

Exponential Families, Kernels & Graphical Models

Vishy

16th May 2007

1 Exponential Family

Exponential families are given by the following probability distribution:

$$P(\mathcal{X}; \theta) = P_0(x) \exp(\langle \phi(x), \theta \rangle - g(\theta)) \quad (1)$$

where...

- \mathcal{X} = space of observations
- $P_0(x)$ = base function/measure - provides a way of measuring/weighting space (constants)
- $\phi(x)$ = sufficient statistics/features - the transformation
- $g(\theta)$ = log partition function - a normalising factor to ensure the distribution sums to one

Please note that the $\langle \dots \rangle$ term represent the inner product: $\langle x, y \rangle_{\mathbb{R}^n} = \sum_{i=1}^n x_i y_i$, where $x, y \in \mathbb{R}^n$. Some nice properties of exponential families are that if we:

- choose sufficient statistics ($\phi(x)$) \Rightarrow choose the family ($\Leftrightarrow g(\theta)$)
- choose theta (θ) \Rightarrow choose the members of family (instance of family)

For statistical machine learning we are interested in both: family (for estimation) and which member (most of what we're going to do). Also note that for any distribution you can think of, it is (or can be written as) a member of the exponential family.

1.1 Example: binomial distribution

This is an example of a coin toss where the probability of getting a heads (P) is given by: $1 \geq P \geq 0$. The probability distribution is $P(x) = p^x(1-p)^{1-x}$, where $x \in \{0, 1\} = \mathcal{X}$. You can prove this is an exponential family by starting with $P(x)$ and applying P^{-1} : $PP^{-1} = \mathbb{I}$. Then it follows that:

$$\begin{aligned} P(x) &= \exp \log P(x) \\ &= \exp \log p^x (1-p)^{1-x} \\ &= \exp(x \log(p) + (1-x) \log(1-p)) \\ &= \exp \left(\left(\begin{matrix} x \\ 1-x \end{matrix} \right), \left(\begin{matrix} \log p \\ \log(1-p) \end{matrix} \right) \right) \\ &= \exp(\langle \phi(x), \theta \rangle) \end{aligned}$$

From the working out above the log partition (normalising) function seems to be missing (it is not 0). The next section explains this function's purpose, and using this information the missing log partition function is worked out to be $-\log \theta$. So the final probability function for a binomial distribution is:

$$P(x) = \exp(\langle -x, \theta \rangle - (-\log \theta)) \quad (2)$$

1.2 Example: gaussian

A Gaussian function is given by:

$$P(x) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(\frac{-(x - \mu)^2}{2\theta^2}\right) \quad (3)$$

Expanding this...

$$\begin{aligned} & \exp\left(\frac{-x^2}{2\theta^2} - \frac{\mu}{2\theta^2} + \frac{2x\mu}{2\theta^2} - \frac{1}{2} \log(2\pi\theta^2)\right) \\ & \exp\left(\left\langle \begin{pmatrix} x \\ x^2 \end{pmatrix}, \begin{pmatrix} \frac{-1}{2\theta^2} \\ \frac{\mu}{\theta^2} \end{pmatrix} \right\rangle - \left(\frac{\mu^2}{2\theta^2} + \frac{1}{2} \log(2\pi\theta^2)\right)\right) \end{aligned}$$

Relating this back to the exponential function formula:

$$\begin{aligned} & \underbrace{\left\langle \begin{pmatrix} x \\ x^2 \end{pmatrix} \right\rangle}_{\text{family}}, \underbrace{\begin{pmatrix} \frac{-1}{2\theta^2} \\ \frac{\mu}{\theta^2} \end{pmatrix}}_{\text{member}} \rangle = \langle -\phi(x), \theta \rangle \\ & \left(\frac{\mu^2}{2\theta^2} + \frac{1}{2} \log(2\pi\theta^2)\right) = g(\theta_1, \theta_2), \text{ which can be "massaged" into } \mu\theta^2 \text{ terms} \end{aligned}$$

If we know what the μ and θ^2 values are, we are able to work out what gaussian distribution it is!

