

The EM Algorithm and its Application to Context Estimation

Hauptseminar Machine Learning for Context Aware Computing

Martin Wagner

wagnerm@in.tum.de

Lehrstuhl für angewandte Softwaretechnik

Institut für Informatik, Technische Universität München

Boltzmannstr. 3, 85748 Garching b. München, Germany

- Parametric techniques are often not sufficient.
- *Idea:* Mix several parametric distributions.
- Parameters of mixing components could be found easily if responsible component for every data point would be known.
- *Next Idea:* Treat this relationship as *complementary data*.
- Now the *Expectation Maximization* algorithm can iteratively yield good solutions.
- An Application to skin color blob tracking is shown.

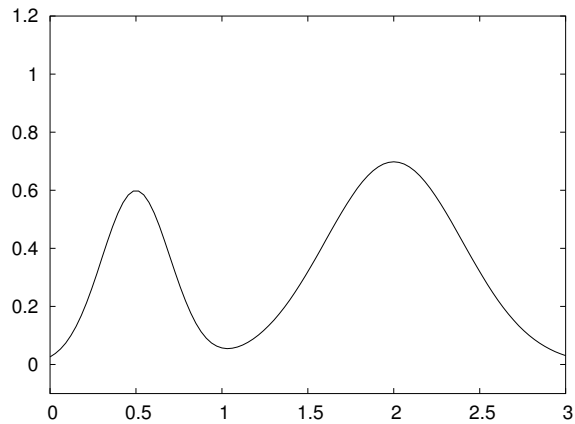
- **Motivation**
- Mixture Models
- The EM Algorithm: Overview
- The EM Algorithm: Expectation Step
- The EM Algorithm: Maximization Step
- Application: Skin Color Blob Tracking

- General problem: Given a data set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, find a suitable distribution modeling the underlying random process $p(\mathbf{x})$.
- We have discussed *parametric models* $p(\mathbf{x}|\theta)$ in detail, especially Gaussian models.
- We will see the power of *nonparametric models* (i.e. histograms, kernel-based methods) in later talks.
- Both techniques have advantages and drawbacks.
- Why not combine them?

- Motivation
- **Mixture Models**
- The EM Algorithm: Overview
- The EM Algorithm: Expectation Step
- The EM Algorithm: Maximization Step
- Application: Skin Color Blob Tracking

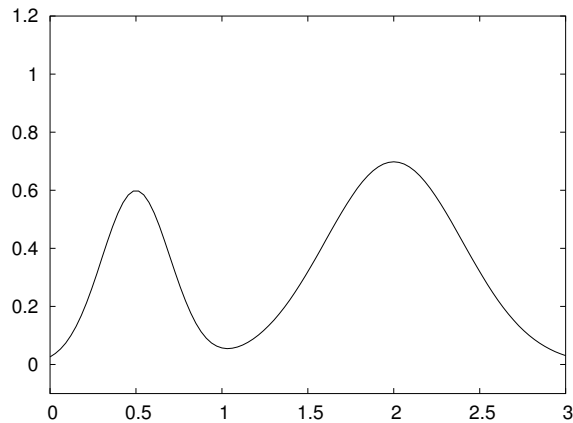
Mixture Models

- How would you model this distribution?



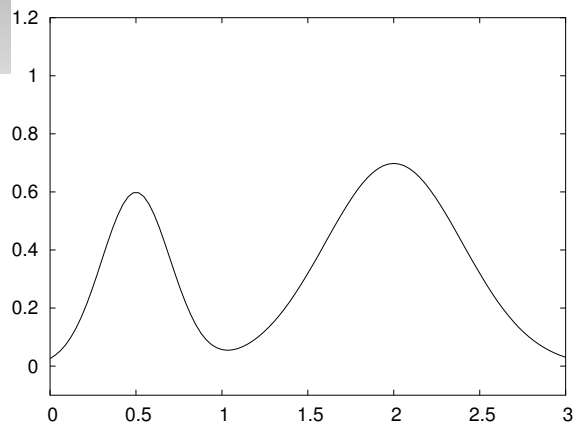
Mixture Models

- How would you model this distribution?
- Mixture of two Gaussians:



$$p(x) = 0.3\mathbf{n}(\mu = 0.5, \sigma = 0.2) + 0.7\mathbf{n}(2.0, 0.4)$$

Mixture Models



- How would you model this distribution?

