Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.438 Algorithms for Inference
Fall 2020

20 Stucture Learning II

In this installation of the notes, we will investigate another technique for learning the structure of an undirected graphical model, by viewing it as a regression problem. In the previous notes, we discussed how moment matching could be used to learn structure. However, such an approach is only computationally possible if there are only a few variables of interest. This gives intuition for the regression techniques we will explore today, where we consider the neighborhood of each node one at a time. Such techniques will have a massive computational difference over those we saw last time.

20.1 Binary graphical model learning as Logistic Regression

Assume that we have N binary random variables $\mathsf{x}_1, \ldots, \mathsf{x}_N$ distributed according to some unknown graph $\mathcal{G}$. Furthermore, assume that the graph can be factorized using only pairwise potentials, so

$$\mathbb{P}_\theta(\mathsf{x}_1 = x_1, \ldots, \mathsf{x}_N = x_N) = \frac{1}{Z(\theta)} \exp\left(\sum_{i=1}^{N} \theta_{ii} x_i + \sum_{i<j} \theta_{ij} x_i x_j\right) \tag{1}$$

Let's start with a simple question: What is the conditional distribution of x_1 given $\mathsf{x}_2, \ldots, \mathsf{x}_N$? Well, we obtain that

$$\begin{aligned}\mathbb{P}_\theta(\mathsf{x}_1 = 1 \mid \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N) &= \frac{\mathbb{P}_\theta(\mathsf{x}_1 = 1, \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)}{\mathbb{P}_\theta(\mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)} \\ &= \frac{\frac{1}{Z(\theta)} \exp(\theta_{11} + \sum_{j=2}^{N} \theta_{1j} x_j + \sum_{i,j>1} \theta_{ij} x_i x_j)}{\mathbb{P}_\theta(\mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)}\end{aligned}$$

Similarly, we can argue that

$$\mathbb{P}_\theta(\mathsf{x}_1 = 0 \mid \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N) = \frac{\frac{1}{Z(\theta)} \exp(\sum_{i,j>1} \theta_{ij} x_i x_j)}{\mathbb{P}_\theta(\mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)}$$

Therefore, the ratio of these two terms is

$$\frac{\mathbb{P}_\theta(\mathsf{x}_1 = 1 \mid \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)}{\mathbb{P}_\theta(\mathsf{x}_1 = 0 \mid \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N)} = \exp\left(\theta_{11} + \sum_{j=2}^{N} \theta_{1j} x_j\right) \tag{2}$$

One may notice that this looks exactly like a *logistic regression* problem, where we are trying to classify x_1 from features $\mathsf{x}_2, \ldots, \mathsf{x}_N$.

20.1.1 Logistic Regression Review

In logistic regression, we have a label $Y \in \{0, 1\}$, features $\mathbf{Z} = (z_1, \ldots, z_L) \in \mathbb{R}^L$, and we would like to predict the label as a function of features. We use a linear predictor $\mathbf{w} \in \mathbb{R}^L$, such that

$$\frac{\mathbb{P}(Y = 1 \mid \mathbf{Z} = \mathbf{z})}{\mathbb{P}(Y = 0 \mid \mathbf{Z} = \mathbf{z})} \propto \exp(\sum_{k=1}^{L} w_k z_k), \tag{3}$$

or, solving,

$$\mathbb{P}(Y = 1 \mid \mathbf{Z} = \mathbf{z}) = \frac{\exp(\sum_{k=1}^{L} w_k z_k)}{1 + \exp(\sum_{k=1}^{L} w_k z_k)} \quad \mathbb{P}(Y = 0 \mid \mathbf{Z} = \mathbf{z}) = \frac{1}{1 + \exp(\sum_{k=1}^{L} w_k z_k)} \tag{4}$$

Given n observations $\mathbf{z}^k = (z_1^k, \ldots, z_L^k)$ and labels y^k for $k = 1, \ldots, n$ we would like to estimate $\mathbf{w}$. We can do so using the maximum likelihood estimation.

