Indian Institute of Technology Kanpur
CS777 Topics in Learning Theory

Scribe: Abhibhav Garg
Instructor: Purushottam Kar
Date: April 22, 2018

LECTURE
22

Probabilistic Modelling-I

1 Introduction

We now move away from alternating minimization methods, and consider probabilistic methods of learning. Broadly speaking, these methods assume that data is generated from some distribution, which is possibly parametric, with the goal being to estimate these parameters from a given sample. This lecture will deal primarily the most basic methods of probabilistic modelling, namely point estimation methods. It will then discuss a very general algorithm known as Majorization Minimization, and will end with an analysis of the EM algorithm.

1.1 Notation

We use $\mathcal{X}$ as usual to denote the domain, and use $p_\theta(x)$ to denote the distribution modelled by $\theta \in \Theta$. We alternatively use $\mathbb{P}[x|\theta]$ for the same. We assume that the training data itself is drawn i.i.d , $x_i \sim p_{\theta^*}(x)$ for some θ^*.

2 Point Estimators

As the name suggests, a point estimator is an estimator that outputs a single $\theta \in \Theta$ as its estimate. Arguably the simplest method for estimating the θ that gave rise to some given data is to find that θ that maximizes the likelihood of the data - given no other information about the model, this is also the best point estimate we can do.

Let $\{x_1, \ldots, x_n\} = X$ denote the data. The likelihood of a particular θ is then given by

$$\mathbb{P}[X|\theta] = \prod_{i=0}^{n} p_\theta(x_i)$$

where the equality follows from the i.i.d assumption. The **Maximum Likelihod Estimate,** θ_{MLE} is then simply that θ that maximizes the above quantity. Note that this is exactly that parameter that maximizes the probability of seeing the given data.

$$\begin{aligned}
\theta_{MLE} &= \arg\max_{\theta \in \Theta} \prod_{i=0}^{n} p_\theta(x_i) \\
&= \arg\max_{\theta \in \Theta} \sum_{i=0}^{n} \log p_\theta(x_i) && \text{(monotonicity)}
\end{aligned}$$

If on the other hand, we also had some prior belief $\mathbb{P}[\theta]$ on the parameter even before looking at the data, this can be naturally Incorporated into the estimator, by maximizing the posterior distribution, given below, instead of the likelihood.

$$\mathbb{P}[\theta|X] = \mathbb{P}[X|\theta]\,\mathbb{P}[\theta] = \mathbb{P}[\theta]\prod_{i=0}^{n} p_\theta(x_i)$$

This gives us the **Maximum A Posteriori Estimate,** θ_{MAP}, given by

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta\in\Theta} \mathbb{P}[\theta]\prod_{i=0}^{n} p_\theta(x_i) \\ &= \arg\max_{\theta\in\Theta} \log\mathbb{P}[\theta] + \sum_{i=0}^{n}\log p_\theta(x_i) && \text{(monotonicity)}\end{aligned}$$

2.1 Latent Variables

In a lot of settings, not all of the generated variables are observed in the form of data. There are some variables that are "hidden" or "latent", and can only be inferred from the observed variables. More formally, p_θ now has support over $\mathcal{X}\times\mathcal{Z}$, instead of just $\mathcal{X}$. The data is generated as $(x_i, z_i) \sim p_{\theta^*}$ but the data that is observed is only $(x_1, \ldots, x_n)$, with the (z_i) hidden. These z_i are the latent variables.

For example, consider the following clustering problem, with two clusters. Let the clusters have means μ_0, μ_1. Further, let the weights of the clusters be w_0, w_1. The parameter θ then looks like $((\mu_0, \mu_1), (w_0, w_1))$. The samples are then generated via the following two step process:

$$\begin{aligned} z_i &\sim Bernoulli(\frac{w_0}{w_0+w_1}) \\ x_i &\sim \mathcal{N}(\mu_{z_i}, \sigma^2)\end{aligned}$$

Here, the z_i, namely the cluster from which a given point was generated from is latent, only the final position is seen.

Similarly, consider the following (noisy) Mixed Regression Problem. Here, there are multiple possible weight vectors. In particular, θ looks like $((c_i), (w_i))$. The data is generated as

$$\begin{aligned} z_i &\sim Bernoulli(\frac{w_i}{\sum w_i}) \\ y_i &\sim \mathcal{N}(w_{z_i}^T x_i, \sigma^2)\end{aligned}$$

Again, along with x_i, only y_i is given to us, and the z_i are hidden.

