Learning with Latent Variables

DD2421

Giampiero Salvi

HT2017

Outline

Recap: Probabilistic Learning Framework

- Maximum Likelihood
  - model parameters are deterministic
  - look for a point estimate θ_{ML}

$$\theta_{\text{ML}} = \arg\max_{\theta} p(\mathcal{D}|\theta) \quad \rightarrow \quad p(x|\theta_{\text{ML}})$$

- Maximum a Posteriori
  - model parameters are stochastic (prior $p(\theta)$)
  - look for a point estimate θ_{MAP}

$$\theta_{\text{MAP}} = \arg\max_{\theta} p(\theta|\mathcal{D}) = \arg\max_{\theta} p(\mathcal{D}|\theta)p(\theta) \quad \rightarrow \quad p(x|\theta_{\text{MAP}})$$

- Bayesian methods
  - model parameters are stochastic (prior $p(\theta)$)
  - use the full posterior $p(\theta|\mathcal{D})$

$$p(x|\mathcal{D}) = \int_{\theta} p(x|\theta)p(\theta|\mathcal{D})d\theta$$

Recap: Probabilistic Models for Classification

$\mathbf{x} \in \mathbb{R}^D$ are the features
$y \in \{y_1, \dots, y_K\}$ is the class identifier (discrete)
Goal: find most likely y_i given $\mathbf{x}$

- Maximum Likelihood:

$$\hat{y}_{\mathsf{ML}} = \arg\max_i p(x|y_i) = \arg\max_i p(x|\theta_i)$$

- Maximum a Posteriori:

$$\hat{y}_{\mathsf{MAP}} = \arg\max_i P(y_i|x) = \arg\max_i p(x|\theta_i)P(y_i)$$

Recap: Probabilistic Models for Regression

$\mathbf{x} \in \mathbb{R}^D$ are the features
$\mathbf{y} \in \mathbb{R}^K$ are the dependent variables (continuous)
Goal: find $p(\mathbf{y}|\mathbf{x})$

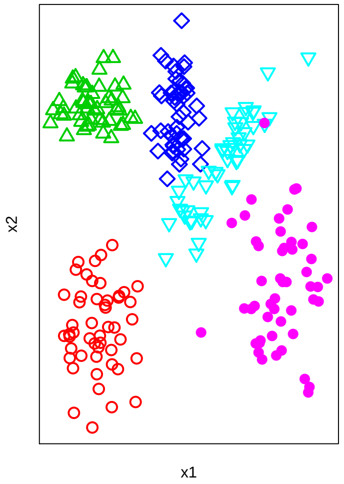
- Maximum Likelihood ($K = 1$):

$$\begin{aligned} y &= \mathbf{w}^T\mathbf{x} + \epsilon \\ p(\epsilon) &= \mathcal{N}(0, \sigma^2) \quad \rightarrow \quad p(y|\mathbf{x}) = \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma^2) \end{aligned}$$

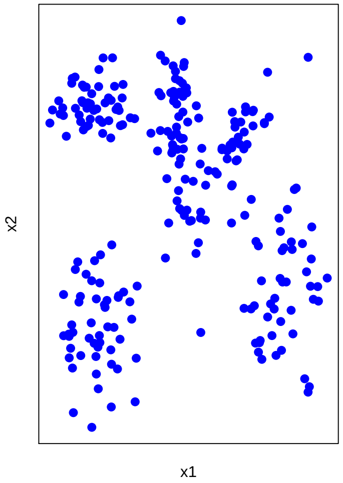
- Maximum a Posteriori:
  - prior over the parameters: $p(\mathbf{w}, \sigma^2)$
  - posterior of the parameters given the data: $p(\mathbf{w}, \sigma^2|\mathcal{D})$

Clustering vs Classification

Classification



Clustering



Fitting complex distributions

We can try to fit a mixture of K distributions:

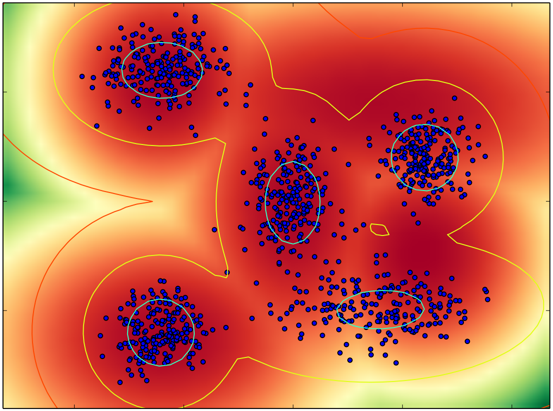
$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

