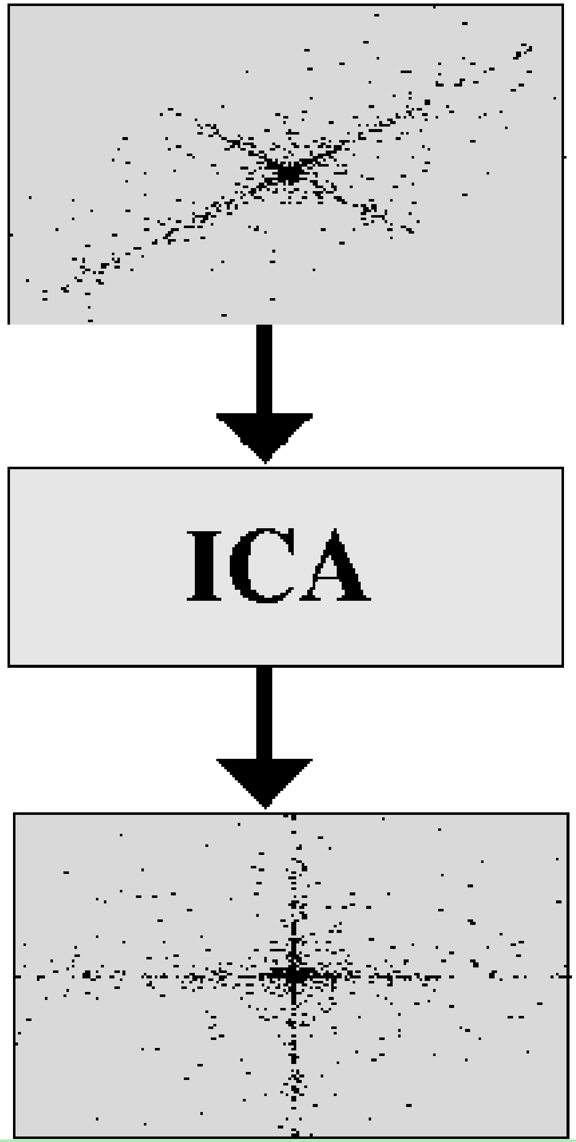
Introduction

Original:

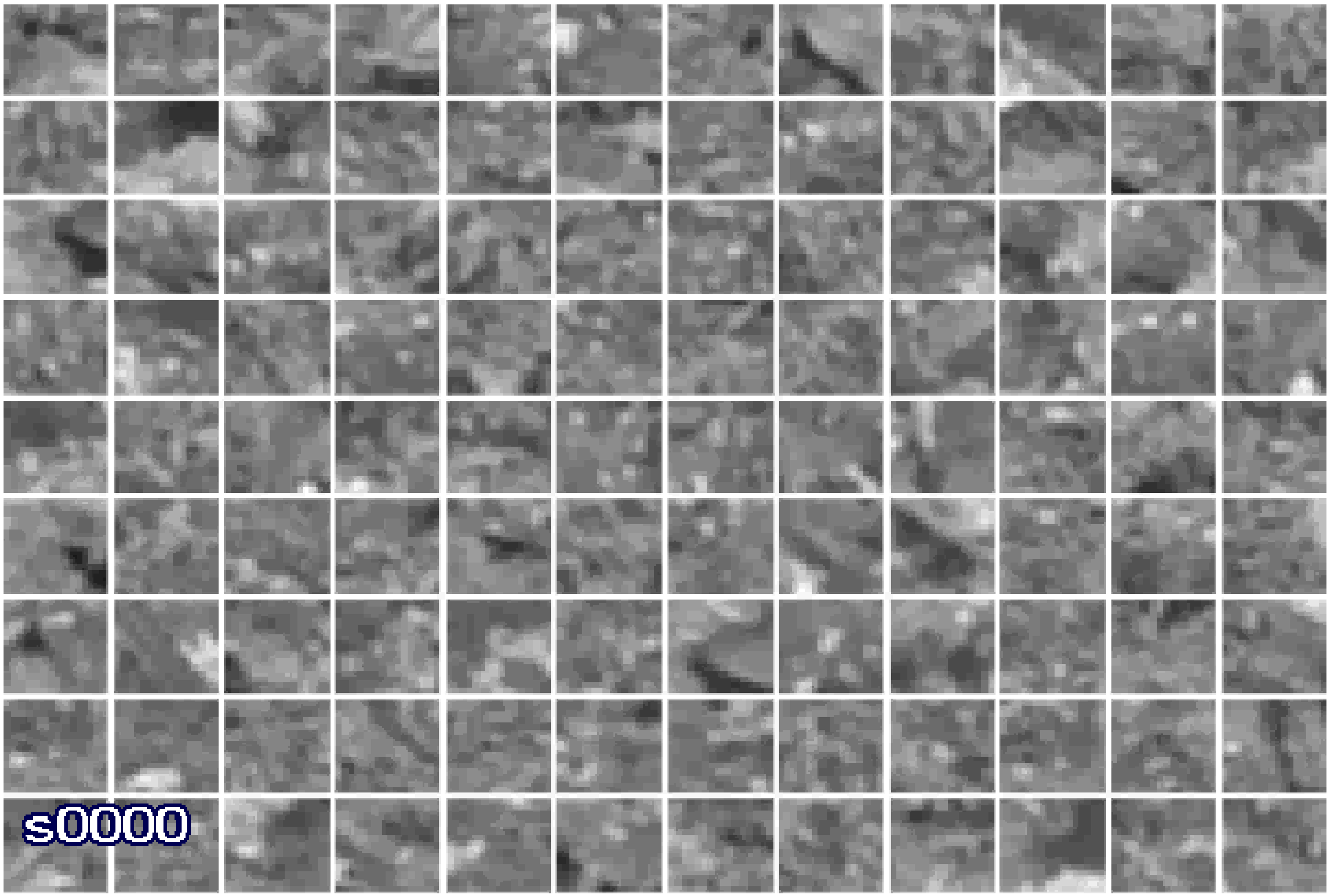
Mixtures:

Demixed by ICA:

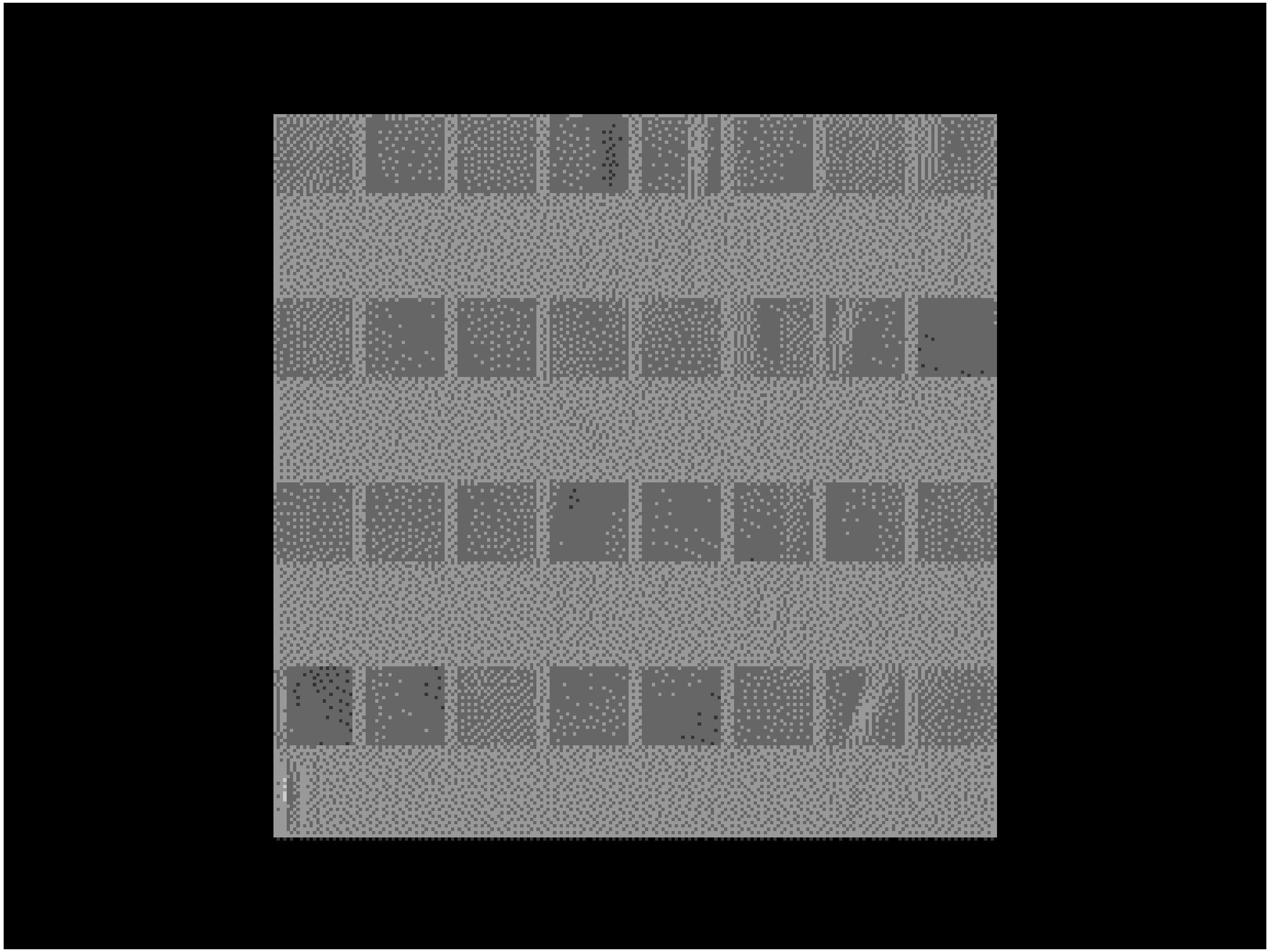
Introduction



ICA: on images



ICA: on video components



Parametric vs. Non-Parametric Models

important step in machine learning is to select a **model class**

parametric models:

- each parameter vector represents a model
- examples:
  - neural networks: parameter are synaptic weights
  - support vector machines
- learning: paths through the parameter space
- disadvantages:
  - different parameterizations of the same function
  - model complexity and class via the parameters

nonparametric models:

- model is locally constant / superimpositions
- Examples:
  - *k*-nearest-neighbor (*k* is hyperparameter – not adjusted)
  - kernel density estimation
  - decision tree
- constant models (rules) must be a priori selected that is hyperparameters must be fixed (*k*, kernel width, splitting rules)

Generative vs. descriptive Models

descriptive model:

- additional description or another representation of the data
- projection methods (PCA, ICA)

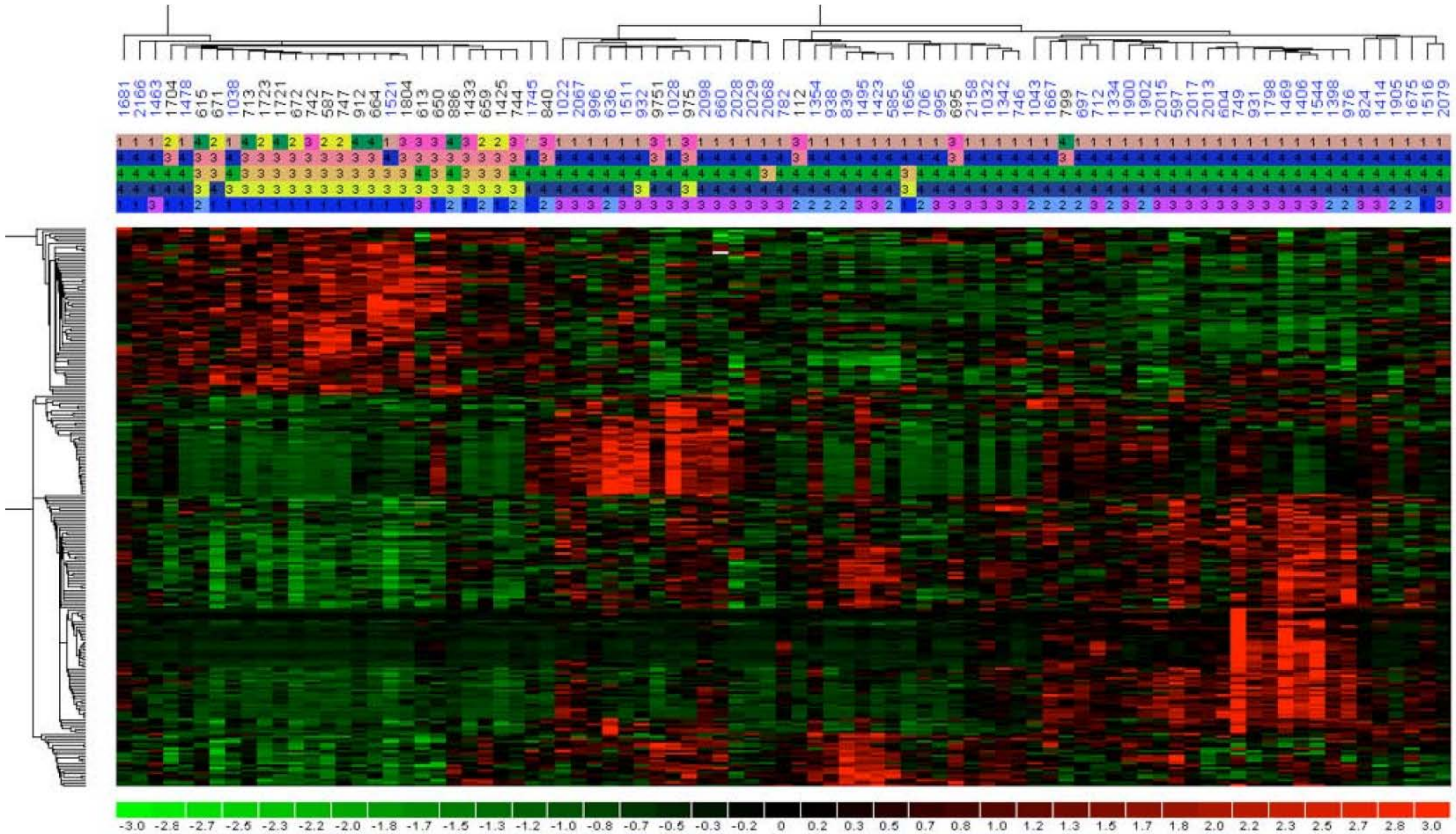
generative model:

- model should produce the distribution observed for the real world data points
- describing or representing random components which drive the process
- prior knowledge about the world or desired model
- predict new states of the data generation process (brain, cell)

Chapter 2

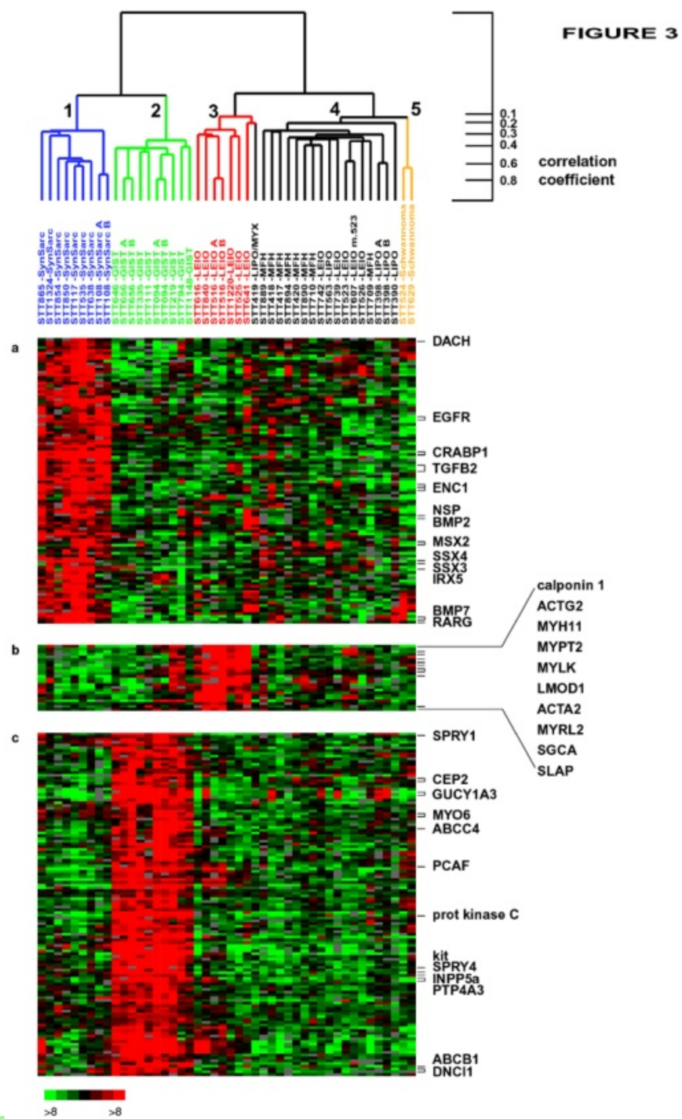
Basic Terms and Concepts

Unsupervised Learning in Bioinformatics



Clustering of microarray data

Unsupervised Learning in Bioinformatics



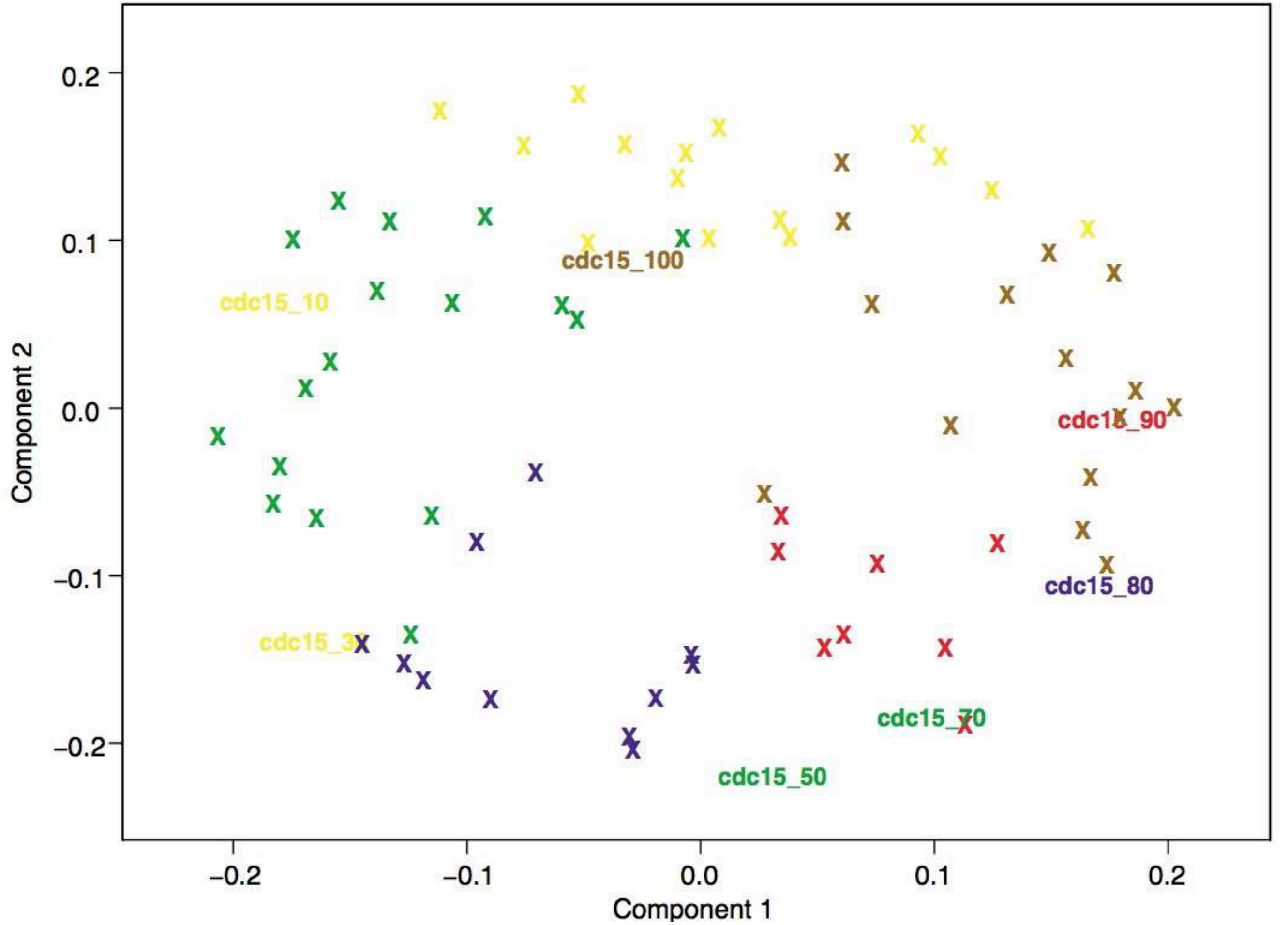
Clustering of microarray data.

