Inference and Representation

Rachel Hodos

New York University

Lab 10, November 18, 2015

Outline

Definition of exponential family

A distribution is in the exponential family if it can be written in the following form:

$$p(\mathbf{x};\eta) = h(\mathbf{x})\exp\{\eta \cdot \mathbf{f}(\mathbf{x}) - \ln Z(\eta)\}$$

Why talk about the exponential family?

- Most distributions you know are in the exponential family
- Maximum entropy solutions (via moment matching)
- Writing in this form can reveal new algorithms
- All distributions in the exponential family have *conjugate* distributions
- Parametrizing in log-linear form can make learning the parameters easier

Examples

(on chalkboard)

MLE for MRFs? Bad news...

- The global normalization constant $Z(\theta)$ kills decomposability:

$$\begin{aligned}
\theta^{ML} &= \arg\max_{\theta} \; \log \prod_{\mathbf{x} \in \mathcal{D}} p(\mathbf{x}; \theta) \\
&= \arg\max_{\theta} \sum_{\mathbf{x} \in \mathcal{D}} \left(\sum_{c} \log \phi_c(\mathbf{x}_c; \theta) - \log Z(\theta) \right) \\
&= \arg\max_{\theta} \left(\sum_{\mathbf{x} \in \mathcal{D}} \sum_{c} \log \phi_c(\mathbf{x}_c; \theta) \right) - |\mathcal{D}| \log Z(\theta)
\end{aligned}$$

- The log-partition function prevents us from decomposing the objective into a sum over terms for each potential
- Solving for the parameters becomes much more complicated

...but wait, there's hope!

$$
\begin{aligned}
\theta^{ML} &= \arg\max_{\theta} \left(\sum_{\mathbf{x} \in \mathcal{D}} \sum_{c} \log \phi_c(\mathbf{x}_c; \theta) \right) - |\mathcal{D}| \log Z(\theta) \\
&= \arg\max_{\mathbf{w}} \left(\sum_{\mathbf{x} \in \mathcal{D}} \sum_{c} \mathbf{w} \cdot \mathbf{f}_c(\mathbf{x}_c) \right) - |\mathcal{D}| \log Z(\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \mathbf{w} \cdot \left(\sum_{\mathbf{x} \in \mathcal{D}} \sum_{c} \mathbf{f}_c(\mathbf{x}_c) \right) - |\mathcal{D}| \log Z(\mathbf{w})
\end{aligned}
$$

- The first term is linear in w
- The second term is also a function of w, and we can compute derivatives in the following way..

Derivative of log-partition function

The derivative of the log-partition function is equal to the expectation of the sufficient statistic vector (i.e. the distribution's marginals):

$$
\begin{aligned}
\partial_{\eta_i} \ln Z(\eta) &= \partial_{\eta_i} \ln \sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\} \\
&= \frac{1}{\sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\}} \partial_{\eta_i} \sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\} \\
&= \frac{1}{\sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\}} \sum_{\mathbf{x}} \partial_{\eta_i} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\} \\
&= \frac{1}{\sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\}} \sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\} \partial_{\eta_i} \eta \cdot \mathbf{f}(\mathbf{x}) \\
&= \frac{1}{\sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\}} \sum_{\mathbf{x}} \exp\{\eta \cdot \mathbf{f}(\mathbf{x})\} f_i(\mathbf{x}) \\
&= \sum_{\mathbf{x}} \frac{\exp\{\eta \cdot \mathbf{f}(\mathbf{x})\}}{\sum_{\hat{\mathbf{x}}} \exp\{\eta \cdot \mathbf{f}(\hat{\mathbf{x}})\}} f_i(\mathbf{x}) = \sum_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) = E_p[f_i(\mathbf{x})].
\end{aligned}
$$

Log partition function is convex!

- Similarly, 2nd derivatives are the 2nd-order moments (i.e. the covariance matrix).
- This is positive semi-definitive, which means that the log-partition function is convex.
- This means we can use any convex optimization algorithm!

Notes on moment matching

$$p(\hat{x}_i, \hat{x}_j; \mathbf{w}^{ML}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} 1[x_i = \hat{x}_i, x_j = \hat{x}_j]$$

- Yesterday we saw that the ML solution had the same moments as our data
- **This does not mean we can (always) estimate the ML** *parameters* **directly from the data**
- Tree-structured MRFs are a special case where we *can* esimate the parameters from the moments (you will show this in your HW)

What is a Gaussian Process?

