Generalized Linear Models

David Rosenberg

New York University

October 29, 2016

Gaussian Regression

- Input space $\mathcal{X} = \mathbf{R}^d$, Output space $\mathcal{Y} = \mathbf{R}$
  - Hypothesis space consists of functions $f : x \mapsto \mathcal{N}\left(w^T x, \sigma^2\right)$.
  - For each x, $f(x)$ returns a particular Gaussian density with variance σ^2 .
  - Choice of w determines the function.
- For some parameter $w \in \mathbf{R}^d$, can write our prediction function as

$$[f_w(x)](y) = p_w(y \mid x) = \mathcal{N}(y \mid w^T x, \sigma^2),$$

  where $\sigma^2 > 0$.
- Given some i.i.d. data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, how to assess the fit?

Gaussian Regression: Likelihood Scoring

- Suppose we have data $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$.
- Compute the model likelihood for $\mathcal{D}$:

$$p_w(\mathcal{D}) = \prod_{i=1}^{n} p_w(y_i \mid x_i) \text{ [by independence]}$$

- Maximum Likelihood Estimation (MLE) finds w maximizing $p_w(\mathcal{D})$.
- Equivalently, maximize the data log-likelihood:

$$w^* = \underset{w \in \mathbf{R}^d}{\arg\max} \sum_{i=1}^{n} \log p_w(y_i \mid x_i)$$

- Let's start solving this!

Gaussian Regression: MLE

- The conditional log-likelihood is:

$$
\begin{aligned}
&\sum_{i=1}^{n} \log p_w(y_i \mid x_i) \\
= \quad &\sum_{i=1}^{n} \log \left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \right] \\
= \quad &\underbrace{\sum_{i=1}^{n} \log \left[\frac{1}{\sigma\sqrt{2\pi}} \right]}_{\text{independent of } w} + \sum_{i=1}^{n} \left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right)
\end{aligned}
$$

- MLE is the w where this is maximized.
- Note that σ^2 is irrelevant to finding the maximizing w.
- Can drop the negative sign and make it a minimization problem.

Gaussian Regression: MLE

- The MLE is

$$w^* = \underset{w \in \mathbf{R}^d}{\arg\min} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

- This is exactly the objective function for least squares.
- From here, can use usual approaches to solve for w^*(linear algebra, calculus, iterative methods etc.)
- NOTE: Parameter vector w only interacts with x by an inner product

Poisson Regression: Setup

- Input space $\mathcal{X} = \mathbf{R}^d$, Output space $\mathcal{Y} = \{0, 1, 2, 3, 4, \ldots\}$
- Hypothesis space consists of functions $f : x \mapsto \text{Poisson}\left(\lambda(x)\right)$.
  - That is, for each x, $f(x)$ returns a Poisson with mean $\lambda(x)$.
  - What function?
- Recall $\lambda > 0$.
- GLMs (and Poisson is a special case) have a linear dependence on x.
- Standard approach is to take

$$\lambda(x) = \exp\left(w^T x\right),$$

  for some parameter vector w.
- Note that range of $\lambda(x) = (0, \infty)$, (appropriate for the Poisson parameter).

Poisson Regression: Likelihood Scoring

- Suppose we have data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.
- Last time we found the log-likelihood for Poisson was:

$$\log p(\mathcal{D}, \lambda) = \sum_{i=1}^{n} [y_i \log \lambda - \lambda - \log(y_i!)]$$

- Plugging in $\lambda(x) = \exp\left(w^T x\right)$, we get

$$\begin{aligned}\log p(\mathcal{D}, \lambda) &= \sum_{i=1}^{n} \left[y_i \log\left[\exp\left(w^T x\right)\right] - \exp\left(w^T x\right) - \log(y_i!)\right] \\ &= \sum_{i=1}^{n} \left[y_i w^T x - \exp\left(w^T x\right) - \log(y_i!)\right]\end{aligned}$$

- Maximize this w.r.t. w to find the Poisson regression.
- No closed form for optimum, but it's concave, so easy to optimize.

