10-708: Probabilistic Graphical Models 10-708, Spring 2014

23 : Kernel Graphic Models

Lecturer: Eric P. Xing *Scribes: Xiang Li, Ran Chen*

1 Review of Hilbert Space Embeddings

1.1 Reproducing Kernel Hilbert Space

A Hilbert Space is a complete vector space equipped with an inner product. It is basically a nice infinite dimensional vector space where lots of things behave like the finite case.

A Reproducing Kernel Hilbert Space(RKHS) is basically a better infinite dimensional vector space where even more things behave like the finite case. RKHS is constructed with a Mercer Kernel, which is a function of two variables, such that:

$$\int\int K(x,y)f(x)f(y)dxdy > 0 \quad \forall f$$

Consider holding one element of the kernel fixed. We get a function of one variable which can be called Feature Function. And the collection of feature functions is called the Feature Map:

$$\phi_x := K(x,\cdot)$$

By this definition, the inner product can be defined as:

$$\langle \phi_x, \phi_y \rangle = \langle K(x,\cdot), K(y,\cdot) \rangle := K(x,y)$$

Now, we consider the set of functions that can be formed with linear combinations of these feature functions:

$$\mathcal{F}_0 = \left\{ f(z) : \sum_{j=1}^{k} \alpha_j \phi_{x_j}(z), \forall k \in \mathbb{N}_+ \text{ and } x_j \in \mathcal{X} \right\}$$

And we define the RKHS $\mathcal{F}$ to the completion of $\mathcal{F}_0$. In such space, we have a good property, which is reproducing property. The inner product of a function f with ϕ_x, evaluates a function at point x can be:

$$\begin{aligned}\langle f, \phi_x \rangle &= \left\langle \sum_j \alpha_j \phi_{x_j}, \phi_x \right\rangle \\ &= \sum_j \alpha_j \langle \phi_{x_j}, \phi_x \rangle \\ &= \sum_j \alpha_j K(x_k, x) \\ &= f(x)\end{aligned}$$

1.2 Embedding Distribution with One Variable

With such good property, we now can think about how to embed the distribution with one variable, which is called Mean Map:

$$\mu_X = \mathbb{E}_X[\phi_X]$$

The empirical estimate of such count defined as blow. However, this is not actually computable from data, since the feature map is infinite. And the x_n here is data point.

$$\hat{\mu}_X = \frac{1}{N} \sum_{n=1}^{N} \phi_{x_n}$$

By this definition, we can prove that:

$$\langle f, \mu_X \rangle = \langle f, \mathbb{E}_{X \sim D}[\phi_X] \rangle = \mathbb{E}_{X \sim D}[f(x)]$$

1.3 Embedding Joint Distribution

Now, we can think about how to embed joint distribution of two variables. In vector space, we have operator $\mathcal{C}$, which is used to map a function in one Hilbert Space to another function in the same or another Hilbert Space. Now we can define the uncentered cross-covariance operator $\mathcal{C}_{YX}$ implicitly such that:

$$\langle g, \mathcal{C}_{YX} f \rangle = \mathbb{E}_{YX}[f(X)g(Y)] \quad \forall f \in \mathcal{F}, \forall g \in \mathcal{G}$$

We should know that f is in one Hilbert Space, and g is in another. The $\mathcal{C}_{YX}$ here is our embedding of the joint distribution of X and Y. Intuitively, $\mathcal{C}_{YX}$ is a operator, just like $P[X, Y]$ is a matrix.

