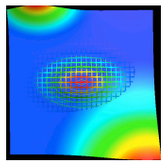


Data Mining algorithms

2017-2018 spring

03.14-21.2016

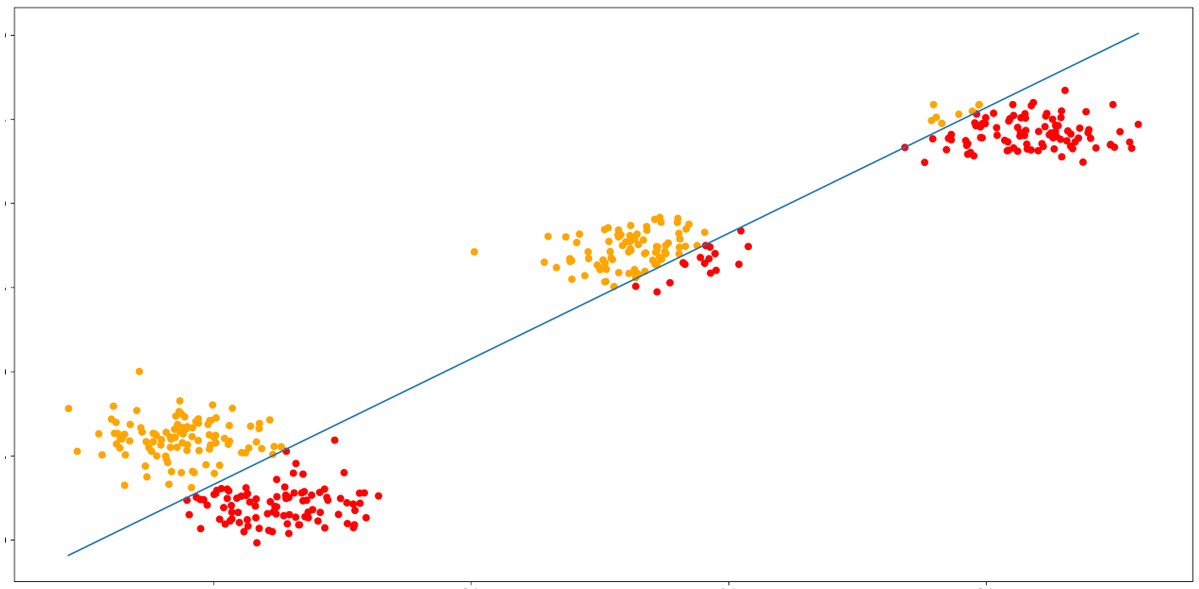
1. Linear separators in python
2. Clustering
3. GMM
4. MRF

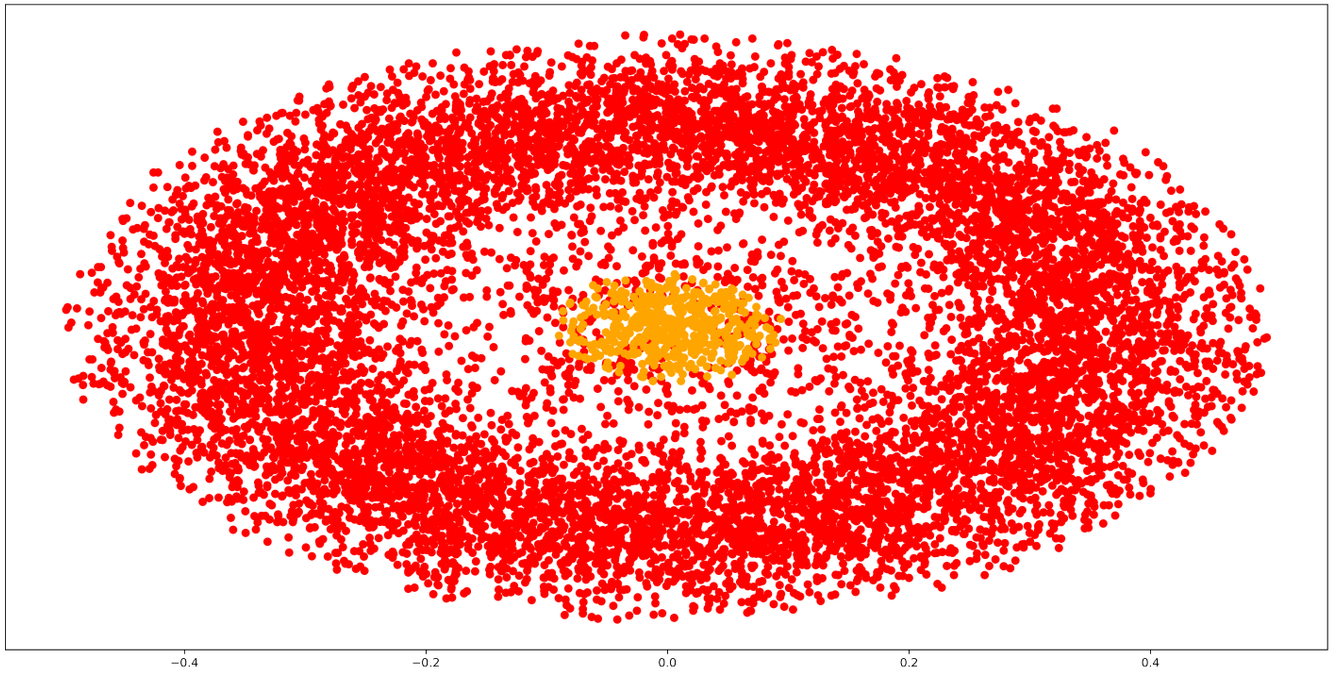
Plan

W1 Februar 7-9: Introduction, kNN, evaluation
W2 Februar 14-16: Evaluation, Decision Trees
W3 Februar 21-23: Linear separators, iPython, VC theorem
W4 Februar 28-march 2: Linear separators, iPython, maximal margin
W5 March 7-9: SVM, VC theorem and Bottou-Bousquet
W6 March 14-16: clustering (hierarchical, density based etc.), GMM, MRF, Apriori and association rules
W7 March 21-23: Recommender systems and generative models
W8 March 28-30: basics of neural networks, Sontag-Maas-Bartlett theorems, Bayes networks
W9 April 4-6: holiday
W10 April 11-13: BN, CNN, MLP, Dropout, Batch normalization
W11 April 18-20: midterm, RNN
W12 April 25-27: LSTM, GRU, attention, Image caption, Turing Machine
W13 May 2-4: RBM, DBN, VAE, GAN
W14 May 9-11: Boosting, Time series
W14 May 16-18: TS, Projects on Friday

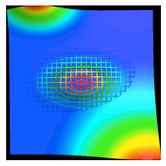
Ipython and linear separators

check out logreg.ipynb

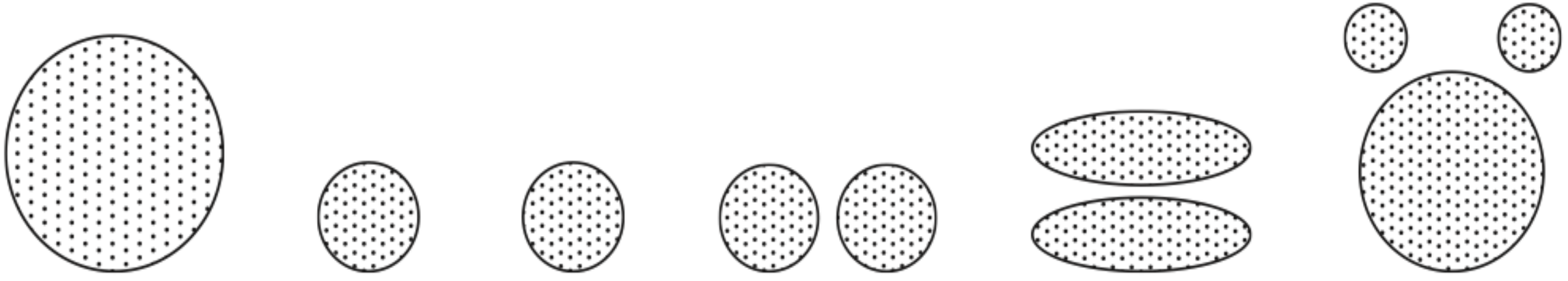




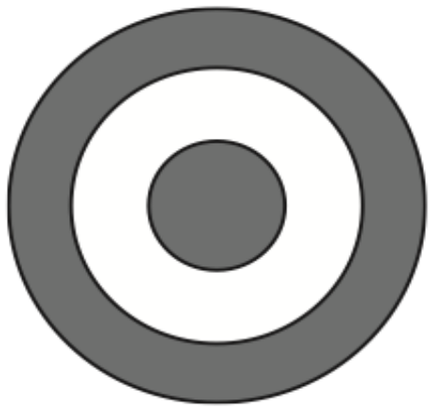
svm_kernels.ipynb



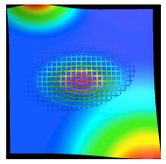
How will perform a lnkage, density or a k-means?



