Exponential Families in Feature Space

Alexander J. Smola
Alex.Smola@anu.edu.au

National ICT Australia
Machine Learning Program
RSISE, The Australian National University

Thanks to Yasemin Altun, Stephane Canu, Thomas Hofmann, and Vishy Vishwanathan
Outline

Exponential Families
- Definition, Examples, Priors
- Inference

Conditionally Multinomial Models
- Gaussian Process Classification
- Multiclass models

Conditionally Normal Models
- Gaussian Process regression
- Heteroscedastic noise

Structured Estimation
- Conditional Random Fields
- Clifford Hammersley decompositions
The Exponential Family

Definition
A family of probability distributions which satisfy

\[ p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta)) \]

Details
- \( \phi(x) \) is called the **sufficient statistic** of \( x \).
- \( g(\theta) \) is the **log-partition function** and it ensures that the distribution integrates out to 1.
Example: Normal Distribution

Engineer’s favorite

\[ p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2\sigma^2} (x - \mu)^2 \right) \quad \text{where} \quad x \in \mathbb{R} =: \mathcal{X} \]

Massaging the math

\[ p(x) = \exp \left( \langle (x, x^2), \theta \rangle - \frac{\mu^2}{2\sigma^2} + \frac{1}{2} \log(2\pi\sigma^2) \right) \]

Using the substitution \( \theta_2 := -\frac{1}{2} \sigma^{-2} \) and \( \theta_1 := \mu \sigma^{-2} \) yields

\[ g(\theta) = -\frac{1}{4} \theta_1 \theta_2^{-1} + \frac{1}{2} \log 2\pi - \frac{1}{2} \log -2\theta_2 \]
Many discrete events
Assume that we $n$ events, each which all may occur with a certain probability $\pi_x$.

Guessing the answer
Use the map $\phi: x \rightarrow e_x$, that is, $e_x$ is an element of the canonical basis $(0, \ldots, 0, 1, 0, \ldots)$ as sufficient statistic.

$$\Rightarrow p(x) = \exp(\langle e_x, \theta \rangle - g(\theta))$$

where the normalization is

$$g(\theta) = \log \sum_{i=1}^{n} \exp(\theta_i)$$
Generating Cumulants

$g(\theta)$ is the normalization for $p(x|\theta)$ Taking the derivative wrt. $\theta$ we can see that

$$\partial_{\theta} g(\theta) = E_{x \sim p(x|\theta)} [\phi(x)]$$
$$\partial^{2}_{\theta} g(\theta) = \text{Cov}_{x \sim p(x|\theta)} [\phi(x)]$$

Good News

$g(\theta)$ is a convex function

Very Good News

$$- \log p(X|\theta) = \sum_{i=1}^{m} -\langle \phi(x_i), \theta \rangle + mg(\theta)$$

is convex. So Maximum Likelihood Estimation is a convex minimization problem.
Tossing a dice

![Graph 1](image1)

![Graph 2](image2)

![Graph 3](image3)

![Graph 4](image4)
Problems with Maximum Likelihood
With not enough data, parameter estimates will be bad.

Prior to the rescue
Often we know where the solution should be.

Normal Prior
Simply assume $\theta \sim \mathcal{N}(0, \sigma^2 1)$.

Posterior

$$- \log p(\theta|X) = \sum_{i=1}^{m} -\langle \phi(x_i), \theta \rangle + g(\theta) + \frac{1}{2\sigma^2} \|	heta\|^2 + \text{const.} - \log p(x_i|\theta) - \log p(\theta)$$

Good News
Minimizing $- \log p(\theta|X)$ is a convex optimization problem.
Tossing a dice with priors

[Diagrams showing probability distributions for different outcomes of a dice roll with priors. Each diagram has bars representing the probability of each face.]
Normal Prior on $\theta$ ...

\[ \theta \sim \mathcal{N}(0, \sigma^2 I) \]

... yields Normal Prior on $t(x) = \langle \phi(x), \theta \rangle$

Distribution of projected Gaussian is Gaussian.

