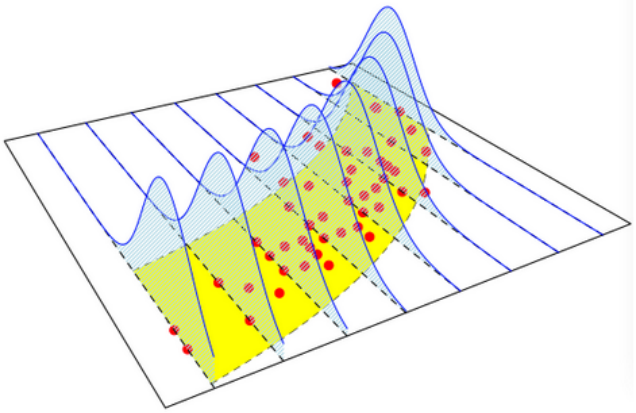


Generalised Linear Models

An overview of theory and implementation of GLMs

Michael Wedgwood

July, 2015

Contents

1 Introduction

This document gives an overview of GLMs (generalised linear models) and some specifics on derivations and implementations.

2 Overview of GLMs

According to Wikipedia

> The generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

For OLS the expected value of y, the target (or response) variable, is linearly related to x, the features (or inputs), and the assumption is that the errors are normally distributed. I.e. for each training example $(x^{(m)}, y^{(m)})$ and parameter (or weight) vector w

$$y^{(m)} = w^T x^{(m)} + \epsilon^{(m)} \quad \text{with} \quad \epsilon^{(m)} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \mathbb{E}[y^{(m)}] = w^T x^{(m)}$$

The GLM generalises this form such that a function (the link function - g) of the expected value of the target variable is linearly related to the features. I.e.

$$g(\mathbb{E}[y^{(m)}]) = w^T x^{(m)}$$

We also assume that each training example is drawn independently (of the other training examples) from the same distribution - i.e. the training examples are i.i.d. And the distribution is some distribution of the exponential family (which includes many of the most common distributions and has some useful properties).

Given this link function we can find the response function (or hypothesis - h) - which is what we want - by taking the inverse. I.e.

$$\mathbb{E}[y^{(m)}] = g^{-1}(w^T x^{(m)}) = h(w^T x^{(m)})$$

Note that some texts show the link function as an inverse i.e. g^{-1} and the response function as g.

You might ask: "Why don't we apply the link function to the target variable, and then solve using linear regression?" And the reason that this doesn't work well is that the link function is a function of $\mathbb{E}[y_i]$ and not of y_i. And we have y_i but we don't have $\mathbb{E}[y_i]$. Take for example a binary setting where we have the y_i value (either 0 or 1) but not the expected value which is in the range $[0, 1]$.

3 Model Form

The three main elements of a GLM model are as follows:

Assumption:

1. $y|x; w \sim$ Exponential Family(η)

   I.e. Given x and w, the distribution of y follows some exponential family distribution with parameter η.

Design choice:

2. The natural parameter η and the inputs x are related linearly.

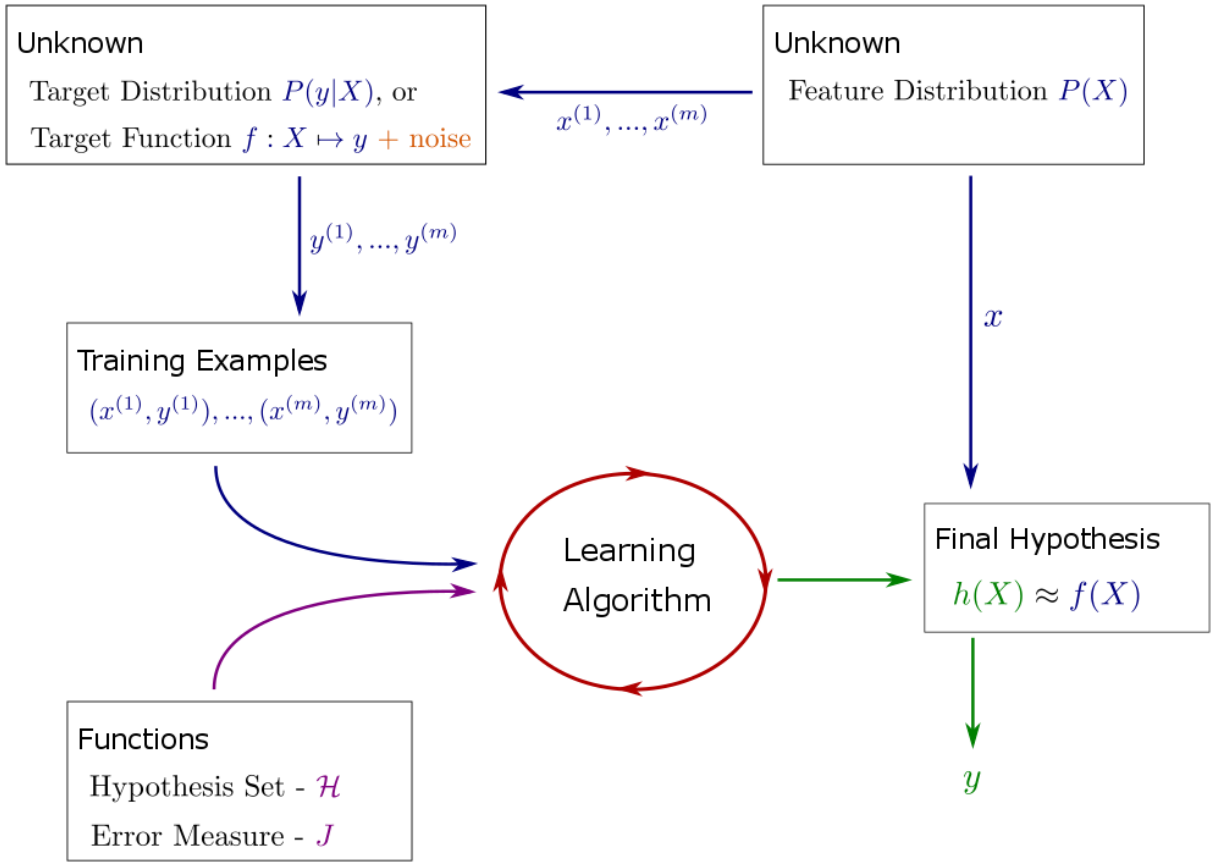
   I.e. $\eta = w^T x \quad$ or $\quad \eta_k = w_k^T x$ (if η is vector-valued)

Goal:

3. Given x, our goal is to predict the expected value of response $T(y)$.

   Typically $T(y) = y$ so we want $h(x) = \mathbb{E}(y|x)$.