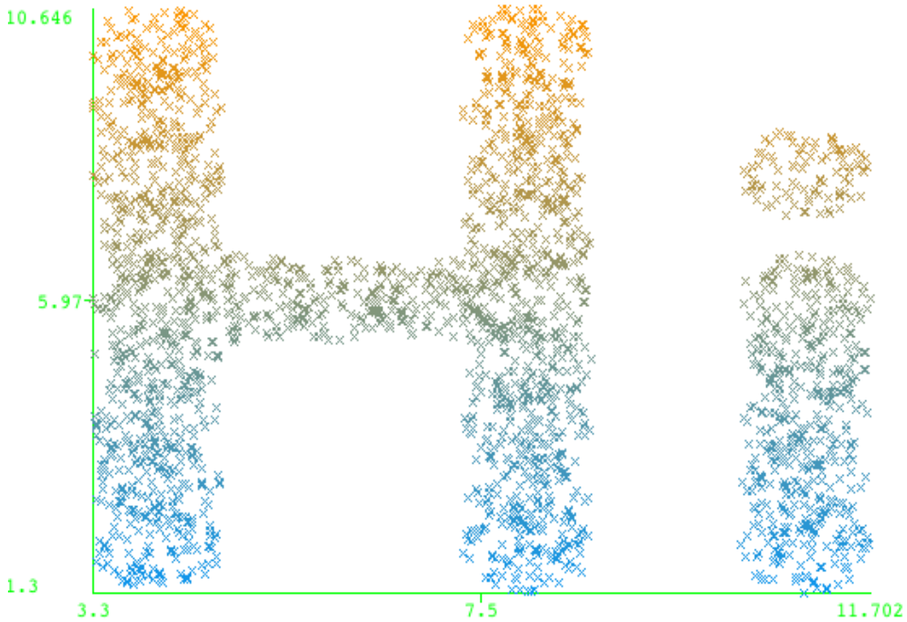
a) b) c) d) e)

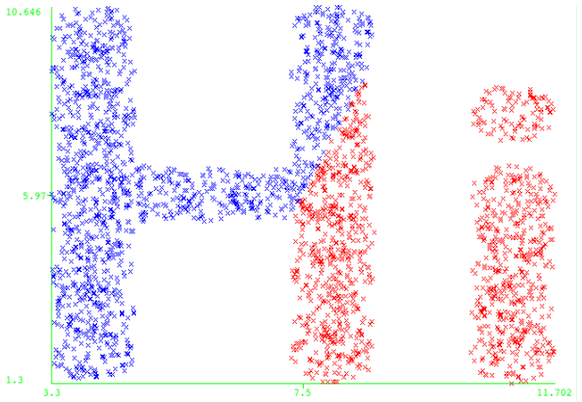


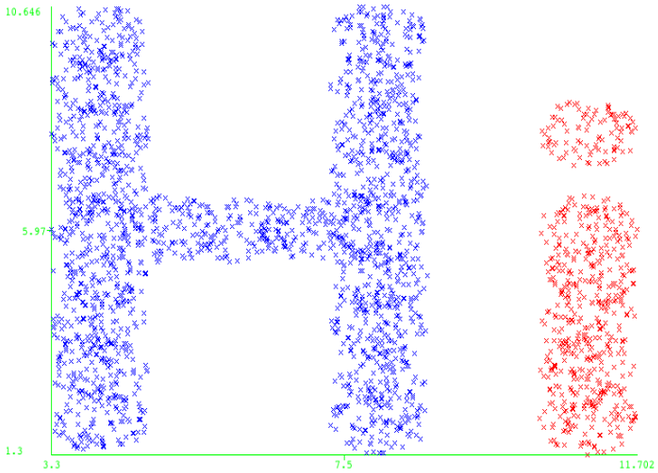
f)

