CS 229 Lecture Four
The Exponential Family

Chris Ré

April 2, 2023

Exponential Family

- Definition and motivation for the exponential family
- Examples
- SoftMax (Multiclass Classification)

> The exponential family unifies inference and learning for many important models

Exponential Family

Rough Idea *"If P has a a special form, then inference and learning come for free"*

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Here y, $a(\eta)$, and $b(y)$ are scalars. $T(y)$ same dimension as η.

Exponential Family

Rough Idea *"If P has a a special form, then inference and learning come for free"*

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Here y, $a(\eta)$, and $b(y)$ are scalars. $T(y)$ same dimension as η.
These terms have names:

- $T(y)$ is called the **sufficient statistic**.
- $b(y)$ is called the **base measure** – does *not* depend on η.
- $a(\eta)$ is called the **log partition function** – does *not* depend on y.

Exponential Family

Rough Idea *"If P has a a special form, then inference and learning come for free"*

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Here y, $a(\eta)$, and $b(y)$ are scalars. $T(y)$ same dimension as η. These terms have names:

- $T(y)$ is called the **sufficient statistic**.
- $b(y)$ is called the **base measure** – does *not* depend on η.
- $a(\eta)$ is called the **log partition function** – does *not* depend on y.

$$1 = \sum_y P(y;\eta) = e^{-a(\eta)} \sum_y b(y)\exp\left\{\eta^T T(y)\right\}$$

Exponential Family

Rough Idea *"If P has a a special form, then inference and learning come for free"*

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Here y, $a(\eta)$, and $b(y)$ are scalars. $T(y)$ same dimension as η. These terms have names:

- $T(y)$ is called the **sufficient statistic**.
- $b(y)$ is called the **base measure** – does *not* depend on η.
- $a(\eta)$ is called the **log partition function** – does *not* depend on y.

$$1 = \sum_y P(y;\eta) = e^{-a(\eta)} \sum_y b(y)\exp\left\{\eta^T T(y)\right\}$$

$$\implies a(\eta) = \log \sum_y b(y)\exp\left\{\eta^T T(y)\right\}$$

Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$P(y;\theta) = \theta^y(1-\theta)^{1-y}.$$

Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$P(y;\theta) = \theta^{y}(1-\theta)^{1-y}.$$

How do we put it in the required form?

$$P(y;\eta) = b(y)\exp\left\{\eta^{T}T(y) - a(\eta)\right\}.$$

Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$P(y;\theta) = \theta^y (1-\theta)^{1-y}.$$

How do we put it in the required form?

$$P(y;\eta) = b(y) \exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

$$\theta^y (1-\theta)^{1-y} = \exp\{y \log \phi + (1-y)\log(1-\phi)\}$$

Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$P(y;\theta) = \theta^y (1-\theta)^{1-y}.$$

How do we put it in the required form?

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

$$\begin{aligned}\theta^y (1-\theta)^{1-y} &= \exp\left\{y\log\phi + (1-y)\log(1-\phi)\right\} \\ &= \exp\left\{y\log\frac{\phi}{1-\phi} + \log(1-\phi)\right\}\end{aligned}$$

Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$P(y;\theta) = \theta^y (1-\theta)^{1-y}.$$

How do we put it in the required form?

$$P(y;\eta) = b(y) \exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

$$\begin{aligned} \theta^y (1-\theta)^{1-y} &= \exp\{y \log \phi + (1-y)\log(1-\phi)\} \\ &= \exp\left\{y \log \frac{\phi}{1-\phi} + \log(1-\phi)\right\} \end{aligned}$$

So then:

$$\eta = \log \frac{\phi}{1-\phi}, T(y) = y, a(\eta) = -\log(1-\phi).$$

We need to show that $a(\eta)$ is indeed a function of η.