The image below shows a general picture of a supervised learning process and gives some context to the GLM model assumptions.



Supervised Learning Process

Each training example input $x^{(m)}$ is drawn from some feature distribution $P(x)$. The target distribution $P(y|x)$ takes as input these x's and produces the data pairs $(x^{(m)}, y^{(m)})$. An assumption is that $P(y|x)$, the conditional distribution of y given the input x, is an exponential family distribution.

The hypothesis $h(x)$ from set $\mathcal{H}$ is derived from the assumption that the natural parameter is linearly related to the inputs i.e. $\eta = w^T x$ and is some function (the link function) of the expectation of y i.e. $g(\mathbb{E}[y]) = \eta = w^T x$. Inverting leads to the hypothesis function $h(x) = g^{-1}(w^T x) = \mathbb{E}[y|x]$.

The error measure (or cost function) J will be derived using the maximum likelihood method of parameter estimation which will rely on the i.i.d. assumption as well as some of the exponential family distribution properties.

4 Vectorisation

Some of the calculations and programming are made simpler with vector and matrix algebra. So it's useful being comfortable with the different vectors and matrices that will be used to programme a GLM.

Typically there will just be a single η parameter so we'll cover that before the general case.

4.1 Single η parameter i.e. η is single valued

Recall that the natural parameter is a linear function of the $n = 1, \ldots, N$ features. I.e.

$$\begin{aligned}
\eta &= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \cdots + w_N x_N \\
&= w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \cdots + w_N x_N \qquad \text{(with } x_0 = 1 \text{ where } w_0 \text{ is the bias)} \\
&= \textstyle\sum_{n=0}^{N} w_n x_n \\
&= x^T w
\end{aligned}$$

The design matrix $X \in \mathbb{R}^{M \times (N+1)}$ is a matrix where each of the $m = 1, \ldots, M$ rows is a single training example and each of the $n = 1, \ldots, N$ columns is an individual feature. Typically we'll add a vector of 1's so that a bias (or intercept) parameter is included.

$$X = \overbrace{\begin{bmatrix}
1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} & \cdots & x_N^{(1)} \\
1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} & \cdots & x_N^{(2)} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} & \cdots & x_N^{(m)} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & x_1^{(M)} & x_2^{(M)} & \cdots & x_n^{(M)} & \cdots & x_N^{(M)}
\end{bmatrix}}^{n = 0 : N \text{ features}} = \left.\begin{bmatrix}
x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \\ \vdots \\ x^{(M)}
\end{bmatrix}\right\} m = 1 : M \text{ training examples}$$

The parameter vector $w \in \mathbb{R}^{(N+1)}$ is a vector where each of the $n = 0, \ldots, N$ rows is an individual feature. (It's written here as transpose for neatness.)

$$w^T = \underbrace{\begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_n & \cdots & w_N \end{bmatrix}}_{n = 0 : N \text{ features}}$$

And this allows us to write the hypothesis function for all training examples as a single expression as follows:

$$Xw = \begin{bmatrix}
(x^{(1)})^T w \\
(x^{(2)})^T w \\
\vdots \\
(x^{(m)})^T w \\
\vdots \\
(x^{(M)})^T w
\end{bmatrix} \qquad \text{and therefore} \qquad h(Xw) = \begin{bmatrix}
h((x^{(1)})^T w) \\
h((x^{(2)})^T w) \\
\vdots \\
h((x^{(m)})^T w) \\
\vdots \\
h((x^{(M)})^T w)
\end{bmatrix} = \begin{bmatrix}
h_w(x^{(1)}) \\
h_w(x^{(2)}) \\
\vdots \\
h_w(x^{(m)}) \\
\vdots \\
h_w(x^{(M)})
\end{bmatrix} \in \mathbb{R}^M$$

4.2 Parameter η is vector valued

Recall that each of the $k = 1, \ldots, K$ natural parameters is a linear function of the $n = 1, \ldots, N$ features. I.e.

$$\begin{aligned}
\eta_k &= w_{0,k} + w_{1,k}x_1 + w_{2,k}x_2 + \cdots + w_{n,k}x_n + \cdots + w_{N,k}x_N \\
&= w_{0,k}x_0 + w_{1,k}x_1 + w_{2,k}x_2 + \cdots + w_{n,k}x_n + \cdots + w_{N,k}x_N \qquad \text{(with } x_0 = 1 \text{ where } w_0 \text{ is the bias)} \\
&= \textstyle\sum_{n=0}^{N} w_{n,k}x_n \\
&= x^T w_k
\end{aligned}$$

The design matrix $X \in \mathbb{R}^{M\times(N+1)}$ is the same as above. The parameters now form a matrix $W \in \mathbb{R}^{(N+1)\times K}$. I.e.

$$W = \underbrace{\left[\begin{array}{cccccc}
w_{1,0} & w_{2,0} & \cdots & w_{k,0} & \cdots & w_{K,0} \\
w_{1,1} & w_{2,1} & \cdots & w_{k,1} & \cdots & w_{K,1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
w_{1,n} & w_{2,n} & \cdots & w_{k,n} & \cdots & w_{K,n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
w_{1,N} & w_{2,N} & \cdots & w_{k,N} & \cdots & w_{K,N}
\end{array}\right]}_{k = 1:K \text{ natural parameters } \eta} \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\end{array}}\right\} n = 0:N \text{ features}$$

And then we can write the product as:

$$XW = \left[\begin{array}{ccccccc}
(x^{(1)})^T w_1 & (x^{(1)})^T w_2 & \ldots & (x^{(1)})^T w_k & \ldots & (x^{(1)})^T w_K \\
(x^{(2)})^T w_1 & (x^{(2)})^T w_2 & \ldots & (x^{(2)})^T w_k & \ldots & (x^{(2)})^T w_K \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
(x^{(m)})^T w_1 & (x^{(m)})^T w_2 & \ldots & (x^{(m)})^T w_k & \ldots & (x^{(m)})^T w_K \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
(x^{(M)})^T w_1 & (x^{(M)})^T w_2 & \ldots & (x^{(M)})^T w_k & \ldots & (x^{(M)})^T w_K
\end{array}\right] \in \mathbb{R}^{M\times K}$$