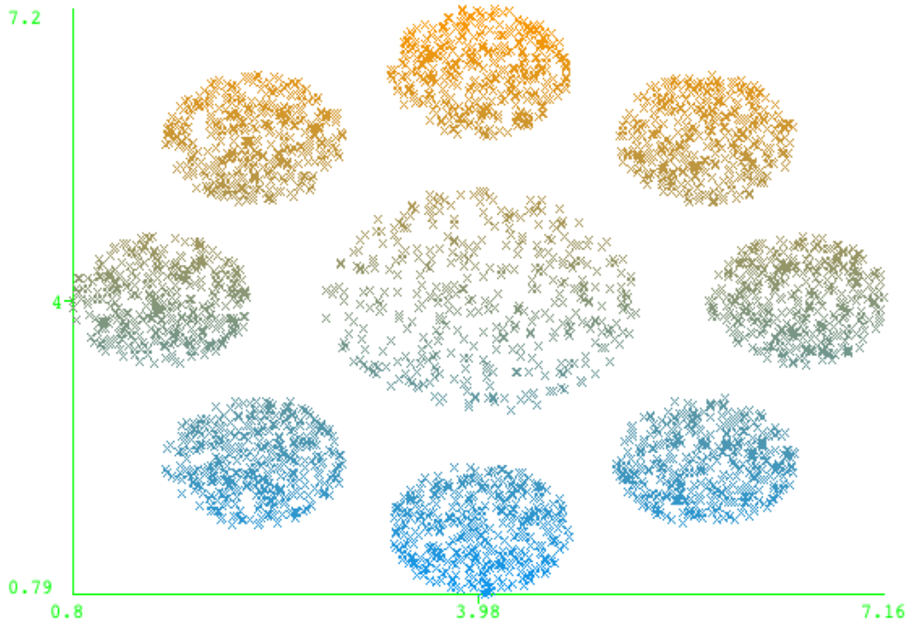


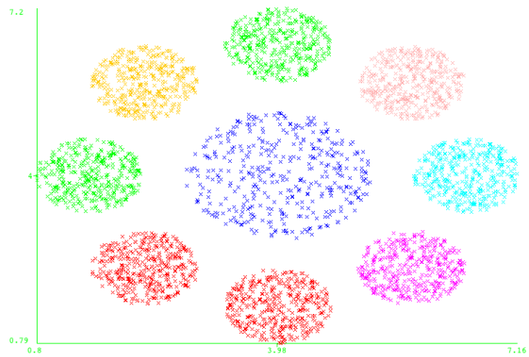
Clustering with weka

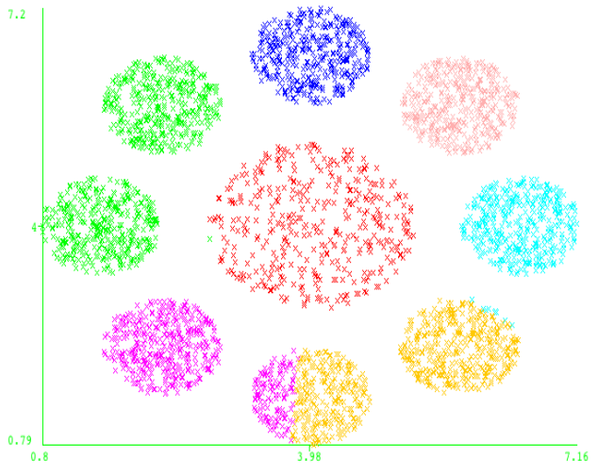


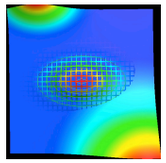




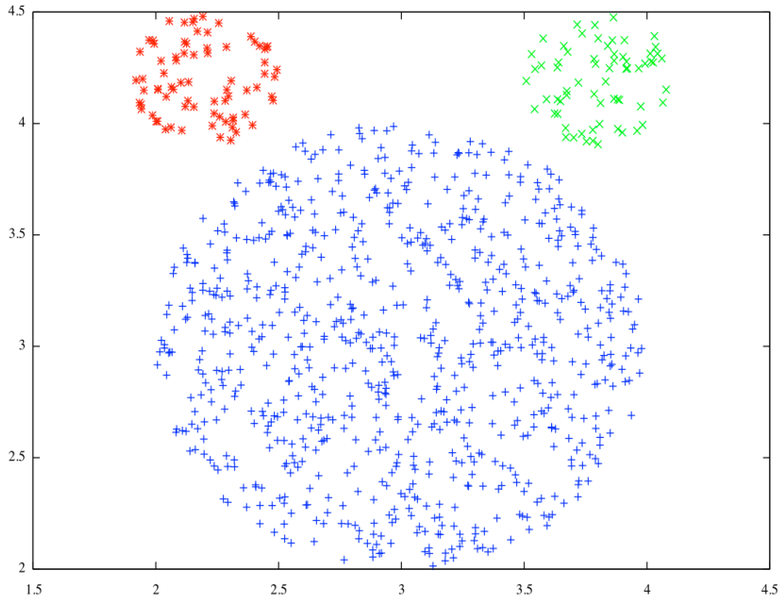




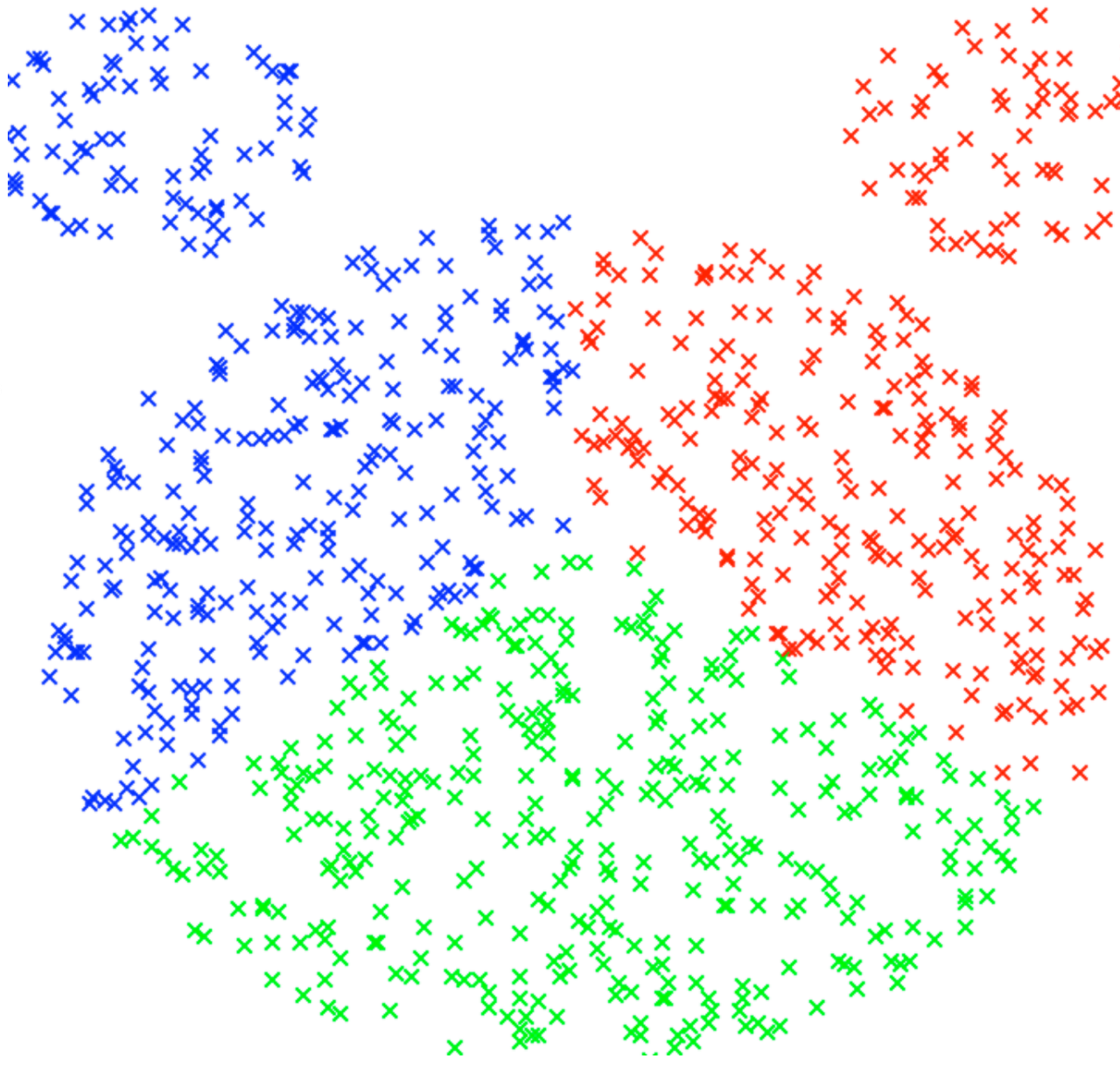


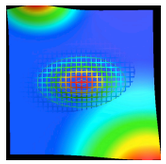


Mickey mouse



Mickey.arff





Generative models

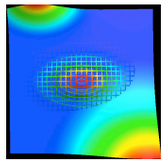
Statistical analysis: determine a probabilistic model to fit a known set of observations.

Formally, we have a set of observations $X = \{x_1, .., x_T\}$ in R^d and a probability density function (pdf) as

$$p(x \mid \theta)$$

where we even presumed that the model is parametric hence the formula (sometimes with “;” not with pipe)

Hence the name: $\theta = \{\theta_1, .., \theta_N\}$ is the parameter set of the density function.



Generative models

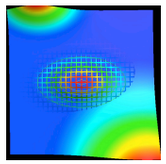
Now let us define the likelihood function to be equal to the probability of observing our sample set X:

$$\mathcal{L}(\theta \mid X = \{x_1, .., x_T\}) = p(X \mid \theta).$$

Our main goal is to estimate the parameter set which maximizing the likelihood function or the natural logarithm of it (log-likelihood) over X, formally

$$\hat{\theta}_{mle} = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta \mid X) = \arg\max_{\theta \in \Theta} \ln \mathcal{L}(\theta \mid X)$$

where we think of X as a constant.



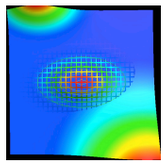
Generative models

This optimization problem is the so-called Maximum Likelihood Estimation (MLE).

If our density function is simple enough, we can calculate the parameters analytically by setting the derivative of the log-likelihood to zero.

Unfortunately, there are important and widely used models where we cannot solve the derivative directly and therefore we need more refined methods to estimate the parameters.

One of them is the Expectation-Maximization [Dempster et al., 1977].

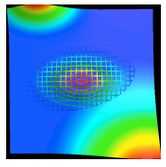


Expectation Maximization

By the EM algorithm we assume that either our set of known observations or our model parameter set has missing latent variables or values.

The EM method is an iterative algorithm:

1. E-step: calculate the expected value of the latent variables using the current estimation of the parameters
2. M-step: we calculate the parameters which maximize the estimated likelihood over the known observations.



Expectation Maximization

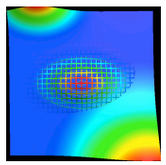
We usually think of the known observations (or the training set) X = $\{x_1, .., x_T\}$ as independent samples drawn from the same distribution, thus the joint probability is

$$p(X = \{x_1, .., x_T\} \mid \theta) = \Pi_{t=1}^{T} p(x_t \mid \theta)$$

Now, let us assume that the missing set of random variables Y exists

-> we define the complete pdf and therefore the complete likelihood as

$$\mathcal{L}(\theta \mid X, Y) = p(X, Y \mid \theta) = p(Y \mid X, \theta) p(X \mid \theta)$$

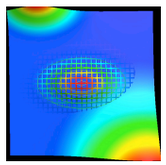


Expectation Maximization

With the left side and the first part of the right side we assume a joint relationship between the missing, latent variables and the known observations.

If we think of Y as a random variable drawn from an underlying distribution, we can define the following supplementary function:

$$Q(\theta, \theta^{(i-1)}) = \mathbf{E}_{Y|X,\theta^{(i-1)}}[\log p(X, Y \mid \theta)]$$
$$= \int_{y \in Y} p(y \mid X, \theta^{(i-1)}) \log p(X, Y \mid \theta) dy$$



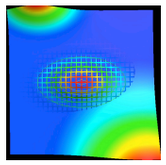
Expectation Maximization

$$Q(\theta, \theta^{(i-1)}) = \mathbf{E}_{Y|X,\theta^{(i-1)}}[\log p(X, Y \mid \theta)]$$
$$= \int_{y \in Y} p(y \mid X, \theta^{(i-1)}) \log p(X, Y \mid \theta) dy$$

The expected value of the complete log-likelihood over Y drawn from a distribution

$$p(y \mid X, \theta^{(i-1)})$$

parametrized by the previous (thus a constant) estimation of the parameters ($\theta^{(i-1)}$) and X, another constant.



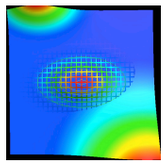
Expectation Maximization

$$Q(\theta, \theta^{(i-1)}) = \mathbf{E}_{Y|X,\theta^{(i-1)}}[\log p(X, Y \mid \theta)]$$

$$= \int_{y \in Y} p(y \mid X, \theta^{(i-1)}) \log p(X, Y \mid \theta) dy$$

With $Q(\theta, \theta^{(i-1)})$ we have a more manageable function to calculate the next estimation of the parameters:

$$\theta^{(i)} = \arg\max_{\theta \in \Theta} Q(\theta, \theta^{(i-1)})$$



Expectation Maximization

Summary:

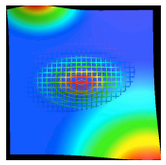
E-step:

$$p(y \mid X, \theta^{(i-1)})$$

M-step:

$$\theta^{(i)} = \arg\max_{\theta \in \Theta} Q(\theta, \theta^{(i-1)})$$

It can be proved that this two-step procedure is guaranteed not to decrease the original likelihood and converge to an unfortunately local maximum [Dempster et al., 1977, McLachlan and Krishnan, 2007].



Gaussian Mixture Model

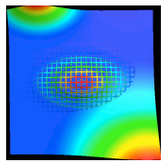
Approximation with a single normal distribution?

1. Poor approximation quality
2. Can prefer observations not in the original sample population

Idea?

Expanding to mixture distributions!

If the number of mixture distributions is finite -> Gaussian Mixture Model



Gaussian Mixture Model

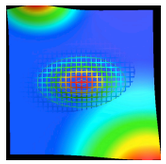
Formally, let N be the number of Gaussian distributions, each in R^d and their positive mixing weights $\omega = \{\omega_1, \omega_2, .., \omega_N\}$ with $N_i=1\ \omega_i = 1$.

Pdf:

$$p(x \mid \Theta) = \sum_{i=1}^{N} \omega_i g_i(x)$$

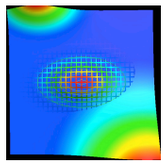
where $\Theta = \{\omega_1,..,\omega_N,\mu_i,..,\mu_N,\Sigma_1,..,\Sigma_N\}$ are the parameters of the mixture and the i-th d-dimensional multivariate normal distribution is

$$g_i(x) = \frac{1}{\sqrt{(2\Pi)^d \mid \Sigma_i \mid}} \exp^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$



Gaussian Mixture Model

Parameters of our mixture model?