Showing $a(\eta)$ is a function of η

We first observe that:

$$\eta = \log \frac{\phi}{1 - \phi} \implies e^{\eta}(1 - \phi) = \phi$$

Showing $a(\eta)$ is a function of η

We first observe that:

$$\eta = \log \frac{\phi}{1-\phi} \implies e^{\eta}(1-\phi) = \phi$$

$$e^{\eta} = (e^{\eta}+1)\phi \implies \phi = \frac{1}{1+e^{-\eta}}$$

Showing $a(\eta)$ is a function of η

We first observe that:

$$\eta = \log \frac{\phi}{1-\phi} \implies e^{\eta}(1-\phi) = \phi$$
$$e^{\eta} = (e^{\eta}+1)\phi \implies \phi = \frac{1}{1+e^{-\eta}}$$

Now, we plug into $\log(1-\phi)$ and we verify:

$$a(\eta) = \log(1-\phi) = \log \frac{e^{-\eta}}{1+e^{-\eta}} = -\log(1+e^{\eta}).$$

Takeaway: We have verfified the Bernoulli distribution is in the exponential family!

Example 2: Gaussians with Fixed Variance $\sigma^2 = 1$

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(y-\mu)^2\right\}.$$

Example 2: Gaussians with Fixed Variance $\sigma^2 = 1$

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(y-\mu)^2\right\}.$$

Can we put it in the exponential family form?

$$P(y;\eta) = b(y) \exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Example 2: Gaussians with Fixed Variance $\sigma^2 = 1$

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(y-\mu)^2\right\}.$$

Can we put it in the exponential family form?

$$P(y;\eta) = b(y) \exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Multiply out the square and group terms:

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-y^2/2\right\} \exp\left\{\mu y - \frac{1}{2}\mu^2\right\}.$$

Example 2: Gaussians with Fixed Variance $\sigma^2 = 1$

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(y-\mu)^2\right\}.$$

Can we put it in the exponential family form?

$$P(y;\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}.$$

Multiply out the square and group terms:

$$P(y;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-y^2/2\right\} \exp\left\{\mu y - \frac{1}{2}\mu^2\right\}.$$

Perfect!

$$\eta = \mu,\ T(y) = y,\ a(\eta) = \frac{1}{2}\eta^2.$$

Takeaway: Normal distribution is in the exponential family.

An Observation . . .

Notice that for a Gaussian with mean μ we had

$$\eta = \mu, T(y) = y, a(\eta) = \frac{1}{2}\eta^2.$$

We observe something peculiar:

$$\partial_\eta a(\eta) = \eta = \mu = \mathbb{E}[y] \text{ and } \partial_\eta^2 a(\eta) = 1 = \sigma^2 = \text{var}(y)$$

That is, derivatives of the log partition function is the expectation and variance. Same for Bernoulli.

Is this true in general?

Log Partition Function

Yes! Recall that

$$a(\eta) = \log \sum_{y} b(y) \exp\left\{\eta^T T(y)\right\}$$

Log Partition Function

Yes! Recall that

$$a(\eta) = \log \sum_{y} b(y) \exp\left\{\eta^T T(y)\right\}$$

Then, taking derivatives

$$\nabla_\eta a(\eta) = \frac{\sum_y T(y) b(y) \exp\{\eta^T T(y)\}}{\sum_y b(y) \exp\{\eta^T T(y)\}} = \mathbb{E}[T(y); \eta]$$

Takeaway: In this way, once we're in the exponential family–we get inference "for free" meaning in the same way for every member.

Log Partition Function

Yes! Recall that

$$a(\eta) = \log \sum_y b(y) \exp\left\{\eta^T T(y)\right\}$$

Then, taking derivatives

$$\nabla_\eta a(\eta) = \frac{\sum_y T(y) b(y) \exp\{\eta^T T(y)\}}{\sum_y b(y) \exp\{\eta^T T(y)\}} = \mathbb{E}[T(y); \eta]$$

Takeaway: In this way, once we're in the exponential family–we get inference "for free" meaning in the same way for every member.

Note: $\nabla_\eta^2 a(\eta) = \mathrm{var}[T(y); \eta]$, you can check!

Some Facts About Exponential Models

- There are many canonical exponential family models:
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet

Some Facts About Exponential Models

- There are many canonical exponential family models:
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet
- In this course, we'll use $T(y) = y$.

Generalized Linear Models (using Exponential Family Models)

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model.

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model. We first we pick a distribution based on y's type.