Let $\phi_X \in \mathcal{F}$ and $\psi_Y \in \mathcal{G}$, then the explicit form of cross-covariance operator is:

$$\mathcal{C}_{YX} = \mathbb{E}_{YX}[\psi_Y \otimes \phi_X]$$

It can be proofed:

$$\begin{aligned}\langle g, \mathcal{C}_{YX} f\rangle &= \langle g, \mathbb{E}_{YX}[\psi_Y \otimes \phi_X] f\rangle \\ &= \mathbb{E}_{YX}[\langle g, [\psi_Y \otimes \phi_X] f\rangle] \\ &= \mathbb{E}_{YX}[\langle g, \langle \phi_X, f\rangle \psi_Y\rangle] \\ &= \mathbb{E}_{YX}[\langle g, \psi_Y\rangle \langle \phi_X, f\rangle] \\ &= \mathbb{E}_{YX}[f(X)g(Y)]\end{aligned}$$

1.4 Embedding Conditional Distribution

By the previous definition, we can not define conditional embedding operator:

$$\mathcal{C}_{Y|X} = \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}$$

And it has such property that:

$$\mathbb{E}_{Y|x}[\phi_Y|x] = \mathcal{C}_{Y|X}\phi_x$$

2 Why We like Hilbert Space Embeddings

The reason we like Hilbert Space is that we can have the same way to deal with operations just like in probabilistic space. We can marginalize and use chain rule in Hilbert Space too.

2.1 Sum Rule and Chain Rule

We can have a simple comparison, Sum Rule:

$$\mathbb{P} = \int_Y \mathbb{P}[X, Y] = \int_Y \mathbb{P}[X|Y]\mathbb{P}[Y] \iff \mu_X = \mathcal{C}_{X|Y}\mu_Y$$

Chain Rule:

$$\mathbb{P}[X, Y] = \mathbb{P}[X|Y]\mathbb{P}[Y] = \mathbb{P}[Y|X]\mathbb{P}[Y] \iff \mathcal{C}_{YX} = \mathcal{C}_{Y|X}\mathcal{C}_{XX} = \mathcal{C}_{X|Y}\mathcal{C}_{YY}$$

We can see from this comparison that these two operations are almost in the same way in both space, which means that we can do the same thing in either way.

2.2 Mean Map and Probability Density Function

$$\mathcal{P}[X] = \mathbb{E}_X[\delta_X] \iff \mu_X = \mathbb{E}_X[\theta_X]$$

3 Kernel Graphic Models

As we can embed marginal distribution, joint distribution and conditional distribution in RKHS space, we can use these embeddings to replace the conditional probability tables we used before in graphic model to build the kernel graphic model. We can also perform inference on kernel graphic model with the sum rule and chain rule in RKHS.

3.1 Inference on Kernel Graphic Models

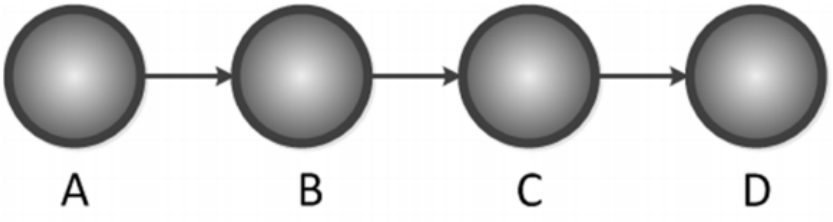

Figure 1: A simple graphic model

Consider a simple graphic model shown in Figure 1. If the variables in this model are discrete, we can parameterize the model using probability vector $\mathcal{P}[A]$ and conditional probability matrix $\mathcal{P}[B|A]$, $\mathcal{P}[C|B]$ and $\mathcal{P}[D|C]$. Then, we can do inference based on these matrix.

For example, if we want to compute $\mathbb{P}[D = d]$, we can firstly compute the marginal probability vector $\mathcal{P}[D]$ by

$$\mathcal{P}[D] = \mathcal{P}[A]\mathcal{P}[B|A]^T\mathcal{P}[C|B]^T\mathcal{P}[D|C]^T$$

Then, we can get $\mathbb{P}[D = d] = \delta_d^T\mathcal{P}[D]$, where δ_d is the evidence vector for d and all elements in δ_d is 0 except the element corresponding to d.