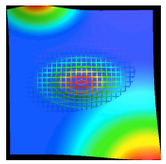


Gaussian Mixture Model

Parameters of our mixture model?

Unfortunately, in practice the number of parameters of our mixture distribution could be really huge.

What are the parts?



Gaussian Mixture Model

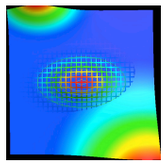
Parameters of our mixture model?

Unfortunately, in practice the number of parameters of our mixture distribution could be really huge.

What are the parts?

If we assume a d-dimensional underlying vector space, our parameter set has three parts:

1. $\omega = \{\omega_1 , .., \omega_N \}$ is an N -dimensional real vector
2. $\mu = \{\mu_1, .., \mu_N \}$ is a set of d-dimensional mean vectors
3. $\Sigma = \{\Sigma_1, .., \Sigma_N \}$ is a set of N covariance matrices each with d^2 elements.



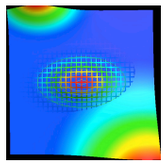
Gaussian Mixture Model

We can reduce the latter -> Nd with diagonal covariance matrices (isotropic Gaussians)

Overall: cardinality(Θ) :=| Θ |= N (1 + 2d)

Vs. k-means?

How to find an element analyticaly?



Gaussian Mixture Model

We can reduce the latter -> Nd with diagonal covariance matrices (isotropic Gaussians)

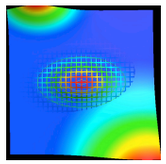
Overall: cardinality(Θ) :=| Θ |= N (1 + 2d)

Vs. k-means?

How to find an element analyticaly?

Derivation? :(

EM ☺️



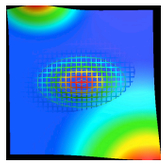
Gaussian Mixture Model

We need a latent, but computable estimation of the probability og the latent variables:

Adjuvant proportion or the the membership probability:

$x_t \in X$ and the i-th Gaussian as

$$\gamma_i(x_t) = \frac{\omega_i g_i(x_t)}{\sum_{j=1}^{N} \omega_j g_j(x_t)}$$



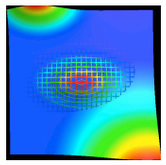
Gaussian Mixture Model

Interpretation:

the probability that sample x_t was generated by the i-th Gaussian distribution

due to the fact that $N_i\ \gamma_i(x_t) = 1$ for all x.

During the E-step we estimate the membership probabilities for the observations using the actual parameters.



Gaussian Mixture Model

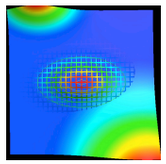
M-step:

use these expected values to determine a better estimation of the parameters.

The smoothness property of the Gaussian Mixtures (and for all the density functions) allow us to optimize over the natural logarithm of the likelihood instead of the likelihood:

$$\mathcal{L}(X) = \log p(X \mid \Theta) = \log \Pi_{t=1}^{T} p(x_t \mid \Theta) = \sum_{t=1}^{T} \log p(x_t \mid \Theta)$$

Gradient?

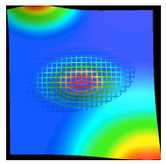


Gaussian Mixture Model

$$\mathcal{L}(X) = \log p(X \mid \Theta) = \log \Pi_{t=1}^{T} p(x_t \mid \Theta) = \sum_{t=1}^{T} \log p(x_t \mid \Theta)$$

Gradient:

$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

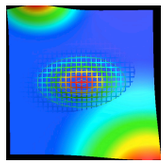


Gaussian Mixture Model

$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

Gradient for the weight:

$$\frac{\partial \mathcal{L}(X)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{1}{p(x_i \mid \Theta)} \frac{\partial p(x_i \mid \Theta)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{1}{\sum_{j=1}^{N} \omega_j g_j(x_t)} \frac{\partial \sum_{j=1}^{N} \omega_j g_j(x)}{\partial \omega_i}$$



Gaussian Mixture Model

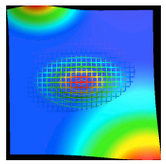
$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

Gradient for the weight:

$$\frac{\partial \mathcal{L}(X)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{1}{p(x_i \mid \Theta)} \frac{\partial p(x_i \mid \Theta)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{1}{\sum_{j=1}^{N} \omega_j g_j(x_t)} \frac{\partial \sum_{j=1}^{N} \omega_j g_j(x)}{\partial \omega_i}$$

$$= \sum_{t=1}^{T} \frac{g_i(x)}{\sum_{j=1}^{N} \omega_j g_j(x_t)}$$

Connection with the membership prob.?

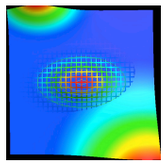


Gaussian Mixture Model

$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

There is a straightforward connection between the membership probability and our gradient:

$$\frac{\partial \mathcal{L}(X)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{g_i(x_t)}{\sum_{j=1}^{N} \omega_j g_j(x_t)} = \sum_{t=1}^{T} \frac{\gamma_i(x_t)}{\omega_i}$$



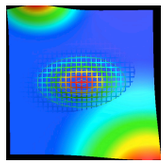
Gaussian Mixture Model

$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

The rest of the gradient vector respect to the mean, under assumption of diagonal covariance matrices (isotropic Gaussian):

$$\frac{\partial \mathcal{L}(X)}{\partial \mu_{id}} = \sum_{t=1}^{T} \frac{\omega_i}{\sum_{j=1}^{N} \omega_j g_j(x_t)} \frac{\partial g_i(x_t)}{\partial \mu_{id}} = \sum_{t=1}^{T} \frac{\omega_i g_i(x_t)}{\sum_{j=1}^{N} \omega_j g_j(x_t)} \frac{(\mu_{id} - x_{td})}{\sigma_{id}^2}$$

$$= \sum_{t=1}^{T} \gamma_i(x_t) \frac{(\mu_{id} - x_{td})}{\sigma_{id}^2}.$$



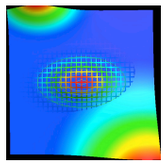
Gaussian Mixture Model

$$\frac{\partial \mathcal{L}(X)}{\partial \theta_i} = \sum_{t=1}^{T} \frac{1}{p(x_t \mid \Theta)} \frac{\partial p(x_t \mid \Theta)}{\partial \theta_i}$$

Similarly, the gradient vector respect to the variance, under assumption of diagonal covariance matrices (isotropic Gaussian):

$$\frac{\partial \mathcal{L}(X)}{\partial \sigma_{id}} = \sum_{t=1}^{T} \frac{\omega_i}{\sum_{j=1}^{N} \omega_j g_j(x_t)} \frac{\partial g_i(x_t)}{\partial \sigma_{id}}$$

$$= \sum_{t=1}^{T} \gamma_i(x_t) \left(\frac{(x_{td} - \mu_{id})^2}{\sigma_{id}^3} - \frac{1}{\sigma_{id}} \right).$$



Gaussian Mixture Model

Summary:

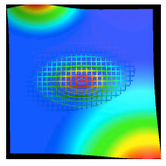
1. Set the parameters of GMM by random.
2. E-step: Estimate the membership probabilities after k iterations:

$$\gamma_i^{(k)}(x_t) = \frac{\omega_i^{(k-1)} g_i^{(k-1)}(x_t)}{\sum_{j=1}^{N} \omega_j^{(k-1)} g_j^{(k-1)}(x_t)}$$

3. M-step: Set the gradients to zero:

$$\mu_{id}^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t) x_{td}}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)} \quad \sigma_{id}^{(k)} = \sqrt{\frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)(x_{td} - \mu_{id}^{(k)})^2}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}}$$

4. Go back to 2



Gaussian Mixture Model

Summary:

1. Set the parameters of GMM by random.
2. E-step: Estimate the membership probabilities after k iterations:

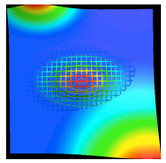
$$\gamma_i^{(k)}(x_t) = \frac{\omega_i^{(k-1)} g_i^{(k-1)}(x_t)}{\sum_{j=1}^{N} \omega_j^{(k-1)} g_j^{(k-1)}(x_t)}$$

3. M-step: Set the gradients to zero:

Are we done?

$$\mu_{id}^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t) x_{td}}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)} \qquad \sigma_{id}^{(k)} = \sqrt{\frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)(x_{td} - \mu_{id}^{(k)})^2}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}}$$

4. Go back to 2



Gaussian Mixture Model

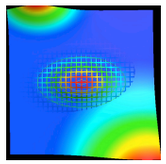
What was missing?

