10-708: Probabilistic Graphical Models, Spring 2015

26 : Spectral GMs

Lecturer: Eric P. Xing *Scribes: Guillermo A Cidre, Abelino Jimenez G.*

1 Introduction

A common task in machine learning is to work with high-dimensional data. To deal with that, generally people assume a structure data based on latent variables. In this kind of models, the inference is not only depending on observed variables, but also on unobserved latent variable.

The insertion of unobserved variables lets us have a good comprehension of the problem that we want to model. However, it is often the case that exact inference is intractable. Throughout the course, we have studied methods like EM algorithm and approximate inference methods, like Markov Chain Monte Carlo (MCMC), which are very popular, but they have several drawbacks. For example, MCMC can be slow, and EM can get stuck in local optima and be slow, as well.
Spectral methods arise as alternative to the standard methods in order to make inference. Thus, this kind of methods attempt only to estimate probabilities over the observable variables. There are many applications where the estimation of probabilities is more important than the latent variables themselves, so spectral methods provide useful estimates in a wide range of domains and models. Indeed, Spectral methods describe a model such that the probabilities of observable variables depend only on statistics over the observable variables themselves, preventing the difficulties which appear in the optimization problem of the standard approaches.

2 Graphical Models and Rank

2.1 Independence and Rank

We can consider the joint distribution of two random variables X_1,X_2 each taking values in $\{1, ..., N\}$. The joint probability of X_1 , X_2 can be fully represented by a $N \times N$ matrix $\mathcal{P}$ such that $\mathcal{P}_{ij} = \mathbb{P}(X_1 = i, X_2 = j)$. Moreover, the marginal probabilities of X_1 can be represented in a vector with N components $P(X_1)$, ie $P(X_1)_n = \mathbb{P}(X_1 = n)$. Similarly, we can define $P(X_2)$. Therefore, if X_1 and X_2 are independent, we have

$$\begin{aligned} \mathcal{P}_{ij} &= \mathbb{P}(X_1 = i, X_2 = j) \\ &= \mathbb{P}(X_1 = i)\mathbb{P}(X_2 = j) \\ &= P(X_1)_i P(X_2)_j \end{aligned}$$

Thus,

$$\mathcal{P} = P(X_1)P(X_2)^T$$

ie, if X_1 and X_2 are independent, then $\mathcal{P}$ is a rank 1 matrix.

On the other hand, we know that in case that X_1 and X_2 are not independent, the rank of $\mathcal{P}$ is at most N. Then, it is reasonable to ask what structure emerges when the rank of $\mathcal{P}$ is lower than N. Let us consider an example.

If X_1 and X_2 are not marginally independent, but they are conditionally independent by X, which has k states, then $rank(\mathcal{P}) \leq k$. Indeed, we can define the matrix $P(X_1|X)$ as the $N \times k$ matrix such that $P(X_1|X)_{ij} = \mathbb{P}(X_1 = i|X = j)$. Similarly we can define $P(X_2|X)$. Therefore, it is easy to prove:

$$\mathcal{P} = P(X_1|X) \cdot diag(P(X)) \cdot P(X_2|X)$$

In the right hand side, all the matrices have rank lower than k, then product must have rank lower than k.

Thus, we can say that latent variables models encode low rank dependencies among variables. To exploit this structure, Linear Algebra offers us a bunch of tools: Rank Computation, Eigenvalues, SVD, Tensor, and so on.

2.2 HMM example

Just for example, we can consider the following HMM, where the observable variables have m states and the hidden variable k states.

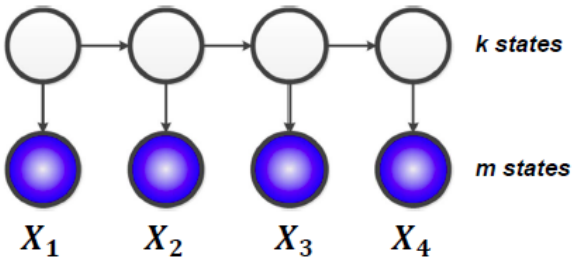

Figure 1: Graphical Model for a Hidden Markov Model.

