Gaussian Mixture Models

John Thickstun

Suppose we have data $x \in \mathbb{R}^d$ sampled from a mixture of K Gaussians with unknown parameters (μ_k, Σ_k) and mixing weights π_k. Formally, we can express the Gaussian mixture model (GMM) with the following generative process:

1. $z \sim \text{Categorical}_\pi(K)$,
2. $x \sim \mathcal{N}(\mu_z, \Sigma_z)$.

The mixture distribution is given by a density

$$p(x) = \int_{\mathcal{Z}} p(x, z)\, dz = \int_{\mathcal{Z}} p(x|z)p(z)\, dz = \sum_{k=1}^{K} \pi_k p(x|z = k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k).$$

Given i.i.d. observations $x_i \sim p$, $i = 1, \ldots, n$, we want to estimate the distribution p.

How do we measure success? In machine learning, we usually care about predictive accuracy, i.e. we want to maximize the expected likelihood that our model $\hat{p}$ assigns to new observations from the distribution:

$$\mathop{\mathbb{E}}_{x \sim p} \log \hat{p}(x). \tag{1}$$

Information theorists refer to the negation of Equation (1) as the cross-entropy. Crucially, when we measure error via cross-entropy, we approximate the expectation using fresh samples $x \sim p$ and *not* the samples that we used to fit the model $\hat{p}$. The machine learning community refers to this separation of samples for training and evaluation as a train/test split. We are interested in measuring the generalization error of our model (i.e. its performance on test).

The overfitting story for generative models is different than the standard story for classification problems. In the machine learning community, we are used to training large over-parameterized models for supervised learning problems that send the classification error to zero on the training data. The cross-entropy cannot be sent to zero, even with a perfect model

$$\mathop{\mathbb{E}}_{x \sim p} -\log \hat{p}(x) = \mathop{\mathbb{E}}_{x \sim p} -\log \frac{\hat{p}(x)}{p(x)} p(x) = H(p) + D(p \parallel \hat{p}) \geq H(p). \tag{2}$$

For a finite-sample training set of size n, assuming the density $p(x)$ is continuous and therefore the training set almost surely contains no duplicate elements, the empirical cross-entropy is maximized by setting $p(x_i) = 1/n$. Therefore, even overfitting to the training set at most achieves $\log n$ empirical cross-entropy. In practice, overfitting the training data to this degree is inadvisable for generative modeling and will result in massive or even infinite generalization error.

Gradient Descent and SGD

The most direct way to achieve small cross-entropy error in Equation 1 is to maximize the likelihood of our observed data $x_1, \ldots, x_n \sim p$. Let θ denote the collective parameters of a K-GMM. The maximum likelihood estimator (MLE) $\hat{p}_{\text{mle}}$ is given by $p_{\hat{\theta}_{\text{mle}}}$ where

$$\hat{\theta}_{\text{mle}} = \arg\min_{\theta} \mathop{\mathbb{E}}_{x \sim p} -\log p_\theta(x) \approx \arg\min_{\theta} \sum_{i=1}^{n} -\log p_\theta(x_i). \tag{3}$$

How do we optimize Equation (3)? One option is gradient descent:

$$\theta^{(k)} = \theta^{(k-1)} + \eta \nabla_\theta \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i). \tag{4}$$

Or we could use the stochastic variant of gradient descent (SGD):

$$\theta^{(k)} = \theta^{(k-1)} + \eta \nabla_\theta \log p_\theta(x_{k \pmod n}). \tag{5}$$

The point of SGD is that $\log p_\theta(x_i)$ is an unbiased estimator of Equation 1. While the SGD updates have a lot of variance compared to the batch estimator $\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i)$, the cost of computing Equation 5 is $O(1)$ whereas the cost of Equation 4 is $O(n)$. For practical large-scale optimizations, we typically work with a tradefoff between GD and SGD known as minibatch SGD. The idea is to use more than one sample, but less than the full dataset, to construct an estimate of Equation (1). This tradeoff is analyzed in detail by Jain et al. [2017].