The mixture parameter is tricky, setting to zero:

$$\frac{\partial \mathcal{L}(X)}{\partial \omega_i} = \sum_{t=1}^{T} \frac{g_i(x_t)}{\sum_{j=1}^{N} \omega_j g_j(x_t)} = \sum_{t=1}^{T} \frac{\gamma_i(x_t)}{\omega_i}$$

Ultimately (as an approx.), the formula to update the mixture weights is just as illustrative as the above expressions:

$$\omega_i^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}{T}$$



Gaussian Mixture Model

E-step: Estimate the membership probabilities after k iterations:

$$\gamma_i^{(k)}(x_t) = \frac{\omega_i^{(k-1)} g_i^{(k-1)}(x_t)}{\sum_{j=1}^{N} \omega_j^{(k-1)} g_j^{(k-1)}(x_t)}$$

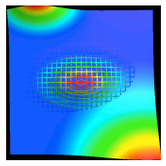
M-step: Set the gradients to zero:

$$\omega_i^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}{T}$$

$$\mu_{id}^{(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t) x_{td}}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}$$

$$\sigma_{id}^{(k)} = \sqrt{\frac{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)(x_{td} - \mu_{id}^{(k)})^2}{\sum_{t=1}^{T} \gamma_i^{(k)}(x_t)}}$$

Are we done yet?

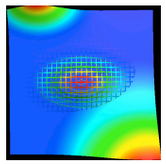


Gaussian Mixture Model

The EM algorithm will alternate between the two steps and as we mentioned in the previous section there are theoretical guarantees of convergence, hence a direct implementation will not work or will be slow in particular cases.

The main reason is that the denominator in the definition of the membership probability can easily underflow even in fp64 (64 bit precision, aka double) and especially in large dimensional spaces.

How can we overpass it?



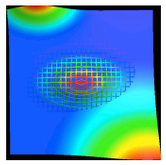
Gaussian Mixture Model

The main reason is that the denominator in the definition of the membership probability can easily underflow even in fp64 (64 bit precision, aka double) and especially in large dimensional spaces.

How can we overpass it?

One solution is to modify the expression. Let us reformulate the value $\omega_i g_i(x)$ as $e^{m_i(x)}$ where

$$m_i(x) = \ln \omega_i - \ln \sqrt{(2\Pi)^d \mid \Sigma_i \mid} - \frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$$



Gaussian Mixture Model

One solution is to modify the expression. Let us reformulate the value $\omega_i g_i(x)$ as $e^{m_i(x)}$ where

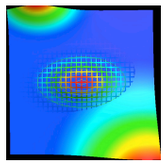
$$m_i(x) = \ln \omega_i - \ln \sqrt{(2\Pi)^d \mid \Sigma_i \mid} - \frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$$

and

$$\gamma_i(x) = \frac{e^{m_i(x)}}{\sum_{j=1}^{N} e^{m_j(x)}}$$

$$= \frac{e^{m_i(x)}}{e^{M(x)} \sum_{j=1}^{N} e^{m_i(x) - M(x)}}$$

where $M(x) = \max_j m_j(x)$ -> At least one exponent is 1!



Gaussian Mixture Model

One solution is to modify the expression. Let us reformulate the value $\omega_i g_i(x)$ as $e^{m_i(x)}$ where

$$m_i(x) = \ln \omega_i - \ln \sqrt{(2\Pi)^d \mid \Sigma_i \mid} - \frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$$

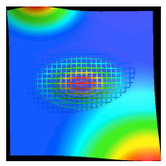
and

$$\gamma_i(x) = \frac{e^{m_i(x)}}{\sum_{j=1}^{N} e^{m_j(x)}}$$

$$= \frac{e^{m_i(x)}}{e^{M(x)} \sum_{j=1}^{N} e^{m_i(x) - M(x)}}$$

where $M(x) = \max_j m_j(x)$ -> At least one exponent is 1!

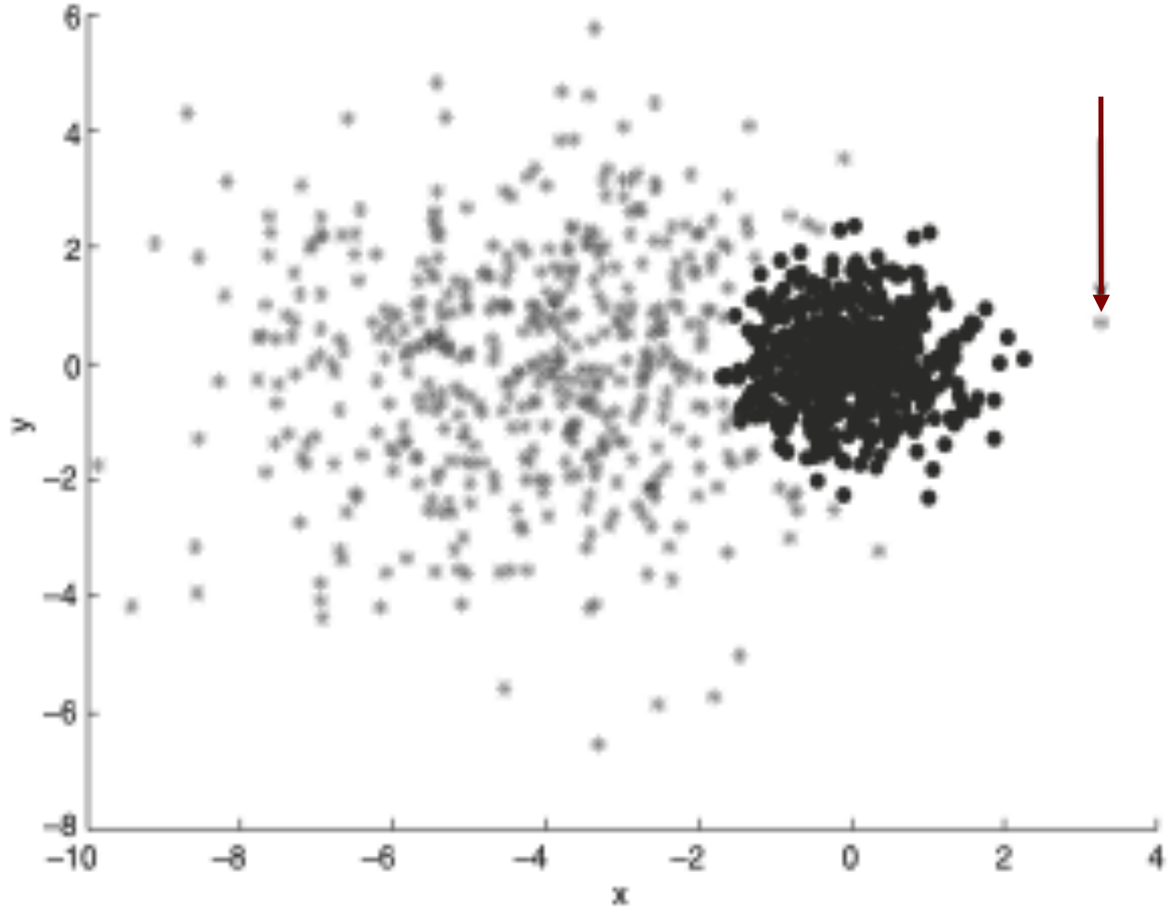
And of course: if one of them is 1 … -> others are zero ☺️



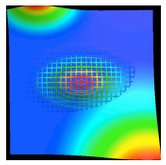
Gaussian Mixture Model

We painted the points from the first cluster to gray and to black from the second.

How can it be that the points on the right are in the gray clusters?



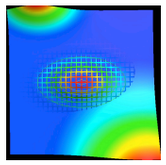
What if we use k-means?



Performance

A) There is a known cost function:
- K-means : RMSE
- GMM: loglikelihood
- DBSCAN: variance or outliers …

b) External knowledge (annotation)
- similarly to classification: F-measure …
- or DT



Performance

Mutual information:

$$MI(K,C)=\sum_{k}\sum_{j} p_{kj}\log\frac{p_{kj}}{p_k\,p_j}$$

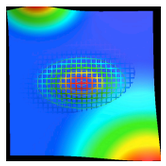
P_{kj} : the prob. of assigning to the j-th cluster while the point is originally from the k-th

Is it a good measure?

We can normalize via entropy:

$$H(K)=\sum -p_k\log p_k \qquad\qquad H(C)=\sum -p_c\log p_c$$

$$NMI(K,C)=\frac{\sum_{k}\sum_{j} p_{kj}\log\frac{p_{kj}}{p_k\,p_j}}{\frac{H(C)+H(K)}{2}}$$



Bag of Visual Words (Csurka et al 2004)