Similarly, we can also compute the joint distribution of two variables, such like $\mathbb{P}[A = a, D = d]$. To do this, we need to compute the joint distribution matrix $\mathcal{P}[A, D]$. However, if we directly use matrix multiplication, A would be integrated out so we can only get $\mathcal{P}[D]$. To solve this problem, we convert $\mathcal{P}[A]$ to a diagonal matrix $\mathcal{P}[\oslash A]$. Then, $\mathcal{P}[A, D]$ could ba calculated using

$$\mathcal{P}[A, D] = \mathcal{P}[\oslash A]\mathcal{P}[B|A]^T\mathcal{P}[C|B]^T\mathcal{P}[D|C]^T$$

And we can finnally get $\mathbb{P}[A = a, D = d] = \delta_a^T\mathcal{P}[A, D]\delta_d$.

If the variables in this model are continues, we can embed them in RKHS using operators $\mu_A/\mathcal{C}_{AA}$, $\mathcal{C}_{B|A}$, $\mathcal{C}_{C|B}$ and $\mathcal{C}_{D|C}$, which can be viewed as infinite dimensional vector or matrix. Then, $\mathbb{P}[D = d]$ can be calculated accoding to its mean map μ_D by

$$\mu_D = \mu_A\mathcal{C}_{B|A}^T\mathcal{C}_{C|B}^T\mathcal{C}_{D|C}^T$$
$$\mathbb{P}[D = d] \propto \phi_d^T\mu_D$$

Similarly, we can compute $\mathbb{P}[A = a, D = d]$ like we did before by computing the cross covariance operator

$\mathcal{C}_{AD}$ first. In specific, we can use the following equations:

$$\mathcal{C}_{AD} = \mathcal{C}_{AA}\mathcal{C}_{B|A}^T\mathcal{C}_{C|B}^T\mathcal{C}_{D|C}^T$$
$$\mathbb{P}[A = a, D = d] \propto \phi_a^T\mathcal{C}_{AD}\phi_d$$

These examples show that inference on kernel graphic model is similar to inference on regular graphic model. Therefore, we can apply the inference algorithm on regular graphic model, such like message passing algorithm to kernel graphic model by replacing the sum-product operations with tensor operations.

3.2 Relationship with Kernel Density Estimation

Now we have already shown that we can build kernel graphic model and perform inference on it. Here, we would like to discuss what we really get with kernel graphic model and how it help us in complicated problems.

Consider evaluating one random variable X at a particular evidence value $\bar{x}$. For a RKHS with RBF kernel, it could be evaluated by

$$< \mu_X, \phi_{\bar{x}} >= \mathbb{E}_X[< \phi_X, \phi_{\bar{X}} >] = \mathbb{E}_X[\exp(\frac{-||X - \bar{x}||_2^2}{\sigma^2})]$$

This is similar to the kernel density estimitor at point $\bar{x}$:

$$\mathbb{P}[X = \bar{x}] \propto \mathbb{E}[\exp(\frac{-||X - \bar{x}||_2^2}{\sigma^2})]$$

So evaluating the mean map at a point could be viewed as an unnormalized kernel density estimate. Yet, kernel density estimate doesn't work well in high dimensional space. For example, in an $\mathcal{O}$ dimensional space, kernel density estimation with RBF kernel is

$$\mathbb{P}[X_{1:\mathcal{O}} = \bar{x}_{1:\mathcal{O}}] \propto \mathbb{E}[\prod_{o=1}^{\mathcal{O}} \exp(\frac{-||X_o - \bar{x}_o||_2^2}{\sigma^2})]$$

This is similar to evaluating a huge covariance operator. Thus, it would suffer from curse of dimensionality and could get inaccurate result.

Fortunately, we could solve this problem using kernel graphic model by taking advantages of the conditional independencies embedded in the graph. In specific, we could factorize the huge covariance operator into several smaller covariance operators or conditional embedding operators according to the graph, so that we can estimate them more efficiently.