2.2 Latent Variable Models are Hard

While the linear regression problem, in the setting without latent variables has a polynomial time closed form solution, adding latent variables in the above manner makes the point estimation for the above problem NP-Hard. This can be seen by looking at the likelihood estimator,

$$\theta_{MLE} = \arg\max_{\theta\in\Theta}\sum_{i=0}^{n}\log p_\theta(x_i)$$

In order to find the above RHS, we need to marginalize over all possible Z. This gives

$$\theta_{MLE} = \arg\max_{\theta \in \Theta} \sum_{i=0}^{n} \log \sum_{z \in \mathcal{Z}} p_\theta(x_i, z)$$

Due to a lack of closed form for the above in most settings, the problem becomes very hard. There are special cases though, such as dimensionality reduction, where this problem is avoided.

2.3 Overcoming The Intractability

Consider the following two observations.

- *Observation 1:* If the latent variables, or atleast a good estimate of the latent variables $\hat{z}_i$ were known to us, then these problems get changed to problems which we have solved before, namely finding the mean/regression in the above examples. In other words,

  $$\theta_{MLE} = \arg\max_{\theta \in \Theta} \sum_{i=0}^{n} \log p_\theta(x_i, \hat{z}_i)$$

  is an easy problem.

- Observation 2 If we knew θ^*, then estimating the z_i is also very simple. Given an x_i, we can just estimate z_i to be that which maximizes the probability of z_i given x_i and θ, using the Bayes rule, as follows

  $$\hat{z}_i = \arg\max_{z \in \mathcal{Z}} \mathbb{P}\left[z | x_i, \theta^*\right] \propto \mathbb{P}\left[x_i | z_i, \theta^*\right] \mathbb{P}\left[z\right]$$

The above two observations makes the problem an obvious candidate to apply our newfound alternating optimization tools on, as follows:

Algorithm 1: ALT-OPT

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad z_i^{t+1} = \arg\max_{z \in \mathcal{Z}} \mathbb{P}\left[z | x_i, \theta^*\right]$
3: $\quad \theta^{t+1} = \arg\max_{\theta \in \Theta} \sum_{i=0}^{n} \log p_\theta(x_i, \hat{z}_i)$
4: **end for**

Roughly, the algorithm first picks that latent variable that is most likely given the current model, then updates the model to match this new latent variable information.

Notice that in the above, in the first step, we pick that z that is most likely. However, we could alternatively pick multiple z, and weigh them based on how likely they are. This way, even if the most likely z, is for some reason (such as poor initialization) not optimal, we are not fully lost. This is called Expectation Maximization. This of course, is considerably more expensive.

3 Majorization Minimization (Minorization Maximization) (MM)

We now move on to another very popular alt-min algorithm, MM. We first discuss the algorithm, before showing that two common algorithms are just MM in disguise.

Assume that we want to minimize $f(x)$ where the domain of x is $\mathcal{X}$. The majorization minimization algorithm works as follows: In each time step, it constructs a proxy for the true minimization problem. This proxy has the following property: at x^t, its value is the same as that of f, and at every other point, is at-least as much as f. This proxy is minimized instead, and this process is repeated.

Algorithm 2: Majorization Minimization

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad g^{t+1} = MAJ(f, x^t)$
3: $\quad x^{t+1} = \arg\min_{x \in \mathcal{X}} g^{t+1}$
4: **end for**

where $MAJ(f, x^t)$ returns a function g^{t+1} with the following three properties:

1. $g^{t+1}(x) \geq f(x), \forall x \in \mathcal{X}$.
2. $g^{t+1}(x^t) = f(x^t)$.
3. g^{t+1} should be "easy" to minimize.

It is easy to see that in every step, the objective is either unchanged or reduced, never increased:

$$f(x^t) = g^{t+1}(x^t) \geq g^{t+1}(x^{t+1}) \geq f(x^{t+1})$$

However, proving that this algorithm actually makes significant progress needs to be proved on a per algorithm basis.

Finally, the Minorization Maximization algorithm works just as expected, where the function used is $MIN(f, x^t)$ which returns a function g^{t+1} with the expected properties, that is, it should be less than or equal to f at all points.