2 Log Partition Function

The log partition function is a normalising function so that the probability will always be one:

$$\begin{aligned} 1 &= \int P_0(x) \exp(\langle \phi(x), \theta \rangle - g(\theta)) dx \\ 1 &= \int P_0(x) \frac{\exp(\langle \phi(x), \theta \rangle)}{\exp(g(\theta))} dx \\ 1 &= \frac{1}{\exp(g(\theta))} \int P_0(x) \exp(\langle \phi(x), \theta \rangle) dx \\ \exp(g(\theta)) &= \int P_0(x) \exp(\langle \phi(x), \theta \rangle) dx \\ g(\theta) &= \log \int_x P_0(x) \exp(\langle \phi(x), \theta \rangle) dx \end{aligned} \quad (4)$$

This is the log partition function. The \int_x term is *expensive* if x is large (because we are integrating exponential which produces a very small number). If we take the gradient (first derivative):

$$\begin{aligned}\partial_\theta g(\theta) &= \int_x P_0(x)\phi(x) \exp(\langle \phi(x), \theta \rangle - g(\theta)) dx \\ &= \mathbb{E}_{P(x;\theta)}[\phi(x)]\end{aligned}$$

If we take the second derivative:

$$\begin{aligned}\partial_\theta^2 g(\theta) &= \int_x P_0(x)\phi(x)(\phi(x) - \partial_\theta(\theta))^T \exp(\langle \phi(x), \theta \rangle - g(\theta)) dx \\ &= \mathbb{E}_{P(x;\theta)}[\phi(x)\phi(x)]^T - \mathbb{E}_{P(x;\theta)}[\phi(x)]\mathbb{E}_{P(x;\theta)}[\phi(x)]^T \\ &= \text{var}_{P(x;\theta)}[\phi(x)]\end{aligned}$$

Following this, taking higher order derivatives produces higher order moments. Using this information we're able to determine that $g(\theta)$ is a convex function! This is a nice function as it has a useful property that if we minimise the convex function the local minimum is the global minimum (there is only one minimum present). There are efficient methods to find the local minimum, which means we find the global minimum of the function.

3 Unconditional Models

Now we will be estimating the density on a set of data points drawn with the same distribution but are independent of each other (iid - independent and identically distributed random variables).

$$P(x; \theta) = \prod_{i=1}^n P(x_i; \theta) \tag{5}$$

Now we do maximum likelihood estimation (MLE). We want to find $\max_\theta P(x; \theta)$, which we can do by finding $\min_\theta -\log P(x; \theta)$:

$$\begin{aligned}\min_\theta -\log P(x; \theta) &= \min_\theta -\sum_{i=1}^n \log P(x_i; \theta) \\ &= \min_\theta \underbrace{ng(\theta)}_{\text{convex}} - \sum_{i=1}^n \underbrace{\langle \phi(x_i), \theta \rangle}_{\text{linear}}\end{aligned}$$

Here, we have a convex – a linear function, which does not invalidate convexity: the function still has a unique global minimum. Lets take the derivatives and set them to zero:

$$\begin{aligned}0 &= \partial_\theta ng(\theta) - \sum_{i=1}^n \langle \phi(x_i), \theta \rangle \\ 0 &= n\partial_\theta g(\theta) - \sum_{i=1}^n \phi(x_i) \\ n\partial_\theta g(\theta) &= \frac{1}{n} \sum_{i=1}^n \phi(x_i) \\ \mathbb{E}_\theta[\phi(x)] &= \hat{u}\end{aligned}$$

A pleasing property is that at the optimum expectation under the exponential family distribution matches the empirical mean.