And the hypothesis matrix as:

$$h(XW) = \left[\begin{array}{ccccccc}
h(w_1^T x^{(1)}) & h(w_2^T x^{(1)}) & \ldots & h(w_k^T x^{(1)}) & \ldots & h(w_K^T x^{(1)}) \\
h(w_1^T x^{(2)}) & h(w_2^T x^{(2)}) & \ldots & h(w_k^T x^{(2)}) & \ldots & h(w_K^T x^{(2)}) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
h(w_1^T x^{(m)}) & h(w_2^T x^{(m)}) & \ldots & h(w_k^T x^{(m)}) & \ldots & h(w_K^T x^{(m)}) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
h(w_1^T x^{(M)}) & h(w_2^T x^{(M)}) & \ldots & h(w_k^T x^{(M)}) & \ldots & h(w_K^T x^{(M)})
\end{array}\right] \in \mathbb{R}^{M\times K}$$

5 Exponential Family Density or Mass Function

Most common distributions (e.g. Gaussian, Bernoulli, Binomial, Multinomial, Poisson, Gamma) are members of the exponential family i.e. they can be expressed in this form (see section 7). Two notable exceptions are the Student t and the Uniform distributions.
The overdispersed exponential family in canonical (or natural) form has the following general form for the density (if continuous) or mass (if discrete) function:

$$\boldsymbol{p(y;\theta,\tau) = b(y,\tau)\ \exp\left(\frac{\theta^T T(y) - a(\theta)}{c(\tau)}\right)}$$

Notation:

θ	natural (or canonical) parameter
τ	dispersion parameter (related to variance)
$T(y)$	response function (often $T(y) = y$)
$a(\theta)$	is the log partition function
$b(y,\tau)$	scaling constant (often $b = 1$)
$e^{-a(\theta)}$	plays the role of a normalisation constant ensuring the distribution sums / integrates over y to 1.

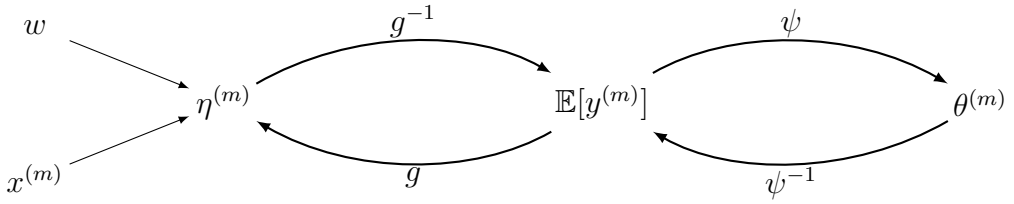
GLM terminology:

$\eta = w^T x$ is a linear function of the inputs

The link function g maps the mean to η i.e. $\eta = g(\mathbb{E}[y])$.

The mean (or response) function g^{-1} maps η to the distributions mean i.e. $\mathbb{E}[y] = g^{-1}(\eta)$.

θ is a function of η (i.e. $\theta = \theta(\eta)$) and the relationship is shown in the diagram below. The function ψ (mapping between θ and the mean) is known for each distribution. The link function g (mapping between η and the mean) is chosen. The only restrictions on choosing a link function is that it must be invertible (i.e. need g^{-1}) and the domain of g (or range of g^{-1}) should match the range of the distribution's mean. If we choose $g = \psi$ then $\eta = g(\psi^{-1}(\theta)) = \theta$. In this case g is the **canonical link function**.



If the canonical link function is chosen then the exponential family distribution can be written as:

$$\boldsymbol{p(y;\eta,\tau) = b(y,\tau)\ \exp\left(\frac{\eta^T T(y) - a(\eta)}{c(\tau)}\right)}$$

5.1 Expected Value and Variance

We can find expressions for the expected value and variance for the general case i.e. for the exponential family. These can then be applied to each individual case.

We'll use the fact that the probability density sums to 1 as well as the definitions of expected value and variance. I.e.

$$\int p(y;\theta,\tau)\,dy = 1 \quad , \quad \mathbb{E}[T(y)] = \int T(y)p(y;\theta,\tau)\,dy \quad , \quad \mathrm{Var}[T(y)] = \int (T(y) - \mathbb{E}[T(y)])^2\, p(y;\theta,\tau)\,dy$$

If the distribution is discrete the integrals can be replaced with summations for the same results.

5.1.1 Expected Value (and Hypothesis)

If θ is scalar i.e. $\theta \in \mathbb{R}^1$ then

$$\begin{aligned}
& \frac{d}{d\theta}\int p(y;\theta,\tau)\,dy && = && \frac{d}{d\theta}1 \\
\Rightarrow \quad & \int \frac{d}{d\theta}\,p(y;\theta,\tau)\,dy && = && 0 \\
\Rightarrow \quad & \int \frac{T(y) - a'(\theta)}{c(\tau)}\,p(y;\theta,\tau)\,dy && = && 0 \\
\Rightarrow \quad & \mathbb{E}[T(y)] - \frac{d}{d\theta}\,a(\theta) && = && 0 \qquad \left(\text{ since } \int a'(\theta)\,p(y;\theta,\tau)\,dy = a'(\theta)\int p(y;\theta,\tau)\,dy = a'(\theta)\ \right) \\
\Rightarrow \quad & && && \boldsymbol{\mathbb{E}[T(y)] = \frac{d}{d\theta}\,a(\theta)}
\end{aligned}$$

And if θ and $T(y)$ are vector valued i.e. $\theta \in \mathbb{R}^k$ and $T(y) \in \mathbb{R}^k$ and $\theta^T T(y) = \theta_1 T(y)_1 + \theta_2 T(y)_2 + \cdots + \theta_k T(y)_k + \cdots + \theta_K T(y)_K$ then for each k

$$\begin{aligned}
& \int \frac{\partial}{\partial\theta_k} p(y;\theta,\tau)\,dy && = && \frac{\partial}{\partial\theta_k}1 \\
\Rightarrow \quad & \int \frac{T(y)_k - \dfrac{\partial}{\partial\theta_k}\,a(\theta)}{c(\tau)}\,p(y;\theta,\tau)\,dy && = && 0 \\
\Rightarrow \quad & \mathbb{E}[T(y)_k] - \frac{\partial}{\partial\theta_k}\,a(\theta) && = && 0
\end{aligned}$$