3.1 IRLS is MM

Consider the case where we want to do regression, but we do not expect the noise to be Gaussian. For example, maybe we have some evidence that the noise has heavy tails. An alternate model for the noise is then a Laplacian likelihood, which looks like

$$P(y^i | x^i, \theta \propto exp\left(-\frac{\left|y^i - \theta^T x^i\right|}{\sigma^2})\right)$$

Thus, to perform a simple MLE estimate, we need to solve

$$\arg\min_{\theta \in \Theta} \sum_{i=0}^{n} \left|y^i - \theta^T x^i\right|$$

Using the facts that

$$a^2 \leq b^2 + a^2$$
$$|a| \leq \frac{b^2}{|a|} + |a| \qquad \text{(When } a \neq 0\text{)}$$

We get

$$\underbrace{\left|y^i - \theta_t^T x^i\right|}_{f(\theta^t)} \leq \underbrace{\frac{1}{\left|y^i - \theta_t^T x^i\right|}\left(y^i - \theta^T x^i\right)^2}_{g(\theta)} + \left|y^i - \theta^T x^i\right|$$

Note that since the last term is a constant, we do not need to include it in g, though it does not change anything. This is IRLS, where the points that fit better are given a greater weightage in subsequent iterations.

3.2 EM (Minorization Maximization)

The fact that EM is basically Minorization Minimization is seen from the following:

$$\begin{aligned}
\underbrace{\sum_{i=0}^{n} \log \sum_{z \in \mathcal{Z}} \mathbb{P}\left[x^i, z|\theta\right]}_{f(\theta)} &= \sum_{i=0}^{n} \log \sum_{z \in \mathcal{Z}} \mathbb{P}\left[z|x^i, \theta^t\right] \frac{\mathbb{P}\left[x^i, z|\theta\right]}{\mathbb{P}\left[z|x^i, \theta^t\right]} \\
&= \sum_{i=}^{n} \log \mathop{\mathbb{E}}_{z \sim \mathbb{P}[.|x^i, \theta^t]} \left[\frac{\mathbb{P}\left[x^i, z|\theta\right]}{\mathbb{P}\left[z|x^i, \theta^t\right]}\right] \\
&\geq \sum_{i=}^{n} \mathop{\mathbb{E}}_{z \sim \mathbb{P}[.|x^i, \theta^t]} \left[\log \frac{\mathbb{P}\left[x^i, z|\theta\right]}{\mathbb{P}\left[z|x^i, \theta^t\right]}\right] \\
&= \underbrace{\sum_{i=1}^{n} \sum_{z \in \mathcal{Z}} \mathbb{P}\left[z|x^i, \theta^t\right] \log \mathbb{P}\left[x^i, z|\theta\right]}_{g^{t+1}} + c
\end{aligned}$$

where the inequality follows by Jensens. The constant c captures the denominator in the second last expression. Note that this is non negative because it is the entropy of a distribution. Thus, g^{t+1} satisfies the properties we expect from a minorizer.

4 Analysis Technique for MM algorithms

We finish this lecture by looking at an analysis technique by Balakrishnan et al. Balakrishnan et al. (2017) which was originally developed for EM. We first set up some notation and definitions. Let

$$M(\theta) = \arg\min(MAJ(f, \theta))$$

Further, let

$$g_{\theta^*} = MAJ(f, \theta^*)$$

Definition 22.1. $g_{\theta^*}(\cdot)$ is α strongly convex if

$$g_{\theta^*}(\theta^1) \geq g_{\theta^*}(\theta^2) + \left\langle \nabla g_{\theta^*}(\theta^2), \theta^1 - \theta^2 \right\rangle + \frac{\alpha}{2}\left\|\theta^2 - \theta^1\right\|_2^2$$

Definition 22.2. $g_{\theta^*}(\cdot)$ is β strongly stable if

$$g_{\theta^*}(M(\theta^1)) \leq g_{\theta^*}(M(\theta^2)) + \left\langle \nabla g_{\theta^*}(M(\theta^2)), M(\theta^1) - M(\theta^2) \right\rangle + \frac{\beta}{2} \left\|\theta^2 - \theta^1\right\|_2 \left\|M(\theta^2) - M(\theta^1)\right\|_2$$

Equivalently, $g_{\theta^*}(\cdot)$ is β strongly stable if

$$\left\|\nabla g_{\theta^*}(M(\theta^1)) - \nabla g_{\theta^*}(M(\theta^2))\right\| \leq \beta \left\|\theta^1 - \theta^2\right\|$$