Clustering Example

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

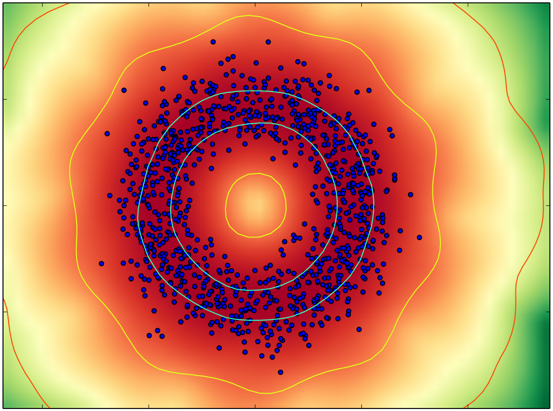
$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k)$$



Not only strictly clustering: Example

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k)$$



Fitting complex distributions

We can try to fit a mixture of K distributions:

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

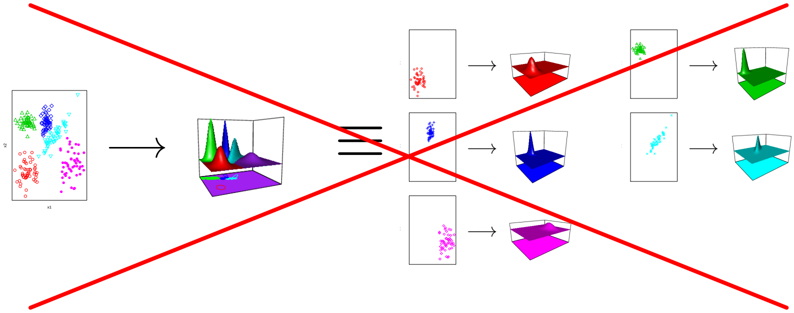
with $\theta = \{\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_K\}$

Problem:

We do not know which point has been generated by which component of the mixture

We cannot optimize $P(\mathbf{x}|\theta)$ directly

No Class Independence Assumption



Expectation Maximization

Fitting model parameters with missing (latent) variables

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

with $\theta = \{\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_K\}$

- very general idea (applies to many different probabilistic models)
- augment the data with the missing variables: h_{ik} probability that each data point x_i was generated by each component of the mixture k
- optimize the Likelihood of the complete data:

$$P(\mathbf{x}, \mathbf{h}|\theta)$$

Heuristic Example: K-means

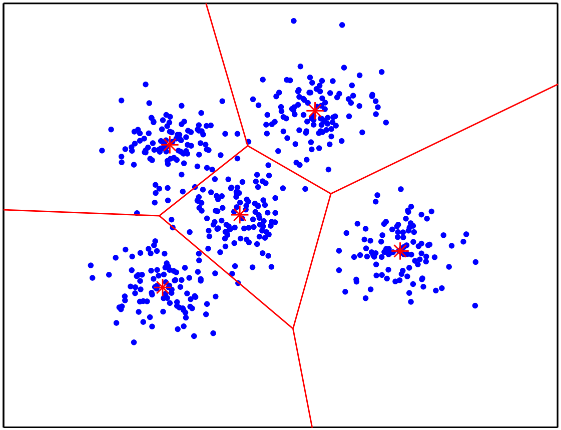
- describes each class with a centroid
- a point belongs to a class if the corresponding centroid is closest (Euclidean distance)
- iterative procedure
- guaranteed to converge
- not guaranteed to find the optimal solution
- used in vector quantization (since the 1950's)

K-means: algorithm

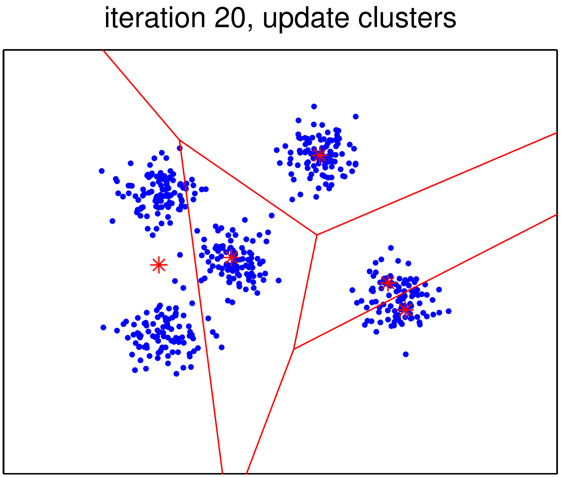
```
**Data:** $k$ (number of desired clusters), $n$ data points $\mathbf{x}_i$
**Result:** $k$ clusters
initialization: assign initial value to $k$ centroids $\mathbf{c}_i$;
repeat
    assign each point $\mathbf{x}_i$ to closest centroid $\mathbf{c}_j$;
    compute new centroids as mean of each group of points;
until centroids do not change;
return $k$ clusters;
```