The mean vanishes

\[ \mathbb{E}_\theta[t(x)] = \langle \phi(x), \mathbb{E}_\theta[\theta] \rangle = 0 \]

The covariance yields

\[ \text{Cov}[t(x), t(x')] = \mathbb{E}_\theta[\langle \phi(x), \theta \rangle \langle \theta, \phi(x') \rangle] = \sigma^2 \langle \phi(x), \phi(x') \rangle \]

\[ := k(x, x') \]

... so we have a Gaussian Process on $x$ ...

with kernel $k(x, x') = \sigma^2 \langle \phi(x), \phi(x') \rangle$. 
Conditional Distributions

Conditional Density

\[
p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))
\]

\[
p(y|x, \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(\theta|x))
\]

Maximum a Posteriori Estimation

\[
-\log p(\theta|X) = \sum_{i=1}^{m} -\langle \phi(x_i), \theta \rangle + mg(\theta) + \frac{1}{2\sigma^2}||\theta||^2 + c
\]

\[
-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}||\theta||^2 + c
\]

Solving the Problem

- Expand \( \theta \) in a linear combination of \( \phi(x_i, y_i) \).
- Solve convex problem in expansion coefficients.
Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass estimator: we have a distribution over $n$ classes which depends on $x$.
- Conditionally Gaussian leads to Gaussian Process regression: we have a normal distribution over a random variable which depends on the location.

**Note:** we estimate mean and variance.

- Conditionally Poisson distributions yields spatial Poisson model.

**Solve the optimization problem**

This is typically convex.

**The bottom line**

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$. 
Sufficient Statistic
We pick $\phi(x, y) = \phi(x) \otimes e_y$, that is
\[ k((x, y), (x', y')) = k(x, x') \delta_{yy'} \text{ where } y, y' \in \{1, \ldots, n\} \]

Kernel Expansion
By the representer theorem we get that
\[ \theta = \sum_{i=1}^{m} \sum_{y} \alpha_{iy} \phi(x_i, y) \]

Optimization Problem
Big mess . . . but convex.
A Toy Example
Noisy Data
Example: GP Regression

Sufficient Statistic (Standard Model)
We pick \( \phi(x, y) = (y\phi(x), y^2) \), that is

\[
k((x, y), (x', y')) = k(x, x')yy' + y^2y'^2 \text{ where } y, y' \in \mathbb{R}
\]

Traditionally the variance is fixed, that is \( \theta_2 = \text{const.} \).

Sufficient Statistic (Fancy Model)
We pick \( \phi(x, y) = (y\phi_1(x), y^2\phi_2(x)) \), that is

\[
k((x, y), (x', y')) = k_1(x, x')yy' + k_2(x, x')y^2y'^2 \text{ where } y, y' \in \mathbb{R}
\]

We estimate mean and variance simultaneously.

Kernel Expansion
By the representer theorem (and more algebra) we get

\[
\theta = \left( \sum_{i=1}^{m} \alpha_i \phi_1(x_i), \sum_{i=1}^{m} \alpha_i \phi_2(x_i) \right)
\]
Mean $k^\top(x)(K + \sigma^2 1)^{-1}y$
Variance $\kappa(x, x) + \sigma^2 - \vec{k}^\top(x)(K + \sigma^2 \mathbf{1})^{-1}\vec{k}(x)$
Optimization Problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} \left[ -\frac{1}{4} \left( \sum_{j=1}^{m} \alpha_{1j} k_1(x_i, x_j) \right)^\top \left( \sum_{j=1}^{m} \alpha_{2j} k_2(x_i, x_j) \right)^{-1} \left( \sum_{j=1}^{m} \alpha_{1j} k_1(x_i, x_j) \right) \right] \\
& \quad - \frac{1}{2} \log \det \left( 2 \sum_{j=1}^{m} \alpha_{2j} k_2(x_i, x_j) \right) - \sum_{j=1}^{m} \left[ y_i^\top \alpha_{1j} k_1(x_i, x_j) + (y_j^\top \alpha_{2j} y_j) k_2(x_i, x_j) \right] \\
& \quad + \frac{1}{2\sigma^2} \sum_{i,j} \alpha_1^\top \alpha_1 k_1(x_i, x_j) + \text{tr} \left[ \alpha_2 \alpha_2^\top \right] k_2(x_i, x_j).
\end{align*}
\]

subject to \(0 \succ \sum_{i=1}^{m} \alpha_{2i} k(x_i, x_j)\).