The Evidence Lower-Bound

The SGD algorithm depended on our ability to evaluate the marginal probability of a sample $p_\theta(x)$. This is relatively easy for GMMs, but for many latent variable models, the integral over the latent state is intractable. In such cases, we can approximate the integral by importance-sampling with a proposal distribution $q(z|x)$:

$$\log p_\theta(x) = \log \mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \left[\frac{p_\theta(x, z)}{q(z|x)} \right] \tag{6}$$

$$= \mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \left[\log \frac{p_\theta(x, z)}{q(z|x)} \right] + D(q(z|x) \parallel p_\theta(z|x)). \tag{7}$$

Dropping the divergence term gives us a lower bound on the likelihood:

$$\log p_\theta(x) \geq \mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \left[\log \frac{p_\theta(x, z)}{q(z|x)} \right] \tag{8}$$

$$= \mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \big[\log p_\theta(x|z) \big] - D(q(z|x) \parallel p(z)). \tag{9}$$

Equality holds exactly when $q(z|x)$ equals the posterior $p_\theta(z|x)$. Equation (8) is sometimes called the "evidence lower-bound" [Kingma and Welling, 2013]. Another way to look at this bound is that we have applied Jensen's inequality to (6).

Expectation-Maximization

We can find the MLE by jointly optimizing the lower bound given by Equation (8) over q and θ:

$$\hat{\theta}_{\text{mle}} = \arg\max_{\theta} \max_{q} \mathop{\mathbb{E}}_{\substack{x\sim p \\ z\sim q(\cdot|x)}} \big[\log p_\theta(x|z)\big] - D(q(z|x) \parallel p_\theta(z)). \tag{10}$$

A natural algorithm for optimizing an objective like (10) is alternating maximization, which is referred to in this setting as Expectation-Maximization (EM). The idea is to repeatedly apply the following two-step update:

1. (E-step) Fixing parameters θ, optimize the proposal distribution q to maximize (10).
2. (M-step) Fixing the proposal distribution q, optimize the parameters θ to maximize (10).

A version of this algorithm was first proposed by Hartley [1958]. It was generalized and given the name EM by Dempster et al. [1977]. The alternating-maximization perspective is discussed in Csiszár and Tusnády [1984], along with an information-theoretic perspective and convergence results.

For some families of latent-variable models, including GMM's and other classical models like HMM's, it is possible to analytically compute the maximizer in the E-step. By observing equation (7), we see that the E-step is maximized by setting $q(z|x)$ to be the posterior $p_\theta(z|x)$. For GMM's, this posterior distribution can be computed by Bayes' rule:

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} = \frac{\pi_z \mathcal{N}(x; \mu_z, \Sigma_z)}{\sum_{k=1}^K \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}. \tag{11}$$

And for GMM's, we can also analytically compute the M-step to find updated parameters θ'. Replacing the expectation over x in (10) with a finite sum over observed data $x_1, \ldots, x_n \sim p$, we can show that the maximum over the parameters $\theta' = (\pi', \mu', \Sigma')$ is given by

$$\pi'_k = \frac{1}{n}\sum_{i=1}^n p_\theta(z_k|x_i), \qquad \mu'_k = \frac{1}{n\pi'_k}\sum_{i=1}^n p_\theta(z_k|x_i)x_i, \qquad \Sigma'_k = \frac{1}{n\pi'_k}\sum_{i=1}^n p_\theta(z_k|x_i)(x_i - \mu'_k)^{\otimes 2}. \tag{12}$$

From a slightly different perspective, we can view EM as an instance of the Minorization-Maximization (MM) algorithmic template [Hunter and Lange, 2004]. For any fixed posterior distribution $p_\theta(z|x)$, the objective (10) is a global lower bound on the likelihood. However, this bound is only tight at θ; EM makes progress on the objective by iteratively maximizing these global lower bounds. A rough analogy can be drawn to methods like Newton, that iteratively maximize a second-order Taylor expansion of the objective. However, the quadratic auxiliary objectives used in Newton's method are not guaranteed to be global lower bounds on the original objective (a valid lower bound can be constructed by cubic regularization [Nesterov and Polyak, 2006], leading to an MM algorithm).

References

I Csiszár and G Tusnády. Information geometry and alternating minimization problems. *Statistics & Decision, Supplement Issue No*, 1, 1984. (document)

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977. (document)

Herman O Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 1958. (document)

David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 2004. (document)

Prateek Jain, Praneeth Netrapalli, Sham M Kakade, Rahul Kidambi, and Aaron Sidford. Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification. *The Journal of Machine Learning Research*, 2017. (document)

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. (document)

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 2006. (document)