K-means: example

iteration 20, update clusters



K-means: sensitivity to initial conditions

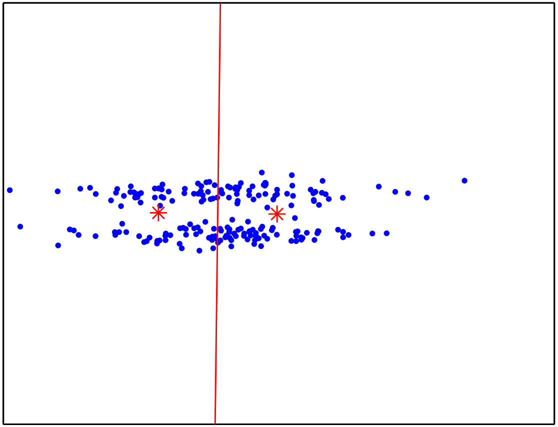


K-means: limits of Euclidean distance

- the Euclidean distance is isotropic (same in all directions in $\mathbb{R}^p$)
- this favours spherical clusters
- the size of the clusters is controlled by their distance

K-means: non-spherical classes

two non–spherical classes



Expectation Maximization

Fitting model parameters with missing (latent) variables

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

$$\text{with } \theta = \{\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_K\}$$

- very general idea (applies to many different probabilistic models)
- augment the data with the latent variables: $h_i \in \{1, \dots, K\}$ assignment of each data point x_i to a component of the mixture
- optimize the Likelihood of the complete data over N data points

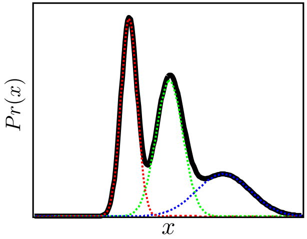
$$P(\mathbf{x}_1, \dots, \mathbf{x}_N, h_1, \dots, h_N|\theta)$$

Mixture of Gaussians

This distribution is a weight sum of K Gaussian distributions

$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$

where $\pi_1 + \cdots + \pi_K = 1$
and $\pi_k > 0$ $(k = 1, \ldots, K)$.



This model can describe **complex multi-modal** probability distributions by combining simpler distributions.

Mixture of Gaussians

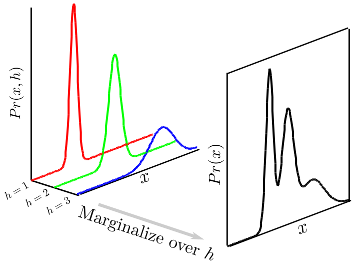
$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$

- Learning the parameters of this model from training data $x_1, \ldots, x_n$ is not trivial - using the usual straightforward maximum likelihood approach.
- Instead learn parameters using the **Expectation-Maximization** (EM) algorithm.

Mixture of Gaussians as a marginalization

We can interpret the Mixture of Gaussians model with the introduction of a discrete hidden/latent variable h and $P(x, h)$:

$$P(x) = \sum_{k=1}^{K} P(x, h = k) = \sum_{k=1}^{K} P(x \mid h = k) P(h = k)$$
$$= \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$



← **mixture density**

Figures taken from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians

Assume: We know the pdf of x has this form:

$$P(x) = \pi_1 \mathcal{N}(x; \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x; \mu_2, \sigma_2^2)$$

where $\pi_1 + \pi_2 = 1$ and $\pi_k > 0$ for components $k = 1, 2$.

Unknown: Values of the parameters (Many!)

$$\Theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2).$$

Have: Observed n samples $x_1, \dots, x_n$ drawn from $P(x)$.

Want to: Estimate Θ from $x_1, \dots, x_n$.

How would it be possible to get them all???