4 Exponential Family & Maximum Entropy

To find this empirical mean ($E_\theta[\phi(x)] = \hat{u}$) we can (if the parameters are known) solve it analytically or use Newton's method, but if we don't have enough data then the solutions are bad. The MLE and maximum entropy are duals of each other, with:

$$\min - \int_x P(x) \log P(x) dx$$

which solves to $E[x] = b$ and also yields an exponential family too. In general it is not a good idea to use MLE as the example following demonstrates:

4.1 Example: laplacian distribution

$$P(x; \theta) = \exp(\langle -x, \theta \rangle - (-\log \theta))$$

We want to satisfy $\partial_\theta g(\theta) = \frac{1}{n} \sum_{i=1}^n \phi(x_i)$. If the features are $-x$:

$$\begin{aligned} \frac{-1}{\theta} &= \frac{1}{n} \sum_{i=1}^n \phi(-x_i) \\ \frac{1}{\theta} &= \hat{x} \\ \theta &= \frac{1}{\hat{x}} \end{aligned}$$

The problem of MLE is that it is data dependant and uses no prior knowledge. What if we sample for a gaussian distribution and all data by chance is sampled from the tail end of the function? Your estimate would then be wrong! With MLE, one bad data point (which could simply be by chance) can detrimentally bias the output. Because of this dependency on data MLE is also subject to over fitting. We want an estimator which can generalise, which can learn! The maximum a posteriori estimate (MAE) is more suitable which uses our prior knowledge if we have it. Suppose:

$$\begin{aligned} P(\theta|x) &= \frac{P(x|\theta)P(\theta)}{P(x)} \\ &\propto P(x|\theta)P(\theta) \end{aligned}$$

If we know some of $P(\theta)$ then we will have a better $P(\theta|x)$. How do we choose a priori? In the absence of any known data we can chose a simple $P(\theta) \sim N(0, \theta^2 \mathbb{I})$ which has a normal distribution and variance along the diagonals. Instead of using the whole distribution we can simply go for the

maximum of the distribution:

$$\begin{aligned} \arg \max_{\theta} P(x|\theta)P(\theta) &= \arg \min_{\theta} -\log P(x|\theta) - \log P(\theta) \\ &= \arg \min_{\theta} \left\{ ng(\theta) - \sum_{i=1}^n \langle \phi(x_i), \theta \rangle + \frac{1}{2\sigma^2} \|\theta\|^2 \right\} \\ \partial_{\theta} g(\theta) - \hat{u} + \frac{1}{n\sigma^2} \theta &= 0 \\ \mathbb{E}_p[\phi(x)] &= \hat{u} - \frac{1}{n\sigma^2} \theta, \text{ with } \frac{1}{n\sigma^2} \theta \rightarrow 0 \text{ (for } n \rightarrow \infty) \end{aligned}$$

So if we have a small number of data points we use prior knowledge. If we have a large number of data points we believe less and less the prior knowledge.

5 Conditional Models

So far we have only used exponential families for density estimation etc. . . With conditional models we now will use them for prediction!

$$P(y|x; \theta) = \exp(\langle \theta(x, y), \theta \rangle - g(\theta|x)) \quad (6)$$

Note that we have now dropped the P_0 base measure and that the function is now in terms of x and y , with the log partition function being: $g(\theta|x) = \log \int_x \exp(\langle \phi(x, y), \theta \rangle) dy$. Now:

$$\begin{aligned} P(Y|x; \theta) &= \prod_i P(y_i|x_i, \theta) \\ \Rightarrow \min_{\theta} -\log P(y|x; \theta) &= \min_{\theta} -\sum_i \log P(y_i|x_i; \theta) \\ &= \min_{\theta} \sum_i g(\theta|x_i) - \langle \phi(x_i, y_i), \theta \rangle \end{aligned}$$

