Candidate Project Statement

Gaussian Mixture Models

Timed project - 7 hour limit

Associated files: `gmmdata.txt`

The well-known Expectation-Maximization (EM) algorithm can be used to learn the parameters of a GMM given a set of M independent and identically distributed (IID) data observations, $X = \{x_i \mid 0 \le i \le M - 1\}$. The algorithm is concisely summarized for 1-dimensional input data as follows:

Step 0. Empirically determine the number of mixture components, K, that will best fit the data.

Step 1. Initialize $P(k)$, μ_k, Θ_k for all K Gaussian components.

Step 2. Compute K posterior probabilities for all M observations:

$$P(k \mid x_i) = P(k)N(x_i; \mu_k, \Theta_k) \Big/ \sum_{r=0}^{K-1} P(r)N(x_i; \mu_r, \Theta_r)$$

$$\text{for } 1 \le k \le K \text{ and } 0 \le i \le M - 1$$

Step 3. Update the parameters of each Gaussian component:

$$P(k) = (1/M) \times \sum_{r=0}^{M-1} P(k \mid x_r)$$

$$\mu_k = \sum_{r=0}^{M-1} x_r P(k \mid x_r) / \sum_{r=0}^{M-1} P(k \mid x_r)$$

$$\Theta_k = \sum_{r=0}^{M-1} (x_r - \mu_k)^2 P(k \mid x_r) / \sum_{r=0}^{M-1} P(k \mid x_r)$$

$$\text{all for } 1 \le k \le K$$

Step 4. If the likelihood function has not sufficiently converged, repeat steps 2 and 3.

Please implement the EM training algorithm for GMMs *from scratch* using a programming language of your choice (but preferably **not** Matlab). The program shall read in the data stored in `gmmdata.txt` (each row is one observation and coordinates are separated by whitespace), iteratively compute the GMM parameter set, and save the model to a file. You should be able to hypothesize the “correct” number of mixture components by inspection of the data.

The content of this document is strictly proprietary and confidential. The document contains privileged information, and is for the sole use of the intended recipient. Any unauthorized review, use, disclosure, or distribution is prohibited. If you are not the intended recipient, please destroy all copies of this document.

Questions:

1. Aside from EM, what other method(s) can be used to learn the parameters of a GMM?
2. What are some known fundamental problems associated with GMM training that should be accounted for in a robust training algorithm?
3. Please explain the relationship between the *K*-means algorithm and EM for GMMs.
4. If one is unable to visualize the training data (say, due to its high dimensionality and/or intractably large volume), please suggest a method for estimating the number of mixture components.
5. Compare and contrast the different techniques for initializing the EM algorithm when training a GMM. Comment on the choice you used in your implementation.

Deliverables:

1. A program, (set of) script(s), executable(s), or a compilable codebase that, when run in the same directory as `gmmdata.txt`, trains a GMM on the data and saves the generated model to a file.
2. A document that describes the result you have obtained and demonstrates, graphically and/or numerically, why the result makes sense.
3. Answers to the five questions posed above.
4. All of the code you wrote.

The content of this document is strictly proprietary and confidential. The document contains privileged information, and is for the sole use of the intended recipient. Any unauthorized review, use, disclosure, or distribution is prohibited. If you are not the intended recipient, please destroy all copies of this document.