Machine Learning
Exercise 11

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

July 10, 2013

1 Max. likelihood estimator for a multinomial distribution

Let X be a discrete variable with domain $\{1, .., K\}$. We parameterize the discrete distribution as

$$P(X=k;\pi) = \pi_k \tag{1}$$

with parameters $\pi = (\pi_1, .., \pi_K)$ that are constrained to fulfill $\sum_k \pi_k = 1$. Assume we have some data $D = \{x_i\}_{i=1}^n$
a) Write down the likelihood $\mathcal{L}(\pi)$ of the data under the model.
b) Prove that the maximum likelihood estimator for π is, just as intuition tells us,

$$\pi_k^{\text{ML}} = \frac{1}{n}\sum_{i=1}^n [x=k] \ . \tag{2}$$

Tip: Although this seems trivial, it is not. The pitfall is that the parameter π is constrained with $\sum_k \pi_k = 1$. You need to solve this using Lagrange multipliers—see Wikipedia or Bishop section 2.2.

2 Gaussians and Singular Value Decomposition

On the course homepage there is a data set `gauss.txt` containing $n = 1000$ 2-dimensional points. Load it in a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times 2}$.
a) Compute the mean (e.g., `mu = 1./n*sum(X,0)`)
b) Center the data ($\tilde{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{1}_n\mu^\top$)
c) Compute the covariance matrix $C = \frac{1}{n}\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}}$. Also compute $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} - \mu\mu^\top$ (using the uncentered data) and compare.
d) Compute the Singular Value Decomposition $C = UDV^\top$. Ouput the eigenvalues (diagonal of D) and eigenvectors (columns of V). Plot the data and the two line segments between μ and $\mu + \sqrt{\lambda_j}v_j$, $j = 1, 2$.

3 Mixture of Gaussians

Download the data set `mixture.txt` from the course webpage, containing $n = 300$ 2-dimensional points. Load it in a data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times 2}$.
a) Implement the EM-algorithm for a Gaussian Mixture on this data set. Choose $K = 3$ and the prior $P(c_i=k) = 1/K$. Initialize by choosing the three means μ_k to be different randomly selected data points x_i (i random in $\{1, .., n\}$) and the covariances $\Sigma_k = \mathbf{I}$ (a more robust choice would be the covariance of the whole data). Iterate EM starting with the first E-step based on these initializations. Repeat with random restarts—how often does it converge to the optimum?
Tip: Store $q(c_i=k)$ as a $K\times n$-matrix with entries q_{ki}; equally $w_{ki} = q_{ki}/\pi_k$. Store μ_k's as $K\times d$-matrix and Σ_k's as $K\times d\times d$-array. Then the M-step update for μ_k is just a matrix multiplication. The update for each Σ_k can be written as $\boldsymbol{X}^\top\text{diag}(w_{k,1:d})\boldsymbol{X} - \mu_k\mu_k^\top$.

b) Do exactly the same, but this time initialize the posterior $q(c_i = k)$ randomly (i.e., assign each point to a random cluster, $q(c_i) = [c_i = \text{rand}(1:K)]$); then start EM with the first M-step. Is this better or worse than the previous way of initialization?