Properties of the problem:

- The problem is convex
- The log-determinant from the normalization of the Gaussian acts as a barrier function.
- We get a semidefinite program.
Natural Parameters

\( \theta_1 \) estimation

\( \theta_2 \) estimation
Joint density and graphical models

Hammersley-Clifford Theorem

\[ p(x) = \frac{1}{Z} \exp \left( \sum_{c \in C} \psi_c(x_c) \right) \]

Decomposition of any \( p(x) \) into product of potential functions on maximal cliques.
Hammersley-Clifford Corollary
Combining the CH-Theorem and exponential families

\[ p(x) = \frac{1}{Z} \exp \left( \sum_{c \in \mathcal{C}} \psi_c(x_c) \right) \]

\[ p(x) = \exp \left( \langle \phi(x), \theta \rangle - g(\theta) \right) \]

we obtain a decomposition of \( \phi(x) \) into

\[ p(x) = \exp \left( \sum_{c \in \mathcal{C}} \langle \phi_c(x_c), \theta_c \rangle - g(\theta) \right) \]

Consequence for Kernels

\[ k(x, x') = \sum_{c \in \mathcal{C}} k_c(x_c, x'_c) \]
Dependence structure between variables

<table>
<thead>
<tr>
<th>Time</th>
<th>$t - 2$</th>
<th>$t - 1$</th>
<th>$t$</th>
<th>$t + 1$</th>
<th>$t + 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

Key Points

- We can drop cliques in $x$: they do not affect $p(y|x, \theta)$.
- Compute $g(\theta|x)$ via dynamic programming.
- Assume stationarity of the model, that is $\theta_c$ does not depend on the position of the clique.
- We only need a sufficient statistic $\phi_{xy}(x_t, y_t)$ and $\phi_{yy}(y_t, y_{t+1})$. 
Conditional Probabilities:

\[
p(y|x, \theta) \propto \prod_{t=1}^{T} \frac{\exp \left( \langle \phi_{xy}(x_t, y_t), \theta_{xy} \rangle + \langle \phi_{yy}(y_t, y_{t+1}), \theta_{yy} \rangle \right)}{M(y_t, y_{t+1})}
\]

So we can compute \( p(y_t|x, \theta) \) and \( p(y_t, y_{t+1}|x, \theta) \) via dynamic programming.

Objective Function:

\[
- \log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2} ||\theta||^2 + c
\]

\[
\partial_\theta - \log p(\theta|X, Y) = \sum_{i=1}^{m} -\phi(x_i, y_i) + \mathbb{E} [\phi(x_i, y_i)|x_i] + \frac{1}{\sigma^2} \theta
\]

We only need \( \mathbb{E} [\phi_{xy}(x_{it}, y_{it})|x_i] \) and \( \mathbb{E} [\phi_{yy}(y_{it}, y_{i(t+1)})|x_i] \).
**Conditional Random Field:** maximize $p(y|x, \theta)$

<table>
<thead>
<tr>
<th>Time</th>
<th>$t - 2$</th>
<th>$t - 1$</th>
<th>$t$</th>
<th>$t + 1$</th>
<th>$t + 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

**Hidden Markov Model:** maximize $p(x, y|\theta)$

<table>
<thead>
<tr>
<th>Time</th>
<th>$t - 2$</th>
<th>$t - 1$</th>
<th>$t$</th>
<th>$t + 1$</th>
<th>$t + 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
</tr>
</tbody>
</table>
Extension: Missing Variables

Basic Idea
We can integrate out over missing variables to obtain

\[ p(y|x, \theta) = \sum_{x_{\text{miss}}} p(y, x_{\text{miss}}|x, \theta) \]

= \sum_{x_{\text{miss}}} \exp(\langle \phi(x, x_{\text{miss}}, y), \theta \rangle - g(\theta|x))

= \exp(g(\theta|x, y) - g(\theta|x))

Big Problem
The optimization is not convex any more. But it still is a difference of two convex functions.