Taking the gradients and setting them to zero:

$$\sum_i \mathbb{E}_{\theta}[\phi(x_i, y_i)] = \sum_i \phi(x_i, y_i)$$

6 Priors & Gaussian Processes

Usually something is known about the distribution of θ . In that case invoke Bayes rule and place a prior as follows:

$$P(\theta|x; y) = \frac{P(y|x; \theta)P(\theta)}{P(y|x)}$$

Now you can encode prior belief in θ via $P(\theta)$. In the absence of any prior knowledge it is typical to assume $P(\theta) \sim N(0, \sigma^2)$. We now show that this prior leads to a GP. What is a GP? Firstly let $t(x)$ be a stochastic process $t: \mathcal{X} \rightarrow \mathbb{R}$. $t(x)$ is a GP $\forall m$ and $\{x_1 \dots x_m\} \subset \mathcal{X}$, and the variables $\{t(x_1) \dots t(x_m)\}$ are normally distributed. Let:

$$\begin{aligned} u(x) &= \mathbb{E}[t(x)] \leftarrow \text{typically zero} \\ k(x, x') &= \text{cov}(t(x), t(x')) \leftarrow \text{typically assumed to be known} \end{aligned}$$

A GP prior is one where we directly place a prior on the space of δ^n . In particular we assume it is a gaussian process. We claim that:

$$\begin{aligned} t(x, y) &= -\log P(y|x; \theta) + g(\theta|x) \text{ is a GP } P(\theta) \sim N(0, \sigma^2) \\ t(x, y) &= \langle \phi(x, y), \theta \rangle \\ \therefore E[t(x, y)] &= \langle \phi(x, y), E[\theta] \rangle \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{cov}(t(x, y), t(x', y')) &= \langle \phi(x, y), \phi(x', y') \rangle_H E[\theta\theta^T] \\ &= \sigma^2 \langle \phi(x, y), \phi(x', y') \rangle_H \\ &= k((x, y), (x', y')) \end{aligned}$$

In other words placing a normal prior on θ yields a GP prior over $t(x, y) = \langle \phi(x, y), \theta \rangle$. Now we maximise the posterior distribution:

$$P(\theta|x; y) \propto P(y|x; \theta)P(\theta)$$

and predict with $t(x, y)$:

$$\text{ie } \min_{\theta} -\log P(y|x; \theta) - \log P(\theta) = \min_{\theta} \frac{1}{2} \|\theta\|^2 + \sum_i g(\theta|x_i) - \langle \phi(x_i, y_i), \theta \rangle$$

Now we take the derivatives and set them to zero:

$$\begin{aligned} \theta &= \sum_i \phi(x_i, y_i) - \partial_{\theta} g(\theta|x_i) \\ &= \sum_i \phi(x_i, y_i) - E_{\theta}[\phi(x_i, y_i)] \\ &= \sum_i \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y) \leftarrow \text{representer theorem!} \end{aligned}$$

$$\begin{aligned} (\text{assuming } \mathcal{Y} \text{ is discrete}) \alpha_{iy} &= P(y|x_i; \theta) \quad y \neq y_i \\ &= 1 - P(y|x_i; \theta) \quad y = y_i \end{aligned}$$

7 Novelty Detection

Estimate $P(x|\theta)$ using samples $\{x_1 \dots x_n\}$:

$$\begin{aligned} yP(x|\theta) < P_0 &\Rightarrow x \text{ is novel} \\ yP(x|\theta) > P_0 &\Rightarrow x \text{ is uninteresting for us} \end{aligned}$$

ie we are interested in modeling:

$$\min \left(\frac{P(x_i|\theta)}{P_0}, 1 \right)$$

If we use $P_0 = \exp(\rho - g(\theta))$ then:

$$\min \left(\frac{\exp(\langle \phi(x), \theta \rangle - g(\theta))}{\exp(\rho - g(\theta))}, 1 \right) = \min(\exp(\langle \phi(x), \theta \rangle - \rho), 1)$$

Now we place a prior and a min:

$$\min_{\theta} \frac{1}{2} \|\theta\|^2 - \max(\rho - \langle \phi(x_i), \theta \rangle, 0) \leftarrow \text{one class svm!}$$