Applying this to all θ_k and $T(y)_k$ we have

$$\boldsymbol{\mathbb{E}[T(y)] = \nabla_\theta\, a(\theta)}$$

5.1.2 Variance

For the variance we use the integral of the second derivative which leads to the second moment.

$$\begin{aligned}
\frac{d^2}{d\theta^2}\, p(y;\theta,\tau) &= \frac{d}{d\theta}\left(\frac{T(y)-a'(\theta)}{c(\tau)}\, p(y;\theta,\tau)\right) \\
&= \frac{-a''(\theta)}{c(\tau)}\, p(y;\theta,\tau) + \left(\frac{T(y)-a'(\theta)}{c(\tau)}\right)^2 p(y;\theta,\tau) \\
&= \frac{-a''(\theta)}{c(\tau)}\, p(y;\theta,\tau) + (\frac{1}{c(\tau)})^2 \left(T(y)-\mathbb{E}[T(y)]\right)^2 p(y;\theta,\tau)
\end{aligned}$$

On integrating we have

$$\int \frac{d^2}{d\theta^2}\, p(y;\theta,\tau)\, dy = \int \frac{-a''(\theta)}{c(\tau)}\, p(y;\theta,\tau)\, dy + \int \left(\frac{1}{c(\tau)}\right)^2 \left(T(y)-\mathbb{E}[T(y)]\right)^2 p(y;\theta,\tau)\, dy = 0$$

$$\Rightarrow \quad -a''(\theta) + \left(\frac{1}{c(\tau)}\right) \mathbb{V}\mathrm{ar}[T(y)] = 0$$

$$\Rightarrow \qquad \boldsymbol{\mathbb{V}\mathrm{ar}[T(y)] = a''(\theta)\, c(\tau)}$$

And if θ and $T(y)$ are vector valued i.e. $\theta \in \mathbb{R}^k$ and $T(y) \in \mathbb{R}^k$, then, following the same steps, we get

$$\boldsymbol{\mathbb{V}\mathrm{ar}[T(y)] = \nabla_\theta^2\, a(\theta)\, c(\tau)}$$

6 Maximum Likelihood

Maximum Likelihood provides a method of estimating the parameters. The broad idea is: Let's say there is some underlying "true" model that we are trying to find and that the training data has been drawn from. Of all the candidate models, we should select the model that is most likely to have created the data.

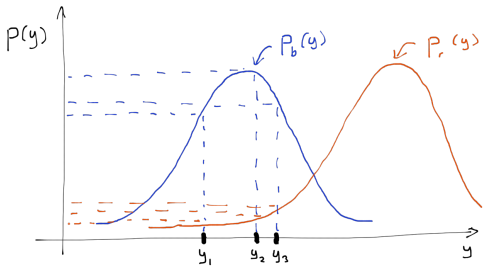
The distribution of y, given the input x, and parametrised by w, is $p(y|x;w)$. So an individual training example $(x^{(m)}, y^{(m)})$ has the probability density (or probability) $p(y^{(m)}|x^{(m)};w)$. The joint density, for a given design matrix X (i.e. the set of all $x^{(m)}$'s), and target vector $\vec{y}$ (i.e. the set of all $y^{(m)}$'s), and parametrised by w, is $p(\vec{y}|X;w)$. If this is instead viewed as a function of w (because we want to manipulate w to find the "right" model) it's called the **likelihood** function:

$$L(w) = L(w; X, \vec{y}) = p(\vec{y}|X;w)$$

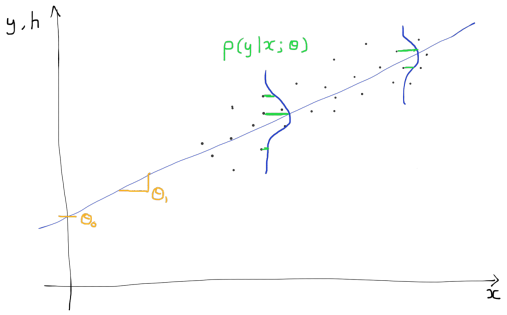
Now if the training data has been drawn independently the joint density can be written as the product of the individual densities:

$$L(w) = \prod_{m=1}^{M} p(y^{(m)}|x^{(m)};w)$$

The principle of Maximum Likelihood says we should choose w so as to make the data as highly probable as possible. I.e. we should choose the w that maximises $L(w)$. The images below show a representation of the principle.



Given the 3 data points $(y1, y2, y3)$, the blue model with distribution $p_b(y)$ ia a more "likely" model than the red because $p_b(y_1)p_b(y_2)p_b(y_3) > p_r(y_1)p_r(y_2)p_r(y_3)$.



For a model with single input x what are the parameters (w_0 the intercept and w_1 the gradient) that maximise the product $\prod_{m=1}^{M} p(y^{(m)}|x^{(m)};w)$.

We can also find the w that maximises any increasing function of $L(w)$ i.e. this will give the same result for w. And often the calculations are easier if, instead of finding the maximum likelihood, we find the maximum log likelihood:

$$\ell(w) = \log L(w) = \log \prod_{m=1}^{M} p(y^{(m)}|x^{(m)};w) = \sum_{m=1}^{M} \log\, p(y^{(m)}|x^{(m)};w)$$

6.1 Cost Function

Another variation is, instead of finding a maximum of the log likelihood function, to find a minimum of a cost function which is the negative log likelihood function (or a function with an equivalent argmin_w). It's also common to average the cost over the training examples (which also does not change argmin_w). I.e. the cost function to minimise is:

$$J(w) = -\tfrac{1}{M} \sum_{m=1}^{M} \log\, p(y^{(m)}|x^{(m)};w)$$

Applying this to the exponential family we have log likelihood function:

$$\begin{aligned}\ell(w) &= \sum_m \log\left(b(y^{(m)},\tau)\,\exp\left(\frac{\theta^{(m)})^T T(y^{(m)}) - a(\theta^{(m)})}{c(\tau)}\right)\right)\\ &= \sum_m \log\, b(y^{(m)},\tau) + \sum_m\left(\frac{\theta^{(m)})^T T(y^{(m)}) - a(\theta^{(m)})}{c(\tau)}\right)\end{aligned}$$

or cost function:

$$J(w) = -\tfrac{1}{M}\sum_m \log\, b(y^{(m)},\tau) - \tfrac{1}{M}\sum_m\left(\frac{\theta^{(m)})^T T(y^{(m)}) - a(\theta^{(m)})}{c(\tau)}\right)$$