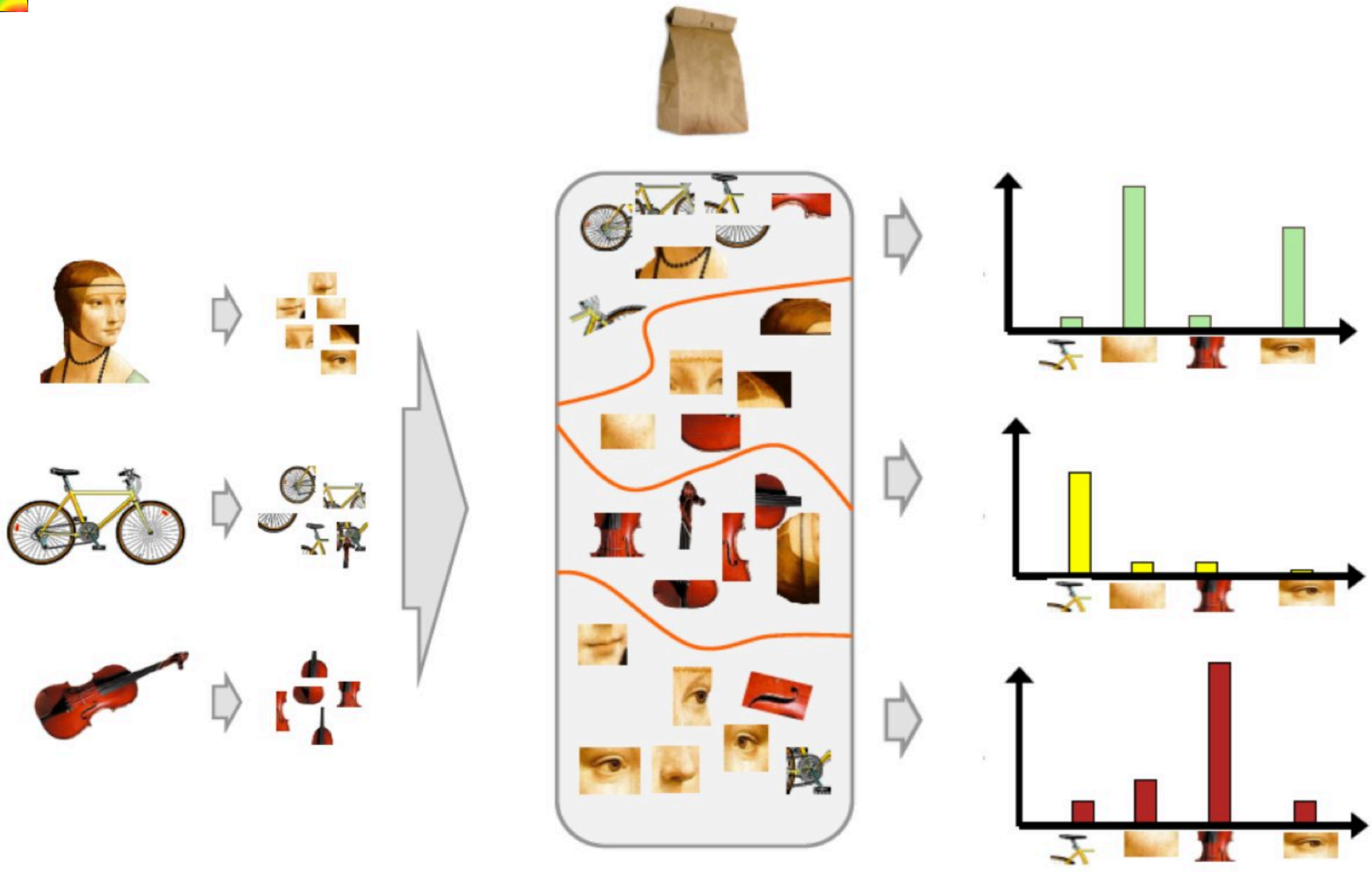
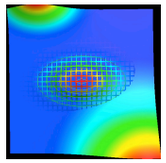
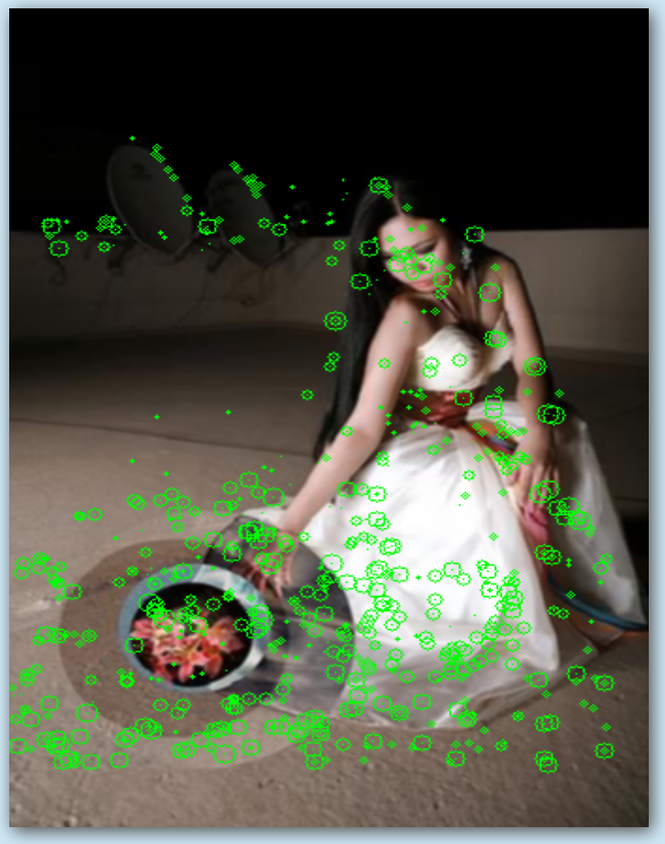