EM for two Gaussians

For each sample x_i introduce a *hidden variable* h_i

$$h_i = \begin{cases} 1 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_1, \sigma_1^2) \\ 2 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_2, \sigma_2^2) \end{cases}$$

and come up with initial values

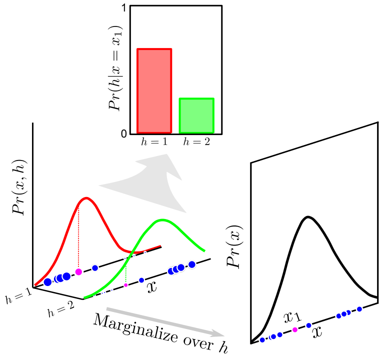
$$\Theta^{(0)} = (\pi_1^{(0)}, \mu_1^{(0)}, \sigma_1^{(0)}, \mu_2^{(0)}, \sigma_2^{(0)})$$

for each of the parameters.

EM is an *iterative algorithm* which updates $\Theta^{(t)}$ using the following two steps...

EM for two Gaussians: E-step

The responsibility of k-th Gaussian for each sample x (indicated by the size of the projected data point)



Look at each sample x along hidden variable h in the E-step

Figure from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians: E-step (cont.)

E-step: Compute the *"posterior probability"* that x_i was generated by component k given the current estimate of the parameters $\Theta^{(t)}$. (responsibilities)

```
for $i = 1, \ldots n$
    for $k = 1, 2$
```

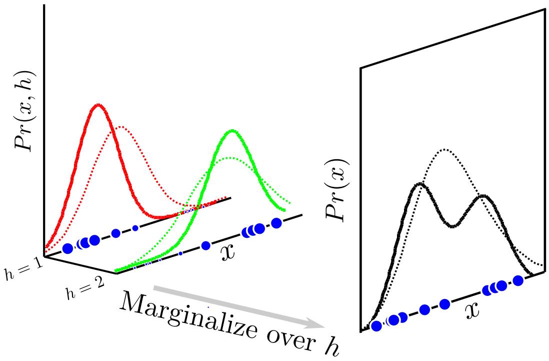
$$\gamma_{ik}^{(t)} = P(h_i = k \,|\, x_i, \Theta^{(t)})$$

$$= \frac{\pi_k^{(t)} \mathcal{N}(x_i; \mu_k^{(t)}, \sigma_k^{(t)})}{\pi_1^{(t)} \mathcal{N}(x_i; \mu_1^{(t)}, \sigma_1^{(t)}) + \pi_2^{(t)} \mathcal{N}(x_i; \mu_2^{(t)}, \sigma_2^{(t)})}$$

Note: $\gamma_{i1}^{(t)} + \gamma_{i2}^{(t)} = 1$ and $\pi_1 + \pi_2 = 1$

EM for two Gaussians: M-step

Fitting the Gaussian model for each of k-th constinuetnt.
Sample x_i contributes according to the responsibility γ_{ik}.



(dashed and solid lines for fit before and after update)

Look along samples x for each h in the M-step

EM for two Gaussians: M-step (cont.)

M-step: Compute the *Maximum Likelihood* of the parameters of the mixture model given out data's membership distribution, the $\gamma_i^{(t)}$'s:

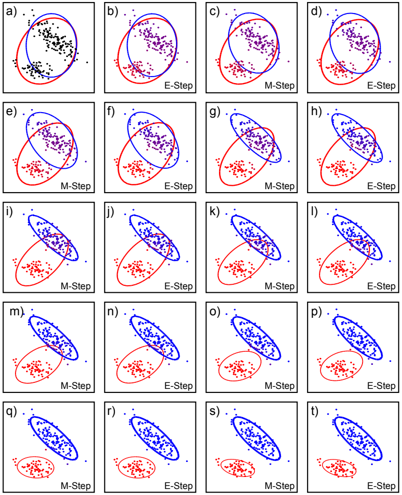
for $k = 1, 2$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)} x_i}{\sum_{i=1}^n \gamma_{ik}^{(t)}},$$

$$\sigma_k^{(t+1)} = \sqrt{\frac{\sum_{i=1}^n \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^n \gamma_{ik}^{(t)}}},$$

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)}}{n}.$$

EM in practice



EM properties

Similar to K-means

- guaranteed to find a local maximum of the complete data likelihood
- somewhat sensitive to initial conditions

Better than K-means

- Gaussian distributions can model clusters with different shapes
- all data points are smoothly used to update all parameters

Summary

If you are interested in learning more take a look at:

C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Verlag 2006.