- Mixture of two Gaussians:

$$p(x) = 0.3\mathbf{n}(\mu = 0.5, \sigma = 0.2) + 0.7\mathbf{n}(2.0, 0.4)$$

- General form of mixture model:

$$p(\mathbf{x}) = \sum_{j=1}^M P(j)p(\mathbf{x}|j)$$

with

$$\sum_{j=1}^M P(j) = 1$$

$$0 \leq P(j) \leq 1$$

$$\int p(\mathbf{x}|j) d\mathbf{x} = 1$$

- Log-likelihood for mixture density:

$$E = \ln(\mathcal{L}(\Theta|\mathcal{X})) = \ln \prod_{n=1}^N p(\mathbf{x}_n|\Theta) = \sum_{n=1}^N \ln \left(\sum_{j=1}^M P(j)p(\mathbf{x}_n|j) \right)$$

- Problem: sum of logarithms of sums is hard to optimize!
- We postulate existence of *unobserved data items* $\mathcal{Z} = \{z_1, \dots, z_N\}$ that tell us from which distribution every \mathbf{x}_n was taken.
- Now the likelihood is easier to optimize:

$$\begin{aligned} E &= \ln(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = \ln(P(\mathcal{X}, \mathcal{Z}|\Theta)) = \sum_{n=1}^N \ln(P(\mathbf{x}_n|z_n)P(z_n)) \\ &= \sum_{n=1}^N \ln(P(z_n)p_{z_n}(\mathbf{x}_n|\theta_{z_n})) \end{aligned}$$

- Motivation
- Mixture Models
- **The EM Algorithm: Overview**
- The EM Algorithm: Expectation Step
- The EM Algorithm: Maximization Step
- Application: Skin Color Blob Tracking

The EM Algorithm: Overview

Expectation step:

- Guess some initialization values for parameters θ_j of mixture models.
- Use θ_j 's to calculate probability distribution of \mathcal{Z} depending on \mathbf{x}_n .
- Compute *expectation* \mathcal{E} of log likelihood $E(\mathcal{X}, \mathcal{Z})$.

Maximization step:

- Maximize $\mathcal{E}(E(\mathcal{X}, \mathcal{Z}))$ with regard to the following parameters:

Distribution of \mathcal{Z} : $P^{new}(j)$

Parameters of mixture model : θ_j

Iterate both steps:

Until a convergence criteria has been reached.

- Motivation
- Mixture Models
- The EM Algorithm: Overview
- **The EM Algorithm: Expectation Step**
- The EM Algorithm: Maximization Step
- Application: Skin Color Blob Tracking

EM Algorithm: Expectation Step (1)

- Start with computation of probability distribution for mixture weights:

$$P^{old}(j|\mathbf{x}_n) = \frac{P^{old}(j)p(\mathbf{x}_n|\theta_j^{old})}{\sum_{l=1}^M P^{old}(l)p(\mathbf{x}_n|\theta_l^{old})}$$

- Compute expectation by trying out all possibilities of \mathcal{Z} :

$$\mathcal{E}(E(\mathcal{X})) = \sum_{z_1=1}^M \sum_{z_2=1}^M \dots \sum_{z_N=1}^M E(\mathcal{X}, \mathcal{Z}) \prod_{n=1}^N P^{old}(z_n|\mathbf{x}_n)$$

EM Algorithm: Expectation Step (2)

- Substitute

$$E(\mathcal{X}, \mathcal{Z}) = \sum_{n=1}^N \ln (P^{new}(z_n)p(\mathbf{x}_n|z_n))$$

- Resulting in

$$\begin{aligned} \mathcal{E}(E(\mathcal{X}, \mathcal{Z})) &= \sum_{n=1}^N \sum_{j=1}^M P^{old}(j|\mathbf{x}_n) \cdot \ln (P^{new}(j)) \\ &+ \sum_{n=1}^N \sum_{j=1}^M P^{old}(j|\mathbf{x}_n) \cdot \ln (p^{new}(\mathbf{x}_n|\theta_j^{new})) \end{aligned}$$

- Both terms are independent of each other and can be maximized individually.

- Motivation
- Mixture Models
- The EM Algorithm: Overview
- The EM Algorithm: Expectation Step
- **The EM Algorithm: Maximization Step**
- Application: Skin Color Blob Tracking

EM Algorithm: Maximization Step (1)

Start with the maximization \mathcal{E} with regard to $P^{new}(j)$:

- Constraint $\sum_{j=1}^M P(j) = 1$ handled by Lagrange multiplier λ .
- Set partial derivation to zero:

$$\frac{\partial}{\partial P^{new}(j)} \left[\sum_{n=1}^N \sum_{j=1}^M P^{old}(j|\mathbf{x}_n) \cdot \ln(P^{new}(j)) + \lambda \left(\sum_{j=1}^M P^{new}(j) - 1 \right) \right] = 0$$

- Resulting in $\lambda = -N$ and

$$P^{new}(j) = \frac{1}{N} \cdot \sum_{n=1}^N P^{old}(j|\mathbf{x}_n)$$

Note: this result is *independent of the underlying distribution*

EM Algorithm: Maximization Step (2)

Now we try to maximize \mathcal{E} with regard to θ_j^{new} .