The normalized log likelihood of our data, $\mathcal{L}(\{y^k, \mathbf{z}^k\}_{k=1}^n, w)$, can be expressed as follows:

$$\begin{aligned}
\mathcal{L}(\{y^k, z^k\}_{k=1}^n, \mathbf{w}) &= \frac{1}{n} \sum_{k=1}^{n} \log \mathbb{P}(y^k \mid \mathbf{z}^k; \mathbf{w}) \\
&= \frac{1}{n} \sum_{k=1}^{n} y_k \log \mathbb{P}(Y = 1 \mid \mathbf{Z} = \mathbf{z}^k; \mathbf{w}) + (1 - y_k) \log \mathbb{P}(Y = 0 \mid \mathbf{Z} = \mathbf{z}^k; \mathbf{w}) \\
&= \frac{1}{n} \sum_{k=1}^{n} \left[y_k \left(\sum_{\ell=1}^{L} w_\ell z_\ell^k \right) - \log(1 + \exp(\sum_{\ell=1}^{L} w_\ell z_\ell^k)) \right]
\end{aligned}$$

Our goal is to maximize $\mathcal{L}(\{y^k, z^k\}_{k=1}^n, \mathbf{w})$, which we can do via gradient ascent. Pick some initialization $\mathbf{w}^{(0)}$. The gradient ascent update is given by

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \frac{1}{t} \nabla \mathcal{L}(\{y^k, z^k\}_{k=1}^n, \mathbf{w}) \tag{5}$$

for a specific parameter w_ℓ, this simplifies as

$$w_\ell^{(t+1)} = w_\ell^{(t)} + \frac{1}{t} \frac{\partial \mathcal{L}}{\partial w_\ell} \tag{6}$$

Now observe that

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_\ell} &= \frac{1}{n} \sum_{k=1}^{n} \left(y^k z_\ell^k - \frac{z_\ell^k \exp(\sum_{\ell=1}^{L} w_\ell z_\ell^k)}{1 + \exp(\sum_{\ell=1}^{L} w_\ell z_\ell^k)} \right) \\
&= \frac{1}{n} \sum_{k=1}^{n} z_\ell^k (y^k - (Y = 1 \mid \mathbf{Z} = \mathbf{z}^k; \mathbf{w}))
\end{aligned}$$

Plugging back in, our gradient ascent update simplifies to

$$w_\ell^{(t+1)} = w_\ell^{(t)} + \frac{1}{t}\left(\frac{1}{n}\sum_{k=1}^{n} z_\ell^k (y^k - (Y = 1 \mid \mathbf{Z} = \mathbf{z}^k; \mathbf{w}))\right) \tag{7}$$

20.1.2 Neighborhood Selection

In comparing to logistic regression, we see that our features are $\mathbf{Z} = (1, x_2, \ldots, x_N)$, our observation is $Y = x_1$, and our weights $\mathbf{w} = (\theta_{11}, \theta 12, \ldots, \theta 1N$. Therefore we can just apply the logistic regression gradient ascent update to obtain an iterative algorithm for the θ_{1i}:

$$\theta_{1i}^{(t+1)} = \theta_{1i}^{(t)} + \frac{1}{t}\left(\frac{1}{n}\sum_{k=1}^{n} x_i^k (x_1^k - \mathbb{P}(\mathsf{x}_1 = 1 \mid \mathsf{x}_2 = x_2^k, \ldots, \mathsf{x}_N = x_N^k; \theta))\right) \tag{8}$$

Note the similarity between this expression and our formula for moment matching in the previous notes. This time, however, we're working with the conditional distribution of x_1 given the other variables, rather than the entire distribution, which makes this update tractable.

Remarks: Assume the underlying graphical model has a maximum degree λ, and also that $\theta_{ij} \leq 1$. This means that $|\theta_1|_1 \leq \lambda$. In our logistic regression viewpoint, this is the constraint that $|\mathbf{w}|_1 \leq 1$.

This suggests the following modified logistic regression problem:

$$\max_{\mathbf{w}} \mathcal{L}(\{y^k, z^k\}_{k=1}^n, \mathbf{w}) - \nu|\mathbf{w}|_1 \tag{9}$$

As before, we can use gradient ascent to solve this problem. In the machine learning literature, this approach of using ℓ_1 regularization is known as *Lasso*, and enforces sparsity in the weight $\mathbf{w}$. The following theorem tells us that using the Lasso for each vertex can allow us to recover the true graph.