Fig.: Li Fei Fei

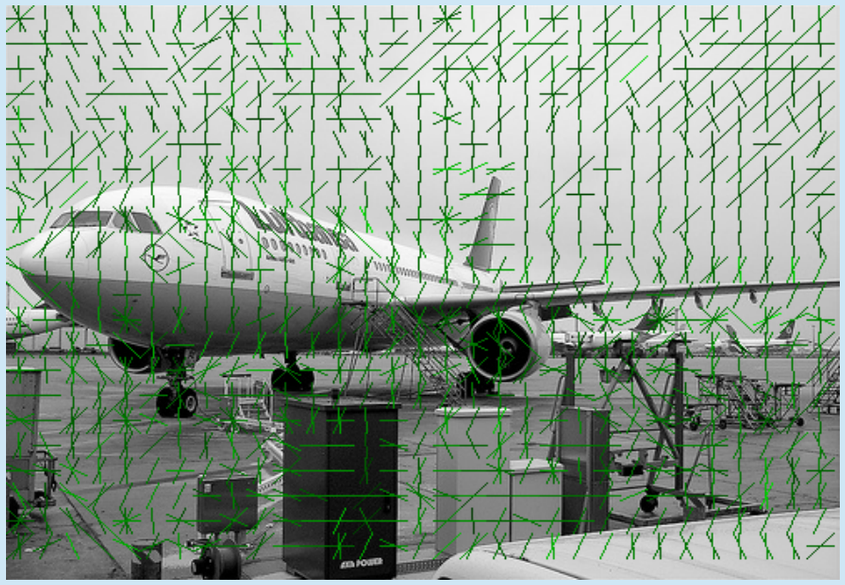


Rigid local descriptors

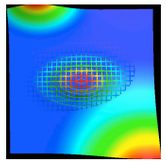
Segmentation
~ 2-300



Region of Interest
~ 1-2k

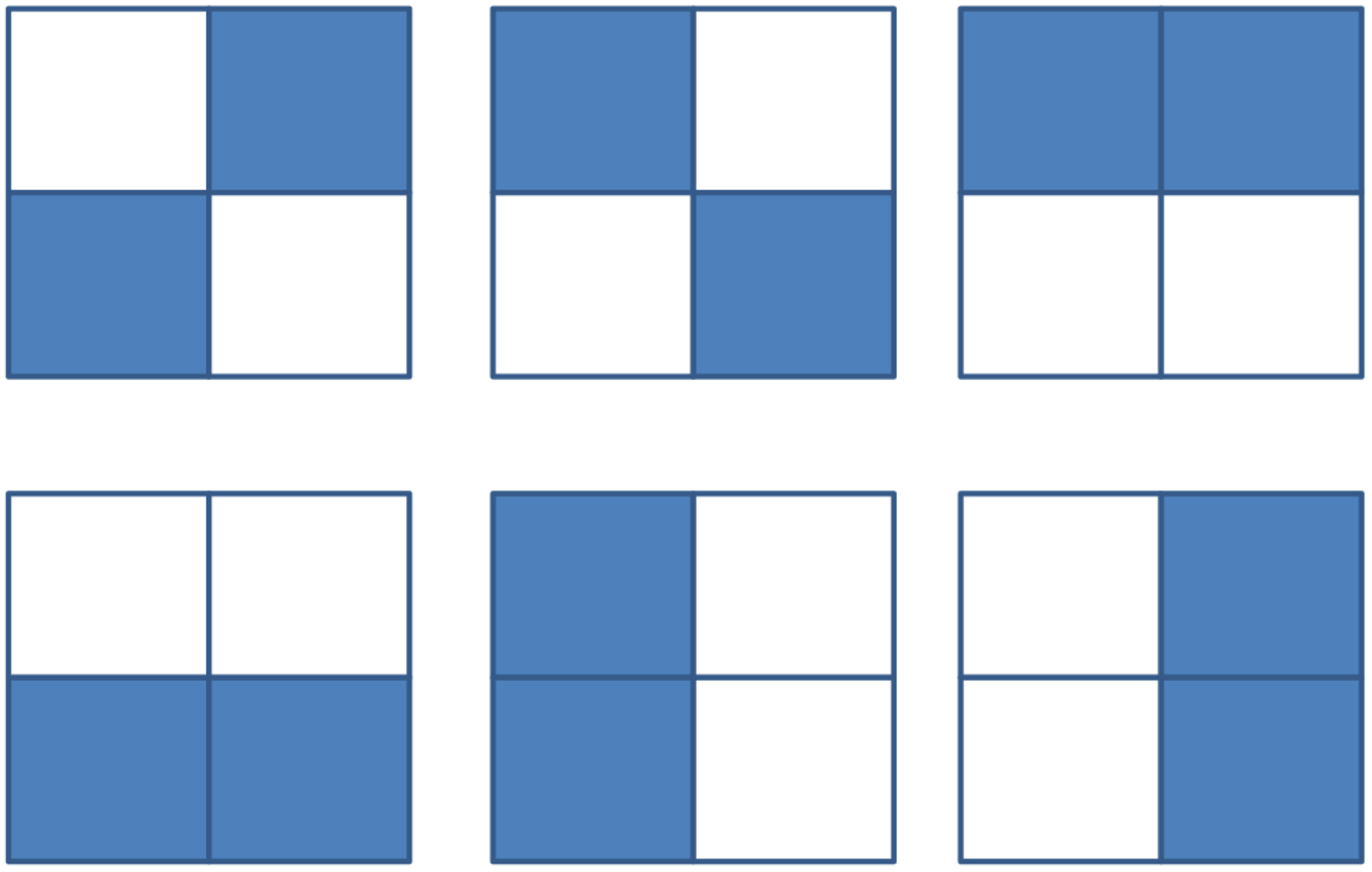


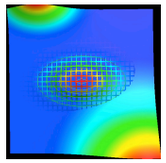
Dense Grid
~ 5k+



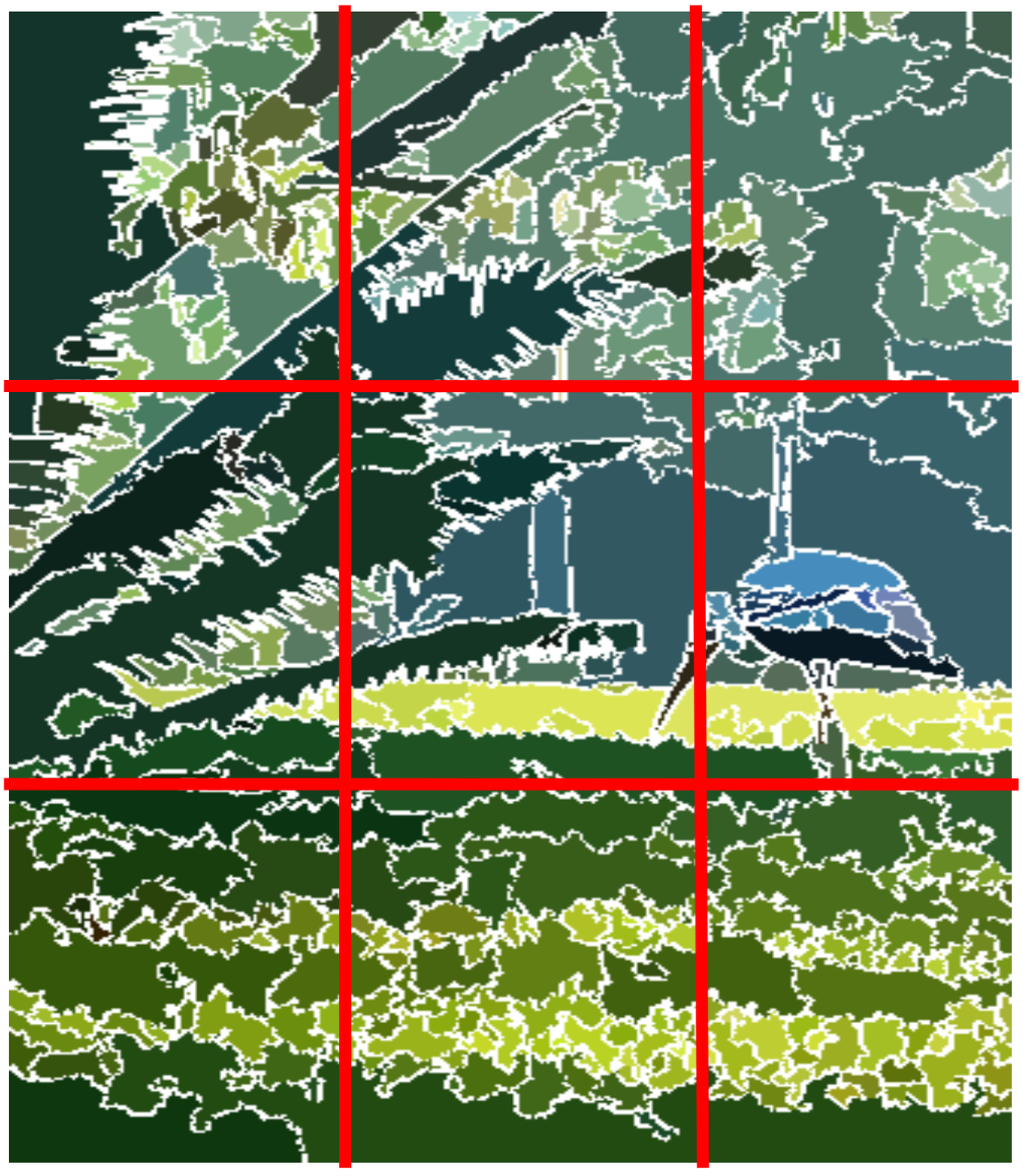
Spatiality

What will k-means or GMM do with the following images?





Spatial Pooling



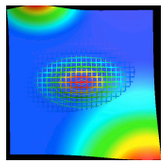
Spatial Pooling
- Rigid splits ☹
- 1x3, 2x2, 4x4

Segmentation

Even NN ☹

Back to square one:
- by GMM we assumed exchangeability

Or …. ☺

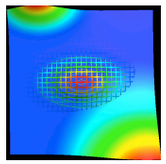


Random Fields

As we mentioned the Gaussian Mixture is powerful method to model the prior distribution of a single observation.

Nevertheless there we can easily think of structures over the samples (for example a website) or samples originated from a complicated structure of sub-samples, such as words or image patches.

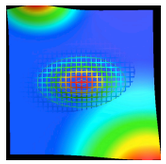
In such a case we can model the overall observation (a set of samples) as a set of random variables each drawn from a prior probability distribution.



Random Fields

If our underlying prior model is a Gaussian Mixture we assume exchangeability for the inner samples of the sample [Perronnin and Dance, 2007].

This conditional independence gives us the advantage of variability in the layout of the sub-samples, although there are some structures where the composition is significant.



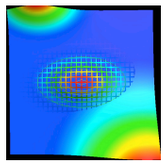
Random Fields

Let us capture the relation between the samples with a graphical model or Random Field:

- the vertices are the set of samples (random variables)
- we connect samples if there is a known connection between them

There are several kinds of Random Fields, among them are the Gaussian and the Markov Random Field.

One of the main characteristics of the Gaussian Random Field is the assumption of conditional independence between the random variables (rough interpretation is a graph without edges).



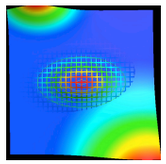
Random Fields

In comparison, by the Markov Random Field we can also capture connections between samples with an undirected graph whilst following both local and global Markov property.

Formally, let be X an observation with T corresponding observations:

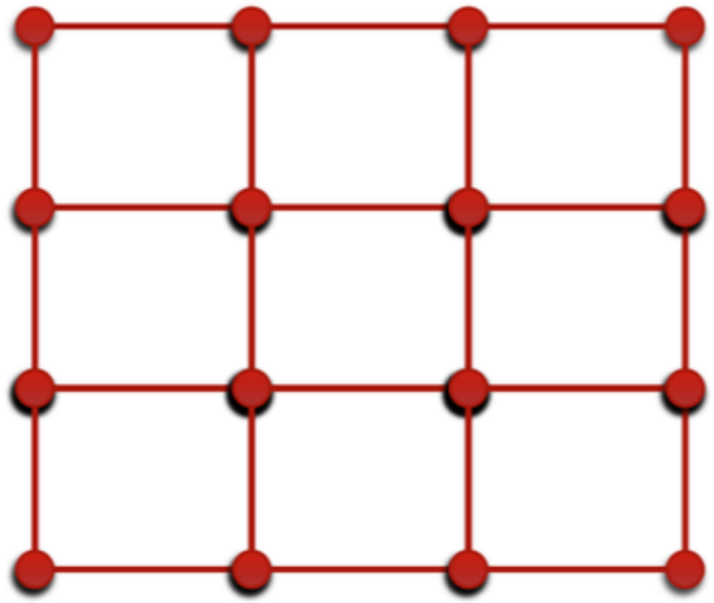
$$X = \{x_1, .., x_T\}$$

For example an image with a set of keypoints, regions or pixels [Geman and Graffigne, 1986, Szirányi et al., 2000].



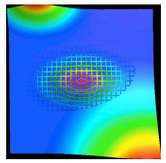
Random Fields

In our case, the Random Field has T vertices and we connect two vertices with an edge if they are neighbours according to our knowledge.