And because $\hat{w} = \text{argmin}_w\, J(w)$ does not depend on b or c i.e.

$$\begin{aligned}\hat{w} &= \underset{w}{\text{argmin}}\ -\tfrac{1}{M}\sum_m \log\, b(y^{(m)},\tau) - \tfrac{1}{M}\sum_m\left(\frac{\theta^{(m)})^T T(y^{(m)}) - a(\theta^{(m)})}{c(\tau)}\right)\\ &= \underset{w}{\text{argmin}}\ -\tfrac{1}{M}\sum_m(\theta^{(m)})^T T(y^{(m)}) - a(\theta^{(m)})) \qquad \text{(since } b \text{ and } c \text{ are constants)}\end{aligned}$$

we can simplify the cost function to:

$$\boldsymbol{J(w) = \tfrac{1}{M}\sum_m a(\theta^{(m)}) - (\theta^{(m)})^T T(y^{(m)})}$$

6.2 Gradient of the Cost Function

To find the parameters (w) that minimise the cost (or maximise the likelihood) we typically use numeric gradient based methods (e.g. gradient descent) so we need the gradient. GLMs have the property of being convex so there is a single global solution which we are guaranteed to find.

If η is scalar then the gradient of each w_n is:

$$\begin{aligned}\frac{\partial}{\partial w_n} J(w) &= \frac{\partial}{\partial w_n}\ \tfrac{1}{M}\sum_m\ a(\theta^{(m)}) - (\theta^{(m)})^T T(y^{(m)})\\ &= \tfrac{1}{M}\sum_m\ \frac{\partial}{\partial\theta^{(m)}}\ \left(a(\theta^{(m)}) - (\theta^{(m)})^T T(y^{(m)})\right)\frac{\partial\theta^{(m)}}{\partial h_w(x^{(m)})}\frac{\partial h_w(x^{(m)})}{\partial\eta^{(m)}}\frac{\partial\eta^{(m)}}{\partial w_n}\\ &= \tfrac{1}{M}\sum_m\ \left(\frac{\partial}{\partial\theta^{(m)}}\,a(\theta^{(m)}) - T(y^{(m)})\right)\frac{\partial\theta^{(m)}}{\partial h_w^{(m)}}\frac{\partial h_w^{(m)}}{\partial\eta^{(m)}}\,x_n^{(m)}\\ &= \tfrac{1}{M}\sum_m\ \left(h_w^{(m)} - T(y^{(m)})\right)\,\theta'(h_w^{(m)})\,h'(\eta_w^{(m)})\,x_n^{(m)}\end{aligned}$$

$\theta'(h_w^{(m)})$ is determined by the distribution i.e. is the same for any choice of link function as $h_w^{(m)} = \mathbb{E}[T(y^{(m)})]$.

$h'_w(\eta^{(m)})$ is determined by the choice of link as the function h is the inverse of the link function.

So the gradient **vector** is:

$$\nabla_w J(w) = \tfrac{1}{M}\begin{bmatrix}\Sigma_m\,(h_w(x^{(m)}) - T(y^{(m)}))\,\theta'(h_w^{(m)})\,h'_w(\eta^{(m)})\,x_0^{(m)}\\ \Sigma_m\,(h_w(x^{(m)}) - T(y^{(m)}))\,\theta'(h_w^{(m)})\,h'_w(\eta^{(m)})\,x_1^{(m)}\\ \vdots\\ \Sigma_m\,(h_w(x^{(m)}) - T(y^{(m)}))\,\theta'(h_w^{(m)})\,h'_w(\eta^{(m)})\,x_N^{(m)}\end{bmatrix} = \tfrac{1}{M}X^T\left[(h(Xw) - T(y))\odot\theta'(h(Xw))\odot h'(Xw)\right]$$

$$\in \mathbb{R}^{(N+1)\times M}\times\mathbb{R}^M$$

$$\in \mathbb{R}^{N+1}$$

If η is vector valued the gradient of each w_n is:

$$\begin{aligned}
\frac{\partial}{\partial w_{n,k}} J(w) &= \tfrac{1}{M} \sum_m \left(h^{(m)}_{w_{*,k}} - T(y^{(m)}) \right) \frac{\partial \theta^{(m)}}{\partial h^{(m)}_{w_{*,k}}} \frac{\partial h^{(m)}_{w_{*,k}}}{\partial \eta^{(m)}_k} x^{(m)}_n \\
&= \tfrac{1}{M} \sum_m \left(h^{(m)}_{w_{*,k}} - T(y^{(m)}) \right) \theta'(h^{(m)}_{w_{*,k}})\, h'_{w_{*,k}}(\eta^{(m)})\, x^{(m)}_n
\end{aligned}$$

And the gradient **matrix** is:

$$\begin{aligned}
\nabla_w J(w) &= \tfrac{1}{M} \begin{bmatrix}
\dots & \Sigma_m (h_{w_{*,k}}(x^{(m)}) - T(y^{(m)}))\, \theta'(h^{(m)}_{w_{*,k}})\, h'_{w_{*,k}}(\eta^{(m)})\, x^{(m)}_0 & \dots \\
\dots & \Sigma_m (h_{w_{*,k}}(x^{(m)}) - T(y^{(m)}))\, \theta'(h^{(m)}_{w_{*,k}})\, h'_{w_{*,k}}(\eta^{(m)})\, x^{(m)}_1 & \dots \\
\vdots & \vdots & \vdots \\
\dots & \Sigma_m (h_{w_{*,k}}(x^{(m)}) - T(y^{(m)}))\, \theta'(h^{(m)}_{w_{*,k}})\, h'_{w_{*,k}}(\eta^{(m)})\, x^{(m)}_N & \dots
\end{bmatrix} \\
&= \tfrac{1}{M} X^T \left[(h(XW) - T(y)) \odot \theta'(h(XW)) \odot h'(XW) \right] \\
&\in \mathbb{R}^{(N+1)\times M} \times \mathbb{R}^{M \times K} \\
&\in \mathbb{R}^{(N+1)\times K}
\end{aligned}$$

Only for ordinary linear regression will we be able to solve $\nabla_w J(w) = 0$ analytically. Typically this will get solved using numeric methods.