- A distribution over functions
- Allows us to do Bayesian estimation of functions
- A generalization of multivariate Gaussians to infinite dimensional space
- Provides explicit representation of uncertainty as a function of input x

Definition of a Gaussian Process

The basic setup:

- Data set $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Inputs $\mathbf{x}_i \in \mathbb{S} \subset \mathbb{R}^D$.
- Outputs $y_i \in \mathbb{R}$.

$$
\begin{aligned}
x_i &\sim p(x) \\
y_i &= f(x_i) + \epsilon_i \\
\epsilon_i &\overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)
\end{aligned}
$$

Definition

f is a Gaussian process if for any collection $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{S}, i = 1, \ldots, n\}$,

$$
\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{pmatrix} \sim \mathcal{N}(\mu(\mathbf{X}), K(\mathbf{X}, \mathbf{X}))
$$

Regression using GP, noise-free

Interpolation/prediction at target locations:

- *(Noise-free observations)* Observe $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, n\}$.
- *(Noisy observations)* Observe $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Want to predict $\mathbf{f}^* = \{f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_k^*)\}$ at $\mathbf{x}^*$.

$$\left.\begin{aligned}
&\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \Big| \mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}^*) \\ K(\mathbf{X}^*,\mathbf{X}) & K(\mathbf{X}^*,\mathbf{X}^*) \end{pmatrix}\right) \\
&\mathbf{f}^* | \mathbf{f}, \mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\Big(K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X})]^{-1}\mathbf{f}, \\
&\qquad K(\mathbf{X}^*,\mathbf{X}^*) - K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X})]^{-1}K(\mathbf{X},\mathbf{X}^*)\Big)
\end{aligned}\right\} \boxed{\text{Prediction with noise-free data}}$$

Regression using GP, general

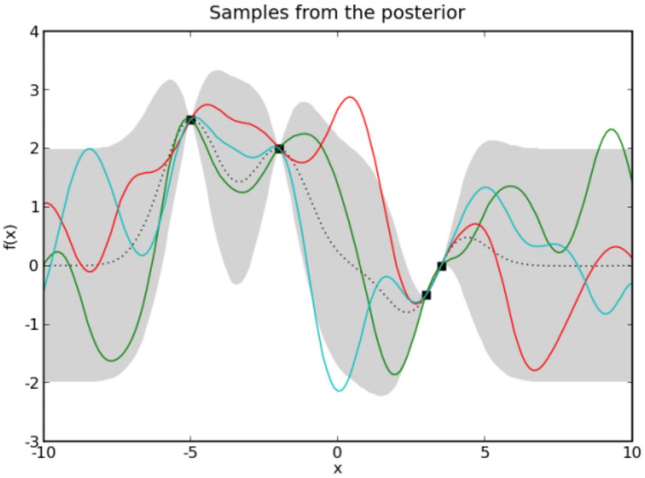
Interpolation/prediction at target locations:

- *(Noise-free observations)* Observe $\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \ldots, n\}$.
- *(Noisy observations)* Observe $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$.
- Want to predict $\mathbf{f}^* = \{f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_k^*)\}$ at $\mathbf{x}^*$.

$$\left.\begin{aligned}
&\begin{pmatrix}\mathbf{f}\\ \mathbf{f}^*\end{pmatrix} |\mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}^*)\\ K(\mathbf{X}^*,\mathbf{X}) & K(\mathbf{X}^*,\mathbf{X}^*)\end{pmatrix}\right)\\
&\mathbf{f}^*|\mathbf{f}, \mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\Big(K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X})]^{-1}\mathbf{f},\\
&\qquad K(\mathbf{X}^*,\mathbf{X}^*) - K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X})]^{-1}K(\mathbf{X},\mathbf{X}^*)\Big)
\end{aligned}\right\} \boxed{\text{Prediction with noise-free data}}$$

$$\left.\begin{aligned}
&\begin{pmatrix}\mathbf{y}\\ \mathbf{f}^*\end{pmatrix} |\mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}K(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}_n & K(\mathbf{X},\mathbf{X}^*)\\ K(\mathbf{X}^*,\mathbf{X}) & K(\mathbf{X}^*,\mathbf{X}^*)\end{pmatrix}\right)\\
&\mathbf{f}^*|\mathbf{y}, \mathbf{X}, \mathbf{X}^* \sim \mathcal{N}\Big(K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}_n]^{-1}\mathbf{y},\\
&\qquad K(\mathbf{X}^*,\mathbf{X}^*) - K(\mathbf{X}^*,\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}_n]^{-1}K(\mathbf{X},\mathbf{X}^*)\Big)
\end{aligned}\right\} \boxed{\text{Prediction with noisy data}}$$

Posterior over functions



Latent GPs

- Can generalize to case where y no longer just a noisy observation of $f(x)$:

$$y_i \sim p(y|f(x_i))$$

$$y_i \overset{ind}{\sim} \text{Bern}\left(\frac{1}{1+\exp(-f(x_i))}\right)$$