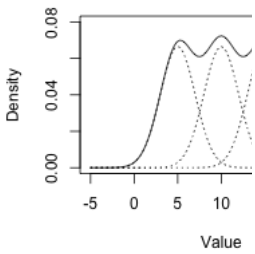
Gaussian Mixture Model

Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.



A Gaussian mixture of three norma

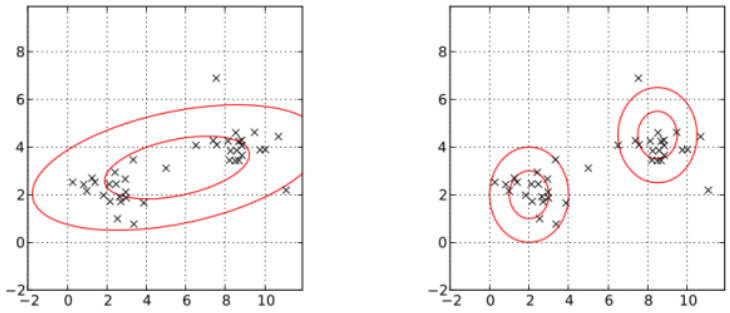
For example, in modeling human height data, height is typically modeled as a normal distribution for each gender with a mean of approximately 5'10" for males and 5'5" for females. Given only the height data and not the gender assignments for each data point, the distribution of all heights would follow the sum of two scaled (different variance) and shifted (different mean) normal distributions. A model making this assumption is an example of a Gaussian mixture model (though in general a GMM may have more than two components. Estimating the parameters of the individual norma distribution components is a canonical problem in modeling data with GMMs.

GMMs have been used for feature extraction from speech data, and have also been used extensively in object tracki multiple objects, where the number of mixture components and their means predict object locations at each frame i sequence.

Contents

Motivation

One hint that data might follow a mixture model is that the data looks multimodal, i.e. there is more than one "peak distribution of data. Trying to fit a multimodal distribution with a unimodal (one "peak") model will generally give a p shown in the example below. Since many simple distributions are unimodal, an obvious way to model a multimodal would be to assume that it is generated by multiple unimodal distributions. For several theoretical reasons, the most used distribution in modeling real-world unimodal data is the Gaussian distribution. Thus, modeling multimodal dat mixture of many unimodal Gaussian distributions makes intuitive sense. Furthermore, GMMs maintain many of the t and computational benefits of Gaussian models, making them practical for efficiently modeling very large datasets.



(Left) Fit with one Gaussian distribution (Right) Fit with Gaussian mixture model with two components

TRY IT YOURSELF

Which of the following data sets is most likely to be well-modeled by a Gaussian mixture model?

- Numbers of pregna many humans
- Student scores on a standardized test
- Speeds across cars j reaching a specific t

The Model

A Gaussian mixture model is parameterized by two types of values, the mixture **component weights** and the compo and **variances/covariances**. For a Gaussian mixture model with K components, the k^{th} component has a mean of μ variance of σ_k for the univariate case and a mean of $\vec{\mu}_k$ and covariance matrix of Σ_k for the multivariate case. The m component weights are defined as ϕ_k for component C_k, with the constraint that $\sum_{i=1}^{K} \phi_i = 1$ so that the total pro distribution normalizes to 1. If the component weights aren't learned, they can be viewed as an a-priori distribution components such that $p(x \text{ generated by component } C_k) = \phi_k$. If they are instead learned, they are the a-poster estimates of the component probabilities given the data.

One-dimensional Model

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

Multi-dimensional Model

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^{\mathrm{T}} {\Sigma_i}^{-1} (\vec{x}-\vec{\mu}_i)\right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

Learning the Model

If the number of components K is known, **expectation maximization** is the technique most commonly used to estin mixture model's parameters. In frequentist probability theory, models are typically learned by using maximum likelih estimation techniques, which seek to maximize the probability, or likelihood, of the observed data given the model p Unfortunately, finding the maximum likelihood solution for mixture models by differentiating the log likelihood and 0 is usually analytically impossible.

Expectation maximization (**EM**) is a numerical technique for maximum likelihood estimation, and is usually used whe form expressions for updating the model parameters can be calculated (which will be shown below). Expectation ma is an iterative algorithm and has the convenient property that the maximum likelihood of the data strictly increases v subsequent iteration, meaning it is guaranteed to approach a local maximum or saddle point.

EM for Gaussian Mixture Models

Expectation maximization for mixture models consists of two steps.

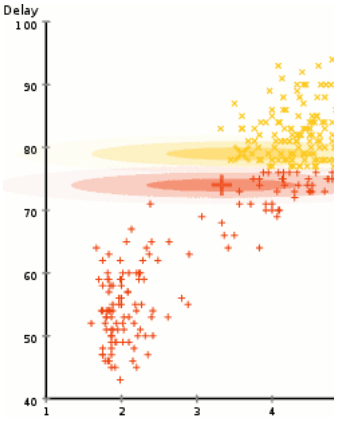
The first step, known as the **expectation** step or **E** step, consists of calculating the expectation of the component assi C_k for each data point $x_i \in X$ given the model parameters ϕ_k, μ_k, and σ_k.

The second step is known as the **maximization** step or **M** step, which consists of maximizing the expectations calcula step with respect to the model parameters. This step consists of updating the values ϕ_k, μ_k, and σ_k.

The entire iterative process repeats until the algorithm converges, giving a maximum likelihood estimate. Intuitively, algorithm works because knowing the component assignment C_k for each x_i makes solving for ϕ_k, μ_k, and σ_k easy knowing ϕ_k, μ_k, and σ_k makes inferring $p(C_k|x_i)$ easy. The expectation step corresponds to the latter case while th maximization step corresponds to the former. Thus, by alternating between which values are assumed fixed, or know maximum likelihood estimates of the non-fixed values can be calculated in an efficient manner.