Representative portions of the tumor specific gene clusters. The spectrum of green to red spots represents the relative centered expression for each gene.

Correlation coefficient bar shown to the right side of the dendrogram indicates the degree of relatedness between branches of the dendrogram.

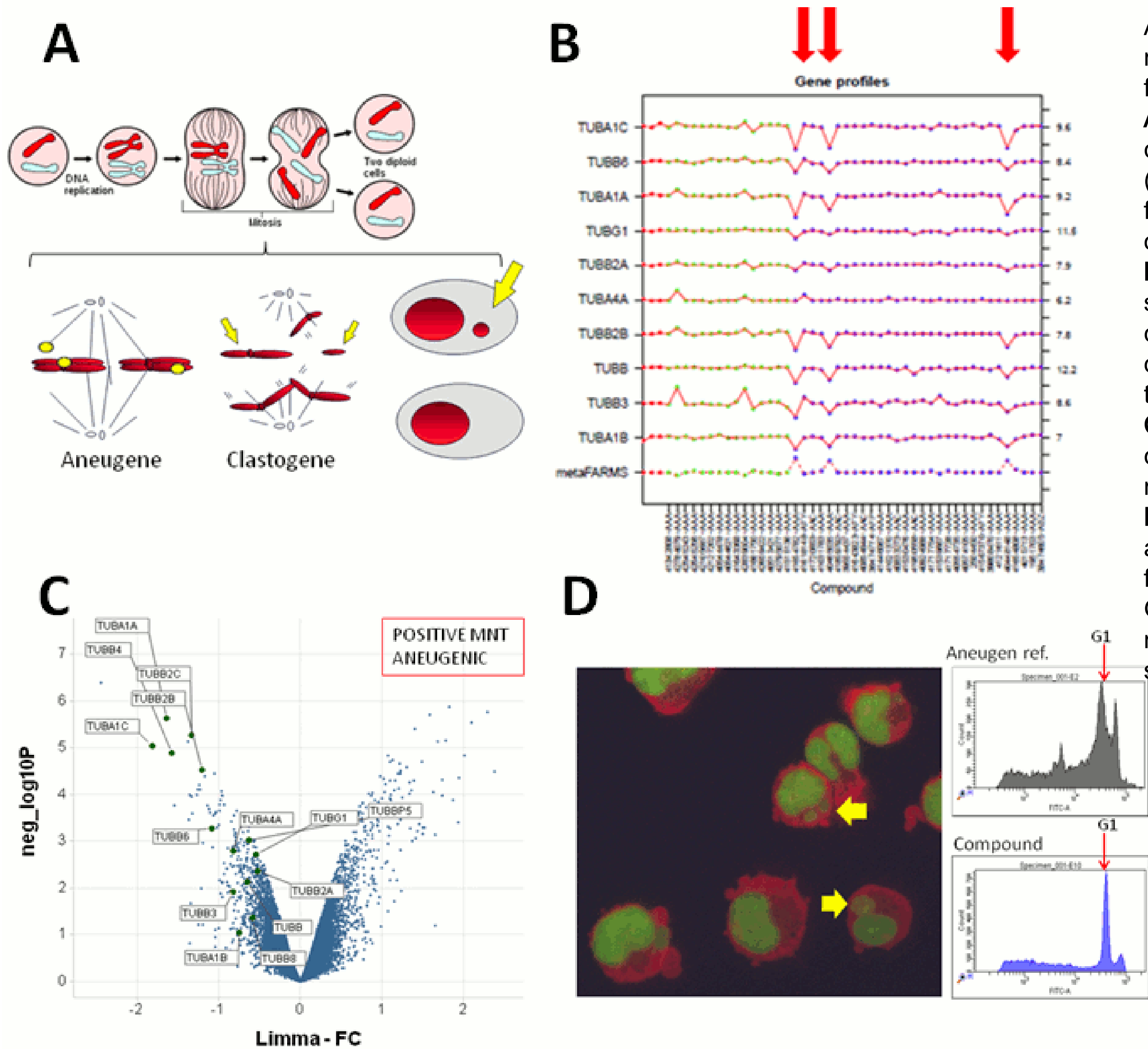
Unsupervised Learning in Bioinformatics

unsupervised methods: visualize dependencies and clusters



Spellmans cell-cycle data: first principal components

Unsupervised Learning in Bioinformatics



An example for a signature of a rare event (micronuclei formation).