4 Learning with Kernel Trick

In this section, we will describe how to learn a kernel graphic model, and how to deal with the infinite dimension problem in the learning procedure. Note that learning for a fully observed graphic model on discrete variables is easy: we just need to count the number of different configurations and fill the conditional distribution table. However, conditional embedding operators could be viewed as infinite dimensional tables, so they cannot be learned directly. To solve this problem, we need to use the kernel trick. More details are shown below.

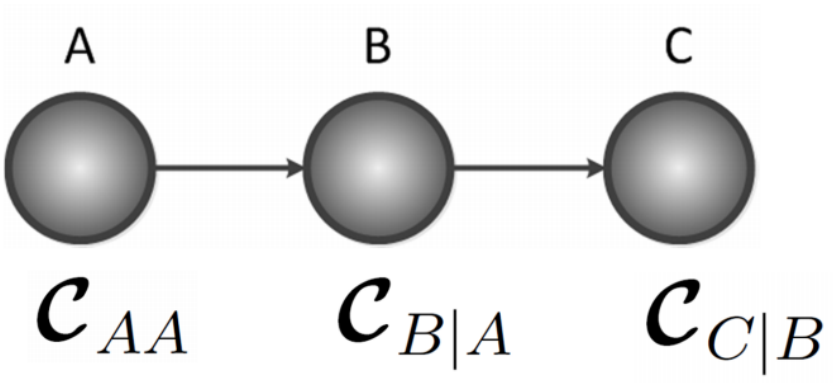

Figure 2: A simpler graphic model

4.1 Learning on Kernel Graphic Models

Consider a simple graphic model shown in Figure 2. To learn this model, we need to estimate operators $\mathcal{C}_{AA}$, $\mathcal{C}_{B|A}$ and $\mathcal{C}_{C|B}$.

To estimate an auto covariance operator like $\mathcal{C}_{XX}$, since $\mathcal{C}_{XX} = \mathbb{E}_X[\phi_X \otimes \phi_X]$, we can estimate it empirically using $\hat{\mathcal{C}}_{XX} = \frac{1}{N}\sum_{n=1}^{N} \phi_X \otimes \phi_X = \frac{1}{N}\Phi_X \otimes \Phi_X^T$, where $\Phi_X = [\phi_{x_1}, \phi_{x_2}, ..., \phi_{x_n}]$ is a matrix with N columns and infinite rows.

To estimate an conditional embedding operator like $\mathcal{C}_{Y|X}$, we can firstly estimate $\mathcal{C}_{XX}$ and $\mathcal{C}_{YX}$, then use the equation $\mathcal{C}_{Y|X} = \mathcal{C}_{YX}\,\mathcal{C}_{XX}^{-1}$. We have already shown how to estimate $\mathcal{C}_{XX}$. For $\mathcal{C}_{YX}$, since we kown that $\mathcal{C}_{YX} == \mathbb{E}_X[\phi_Y \otimes \phi_X]$, an empirical estimitor would be $\hat{\mathcal{C}}_{YX} = \frac{1}{N}\sum_{n=1}^{N} \phi_Y \otimes \phi_X = \frac{1}{N}\Phi_Y \otimes \Phi_X^T$. Then, we can get that $\hat{\mathcal{C}}_{Y|X} = \frac{1}{N}\Phi_X \otimes \Phi_X^T(\frac{1}{N}\Phi_Y \otimes \Phi_X^T + \lambda I)^{-1}$. Note that we add a regularizer term here to make the matrix invertible.

Finally, we can estimate $\mathcal{C}_{AA}$, $\mathcal{C}_{B|A}$, $\mathcal{C}_{C|B}$ as follows:

$$\begin{aligned}
\hat{\mathcal{C}}_{AA} &= \frac{1}{N}\Phi_A \otimes \Phi_A^T \\
\hat{\mathcal{C}}_{B|A} &= \Phi_B(\Phi_A^T\Phi_A + \lambda N I)^{-1}\Phi_A^T \\
\hat{\mathcal{C}}_{C|B} &= \Phi_C(\Phi_B^T\Phi_B + \lambda N I)^{-1}\Phi_B^T
\end{aligned}$$

Note that Φ_X is a matrix with infinite rows, and $\hat{\mathcal{C}}_{Y|X}$ is a matrix with infinite columns and rows. Therefore, these equations cannot be directly used for calculation, and we need to use kernel trick to solve this problem.