Algorithm for Univariate Gaussian Mixture Models

The expectation maximization algorithm for Gaussian mixture models starts with an initialization step, which assigns model parameters to reasonable values based on the data. Then, the model iterates over the expectation (E) and maximization (M) steps until the parameters' estimates converge, i.e. for all parameters θ_t at iteration t, $|\theta_t - \theta_{t-1}| \leq \epsilon$ for some user-defined tolerance ϵ. A graphic of the EM algorithm in action for a two-component, bivariate Gaussian mixture model is displayed on the right.



The EM algorithm updating the parameters o bivariate Gaussian mixture model.[2]

The EM algorithm for a univariate Gaussian mixture model with K components is described below. A variable denoted $\hat{\theta}$ denotes an estimate for the value θ. All equations can be derived algebraically by solving for each parameter as outlined in the section above titled *EM for Gaussian Mixture Models*.

Initialization Step:

- Randomly assign samples without replacement from the dataset $X = \{x_1, ..., x_N\}$ to the component mean es $\hat{\mu}_1, ..., \hat{\mu}_K$. E.g. for $K = 3$ and $N = 100$, set $\hat{\mu}_1 = x_{45}, \hat{\mu}_2 = x_{32}, \hat{\mu}_3 = x_{10}$.
- Set all component variance estimates to the sample variance $\hat{\sigma}_1^2, ..., \hat{\sigma}_K^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$, where $\bar{x}$ is the s mean $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$.
- Set all component distribution prior estimates to the uniform distribution $\hat{\phi}_1, ..., \hat{\phi}_K = \frac{1}{K}$.

Expectation (E) Step:

Calculate $\forall i, k$

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i \mid \hat{\mu}_k, \hat{\sigma}_k)}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i \mid \hat{\mu}_j, \hat{\sigma}_j)},$$

where $\hat{\gamma}_{ik}$ is the probability that x_i is generated by component C_k. Thus, $\hat{\gamma}_{ik} = p(C_k|x_i, \hat{\phi}, \hat{\mu}, \hat{\sigma})$.

Maximization (M) Step:

Using the $\hat{\gamma}_{ik}$ calculated in the expectation step, calculate the following in that order $\forall k$:

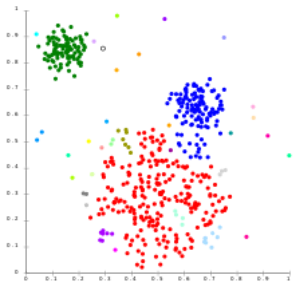
- $\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$
- $\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$
- $\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$.

When the number of components K is not known a priori, it is typical to guess the number of components and fit th the data using the EM algorithm. This is done for many different values of K. Usually, the model with the best trade- between fit and number of components (simpler models have fewer components) is kept.

The EM algorithm for the multivariate case is analogous, though it is more complicated and thus is not expounded h

Unsupervised Learning

Once the EM algorithm has run to completion, the fitted model can be used to perform various forms of inference. T most common forms of inference done on GMMs are **density estimation** and **clustering**.



Clustering using a Gaussian mixture model. Each color represents a different cluster according to the model.[3]

Density Estimation

Since the GMM is completely determined by the parameters of its individual compon fitted GMM can give an estimate of the probabilities of both in-sample and out-of-sa points, known as density estimation. Furthermore, since numerically sampling from ar Gaussian distribution is possible, one can easily sample from a GMM to create synthe

Sampling from a GMM consists of the following steps:

1. Sample the Gaussian component according to the distribution defined by $p(C_s)$
2. Sample x from the distribution for component C_s, according to the distribution d $\mathcal{N}(x \mid \mu_s, \sigma_s)$.

Clustering

Using Bayes' theorem and the estimated model parameters, one can also estimate the posteriori component assign probability. Knowing that a data point is likely from one component distribution versus another provides a way to lea where cluster assignment is determined by the most likely component assignment. Clustering has many uses in mac learning, ranging from tissue differentiation in medical imaging to customer segmentation in market research.

Given a univariate model's parameters, the probability that a data point x belongs to component C_i is calculated us theorem:

$$p(C_i \mid x) = \frac{p(x, C_i)}{p(x)} = \frac{p(C_i)p(x \mid C_i)}{\sum_{j=1}^{K} p(C_j)p(x \mid C_j)} = \frac{\phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(x \mid \mu_j, \sigma_j)}.$$

Recent Applications

GMMs have been used recently for feature extraction from speech data for use in speech recognition systems[4]. The been used extensively in object tracking of multiple objects, where the number of mixture components and their me object locations at each frame in a video sequence[5]. The EM algorithm is used to update the component means ove the video frames update, allowing object tracking.

See Also

- Normal Distribution
- Multivariate Normal Distribution

References

1. , S. *Gaussian-mixture-example*. Retrieved June 13, 2012, from https://commons.wikimedia.org/wiki/File:Gaussian-r example.svg
2. , C. *EM_Clustering_of_Old_Faithful_data*. Retrieved August 1, 2012, from https://commons.wikimedia.org/wiki/File:EM_Clustering_of_Old_Faithful_data.gif
3. , C. *SLINK-Gaussian-data*. Retrieved October 23, 2011, from https://commons.wikimedia.org/wiki/File:SLINK-Gauss data.svg
4. Deng, L. (2014). *Automatic Speech Recognition- A Deep Learning Approach* (pp. 6-8). Springer.
5. Santosh, D. (2013). Tracking Multiple Moving Objects Using Gaussian Mixture Model. *International Journal of Soft Computing and Engineering*, *3-2*, 114-119.