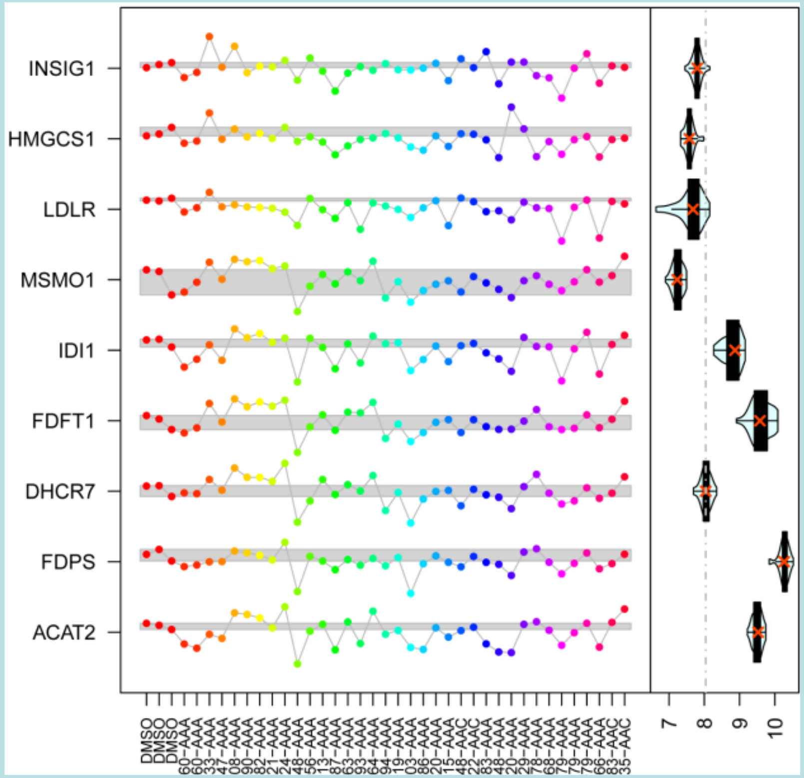
A. Genotoxic compounds can cause chromosomal breaks (aneugene) or affect the formation of the mitotic spindle or microtubuli (clastogene).

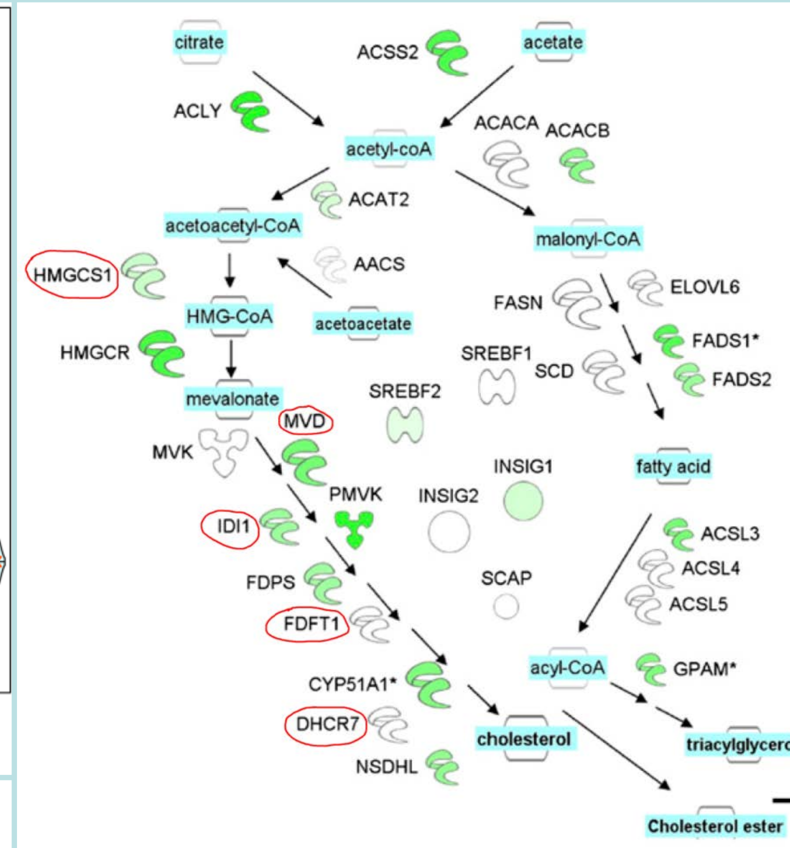
B. The gene expression signature of only three compounds (red arrows) show down-regulation of several tubulin-genes.

C. Volcano plot of one compound showing a down-regulation of tubulin genes.

D. Microscopic and FACScan analysis confirmed micronuclei formation (yellow arrows) and G1-cell cycle arrest indicating microtubuli-based chromosome segregation.