Linear Probabilistic Classifiers

- Setting: $\mathcal{X} = \mathbf{R}^d$, $\mathcal{Y} = \{0, 1\}$
- For each $X = x$, $p(Y = 1 \mid x) = \theta$. (i.e. Y has a Bernoulli(θ) distribution)
- θ may vary with x.
- For each $x \in \mathbf{R}^d$, just want to predict $\theta \in [0, 1]$.
- Two steps:

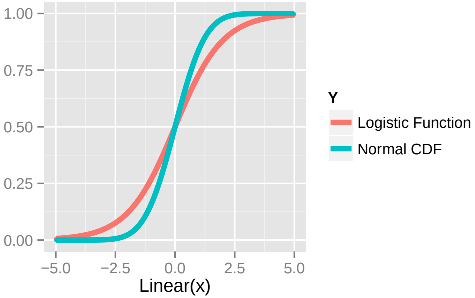
$$\underbrace{x}_{\in \mathbf{R}^D} \mapsto \underbrace{w^T x}_{\in \mathbf{R}} \mapsto \underbrace{f(w^T x)}_{\in [0,1]},$$

  where $f : \mathbf{R} \to [0, 1]$ is called the **transfer** or **inverse link** function.
- Probability model is then

$$p(Y = 1 \mid x) = f(w^T x)$$

Inverse Link Functions

- Two commonly used "inverse link" functions to map from $w^T x$ to θ:



- Logistic function $\Longrightarrow$ Logistic Regression
- Normal CDF $\Longrightarrow$ Probit Regression

Multinomial Logistic Regression

- Setting: $\mathcal{X} = \mathbf{R}^d$, $\mathcal{Y} = \{1, \ldots, K\}$
- The numbers $(\theta_1, \ldots, \theta_c)$ where $\sum_{c=1}^{K} \theta_c = 1$ represent a
  - "**multinoulli**" or "**categorical**" distribution.
- For each x, we want to produce a distribution on the K classes.
- That is, for each x and each $y \in \{1, \ldots, K\}$, we want to produce a probability

$$p(y \mid x) = \theta_y,$$

  where $\sum_{y=1}^{K} \theta_y = 1$.

Multinomial Logistic Regression: Classic Setup

- Classically we write multinomial logistic regression (cf. KPM Sec. 8.3.7):

$$p(y \mid x) = \frac{\exp\left(w_y^T x\right)}{\sum_{c=1}^{K} \exp\left(w_c^T x\right)},$$

  where we've introduced parameter vectors $w_1, \ldots, w_K \in \mathbf{R}^d$.
- The log of this likelihod is concave and straightforward to optimize.

More Convenient to Flatten This

- Dropping proportionality constant $Z(x) = \sum_{c=1}^{K} \exp\left(w_c^T x\right)$, we have

$$
\begin{aligned}
p(y \mid x) &\propto \exp\left(w_y^T x\right) \\
&= \exp\left(\sum_{c=1}^{K} 1(y=c) w_c^T x\right) \\
&= \exp\left(\sum_{c=1}^{K} 1(y=c) \left[\sum_{j=1}^{d} (w_c)_j x_j\right]\right) \\
&= \exp\left(\sum_{i=1}^{K} \sum_{j=1}^{d} (w_c)_j \underbrace{1(y=c) x_j}\right)
\end{aligned}
$$

- Create a "feature" for every term $1(y=c)x_j$, for $c \in \{1, \ldots, k\}$.
- Define feature function

$$g_r(x,y) = 1(y=c)x_j.$$

More Convenient to Flatten This

- So

$$
\begin{aligned}
p(y \mid x) &\propto \exp\left(\sum_{i=1}^{K}\sum_{j=1}^{d}(w_c)_j \underbrace{1(y=c)x_j}\right) \\
&= \exp\left(\sum_{r=1}^{R}\mu_r g_r(x,y)\right).
\end{aligned}
$$

- What is R? What are the μ_r's
- $R = kd$ and μ_r's are just some flattening of $w_1, \ldots, w_K$ into a single vector.