- We assume $y \mid x; \theta$ distributed as an exponential family.
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model. We first we pick a distribution based on y's type.

- We assume $y \mid x; \theta$ distributed as an exponential family.
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet
- Our model is *linear* beacuse we make the natural parameter $\eta = \theta^T x$ in which $\theta, x \in \mathbb{R}^{d+1}$.

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model. We first we pick a distribution based on y's type.

- We assume $y \mid x; \theta$ distributed as an exponential family.
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet
- Our model is *linear* beacuse we make the natural parameter $\eta = \theta^T x$ in which $\theta, x \in \mathbb{R}^{d+1}$.

inference $h_\theta(x) = \mathbb{E}[y \mid x; \theta]$ is the **output**.

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model. We first we pick a distribution based on y's type.

- We assume $y \mid x; \theta$ distributed as an exponential family.
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet
- Our model is *linear* beacuse we make the natural parameter $\eta = \theta^T x$ in which $\theta, x \in \mathbb{R}^{d+1}$.

inference	$h_\theta(x) = \mathbb{E}[y \mid x; \theta]$ is the **output**.
learn	$\max_\theta \log p(y \mid x; \theta)$ by maximum likelihood.

Generalized Linear Models: Recipe

We're given features $x \in \mathbb{R}^{d+1}$ and a target y. We want a model. We first we pick a distribution based on y's type.

- We assume $y \mid x; \theta$ distributed as an exponential family.
  - Binary $\mapsto$ Bernoulli
  - Multiple Classses $\mapsto$ Multinomial
  - Real $\mapsto$ Gaussian
  - Counts $\mapsto$ Poisson
  - $\mathbb{R}_+ \mapsto$ Gamma, Exponential
  - Distributions $\mapsto$ Dirichlet
- Our model is *linear* beacuse we make the natural parameter $\eta = \theta^T x$ in which $\theta, x \in \mathbb{R}^{d+1}$.

inference $h_\theta(x) = \mathbb{E}[y \mid x; \theta]$ is the **output**.

learn $\max_\theta \log p(y \mid x; \theta)$ by maximum likelihood.

algorithm: SGD $\theta^{(t+1)} = \theta^{(t)} + \alpha \left(y^{(i)} - h_{\theta^{(t)}}(x^{(i)}) \right) x^{(i)}$.

Terminology for Exponential Family

Lots of names for parameters floating around...

$$\underline{\text{Model Parameter}} \qquad\qquad \underline{\text{Natural Parameter}} \qquad\qquad \underline{\text{Canonical}}$$

$$\theta \quad \xmapsto{\theta^T x} \quad \eta \quad \xmapsto{g} \quad \begin{array}{rl} \phi : & \text{Bernoulli} \\ \mu : & \text{Gaussian} \\ \lambda : & \text{Poisson} \end{array}$$

- We move from the **model parameters** (θ) to the **natural parameters** (ν) via a linear function $\theta^T x$.
- g is **canonical response** or **link** function
- Note sometimes g^{-1} is called the link function.
- Logistic regression $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ with $g(z) = \frac{1}{1+e^{-z}}$
- Gaussian $h_\theta(x) = \mu = \theta^T x$

A Quick and Dirty Intro to Multiclass Classification.
This technique is *the daily workhorse of modern AI/ML*

Multiclass

Suppose we want to choose among k discrete values, e.g., $\{\text{'Cat'}, \text{'Dog'}, \text{'Car'}, \text{'Bus'}\}$ so $k = 4$.

We encode with **one-hot** vectors i.e. $y \in \{0,1\}^k$ and $\sum_{j=1}^{k} y_j = 1$.

$$\underset{\text{'Cat'}}{\begin{pmatrix}1\\0\\0\\0\end{pmatrix}} \quad \underset{\text{'Dog'}}{\begin{pmatrix}0\\1\\0\\0\end{pmatrix}} \quad \underset{\text{'Car'}}{\begin{pmatrix}0\\0\\1\\0\end{pmatrix}} \quad \underset{\text{'Bus'}}{\begin{pmatrix}0\\0\\0\\1\end{pmatrix}}$$

A prediction here is actually a *distribution* over the k classes. This leads to the SoftMax function described below (derivation in the notes!). That is our hypothesis is a vector of k values:

$$P(y = j | x; \bar{\theta}) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}.$$

Here each θ_j has the *same dimension* as x, i.e., $x, \theta_j \in R^{d+1}$ for $j = 1, \ldots, k$.