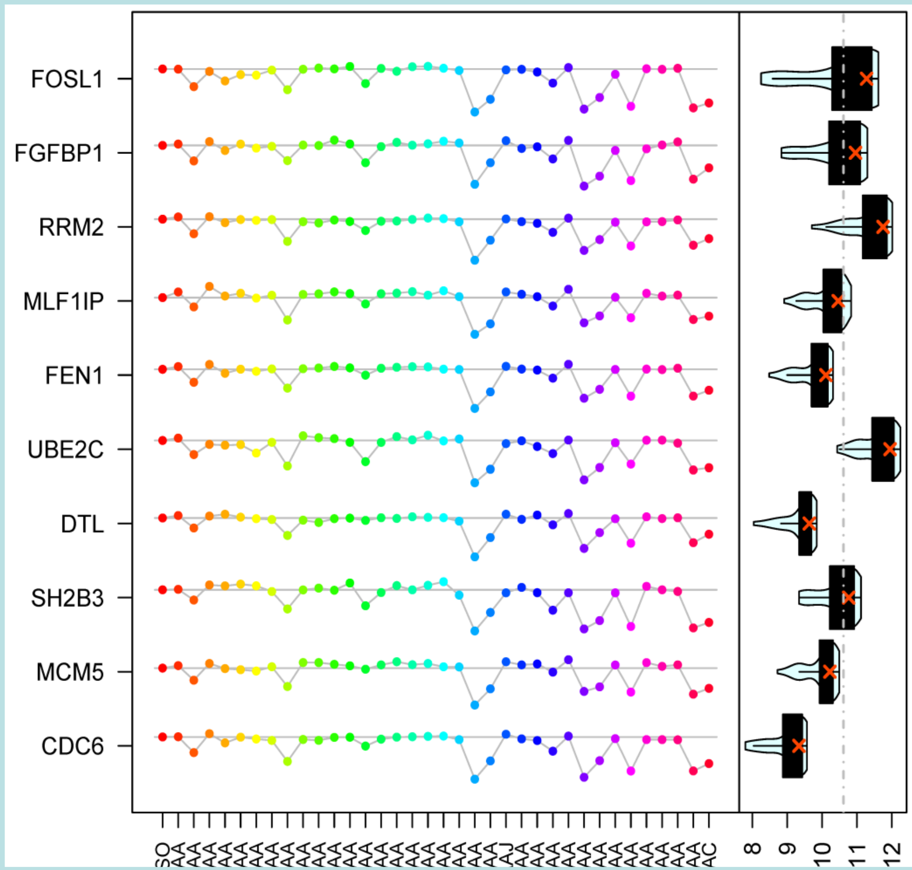
Unsupervised Learning in Bioinformatics

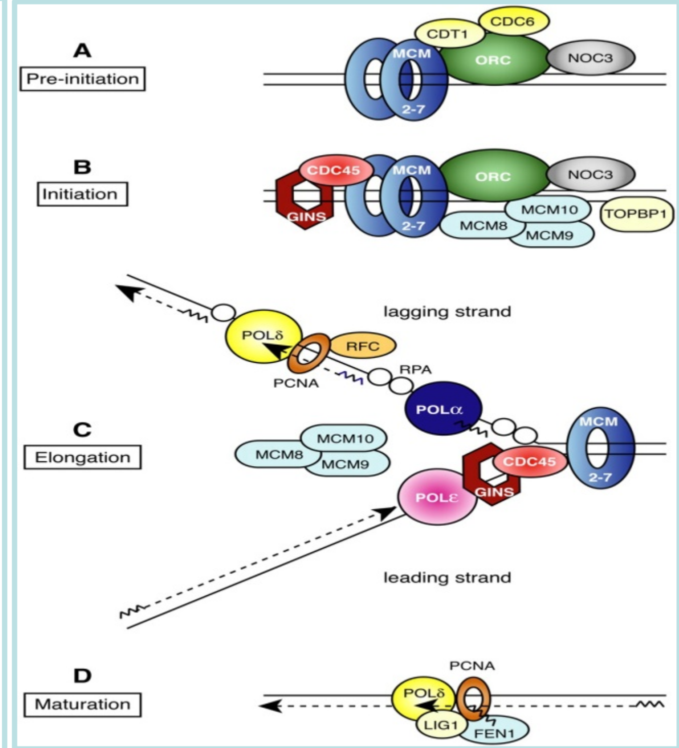




left panel: Biclustering results of gene expression data from a cell line where a compound was added that affects metabolic pathways. **right panel:** The genes HMGCS1, IDI1, FDFT1, DHCR7 of the bicluster code for proteins that belong to the SREBP cholesterol metabolism pathway. FABIA was capable to identify this bicluster of 9 genes activated by few compounds in a data set of tens of thousands of genes.

Unsupervised Learning in Bioinformatics





left panel: Biclustering result of gene expression of a cancer cell line to which a compound has been added. **right panel**: The genes CDC6, MCM5, FEN1 are coding for proteins that participatie at DNA replication complex. The other bicluster genes code for proteins that initiate or are involved DNA replication (MLF1IP → chromosome segregation; RRM2 → DNA synthesis; DTL → regulation of DNA replication).

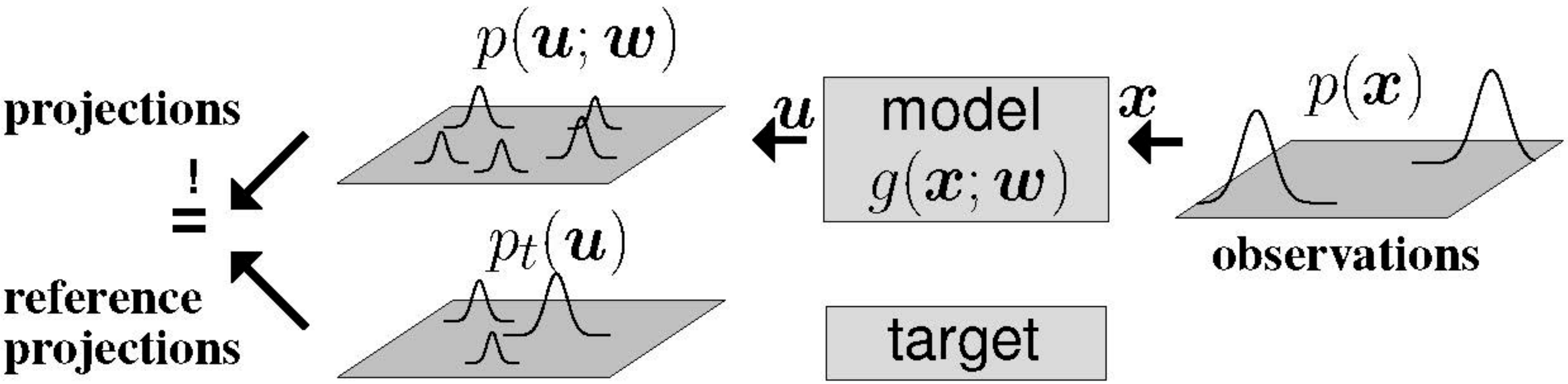
Unsupervised Learning Categories

unsupervised categories:

- generative framework: density estimation, hidden Markov models and objectives are maximum likelihood, maximum a posteriori, expectation maximization
- recoding framework: projection methods, pca, ica and objectives are maximal variance, orthogonal, independence, maximum entropy

Projection Methods

projection methods project the data into a space with desired properties



Projection Methods

BIOINF

- Principal Component Analysis (PCA): projection to a low dimensional space under maximal information conservation