4.2 Why Kernel Trick Works

In the previous equations, we need to use the infinite dimensional matrix Φ_X in two ways: to calculate $\Phi_X^T\Phi_X$ and to calculate $\Phi_X\Phi_X^T$. It can be shown that $\Phi_X^T\Phi_X = K_{XX}$, where

$$k_{XX} = \begin{pmatrix} < \phi_{x_1}, \phi_{x_1} > & ... & < \phi_{x_1}, \phi_{x_N} > \\ ... & ... & ... \\ < \phi_{x_N}, \phi_{x_1} > & ... & < \phi_{x_N}, \phi_{x_N} > \end{pmatrix}$$

So that $\Phi_X^T\Phi_X$ is a finite matrix. For $\Phi_X\Phi_X^T$, it is an infinite dimensional matrix. However, since Φ_X only have finite columns, the rank of $\Phi_X\Phi_X^T$ is finite, so that we can deal with it using kernel in the inference step, and do not need to write it out explicitly.

4.3 Using Kernel Trick

Kernel trick tell us that although $\mathcal{C}_{AA}$, $\mathcal{C}_{B|A}$, $\mathcal{C}_{C|B}$ can not be explicitly written out, we can calculate their inner product in the inference step. For example, if we want to calculate $\hat{\mathcal{C}}_{AC}$, we can use

$$\begin{aligned} \hat{\mathcal{C}}_{AC} &= \hat{\mathcal{C}}_{AA}\hat{\mathcal{C}}_{B|A}^T\hat{\mathcal{C}}_{C|B}^T \\ &= \frac{1}{N}\Phi_A\Phi_A^T\Phi_A(\Phi_A^T\Phi_A + \lambda NI)^{-1}\phi_B^T\phi_B(\phi_B^T\phi_B + \lambda NI)^{-1}\Phi_C^T \\ &= \frac{1}{N}\Phi_A K_{AA}(K_{AA} + \lambda NI)^{-1}K_{BB}(K_{BB} + \lambda NI)^{-1}\Phi_C^T \end{aligned}$$

Then, to evaluate its value for $A = a$ and $C = c$, we can use

$$\begin{aligned} \phi_a^T\hat{\mathcal{C}}_{AC}\phi_c &= \frac{1}{N}\phi_a\Phi_A K_{AA}(K_{AA} + \lambda NI)^{-1}K_{BB}(K_{BB} + \lambda NI)^{-1}\Phi_C^T\phi_c \\ &= \frac{1}{N}K_{AA}(1:N,a)K_{AA}(K_{AA} + \lambda NI)^{-1}K_{BB}(K_{BB} + \lambda NI)^{-1}K_{CC}(1:N,c) \end{aligned}$$

where

$$K_{XX}(1:N,x) = \begin{pmatrix} K(x_1,x) \\ ... \\ K(x_N,x) \end{pmatrix}$$

It can be seen that all Φ and $\mathcal{C}$ disappear in this final equation. So that $\phi_a^T\hat{\mathcal{C}}_{AC}\phi_c$ can be evaluated from the data.

References

[1] Alex Smola et al. "A Hilbert space embedding for distributions". In: *Algorithmic Learning Theory*. Springer. 2007, pp. 13–31.

[2] Le Song, Arthur Gretton, and Carlos Guestrin. "Nonparametric tree graphical models". In: *International Conference on Artificial Intelligence and Statistics*. 2010, pp. 765–772.

[3] Le Song et al. “Hilbert space embeddings of conditional distributions with applications to dynamical systems”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 961–968.

[4] Le Song et al. “Kernel belief propagation”. In: *arXiv preprint arXiv:1105.5592* (2011).

[5] Le Song et al. “Learning via Hilbert space embedding of distributions”. In: (2007).