Unlike the previous analysis, we want the ratio $\frac{\beta}{\alpha} \leq 1$. The reason for this will become clear shortly. Further, for brevity, we write g in place of g_{θ^*}. We now have

$$g(M(\theta^t)) \geq g(M(\theta*)) + \frac{\alpha}{2} \left\|\theta^t - \theta^*\right\|_2^2$$
$$(\text{SC, } \nabla g(M(\theta^*)) = 0)$$
$$g(M(\theta^*)) \geq g(M(\theta^t)) + \left\langle \nabla g(M(\theta^t)), M(\theta^*) - M(\theta^t) \right\rangle + \frac{\alpha}{2} \left\|M(\theta^t) - M(\theta^*)\right\|_2^2 \qquad (\text{SC})$$

Adding the above two, and applying Cauchy Schwartz gives us

$$\left\|\nabla g(M(\theta^t))\right\| \left\|M(\theta^*) - M(\theta^t)\right\| \geq \alpha \left\|M(\theta^t) - M(\theta^*)\right\|_2^2 \tag{1}$$

Using strong stability we now get

$$\left\|\nabla g(M(\theta^t)) - \nabla g(M(\theta^*))\right\| = \left\|\nabla g(M(\theta^t))\right\| \tag{2}$$
$$\leq \beta \left\|\theta^t - \theta^*\right\| \tag{3}$$

Combining the above gives us

$$\left\|M(\theta^t) - M(\theta^*)\right\|_2 = \left\|M(\theta^t)\right\|_2$$
$$\leq \frac{\beta}{\alpha} \left\|\theta^t - \theta^*\right\|$$

Since $M(\theta^t) = \theta^{t+1}$ by definition, this tells us that the distance from the optimal contracts by a constant factor after every iteration, guaranteeing quick convergence.

4.1 Gradient Descent

To finish off this lecture, we show that gradient descent can be analyzed using the above technique. Assume that we are performing gradient descent on f, which is α strongly convex and β strongly smooth. Set

$$g^{t+1}(\theta) := f(\theta^t) + \left\langle \nabla f(\theta^t, \theta - \theta^t \right\rangle + \frac{1}{2\eta} \left\|\theta - \theta^t\right\|_2^2$$

We make our usual assumption that $\eta \leq \frac{1}{\beta}$. It follows by definition that $g^{t+1}(\theta) \geq f(\theta)$. Further, it is not hard to see that the optimizer of $g^{t+1}(\theta)$, namely θ^{t+1} is exactly $\theta^{t+1} = \theta^t - \eta \nabla f(\theta^t)$. Thus, with this interpretation, gradient descent is exactly MM. We have

$$g_{\theta^*}(\theta) = \frac{1}{2\eta} \left\|\theta - \theta^*\right\|_2^2$$
$$\Rightarrow \nabla g_{\theta^*}(\theta) = \frac{1}{\eta} \left\|\theta - \theta^*\right\|_2$$

which gives us strong convexity, with parameter $\hat{\alpha} = \frac{1}{2\eta}$

Now note that

$$\begin{aligned}
\left\|\nabla g_{\theta^*}(M(\theta^1)) - \nabla g_{\theta^*}(M(\theta^2))\right\| &= \frac{1}{\eta}\left\|\theta^1 - \eta\nabla f(\theta^1) - \theta^2 - \eta\nabla f(\theta^2)\right\|_2^2 \\
&= \frac{1}{\eta}\left\|\theta^1 - \theta^2\right\|_2^2 + \eta\left\|\nabla f(\theta^1) - \nabla f(\theta^2)\right\|_2^2 - 2\left\langle\theta^1 - \theta^2, \nabla f(\theta^1) - \nabla f(\theta^2)\right\rangle \\
&\leq \frac{1}{\eta}\left\|\theta^1 - \theta^2\right\|_2^2 + \frac{\beta}{\eta}\left\|\theta^1 - \theta^2\right\|_2^2 + \alpha\left\|\theta^1 - \theta^2\right\|_2^2 \\
&\leq \frac{1}{2\eta}\left(1 - \frac{\alpha}{\beta}\right)\left\|\theta^1 - \theta^2\right\|_2^2
\end{aligned}$$

Where the inequality follows from the assumed properties of the function. This gives us an effective stability parameter $\hat{\beta} = \frac{1}{2\eta}\left(1 - \frac{\alpha}{\beta}\right)$ and thus allows us to apply the previous discussed method.

References

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Ann. Statist.*, 45(1):77–120, 02 2017. doi: 10.1214/16-AOS1435. URL `https://doi.org/10.1214/16-AOS1435`.