Theorem 1. *(Klivans, Meka 2017) Assume that the maximum degree of the graph is λ, and that all probabilities $\mathbb{P}(\mathsf{x}_1 = 1 \mid \mathsf{x}_2 \ldots \mathsf{x}_N; \theta^*)$ lie in $[\delta, 1-\delta]$ for some $\delta > 0$. Then, with $n = O(\log N \cdot \frac{1}{\epsilon^2})$ samples, this procedure produces an estimate $\hat{\theta}$ satisfying $\|\hat{\theta} - \theta^*\|_\infty \leq \epsilon$, in runtime $\tilde{O}(N^2)$.*

To use this method to recover the true underlying graph, if we have a lower bound on the entries of θ^* we can just choose ϵ less than that lower bound, and then threshhold any smaller entries to 0.

20.2 Generic graphical model learning as Regression

We next see if we can extend this approach beyond the binary setting. Let's say our distribution is given by

$$\mathbb{P}_\theta(\mathsf{x} = \mathbf{x}) = \frac{1}{Z(\theta)} \exp(\sum_{i,j} \theta_{ij} \mathsf{x}_i \mathsf{x}_j),$$

where now we let $\mathsf{x}_i \in \mathcal{X} = [0, 1]$. Our goal is to learn θ from data $(x_1^k, \ldots, x_N^k)_{k=1}^n$

By the identical calculation as above, we see that the conditional of x_1 is given by

$$\mathbb{P}_\theta(\mathsf{x}_1 = a \mid \mathsf{x}_2 = x_2, \ldots, \mathsf{x}_N = x_N) = \frac{1}{Z_1(x_2, \ldots, x_N)} \exp\left(\theta_{11} a^2 + a \sum_{j=2}^N \theta_{1j} x_j\right) \quad (10)$$

where the normalization constant is given by

$$Z_1(x_2, \ldots, x_N) = \int_0^1 \exp\left(\theta_{11} a^2 + a \sum_{j=2}^N \theta_{1j} x_j\right) da \quad (11)$$

Let's assume that $\theta_{ii} = 0$ for all i. The integral evaluates as:

$$Z_1(x_2, \ldots, x_N) = \int_0^1 \exp\left(a \sum_{j=2}^N \theta_{1j} x_j\right) da = \frac{\exp\left(\sum_{j=2}^N \theta_{1j} x_j\right) - 1}{\sum_{j=2}^N \theta_{1j} x_j} \quad (12)$$

We can then just do maximum likelihood estimation as we did previously. Using our notation from the previous section, where $\mathsf{x}_1 = Y, (\mathsf{x}_2, \ldots, \mathsf{x}_N) = \mathbf{Z}$, and $(\theta_{12}, \ldots, \theta 1N) = \mathbf{w}$, we obtain that

$$\mathbb{P}(Y = y \mid \mathbf{Z} = \mathbf{z}; \mathbf{w}) = \frac{1}{Z(\mathbf{z})} \exp\left(y \sum_{\ell}^L w_\ell z_\ell\right)$$
$$= \frac{(\sum_\ell^L w_\ell z_\ell) \cdot \exp(y \sum_\ell^L w_\ell z_\ell)}{\exp(\sum_\ell^L w_\ell z_\ell) - 1}$$

Given n data points, we can write the normalized log likelihood as

$$\begin{aligned}
\mathcal{L}(\{y^k, z^k\}_{k=1}^n, \mathbf{w}) &= \frac{1}{n} \sum_{k=1}^n \log \mathbb{P}(Y = y_k \mid \mathbf{Z} = \mathbf{z}; \mathbf{w}) \\
&= \sum_{k=1}^n \left(y^k \sum_\ell^L w_\ell z_\ell^k\right) + \sum_{k=1}^n \log\left(\sum_\ell^L w_\ell z_\ell^k\right) - \sum_{k=1}^n \log\left(\exp\left(\sum_\ell^L w_\ell z_\ell^k\right) - 1\right)
\end{aligned}$$

As before, we can compute the gradient of likelihood, and use gradient ascent to maximize this quantity. Similarly, if our goal is to learn a sparse graph, this is equivalent to $|\mathbf{w}|_1$ being small and thus we can add a $-\nu |w|_1$ penalty