- Solve via Concave-Convex procedure (i.e. EM)
- Use fancy DC programming (to do)
Basic Idea

Instead of minimizing $-\log p(y|x, \theta)$ optimize a trimmed log likelihood ratio

$$R(x, y, \theta) := \log \frac{p(y|x, \theta)}{\max_{y' \neq y} p(y|x, \theta)}$$

$$= \langle \phi(x, y), \theta \rangle - \max_{y' \neq y} \langle \phi(x, y'), \theta \rangle$$

Minimizing $\min(\rho - R(x, y, \theta), 0)$ gives the large-margin criterion.

Technical Detail

For sequences finding the best and second-best sequence is done by dynamic programming. We get the Maximum-Margin-Markov Networks.
Basic Idea
For correct classification it is sufficient if the log-likelihood ratio $R(x, y, \theta) > 0$.

Algorithm
- Initialize $\theta = 0$
- Repeat
  - If $R(x_i, y_i, \theta) < 0$ update $\theta \leftarrow \theta + (\phi(x_i, y_i) - \phi(x_i, y^*))$
- Until all $R(x_i, y_i, \theta) > 0$

Convergence
The perceptron algorithm converges in $\frac{\|\theta\|^2}{\max\|\phi(x_i, y)\|^2}$ updates.
Semi-supervised learning
We have both the training set $X, Y$ and the test patterns $X'$ available at estimation time. Can we take advantage of this additional information (aka “transduction”)?

Partially labeled data
Some observations may have uncertain labels, i.e., $y_i \in Y_i \subseteq Y$ (such as $y_i \in \{\text{apple, oranges}\}$ but $y_i \neq \text{pear}$). Can we use the observations and also infer labels?

Clustering
Here we have no label information at all. The goal is to find a plausible assignment of $y_i$ such that similar observations tend to share the same label.

Key Idea
We maximize the likelihood $p(y|t, X)$ over $t$ and $y$. 

Extension: Partial Labels
Interacting Agents
We have a set of agents which only interact with their neighbors.

Junction Tree
Can use distributed algorithm to find junction tree based on local neighborhood structure. This assumes “nice” structure in the neighborhood graph.

Local Message Passing
Use the Generalized Distributive Law, if junction tree is thin enough. Messages are expectations of $\phi_c(x_c)$.

Alternative
When no junction tree exists, just use loopy belief propagation. And hope . . .
Summary

- Sufficient statistic leads to kernel via
  \[ k(x, x') = \langle \phi(x), \phi(x') \rangle \]

- Maximum a posteriori is convex problem

- Conditioning turns simple models into fancy nonparametric estimators, such as
  - Normal distribution \(\Rightarrow\) regression
  - Multinomial distribution \(\Rightarrow\) multiclass classification
  - Structured statistic \(\Rightarrow\) CRF
  - Poisson distribution \(\Rightarrow\) spatial disease model
  - Latent category \(\Rightarrow\) clustering model
Shameless Plugs

We are hiring. For details contact
Alex.Smola@nicta.com.au (http://www.nicta.com.au)

Positions
- PhD scholarships
- Postdoctoral positions, Senior researchers
- Long-term visitors (sabbaticals etc.)

More details on kernels
- http://www.kernel-machines.org
  Schölkopf and Smola: Learning with Kernels

Machine Learning Summer School
- http://canberra05.mlss.cc
  MLSS’05 Canberra, Australia, 23/1-5/2/2005