6.3 Hessian of the Cost Function

Some numeric methods (e.g. Newton's method) make use of the 2^{nd} partial derivatives.

For scalar η the Hessian matrix $H \in \mathbb{R}^{(N+1)\times(N+1)}$ is defined so that each element $H_{i,j} = \dfrac{\partial^2}{\partial w_i \partial w_j} J(w)$. I.e.

$$\begin{aligned}
H_{i,j} &= \frac{\partial^2}{\partial w_i \partial w_j} J(w) \\
&= \frac{\partial}{\partial w_i} \tfrac{1}{M} \sum_m \left(h^{(m)}_w - T(y^{(m)}) \right) \theta'(h^{(m)}_w)\, h'_w(\eta^{(m)})\, x^{(m)}_j \\
&= \tfrac{1}{M} \sum_m \frac{\partial}{\partial \eta^{(m)}} \left(h_w(x^{(m)}) - T(y^{(m)}) \right) \frac{\partial}{\partial w_i} \eta^{(m)}(w) \,.\, x^{(m)}_j \\
&= \tfrac{1}{M} \sum_m \frac{\partial}{\partial \eta^{(m)}} \left(h_w(x^{(m)}) \right) \frac{\partial}{\partial w_i} w^T x^{(m)} \,.\, x^{(m)}_j \\
&= \tfrac{1}{M} \sum_m h'_w(x^{(m)})\, x^{(m)}_i\, x^{(m)}_j
\end{aligned}$$

A way to vectorise this matrix is to create a diagonal matrix (i.e. all entries zero except the diagonal) D such that $D^{(m)} \equiv D_{m,m} = h'_w(x^{(m)})$. Then we have

$$
\begin{aligned}
H &= \tfrac{1}{M}\begin{bmatrix}
\Sigma_m x_0^{(m)} D^{(m)} x_0^{(m)} & \Sigma_m x_0^{(m)} D^{(m)} x_1^{(m)} & \dots & \Sigma_m x_0^{(m)} D^{(m)} x_N^{(m)} \\
\Sigma_m x_1^{(m)} D^{(m)} x_0^{(m)} & \Sigma_m x_1^{(m)} D^{(m)} x_1^{(m)} & \dots & \Sigma_m x_1^{(m)} D^{(m)} x_N^{(m)} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_m x_N^{(m)} D^{(m)} x_0^{(m)} & \Sigma_m x_N^{(m)} D^{(m)} x_1^{(m)} & \dots & \Sigma_m x_N^{(m)} D^{(m)} x_N^{(m)}
\end{bmatrix} \\
&= \tfrac{1}{M}\begin{bmatrix}
\text{——} & x_0 & \text{——} \\
\text{——} & x_1 & \text{——} \\
 & \vdots & \\
\text{——} & x_N & \text{——}
\end{bmatrix}
\begin{bmatrix}
D^{(1)} & & & \\
 & D^{(1)} & & \\
 & & \ddots & \\
 & & & D^{(M)}
\end{bmatrix}
\begin{bmatrix}
| & | & & | \\
x_0 & x_1 & \dots & x_N \\
| & | & & |
\end{bmatrix} \\
&\qquad \in \mathbb{R}^{(N+1)\times M} \qquad\qquad \in \mathbb{R}^{M\times M} \qquad\qquad \in \mathbb{R}^{M\times(N+1)} \\
&= \tfrac{1}{M} \qquad X^T \qquad\qquad D \qquad\qquad X
\end{aligned}
$$

If η is vector valued then the Hessian is a 3-dimensional array $H \in \mathbb{R}^{(N+1)\times(N+1)\times K}$. Each "slice" of the array along the 3^{rd} dimension is $H_k = X^T D_k X$ where $D_k^{(m)} = h'_{w_k}(x^{(m)})$.

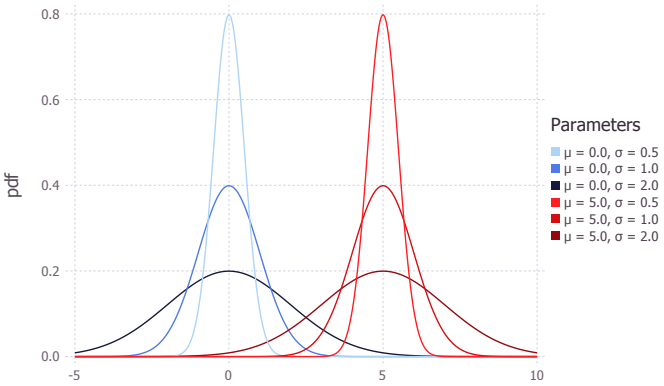
7 Common Distributions

7.1 Gaussian

$$\boldsymbol{y \mid x; w \sim \mathcal{N}(\mu, \sigma^2)}$$

Attributes:

Support:	$y \in \mathbb{R}$
Parameters:	μ (mean)
	$\sigma > 0$ (standard deviation)
Mean:	$\mathbb{E}(y \mid \mu, \sigma^2) = \mu$
Variance:	$\mathbb{V}\text{ar}(y \mid \mu, \sigma^2) = \sigma^2$



There are two versions of the parameterisation for GLMs - 1-parameter (with constant / known variance) and 2-parameter (with unknown variance).

7.1.1 1-Parameter

Mapping probability density function (pdf) to to exponential family general form:

$$\begin{aligned} p(y; \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2} \right) \\ &= \underbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\frac{1}{2}y^2}{\sigma^2} \right)}_{b(y,\tau)} \exp\left(\frac{\overbrace{\mu}^{\theta} \overbrace{y}^{T(y)} - \overbrace{\frac{1}{2}\mu^2}^{a(\theta)}}{\underbrace{\sigma^2}_{c(\tau)}} \right) \end{aligned}$$

$$T(y) = y \quad , \quad \theta = \mu \quad , \quad a(\theta) = \tfrac{1}{2}\mu^2 \quad , \quad b(y,\tau) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2} \right) \quad , \quad c(\tau) = \sigma^2$$

$$\mu = \theta \qquad = \tfrac{1}{2}\theta^2$$

Cost:

$$J(w) = \tfrac{1}{M} \sum_m a(\theta^{(m)}) - (\theta^{(m)})^T T(y^{(m)}) = \tfrac{1}{M} \sum_m \tfrac{1}{2}(h_w(x^{(m)})^2 - h_w(x^{(m)}) y^{(m)}$$