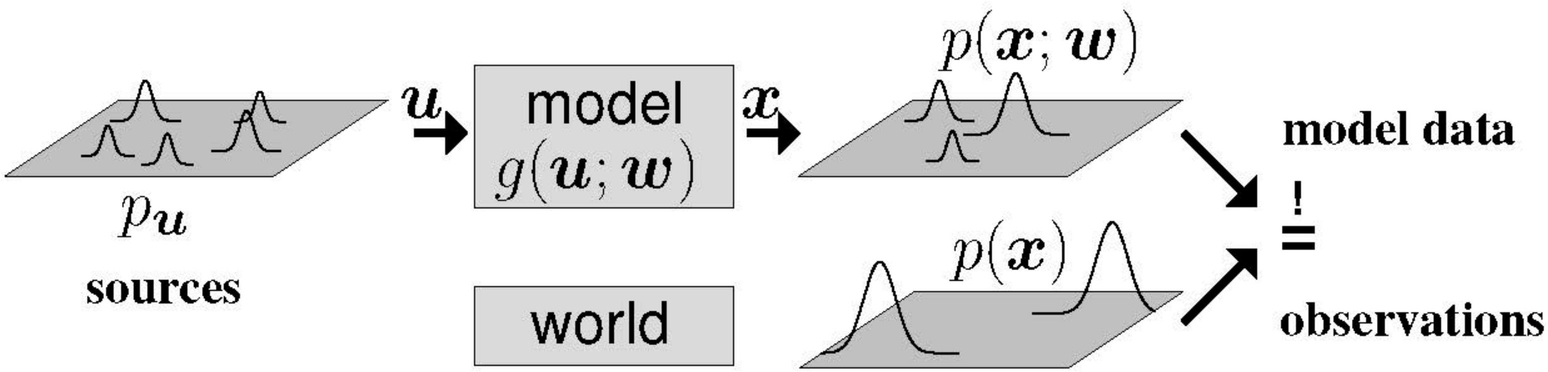
- Independent Component Analysis (ICA): projection into a space with statistically indpendent components (factorial code)
  →often characteristics of a factorial distribution are optimized:
  - maximal entropy (given variance)
  - cummulants

  →or prototype distributions should be matched:
  - product of special super-Gaussians

- Projection Pursuit: components are maximally non-Gaussian

Generative Models

a generative model simulates the world and produces the same data



Generative Models

- data generation process is probabilistic: underlying distribution

- generative model attempts at approximation this distribution

- loss function the distance between model output distribution and the distribution of the data generation process

- examples: factor analysis, latent variable models, Boltzmann machines, hidden Markov models

Parameter Estimation

Generative models estimate the optimal parameter given a parametrized model class

Data are generated from a model of the class: **find this model**

- model class known
- task: estimate actual parameters
- loss: difference between true and estimated parameter
- evaluate estimator: expected loss

Mean Squared Error, Bias, and Variance

Theoretical concepts of parameter estimation

- training data: $\{\boldsymbol{x}\} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\}$

  simply $\boldsymbol{X} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l)^T$ (the matrix of training data)

- true parameter vector: $\boldsymbol{w}$

- estimate of $\boldsymbol{w}$: $\hat{\boldsymbol{w}}$

Mean Squared Error, Bias, and Variance

- **unbiased** estimator: $$\mathrm{E}_{\boldsymbol{X}} \hat{\boldsymbol{w}} \ = \ \boldsymbol{w}$$

  on average (over training set) the true parameter is obtained

- **bias**: $$b(\hat{\boldsymbol{w}}) \ = \ \mathrm{E}_{\boldsymbol{X}} \hat{\boldsymbol{w}} \ - \ \boldsymbol{w}$$

- **variance**: $$\mathrm{var}(\hat{\boldsymbol{w}}) \ = \ \mathrm{E}_{\boldsymbol{X}} \left((\hat{\boldsymbol{w}} \ - \ \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\hat{\boldsymbol{w}} \ - \ \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) \right)$$

- **mean squared error** (MSE, different to supervised loss):

$$\mathrm{mse}(\hat{\boldsymbol{w}}) \ = \ \mathrm{E}_{\boldsymbol{X}} \left((\hat{\boldsymbol{w}} \ - \ \boldsymbol{w})^T (\hat{\boldsymbol{w}} \ - \ \boldsymbol{w}) \right)$$

  expected squared error between the estimated and true parameter

Objective: **minimize MSE!**

Mean Squared Error, Bias, and Variance

$$\boxed{\mathrm{mse}(\hat{\boldsymbol{w}})} = \mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \boldsymbol{w})^T (\hat{\boldsymbol{w}} - \boldsymbol{w}) \right) =$$

$$\mathrm{E}_{\boldsymbol{X}}\left(((\hat{\boldsymbol{w}} - \underbrace{\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) + (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})}_{} - \boldsymbol{w}))^T \right.$$

zero

$$\left. ((\hat{\boldsymbol{w}} - \underbrace{\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) + (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})}_{} - \boldsymbol{w})) \right) =$$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) - \right.$$

$$2\ (\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) +$$

$$\left. (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})^T (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) \right) =$$

Only $\hat{\boldsymbol{w}}$ depends on $\boldsymbol{X}$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) \right) +$$

$$(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})^T (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) =$$

$$\boxed{\mathrm{var}(\hat{\boldsymbol{w}}) + b^2(\hat{\boldsymbol{w}})}$$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) \right) =$$

$$(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) = 0$$

Maximum Likelihood

- One of the major objectives if learning generative models
- It has certain theoretical properties
- Theoretical concepts like efficient estimator or biased estimator are introduced
- Even supervised methods can be viewed as special case of maximum likelihood
- ML is asymptotically efficient and unbiased
- ML does everything right and this efficiently (enough data)

Maximum Likelihood Estimator

The likelihood $\mathcal{L}$ of the data set $\{\boldsymbol{x}\} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\}$:

$$\mathcal{L}(\{\boldsymbol{x}\};\boldsymbol{w}) = p(\{\boldsymbol{x}\};\boldsymbol{w})$$

probability of the model $p(\boldsymbol{x};\boldsymbol{w})$ to produce the data

iid (independent identical distributed) data:

$$\mathcal{L}(\{\boldsymbol{x}\};\boldsymbol{w}) = p(\{\boldsymbol{x}\};\boldsymbol{w}) = \prod_{i=1}^{l} p(\boldsymbol{x}^i;\boldsymbol{w})$$

Negative log-likelihood:

$$-\ln \mathcal{L}(\{\boldsymbol{x}\};\boldsymbol{w}) = -\sum_{i=1}^{l} \ln p(\boldsymbol{x}^i;\boldsymbol{w})$$

Properties of Maximum Likelihood Estimator

MLE:

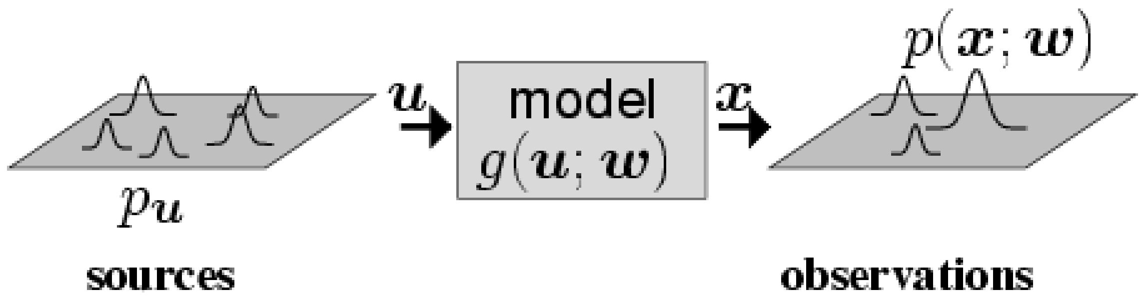
- invariant under parameter change
- asymptotically unbiased and efficient → asymptotically optimal
- consistent for zero CRLB

Expectation Maximization

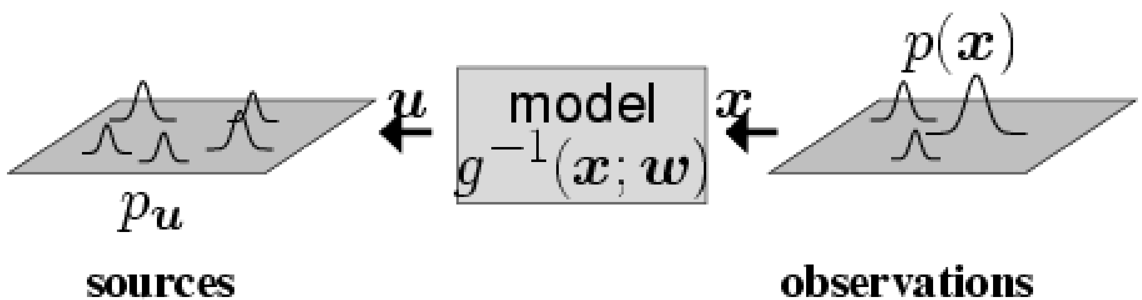
likelihood can be optimized by gradient descent methods

likelihood cannot be computed analytically:
- hidden states
- many-to-one output mapping
- non-linearities

Generative Model



Likelihood



$$p(x; w) = \int_U p_u(u)\, \delta(x = g(u; w))\, du$$

Expectation Maximization

- hidden variables, latent variables, unobserved variables u
- likelihood is determined by all u mapped to x

Expectation Maximization

BIOINF

Expectation Maximization (EM) algorithm:

- joint probability $p(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{w})$ is easier to compute than likelihood
- estimate $p(\boldsymbol{u} \mid \boldsymbol{x}; \boldsymbol{w})$ by $Q(\boldsymbol{u} \mid \boldsymbol{x})$

$$\boxed{\ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w})} = \ln p(\{\boldsymbol{x}\}; \boldsymbol{w}) = \ln \int_U p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} =$$

$$\ln \int_U \frac{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})} p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} \boxed{\geq}$$

Jensen's inequality

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})} \, d\boldsymbol{u} =$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} -$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \, d\boldsymbol{u} =$$

$$\boxed{\mathcal{F}(Q, \boldsymbol{w})}$$

Expectation of log joint probability is easy for exponential family

Expectation Maximization

EM algorithm is an iteration between E-step and M-step:

E-step:

$$Q_{k+1} = \arg\max_{Q} \mathcal{F}(Q, \boldsymbol{w}_k)$$

M-step:

$$\boldsymbol{w}_{k+1} = \arg\max_{\boldsymbol{w}} \mathcal{F}(Q_{k+1}, \boldsymbol{w})$$

Expectation Maximization

EM increases the lower bound in both steps

Beginning of the M-step: $\mathcal{F}(Q_{k+1}, \boldsymbol{w}_k) \; = \; \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_k)$

E-step does not change the parameters

$$\ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_k) \; = \; \mathcal{F}(Q_{k+1}, \boldsymbol{w}_k) \; \leq \; \mathcal{F}(Q_{k+1}, \boldsymbol{w}_{k+1}) \; \leq \; \mathcal{F}(Q_{k+2}, \boldsymbol{w}_{k+1}) \; = \; \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_{k+1})$$

EM algorithm:

- hidden Markov models
- mixture of Gaussians
- factor analysis
- independent component analysis

Maximum Entropy

maximum entropy probability distribution:

- maximal entropy given a class of distributions
- minimal prior assumptions
- physical systems converge to maximal entropy configurations
- most likely observed solution
- connection: statistical mechanics and information theory

principle of maximum entropy first expounded by E.T. Jaynes in 1957

Maximum Entropy

Entropy $$H = - \sum_{k \geq 1} p_k \log p_k$$

$$p_k \log p_k = 0 \text{ for } p_k = 0$$

Examples:

- normal distribution: given mean and standard deviation
- uniform distribution: supported in the interval $[a, b]$
- exponential distribution: given mean in $[0, \infty]$

Maximum Entropy

Not all classes of distributions contain a maximum entropy distribution:

- arbitrarily large entropy: distributions with mean
- entropies of a class are bounded from above but not attained: distributions with mean zero, second moment one, and third moment one

Maximum Entropy Solution

Constraints:

$$\sum_{i=1}^{n} p(x_i)\ f_k(x_i) \ = \ F_k \qquad k = 1, \ldots, m$$

$$\sum_{i=1}^{n} p(x_i) \ = \ 1$$

Solution, the Gibbs distribution

$$p(x_i) \ = \ \frac{1}{Z(\lambda_1, \ldots, \lambda_m)} \ \exp\left(\lambda_1 f_1(x_i) \ + \ \cdots \ + \ \lambda_m f_m(x_i)\right)$$

with partition function

$$Z(\lambda_1, \ldots, \lambda_m) \ = \ \sum_{i=1}^{n} \exp\left(\lambda_1 f_1(x_i) \ + \ \cdots \ + \ \lambda_m f_m(x_i)\right)$$

The Lagrange multipliers are determined by the equation system

$$F_k \ = \ \frac{\partial}{\partial \lambda_k} \log Z(\lambda_1, \ldots, \lambda_m)$$