Multiclass Image: Picture in Class

$$P(y = j | x; \theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}.$$

Quick Comments on Presentation

- *Check for home:* does $k = 2$ case agree with logistic regression?

$$P(y = j|x; \theta) = \frac{e^{\theta_j^T x}}{e^{\theta_1^T x} + e^{\theta_2^T x}}$$

Hint: Given (θ_1, θ_2) for a two class model, compare with logistic regression with the model $\theta_1 - \theta_2$.

- For general k, a probability estimate for any $k - 1$ classes determines the other class (since estimates must sum to 1).

Quick Comments on Presentation

- *Check for home:* does $k = 2$ case agree with logistic regression?

$$P(y = j | x; \theta) = \frac{e^{\theta_j^T x}}{e^{\theta_1^T x} + e^{\theta_2^T x}}$$

Hint: Given (θ_1, θ_2) for a two class model, compare with logistic regression with the model $\theta_1 - \theta_2$.

- For general k, a probability estimate for any $k - 1$ classes determines the other class (since estimates must sum to 1).
- With this observation (and some notation!), you can run the machine from this lecture: multinomials are in the exponential family, and it tells us how to do inference, training, etc.

How do you train multiclass? (Picture Version)

$$P(y = j|x;\theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}.$$

Intuitively, we maximize the probability of the given class.

How do you train multiclass?

Fixing x and θ, our output is a vector $\hat{p} \in \mathbb{R}_+^k$ s.t. $\sum_{j=1}^k \hat{p}_j = 1$.

$$\hat{p}_j = P(y = j | x; \theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^k \exp(\theta_i^T x)}.$$

Formally, we maximize the probability of the given class!

How do you train multiclass?

Fixing x and θ, our output is a vector $\hat{p} \in \mathbb{R}_+^k$ s.t. $\sum_{j=1}^k \hat{p}_j = 1$.

$$\hat{p}_j = P(y = j | x; \theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^k \exp(\theta_i^T x)}.$$

Formally, we maximize the probability of the given class!
We can view as $\textsc{CrossEntropy}$:

$$\textsc{CrossEntropy}(p, \hat{p}) = -\sum_j p(x = j) \log \hat{p}(x = j).$$

Here, p is the label, which is a one-hot vector.

How do you train multiclass?

Fixing x and θ, our output is a vector $\hat{p} \in \mathbb{R}^k_+$ s.t. $\sum_{j=1}^k \hat{p}_j = 1$.

$$\hat{p}_j = P(y = j | x; \theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^k \exp(\theta_i^T x)}.$$

Formally, we maximize the probability of the given class!
We can view as $\textsc{CrossEntropy}$:

$$\textsc{CrossEntropy}(p, \hat{p}) = -\sum_j p(x = j) \log \hat{p}(x = j).$$

Here, p is the label, which is a one-hot vector.Thus, if the label is i, this formula reduces to:

$$-\log \hat{p}(x = i) = -\log \frac{\exp(\theta_i^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)}.$$

We minimize this–and you've seen the movie, it works the same as the others!

How do you train multiclass? (Label Smooth)

$$P(y = j | x; \theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}.$$

Maximize the probability of the given class!

How do you train multiclass? (Label Smooth)

$$P(y = j|x;\theta) = \frac{\exp(\theta_j^T x)}{\sum_{i=1}^{k} \exp(\theta_i^T x)}.$$

Maximize the probability of the given class!

Plugin into $\textsc{CrossEntropy}$, and we're in good shape! Why might label smoothing help?

Summary of Exponential Family Models

- We saw exponential families that gave us a path to generalize to a wider set of models.
- We saw CrossEntropy and SoftMax, which are ML/AI people use *every* day.
- I mentioned label smoothing because your data and model are always at least *little* wrong.