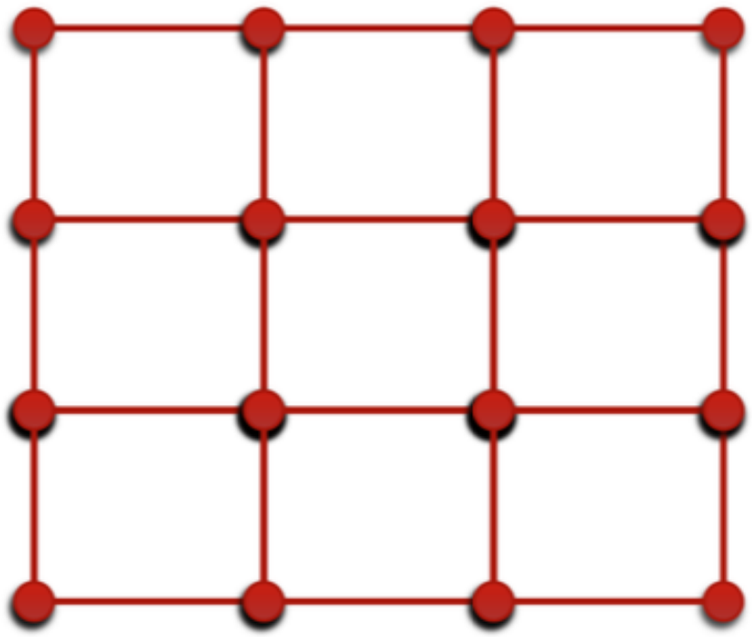


Three type of Markov properties:

- Local
- Global
- pairwise

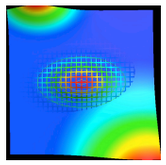


Random Fields



The local Markov property means that an observation is conditionally independent of the non-neighbour observations:

$$p(x_i|X = \{x_1, ..x_{i-1}, x_{i+1}.., x_T\}, \theta) = p(x_i|N_{x_i}, \theta)$$



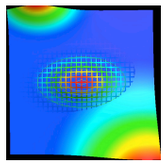
Random Fields

The local Markov property means that an observation is conditionally independent of the non-neighbour observations:

$$p(x_i|X = \{x_1, ..x_{i-1}, x_{i+1}.., x_T\}, \theta) = p(x_i|N_{x_i}, \theta)$$

N_{xi} is the neighbourhood of x_i , the set of nodes adjacent to x_i.

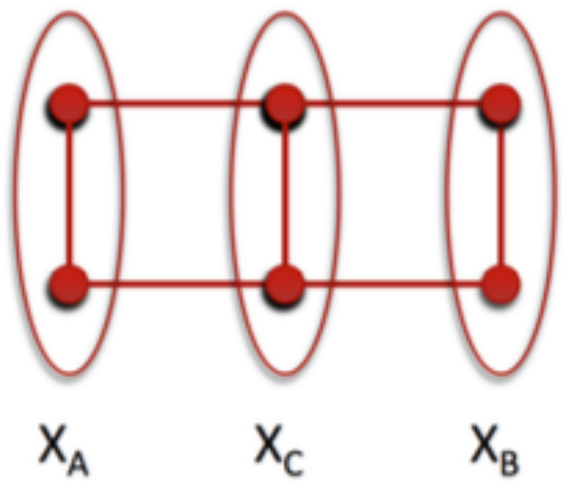
The global Markov property denotes that any two disjoint subsets X_A,X_B $\subset$ X are conditionally independent given a non-empty separate set X_C .



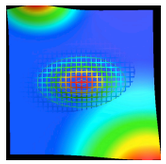
Random Fields

The global Markov property denotes that any two disjoint subsets X_A,X_B $\subset$ X are conditionally independent given a non-empty separate set X_C .

-> any path between each node from X_A to any node in X_B will include at least one node from X_C .



OR in other words if we remove X_C from the graph there will be no paths connecting X_A and X_B.



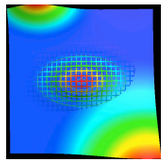
Random Fields

The smallest set of nodes for a node, which is making the node conditionally independent from all other nodes in the graph, is called the Markov blanket of the node.

This set is equivalent with the neighbourhood of the node.

The last property is the pairwise Markov property:

if two separate nodes are not immediate neighbours then they are conditionally independent given the rest of the nodes in the graph [Hammersley and Clifford, 1971].

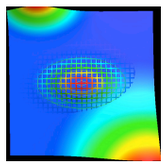


Random Fields

The Hammersly-Clifford theorem [Hammersley and Clifford, 1971] states that the joint probability has a Gibbs distribution form,

$$P(X \mid \theta) = \frac{e^{U(X|\Theta)}}{Z(\theta)}$$

where U(X | Θ) called as the energy function.



Random Fields

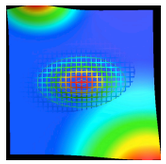
And

$$Z(\theta) = \int_{X \in \mathcal{X}} \mathrm{e}^{U(X|\theta)} \mathrm{d}X$$

is the partition function (or normalization constant)

the expected value of the energy function over our generative model.

Note: what if we define the energy function as the natural logarithm of a pdf?



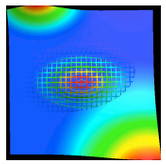
Random Fields

Note: what if we define the energy function as the natural logarithm of a pdf?

Z is trivially equal to 1 and therefore we get back the original pdf as expected.

According to [Hammersley and Clifford, 1971, Besag, 1974] if our MRF can be factorized over the set of cliques (C_X) in the graph than our pdf has a from of

$$p(X \mid \theta) = \Pi_{c \in C_X} p(c \mid \theta) = \frac{1}{Z} e^{\sum_{c \in C_X} U(c|\theta)}$$



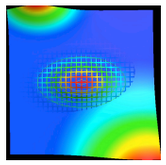
Random Fields

Compared to GMM:

Estimation of the parameters rather depends on the energy function and consequently on the normalization constant.

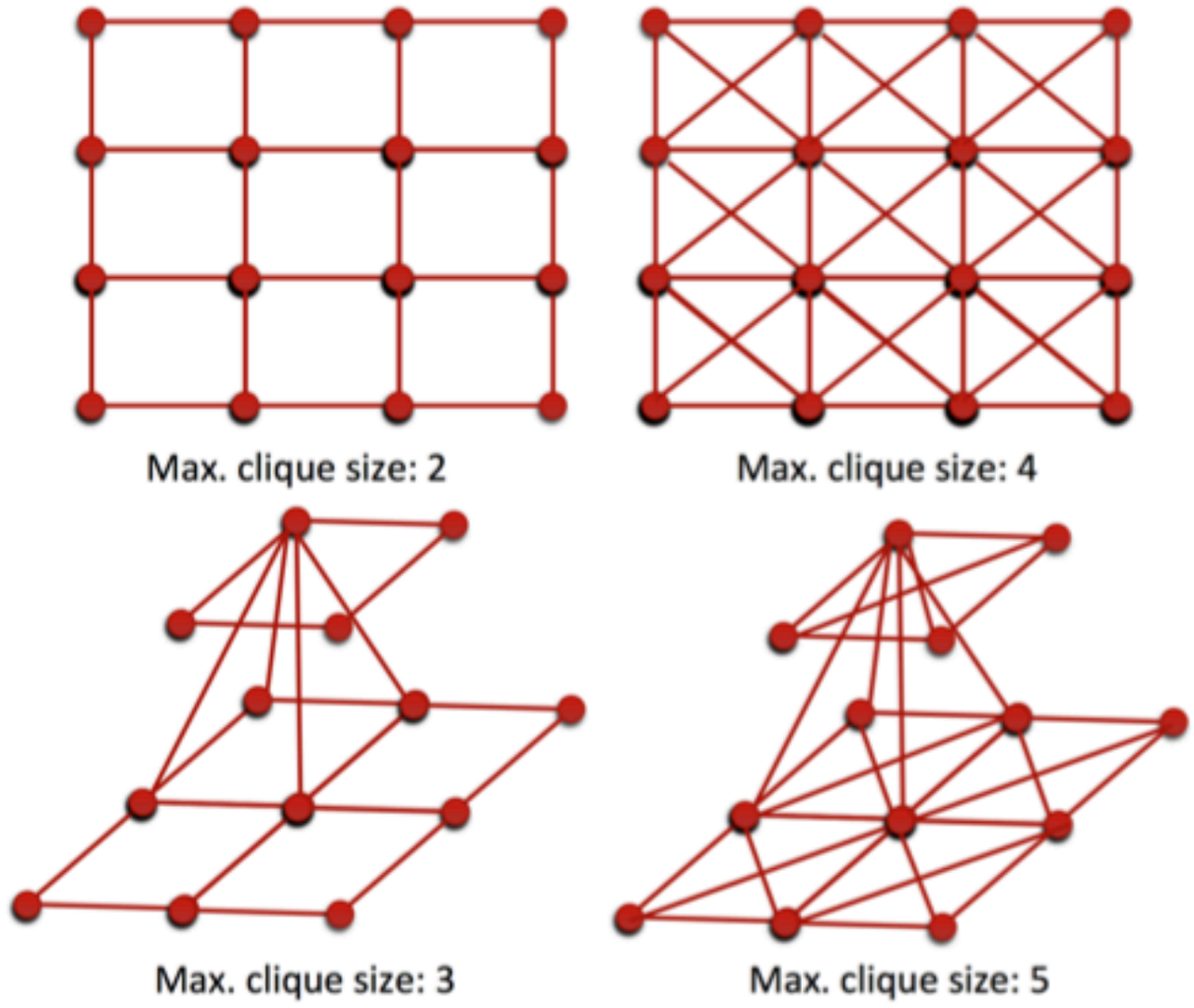
Despite a wide variety of methods can be used to determine the parameters:

- Inference (though the Maximum-a-Posteriori inference is NP-hard [Taskar et al., 2004])
- Simulated annealing [Geman and Graffigne, 1986]
- Maximum Likelihood may be an option (with proper energy functions)



Random Fields

Example: Images



Fisher Information: Riemannian metric over generative models, the gain over GMM is 1-5 %