8 Log Odds Ratio

$$R(x, y; \theta) = \log \frac{P(y|x; \theta)}{\max_{y' \neq y} P(y'|x; \theta)} \quad (7)$$

The $P(y|x; \theta)$ is a correct class label, while the $\max_{y' \neq y} P(y'|x; \theta)$ is an incorrect class label of the next best class. We want to not only place max probability on the correct class label, but also want to ensure that all other class labels get small probability. In the case of the exponential family:

$$\begin{aligned} \log P(y|x; \theta) - \max_{y' \neq y} \log P(y'|x; \theta) &\geq 1 \\ \min_{y' \neq y} \langle \phi(x, y), \theta \rangle - \langle \phi(x, y'), \theta \rangle &\geq 1 \\ \min_{y' \neq y} \langle \phi(x, y) - \phi(x, y'), \theta \rangle &\geq 1 \end{aligned}$$

If we require $R(x, y; \theta) \geq 1$ then our loss is $\max(1 - R(x, y; \theta), 0)$ or by placing priors we have $\min \frac{1}{2} \|\theta\|^2$ which solves to $R(x, y; \theta) \geq 1$.

9 Kernel Functions

Many different ways of viewing them, so a simplified version offers:

$$\begin{aligned} K(x, x') &: \mathcal{X} \cdot \mathcal{X} \rightarrow \mathbb{R} \\ K(x, x') &= \langle \phi(x), \phi(x') \rangle \end{aligned}$$

Instead of having to map into a high dimensional space and perform a dot product (bad), we can simply put x, x' into K which will spit out an answer for us (good). Lets look at an optimisation problem.

$$\begin{aligned} \mathcal{L}(\theta, \alpha) &= \frac{1}{2} \|\theta\|^2 - \sum_i \alpha_i (y_i \langle \theta, \phi(x_i) \rangle - 1) \\ \partial_{\theta} \mathcal{L}(\theta, \alpha) &= \theta - \sum_i \alpha_i y_i \phi(x_i) = 0 \\ \theta &= \sum_i \alpha_i y_i \phi(x_i) \leftarrow \text{special form of the representer theorem} \\ \partial_{\theta} &= y_i \langle \theta, \phi(x_i) \rangle = 1 \end{aligned}$$

Now lets take $\theta = \sum \alpha_i y_i \phi(x_i)$ and put it back into $\mathcal{L}(\theta, \alpha) = \frac{1}{2} \|\theta\|^2 - \sum_i \alpha_i (y_i < \theta, \phi(x_i) > -1)$:

$$\sum \alpha_i - \sum_{i,j} \alpha_i \alpha_j < \phi(x_i), \phi(x_j) > y_i y_j$$

which solves to $\alpha_i \geq 0 \forall i$. We have always been assuming our data is perfect; if it is corrupted however we need to add slack:

10 Graph Structured Losses

To introduce slack write $R(x, y, \theta) \geq 1 - \epsilon$. If more than two labels we have $\Delta(y, y')$ - a graph structured loss on the output. Two ways of rescaling this:

$$R(x, y, \theta) \geq 1 - \frac{\epsilon}{\Delta(y, y')} \forall y' \leftarrow \text{tsochantaridis}$$

$$R(x, y, \theta) \geq \Delta(y, y') - \epsilon \forall y' \leftarrow \text{taskar}$$

in short: $R(x, y, \theta) \geq \Delta(y, y')^B \left(1 - \frac{\epsilon}{\Delta(y, y')}\right)$, where $B = 0$: tsochantaridis, $B = 1$: taskar

Also adding this slack creates a soft margin svm:

$$\begin{aligned} \min \frac{1}{2} \|\theta\|^2 + c \sum_i \epsilon_i \\ \text{with respect to} \\ < \phi(x_i, y) - \phi(x_i, y'), \theta > \geq 1 - \epsilon_i \\ \epsilon_i \geq 0 \end{aligned}$$