Mastering Machine Learning CH3 pt 2

Generative Gaussian Mixture

The first model to discuss us called Generative Gaussian Mixture and it aims to model the data generating process Pdata using a sum of wighted Gaussian distributions. Since the model is generative, its structure allows us not only to cluster the existing into well-defined regions (represented as Gaussians), but also To output the probability of any new data point to belong to each of the classes. This model is very flexible, and can be applied to solve all those problems where it's necessary perform a clustering and a classification at the same time, obtaining The assignment probability vector that determines the likelihood of a point to be generated by a specific Gaussian dist.

Generative Gaussian Mixture Theory

GGM is an inductive algorithm for inductive algorithm for semi-supervised Classification and clustering that’s aimed at modeling the conditional Probability p(X(hat),y(hat)) given both a labeled and an unlabeled dataset

(in this case, we are sure that the knowledge p(X hat)of is helpful because we are p(y hat|X hat)going to derive using Bayes' theorem).

GGM's may help in:
finding a model that explains the structure of the existing data points
giving the ability to output the prob. Of new points.

For example, an anomaly detection system can be modeled starting from a dataset of normal and malicious activities. A Generative Gaussian Mixture will be able to distinguish between them, and to answer the question "Is a new data point representing an activity either normal or malicious?" by providing the probability of both cases.

Let's suppose we have a labeled dataset {Xl,Yl} containing N datapoints (from Pdata) and an unlabeled dataset Xu containing M >> N (not necessary) Points (drawn from the marginal distriution f(theta)), however we have to create a real semi-supervied scenario, with only few labeled samples

Our goal is to determine p(X hat, y hat) dist. Using generative model and then to obtain the conditional dist. p(y hat |X hat). It's possible to use different priors but are now employing multivariate Gaussians to our data:

$$f(\bar{x};\ \bar{\mu};\Sigma) = \frac{1}{\sqrt{\det 2\pi\Sigma}} e^{-\frac{(\bar{x}-\bar{\mu})^T\Sigma^{-1}(\bar{x}-\bar{\mu})}{2}}$$

Our model parameters are means and covariance matrices for all Gaussians. In other contexts, it's possible to use binomial or multinomial distributions. However, the procedure doesn't change

So let's assume it's possible to approximate p(X hat| y hat) with a parametrized distribution p(X hat |y hat; theta) we can achieve it by minimizing the Kullback-Leibler divergence bet. 2 dists.:

$$\underset{\bar{\theta}}{\operatorname{argmin}}\, D_{KL}(p(\bar{x}|y)||p(\bar{x}|y;\ \bar{\theta})) = \sum_i p(\bar{x}_i|y_i)\log\frac{p(\bar{x}_i|y_i)}{p(\bar{x}_i|y_i;\ \bar{\theta})}$$

On one of the following chapters we are going to show that this is equivalent to maximizing the likelihood of the dataset. To obtain the likelihood, it's necessary to define the number of expected Gaussians (which is known from the labeled samples) and a weight-vector that represents the marginal probability of a specific Gaussian:

$$\bar{w} = (p(y=1), p(y=2), \ldots, p(y=M))$$

Using Bayes' theorem, we get:

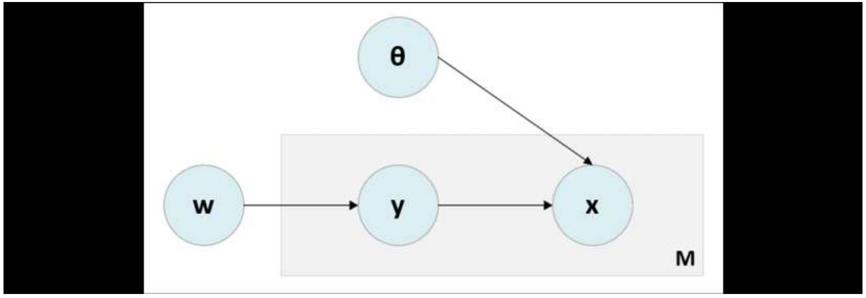
$$p(y_i|\bar{x}_j;\ \bar{\theta}, \bar{w}) \sim w_i p(\bar{x}_j|y_i;\ \bar{\theta})$$

we can get an expression for the conditional distribution of the points X given the parameter vector and the weight vector W hat :

$$p(\bar{x}_j|y_i;\ \bar{\theta}) = \sum_i w_i p(\bar{x}_j|\bar{y}_i;\ \bar{\theta})$$

it's easy to understand the role of each Gaussian in determining
the probability of a new point.

The model can be also quickly visualized using the plate notation, as shown in the following figure, where the rectangle represents a repeating block (in this case, repeating M times) and
the circles represent variables that are conditionally linked by arrows



now consider the complete expression of p(yi | Xj hat; theta hat,w hat) :

$$p(y_i|\bar{x}_j;\ \bar{\theta}, \bar{w}) = \frac{w_i p(\bar{x}_j|y_i;\ \bar{\theta})}{\sum_i w_i p(\bar{x}_j|y_i;\ \bar{\theta})}$$

Since we're working with both labeled and unlabeled samples the previous formula has
a double Interpretation:

For unlabeled samples, it's computed by multiplying the ith Gaussian weight times the
Probability p(Xi hat) relative to the ith Gaussian distribution.

For unlabeled samples it can be represented by a vector p hat =(0,0,..,1,0,0) in this
Way we force our model to trust the labeled samples , to find the best
parameter values that maximize the likelihood of the whole dataset.