We can define the matrix $\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}})$ the probability matrix, where the rows are corresponding the the possible values of (X_1, X_2) and the columns to the possible values of (X_3, X_4). With the previous information, we can say that $\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}})$ is rank lower than k. Indeed, we have to notice that

$$\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}}) = P(X_{\{1,2\}}|H_2) \cdot diag(P(H_2)) \cdot P(X_{\{3,4\}}|H_2)^T$$

And, as before, in the right hand side we have only matrices with rank lower than k. Notice that, with this factorization, we have expressed a matrix with depends on 4 variables, in a product of matrices which depend on 3, 1 and 3 variables respectively. However, we still have the dependency on hidden variables.

To resolve this, we just have to observe that this kind of factorizations are not unique. For example, if $M = LR$, we also can write $M = LSS^{-1}R$. So, if we consider the factor L and R, we can also consider LS and $S^{-1}R$. This observation is really useful for the Spectral Method, because we can make factorizations using only observable variables.

Continuing with our example, and using the same argument as before, we wan say

$$\mathcal{P}(X_{\{1,2\}}, X_3) = P(X_{\{1,2\}}|H_2) diag(P(H_2)) P(X_3|H_2)^T$$

and

$$\mathcal{P}(X_2, X_{\{3,4\}}) = P(X_2|H_2) diag(P(H_2)) P(X_{\{3,4\}}|H_2)^T$$

Additionally, noting that

$$\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}}) = P(X_{\{1,2\}}|H_2) \cdot diag(P(H_2)) \cdot P(X_{\{3,4\}}|H_2)^T$$

and

$$\mathcal{P}(X_2, X_3) = P(X_2|H_2)diag(P(H_2))P(X_3|H_2)^T$$

we have

$$\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}}) = \mathcal{P}(X_{\{1,2\}}, X_3)\mathcal{P}(X_2, X_3)^{-1}\mathcal{P}(X_2, X_{\{3,4\}})$$

ie, we have expressed a matrix depending on 4 variables, as a product of 3 matrices depending on 3, 2 and 3 variables respectively, and all of them observable. This last fact is really important, and is an advantage compared with the EM algorithm, which is based on hidden variables. However, this kind of factorizations may give inconsistent results, for example, giving negative probabilities.

Nevertheless, there are different factorizations for the same structure. For instance,

$$\mathcal{P}(X_{\{1,2\}}, X_{\{3,4\}}) = \mathcal{P}(X_{\{1,2\}}, X_4)\mathcal{P}(X_1, X_4)^{-1}\mathcal{P}(X_1, X_{\{3,4\}})$$

3 Relationship to original factorization

We can write the new factorization as the old one with an invertible term. Let

$$\begin{aligned} M &= \mathcal{P}\left[X_{\{1,2\}}, X_{\{3,4\}}\right] \\ L &= \mathcal{P}\left[X_{\{1,2\}}, H_2\right] \mathcal{P}\left[\varnothing H_2\right] \\ R &= \mathcal{P}\left[X_{\{3,4\}} \mid H_2\right]^T \end{aligned}$$

where $M = LR$.

Choose invertible $S = \mathcal{P}\left[X_3 \mid X_2\right]$ rewrite rewrite $M = LSS^{-1}R$ where

$$\begin{aligned} LS &= \mathcal{P}\left[X_{1,2}, X_3\right] \\ S^{-1}R &= \mathcal{P}\left[X_3, X_2\right]^{-1} \mathcal{P}\left[X_2, X_{\{3,4\}}\right]. \end{aligned}$$

The new factorization uses one less factor in each matrix than in the full joint $\mathcal{P}\left[X_{\{1,2\}}, X_{\{3,4\}}\right]$ and is only variables of the observed data.

We can also use this new factorization to decompose any latent tree. See 5

4 Training

To train our model we replace each probability matrix with its MLE. For the discrete case, the MLE matrices correspond to frequency counts.

Using the MLE estimate for each probability matrix is a consistent estimator of the joint. However, we have to invert one of the probability matrices so we lose some statistical efficiency. The EM algorithm never has to invert a probability matrix so it is more statistically efficient than spectral learning.

5 Generalizing to more variables

We can recursively apply the factorization from the left to the right to decrease the factors until they all have at most 3 factors.