And the following is more numerically convenient and equivalent (same argmin_w and gradient):

$$J(w) = \tfrac{1}{2M} \sum_m (h_w(x^{(m)}) - y^{(m)})^2$$

Gradient:

$$\frac{\partial}{\partial w_n} J(w) = \tfrac{1}{M} \sum_m \left(h_w(x^{(m)}) - T(y^{(m)}) \right) x_n^{(m)} = \tfrac{1}{M} \sum_m \left(h_w(x^{(m)}) - y^{(m)} \right) x_n^{(m)}$$

Cost: $J(w) = \frac{1}{M}\sum_m a(\theta^{(m)}) - (\theta^{(m)})^T T(y^{(m)}) = \frac{1}{M}\sum_m \frac{1}{2}(h_w(x^{(m)})^2 - h_w(x^{(m)})y^{(m)}$

Gradient: $\mathbb{E}(y \mid x; w) = \mu = a'(\eta) = g(\eta) = \eta = w^T x$

Hessian: $\mathbb{V}\mathrm{ar}(y \mid \mu, \sigma^2) = a''(\eta)\, c(\tau) = \sigma^2$

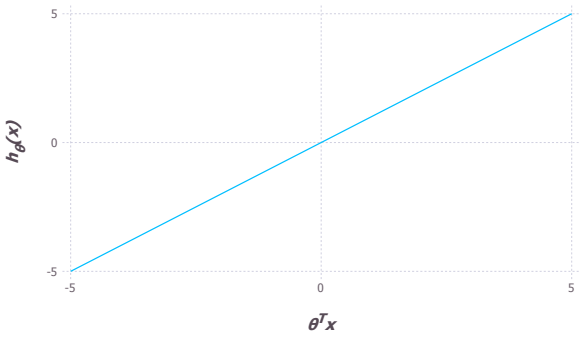
Link: $g^{-1}(\mu) = \eta = \mu$ (i.e. link function is the identity function)

Expected Value: $\mathbb{E}(y \mid x; w) = \mu = a'(\eta) = g(\eta) = \eta = w^T x$

Variance: $\mathbb{V}\mathrm{ar}(y \mid \mu, \sigma^2) = a''(\eta)\, c(\tau) = \sigma^2$

So the hypothesis is the linear function:

$$\boldsymbol{h_w(x) = w^T x}$$



7.1.2 2-Parameter

Mapping probability density function (pdf) to to exponential family general form:

$$\begin{aligned}
p(y;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2\sigma^2} + \frac{2\mu y}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} - \log\sigma\right) \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(\frac{\mu y}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} - \log\sigma\right) \\
&= \underbrace{\frac{1}{\sqrt{2\pi}}}_{b(y,\tau)}\exp\left(\underbrace{\begin{bmatrix}\frac{\mu}{\sigma^2} & \frac{-1}{2\sigma^2}\end{bmatrix}}_{\eta^T}\underbrace{\begin{bmatrix}y\\ y^2\end{bmatrix}}_{T(y)} - \underbrace{\left(\frac{\mu^2}{2\sigma^2} + \log\sigma\right)}_{a(\eta)}\right)
\end{aligned}$$

$$T(y) = \begin{bmatrix} y \\ y^2 \end{bmatrix} \quad , \quad \eta = \begin{bmatrix} \frac{\mu}{\sigma^2} \\ \frac{-1}{2\sigma^2} \end{bmatrix} \quad , \quad a(\eta) = \frac{\mu^2}{2\sigma^2} + \log\sigma \quad , \quad b(y,\tau) = \frac{1}{\sqrt{2\pi}} \quad , \quad c(\tau) = 1$$

$$= \frac{-\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2)$$

$$\mathbb{E}[y] = \frac{\partial}{\partial\eta_1}a(\eta) = \frac{-2\eta_1}{4\eta_2} = \eta_1\frac{-1}{2\eta_2} = \frac{\mu}{\sigma^2}\sigma^2 = \mu$$

$$\mathbb{E}[y^2] = \frac{\partial}{\partial\eta_2}a(\eta) = \frac{\eta_1^2}{4\eta_2^2} - \frac{1}{2\eta_2} = \frac{\mu^2}{\sigma^2}\frac{4\sigma^4}{4} + \sigma^2 = \mu^2 + \sigma^2$$

Hypothesis functions:

$$h_1 = \frac{-\eta_1}{2\eta_2} = -\frac{1}{2}\frac{w_1^T x}{w_2^T x}$$

$$h_2 = \frac{\eta_1^2}{4\eta_2^2} - \frac{1}{2\eta_2} = \frac{\eta_1^2 - 2\eta_2}{4\eta_2^2} = \frac{(w_1^T x)^2 - 2w_2^T x}{4(w_2^T x)^2}$$

TODO - INCOMPLETE

Non Canonical

Log link:

$$\boldsymbol{y \mid x; w \sim \mathcal{N}(\log(\mu), \sigma^2)}$$

$$g(\mathbb{E}[y|x; w]) = \log(\eta)$$

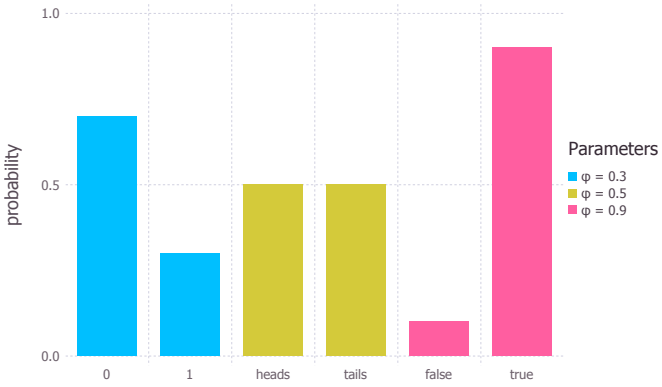
TODO - INCOMPLETE

7.2 Bernoulli

$$\boldsymbol{y \mid x; w \sim \text{Bernoulli}(\phi)}$$

Attributes:

Support:	$y \in \{0, 1\}$
Parameter:	$\phi \in [0, 1]$ (mean)
Mean:	$\mathbb{E}(y \mid \phi) = \phi$
Variance:	$\mathbb{V}\text{ar}(y \mid \phi) = \phi(1 - \phi)$



