TDT4173: Machine Learning and Case-Based Reasoning

Assignment 4

March 10, 2016

- **Delivery deadline: March 31, 2016** by 22:00.
- Solutions must be submitted individually.
- Deliver your solution on *itslearning* before the deadline.
- Please upload your report as a PDF file, and package your code into an archive (e.g. zip, rar, tar).
- The programming tasks may be completed in the programming language of your choice.
- Your code is part of your delivery, so please make sure that your code is well-documented and as readable as possible.

Learning Objectives: Gain insight into (a) Bayesian learning, and (b) how the Expectation Maximization (EM) algorithm can be used to fit a two component Gaussian mixture model.

1 Theory

Bayesian Learning

1. Define the notion of likelihood. What is the difference between a *maximum a posteriori* hypothesis (h_{MAP}) and a *maximum likelihood* hypothesis (h_{ML})? Is it possible that a hypothesis that is h_{MAP} is not h_{ML}? Justify your opinion. If your answer is *yes*, give an example where h_{MAP} and h_{ML} are two different hypotheses.
2. Describe the algorithms for building an *OptimalBayes* and a *NaiveBayes* classifier. Is the *BruteForceMap* learning algorithm of any use here? Explain. Is an *OptimalBayes* classifier an example of a *NaiveBayes* classifier, or is the case the reverse (or none)?
3. Compare the *OptimalBayes* and *NaiveBayes* algorithms in terms of:
    (a) Considered hypothesis space.
    (b) Performance.
    (c) Computational cost.
4. How would you improve *OptimalBayes* and *NaiveBayes* in terms of the three facets listed above?

5. What is the difference between a *NaiveBayes* classifier and a Bayesian belief network? What is the relation between these two methods?

2 Programming

Expectation Maximization

In this programming assignment you are required to implement the Expectation Maximization (EM) algorithm for Gaussian mixtures[1]. Assume that the sample data consist of two 1D Gaussians with equal mixing coefficients, where each Gaussian has variance equal to 1, but they have different expectations. The sample data can be visualised as a histogram.

The EM algorithm is comprised of two main parts: the E step and the M step, that are run once per iteration. The first step calculates the expected value of each hidden variable given the current hypothesis (expectation in our case). This can be seen on the left-hand side of Equation 1, where x_i is the ith sample, k is the number of Gaussian components, μ is the mean value, σ is the standard deviation, and $pdf_{\mathcal{N}}$ is the probability density function for a univariate Gaussian distribution.

$$E[z_{ij}] = \frac{pdf_{\mathcal{N}}(x_i, \mu_j, \sigma)}{\sum_{n=1}^{k} pdf_{\mathcal{N}}(x_i, \mu_n, \sigma)} \qquad pdf_{\mathcal{N}}(x, \mu, \sigma) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \tag{1}$$

The second step calculates maximum likelihood assuming that the hidden variables take on the expected values from the E step. This can be seen in Equation 2, where m is the number of samples.

$$\mu_j = \frac{\sum_{i=1}^{m} E[z_{ij}]x_i}{\sum_{i=1}^{m} E[z_{ij}]} \tag{2}$$

1. Implement the EM algorithm for Gaussian mixtures.
2. Run your program on the sample data provided in a separate text file.
   (a) Show initial parameters, comment on your choice.
   (b) Show intermediate result after the 5th, 10th iterations and show the final result. Discuss your result.
   (c) Visualise the sample data and overlay the Gaussians you have fitted.

[1] In literature, a Gaussian mixture model (GMM) may sometimes be referred to as a mixture of Gaussians (MoG). In general, a MoG is simply a weighted sum or *mixture* of k Gaussian distributions with different means and covariances.