More Convenient to Flatten This

- Why did we do this?
- Computational Reason:
    - To plug into optimization algorithm, easier to have a single parameter vector.
    - Original version had K parameter vectors.
- Conceptual Reason:
    - Introduce the idea of “features” that depend jointly on input and output.
    - These “features” measure “compatibility” between input and particular label.
    - We could call them “compatibility functions”, but we usually call them features.
- Example from natural language processing: (Part-of-speech tagging)

$$g_r(y,x) = \begin{cases} 1 & \text{if } y = \text{"NOUN" and } x_i = \text{"apple"} \\ 0 & \text{otherwise} \end{cases}$$

Natural Exponential Families

- $\{p_\theta(y) \mid \theta \in \Theta \subset \mathbf{R}^d\}$ is a family of pdf's or pmf's on $\mathcal{Y}$.
- The family is a **natural exponential family** with parameter θ if

$$p_\theta(y) = \frac{1}{Z(\theta)} h(y) \exp\left[\theta^T y\right].$$

- $h(y)$ is a **nonnegative** function called the **base measure.**
- $Z(\theta) = \int_{\mathcal{Y}} h(y) \exp\left[\theta^T y\right]$ is the **partition function**.
- The **natural parameter space** is the set $\Theta = \{\theta \mid Z(\theta) < \infty\}$.
    - the set of θ for which $\exp\left[\theta^T y\right]$ can be normalized to have integral 1
- θ is called the **natural parameter**.
- Note: In exponential family form, family typically has a different parameterization than the "standard" form.

Specifying a Natural Exponential Family

- The family is a **natural exponential family** with parameter θ if

$$p_\theta(y) = \frac{1}{Z(\theta)} h(y) \exp\left[\theta^T y\right].$$

- To specify a natural exponential family, we need to choose $h(y)$.
  - Everything else is determined.
- Implicit in choosing $h(y)$ is the choice of the support of the distribution.

Natural Exponential Families: Examples

The following are univariate natural exponential families:

1. Normal distribution with known variance.
2. Poisson distribution
3. Gamma distribution (with known k parameter)
4. Bernoulli distribution (and Binomial with known number of trials)

Example: Poisson Distribution

- For Poisson, we found the log probability mass function is:

$$\log\left[p(y;\lambda)\right] = y\log\lambda - \lambda - \log\left(y!\right).$$

- Exponentiating this, we get

$$p(y;\lambda) = \exp\left(y\log\lambda - \lambda - \log\left(y!\right)\right).$$

- If we reparametrize, taking $\theta = \log\lambda$, we can write this as

$$\begin{aligned} p(y,\theta) &= \exp\left(y\theta - e^{\theta} - \log\left(y!\right)\right) \\ &= \frac{1}{y!}\frac{1}{e^{e^{\theta}}}\exp\left(y\theta\right), \end{aligned}$$

  which is in natural exponential family form, where

$$\begin{aligned} Z(\theta) &= \exp\left(e^{\theta}\right) \\ h(y) &= \frac{1}{y!}. \end{aligned}$$

- $\theta = \log\lambda$ is the **natural parameter**.

Generalized Linear Models [with Canonical Link]

- In GLMs, we first choose a natural exponential family.
  - (This amounts to choosing $h(y)$.)
- The idea is to plug in $w^T x$ for the natural parameter.
- This gives models of the following form:

$$p_\theta(y \mid x) = \frac{1}{Z(w^T x)} h(y) \exp\left[(w^T x) y\right].$$

- This is the form we had for Poisson regression.
- **Note**: This is very convenient, but **only works if $\Theta = \mathbf{R}$.**

Generalized Linear Models [with General Link]

- More generally, choose a function $\psi : \mathbf{R} \to \Theta$ so that

$$x \mapsto w^T x \mapsto \psi(w^T x),$$

  where $\theta = \psi(w^T x)$ is the natural parameter for the family.
- So our final prediction (for one-parameter families) is:

$$p_\theta(y \mid x) = \frac{1}{Z(\psi(w^T x))} h(y) \exp\left[\psi(w^T x) y\right].$$