For example:

$$\mathcal{P}\left[X_{\{1,2\}}, X_{\{3,4,5\}}\right] = \mathcal{P}\left[X_{\{1,2\}}, X_3\right] \mathcal{P}\left[X_2, X_3\right]^{-1} \mathcal{P}\left[X_2, X_{\{3,4,5\}}\right].$$

Rewrite $\mathcal{P}\left[X_2, X_{\{3,4,5\}}\right]$ as $\mathcal{P}\left[X_{\{2,3\}}, X_{\{4,5\}}\right]$ and decompose

$$\mathcal{P}\left[X_{\{2,3\}}, X_{\{4,5\}}\right] = \mathcal{P}\left[X_{\{2,3\}}, X_4\right] \mathcal{P}\left[X_3, X_4\right]^{-1} \mathcal{P}\left[X_3, X_{\{4,5\}}\right].$$

6 Fixing the inverse

In the factorization we assumed that $\mathcal{P}\left[X_2, X_3\right]$ was invertible but this is not generally the case.

For $\mathcal{P}\left[X_2, X_3\right]$ to be invertible, we need each matrix in the factorization

$$\mathcal{P}\left[X_2, X_3\right] = \mathcal{P}\left[X_2 \mid H_2\right] \mathcal{P}\left[\varnothing H_2\right] \mathcal{P}\left[X_3 \mid H_2\right]^T$$

to be full rank.

As before let m be the number of observables and k the number of hidden states.

6.1 $k \leq m$

When $k \leq m$, we can project $\mathcal{P}\left[X_2, X_3\right]$ to a lower dimensional space and invert it in that space. Let U and V be the top k left and right singular vectors, respectively. We can approximate $\mathcal{P}\left[X_2, X_3\right]^{-1}$ by $V\left(U^T \mathcal{P}\left[X_2, X_3\right] V\right)^{-1} U^T$.

6.2 $k > m$

This case the factors of $\mathcal{P}\left[X_2, X_3\right]$ are not full rank so are not invertible. We can't project to a lower dimensional space without sacrificing accuracy.

Intuitively, large k and small m indicates long term dependencies. Below is an example that expresses this intuition.

Let $S = X$ or $S = Y$ with probability $1/2$. We fix a number n. We print S followed by n As. We repeat this process forever.

As we make n bigger the dependencies in the model get bigger because each S gets farther away.

For $n > 2$, we have that the number of hidden states is bigger than the number of observable states. Thus in this case, a large number of hidden states implies that we have long term dependencies.

7 Spectral Learning

We can reduce the rank of $\mathcal{P}[X_2, X_3]$ so that it is at most m using singular value decomposition but we lose our information on long dependencies between variables.

Is there a way to some how keep these long term dependencies? The entries of $\mathcal{P}[X_2, X_3]$ are normalized counts. Given an assignment of variables, Let $\delta(X)$ be an indicator vector for the value of X. For example, lets say that there are only 2 observable states 1 and 2 so

$$\delta(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\delta(2) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Define the tensor product for vectors as $v \otimes w = vw^T$. Notice that

$$(\delta(X_2) \otimes \delta(X_3))_{ij} = (\delta(X_2))_i (\delta(X_3))_j.$$

For a given assignment, $\delta(X_2) \otimes \delta(X_3)$ tells us where to include counts in the joint of $\mathcal{P}[X_2, X_3]$ so

$$\mathcal{P}[X_2, X_3] = \mathrm{E}_{X_2, X_3}[\delta(X_2) \otimes \delta(X_3)].$$

We can interpret δ as indicator features. Therefore, we can replace these δs with a more complicated feature function ϕ.

We can rewrite the full joint as

$$\mathcal{P}\left[X_{\{1,2\}}, X_{\{3,4\}}\right] = \mathrm{E}\left[\delta(X_{\{1,2\}}) \otimes \phi(X_3)\right] V \left(U^T \mathrm{E}\left[\phi(X_2) \otimes \phi(X_3)\right] V\right]^{-1} U^T \mathrm{E}\left[\phi(X_2) \otimes \delta(X_{\{3,4\}})\right].$$

8 Connection to Hilbert Space Embeddings

We can use the kernel trick to deal with infinite dimensional feature functions! We deal with infinite dimensional vectors but we can apply the kernel trick to deal with finite matrices that depend on the size of the sample data. This is beyond the scope of the lecture.

9 Summary

Experimentally, the spectral method performs comparatively to the EM algorithm but is much faster.

Table 1: Pros and cons of each method

EM	Spectral
More statistically efficient	Less statistically efficient
Contains local optima	Local-optima free
Easier to extend to new models	Unknown whether its possible to extend to loopy models
Lacks theoretical guarantees	Provably consistent
Does not deal with negative numbers	Problems with negative numbers in MLE
Slow	Very fast
Difficult to include long dependencies	Handles long dependencies well
Generalizes poorly to continuous variables	Generalizes well to continuous domains