With this distinction we consider the log-likelihood function where the term
Fw(yi| X hat j) has been substituted by a per sample weight

$$L(\bar{\theta};\ \bar{w}) = \sum_j \log \sum_i f_w(y_i|\bar{x}_j)\, p(\bar{x}_j|y_i;\ \bar{\theta}) = \sum_j \log \sum_i w_i\, p(\bar{x}_j|y_i;\ \bar{\theta})$$

$Fw(yi \mid X\ hat\ j) = Wi$

Is computed according to the previously explained method

$$\qquad j \qquad i \qquad\qquad\qquad j \qquad i$$

$Fw(yi \mid X\ hat\ j) = Wi$

$p(y_i|\bar{x}_j;\ \bar{\theta},\bar{w})$ Is computed according to the previously explained method

The parameters of the Gaussians are updated using these rules:

$$\begin{cases} w_i = \dfrac{\sum_j p(y_i|\bar{x}_j;\ \bar{\theta},\ \bar{w})}{N} \\ \bar{\mu}_i = \dfrac{\sum_j [p(y_i|\bar{x}_j;\ \bar{\theta},\ \bar{w})\bar{x}_j]}{\sum_j p(y_i|\bar{x}_j;\ \bar{\theta},\ \bar{w})} \\ \Sigma_i = \dfrac{\sum_j \left[p(y_i|\bar{x}_j;\ \bar{\theta},\ \bar{w})(\bar{x}_j - \bar{\mu}_i)(\bar{x}_j - \bar{\mu}_i)^T\right]}{\sum_j p(y_i|\bar{x}_j;\ \bar{\theta},\ \bar{w})} \end{cases}$$

N is the total number of samples. The procedure must be iterated until the parameters stop modifying or the modifications are lower than a fixed threshold.

Example of Generative Gaussian Mixture

This Example will in a notebook

Summary of Generative Gaussian Mixture

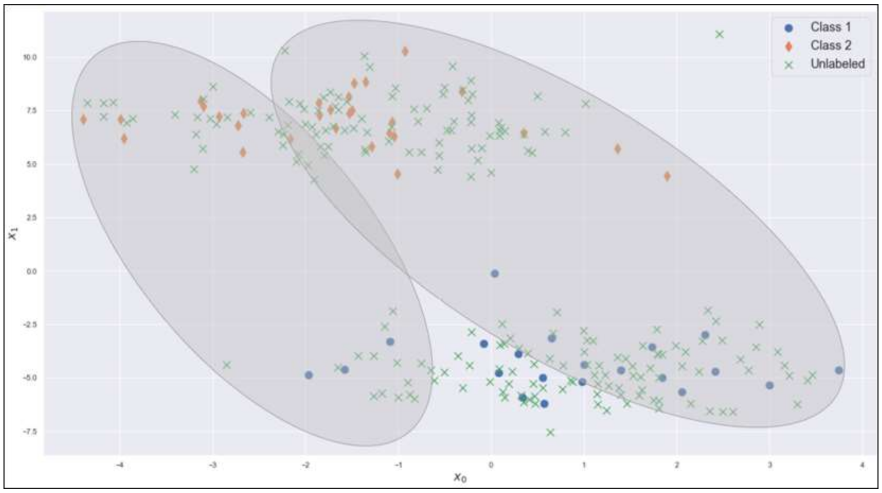
Generative Gaussian Mixtures are models that can learn the structure of a dataset And output the probability of any data point. They are based on both labeled and unlabeled samples, which are assumed to be equally trustworthy that is to say The unlabeled points contribute to the final positioning of the gaussians as like the labeled ones.
As we're discussing in the next section, this condition is not always met and it's necessary to introduce a slight modification in the algorithm.

This algorithm is very fast and produces excellent results in terms of density estimation.

In The previous ex we have considered a single log-likelihood for both labeled And unlabeled samples:

$$L(\bar{\theta};\ \bar{w}) = \sum_j \log \sum_i f_w(y_i|\bar{x}_j)\, p(\bar{x}_j|y_i;\ \bar{\theta}) = \sum_j \log \sum_i w_i\, p(\bar{x}_j|y_i;\ \bar{\theta})$$

This is equivalent to saying that we trust the unlabeled points just like the labeled ones.
However in contexts, this assumption can to completely wrong estimations as in the figure:



Biased final Gaussian mixture configuration

In this case, the means and covariance matrices of both Gaussian distributions have been biased by the unlabeled points and the resulting density estimation is clearly wrong.

If the unlabeled points have a fair contribution,
the mean vectors and covariance matrices should be relatively similar in both cases (for example, the norms of the differences are expected to be smaller than a predefined threshold that might be set equal to 1/10 of the largest element). Moreover, it's possible to compare the non-diagonal elements of the covariance matrices to check whether the orientations are extremely different.
In both cases, a large difference highlights a dominance
of the unlabeled samples over the labeled ones, and the final log-likelihood is often less than the expected one

When this phenomenon happens, the best thing to do is to consider a double weighted log-likelihood. If the first N samples are labeled and the following M are unlabeled, the log-likelihood can be expressed as follows:

$$L(\bar{\theta};\, \bar{w}) = \sum_{j=1}^{N} \log \sum_{i} p(y_i|\bar{\theta})\, p(\bar{x}_j|y_i;\, \bar{\theta}) + \lambda \sum_{j=N+1}^{N+M} \log \sum_{i} w_i\, p(\bar{x}_j|y_i;\, \bar{\theta})$$

We first compute the mean and covariance of the labeled points then we
Compute the mean and covariance of the unlabeled points with adjusted weight
Decrease the importance of the unlabeled points.

As we explained before, there are many potential rules of thumb to determine the effect of the unlabeled sample over the labeled one. A possible strategy to find the optimal lambda could be based on the cross-validation performed on the labeled dataset Another more complex approach is to consider different increasing values of lambda, and pick the first one where the log-likelihood is the maximum. In both cases, the goal is to find a value that avoids the dominance of the unlabeled samples, and at the same time, doesn't overestimate the role of the distribution

---