- In general, this is extremely hard.
- We assume a very simplistic model: a d -dimensional Gaussian with covariance matrix $\Sigma = \sigma^2 \cdot \mathbf{I}$. Now $\theta_j^{new} = (\mu_j, \sigma_j)$ and

$$p^{new}(\mathbf{x}_n | \theta_j^{new}) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left(-\frac{\|\mathbf{x}_n - \mu_j\|^2}{2\sigma_j^2}\right)$$

- Again, we have the Lagrange multiplier $\lambda = -N$, resulting in:

$$\begin{aligned}\mu_j^{new} &= \frac{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n)} \\ \sigma_j^{new2} &= \frac{1}{d} \cdot \frac{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n) \|\mathbf{x}_n - \mu_j^{new}\|^2}{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n)}\end{aligned}$$

EM Algorithm: Maximization Step (3)

Computation of maximization is also possible with full covariance matrix:

$$P^{old}(j|\mathbf{x}_n) = \frac{P^{old}(j) \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x}_n - \mu_j^{old})^t \Sigma_j^{-1} (\mathbf{x}_n - \mu_j^{old})\right)}{\sum_{l=1}^M P^{old}(l) \frac{1}{(2\pi)^{d/2} |\Sigma_l|^{1/2}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x}_n - \mu_l^{old})^t \Sigma_l^{-1} (\mathbf{x}_n - \mu_l^{old})\right)}$$

$$\mu_j^{new} = \frac{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n)}$$

$$\Sigma_j^{new} = \frac{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n) (\mathbf{x}_n - \mu_j^{new})(\mathbf{x}_n - \mu_j^{new})^t}{\sum_{n=1}^N P^{old}(j|\mathbf{x}_n)}$$

- Motivation
- Mixture Models
- The EM Algorithm: Overview
- The EM Algorithm: Expectation Step
- The EM Algorithm: Maximization Step
- **Application: Skin Color Blob Tracking**

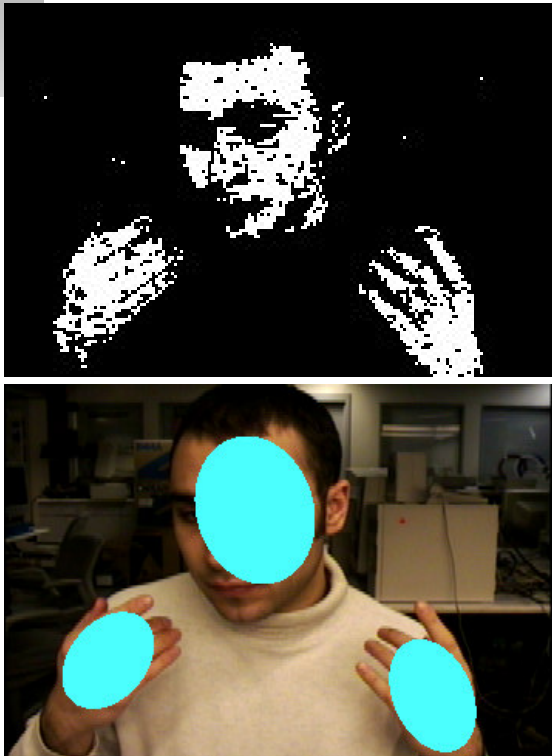
Application: Skin color detection

Idea: Every pixel is composed of three parameters, *Red*, *Green* and *Blue*. For skin color, we could model the distribution probability individually in each of these parameters

Algorithm:

1. Set up mixture model consisting of $M = 3$ 3D Gaussians.
2. Label some sample images by hand (click on skin colored pixels).
3. Let the EM algorithm determine the weights of the mixture model and the parameters (μ_j, σ_j) of the Gaussian distributions.
4. Pre-compute a lookup table containing the probability of being skin color for every color in RGB space.
5. Online classification is now easy: do a lookup and classify current pixel if its probability is above a threshold.

Application: Skin color blob tracking



Algorithm:

1. Extract skin colored pixels from image.
2. Set up Gaussian mixture model with three 2D Gaussians.
3. Feed the skin pixel's locations (x, y) in the EM algorithm.

Result: Three 2D Gaussians describing head and hands of a person on the image.

Thank you

... any Questions? ...