Mapping probability mass function (pmf) to exponential family:

$$\begin{aligned}
p(y; \phi) &= \begin{cases} \phi & \text{for } y = 1 \\ 1 - \phi & \text{for } y = 0 \end{cases} \\
&= \phi^y (1 - \phi)^{1-y} \\
&= \exp(y \log \phi + (1 - y) \log(1 - \phi)) \\
&= \exp(\underbrace{\log\left(\frac{\phi}{1-\phi}\right)}_{\theta} \underbrace{y}_{T(y)} + \underbrace{\log(1 - \phi))}_{a(\theta)}
\end{aligned}$$

$$T(y) = y \quad , \quad \theta = \log\left(\frac{\phi}{1-\phi}\right) \quad , \quad a(\theta) = -\log(1 - \phi) \quad , \quad b(y, \tau) = 1 \quad , \quad c(\tau) = 1$$

$$\phi = \mathrm{e}^\theta / (1 + \mathrm{e}^\theta) \qquad = \quad \log(1 + \mathrm{e}^\theta)$$

Canonical Link ($\psi = g,\ \eta = \theta$)

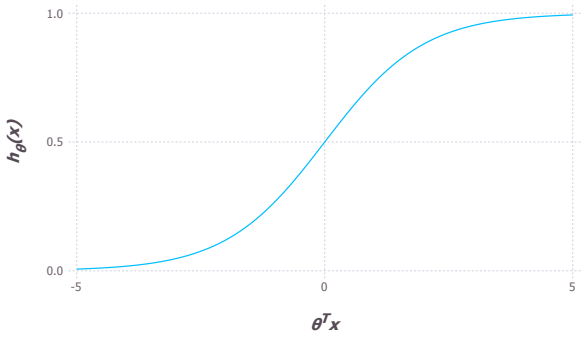
Link: $\eta = g^{-1}(\phi) = \log\left(\frac{\phi}{1-\phi}\right)$ (link function is the logit function)

Expected Value: $\phi = g(\eta) = a'(\eta) = \frac{\mathrm{e}^\eta}{1 + \mathrm{e}^\eta} = \frac{1}{1 + \mathrm{e}^{-\eta}} = \frac{1}{1 + \mathrm{e}^{-w^T x}}$ (mean function is the logistic function)

Variance: $\mathbb{V}\text{ar}(y \mid \phi) = \phi(1 - \phi) = a''(\eta)\, c(\tau) = \frac{e^\eta}{(1 + e^\eta)^2}$

So the hypothesis is the logistic function:

$$\boldsymbol{h_w(x) = \frac{1}{1 + \mathrm{e}^{-w^T x}}}$$

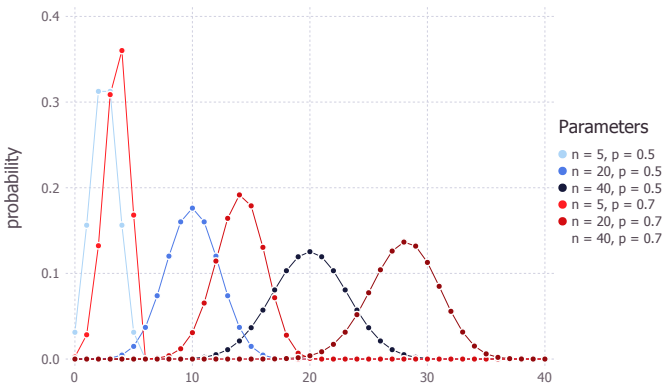


7.3 Binomial

$$\boldsymbol{y \mid x; w \sim \text{Binomial}(n, p)}$$

Attributes:

Support:	$y \in \{0, ..., n\}$
Parameters:	$n \in \mathbb{Z}^+$ (# of trials)
	$p \in [0, 1]$ (success prob.)
Mean:	$\mathbb{E}(y \mid n, p) = np$
Variance:	$\mathbb{V}\text{ar}(y \mid n, p) = np(1-p)$



Mapping probability mass function (pmf) to exponential family general form:

$$\begin{aligned}
p(y; n, p) &= \binom{n}{y} p^y (1-p)^{(n-y)} \\
&= \binom{n}{y} \exp(y \log(p) + (n-y) \log(1-p)) \\
&= \underbrace{\binom{n}{y}}_{b(y,\tau)} \exp(\underbrace{\log\left(\frac{p}{1-p}\right)}_{\eta} \underbrace{y}_{T(y)} + \underbrace{n \log(1-p)}_{a(\eta)})
\end{aligned}$$

$$T(y) = y \quad , \quad \eta = \log\left(\frac{p}{1-p}\right) \quad , \quad a(\eta) = -n \log(1-p) \quad , \quad b(y,\tau) = \binom{n}{y} \quad , \quad c(\tau) = 1$$

$$= n \log(1 + \mathrm{e}^{\eta})$$

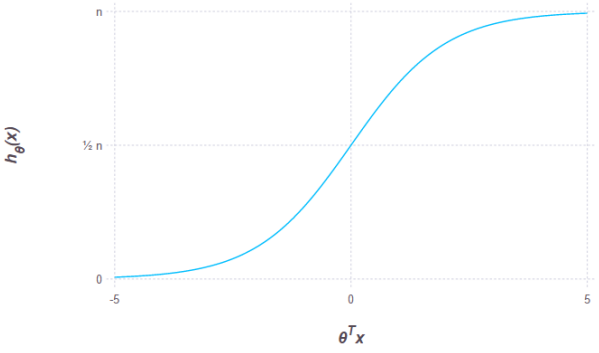
Link: $g^{-1}(np) = \log\left(\frac{p}{1-p}\right) = \eta$ (i.e. link function is the logit function)

Expected Value: $\mathbb{E}(y \mid x; w) = np = a'(\eta) = g(\eta) = n \frac{\mathrm{e}^{\eta}}{1+\mathrm{e}^{\eta}} = \frac{n}{1+\mathrm{e}^{-\eta}} = \frac{n}{1+\mathrm{e}^{-w^T x}}$

Variance: $\mathbb{V}\text{ar}(y \mid n, p) = a''(\eta)\, c(\tau) = n \frac{e^{\eta}}{(1+e^{\eta})^2} = np(1-p)$

So the hypothesis is the sigmoid / logistic function multiplied by the number of trials (note that we need to know the number of trials):

$$\boldsymbol{h_w(x) = \frac{n}{1 + \mathrm{e}^{-w^T x}}}$$

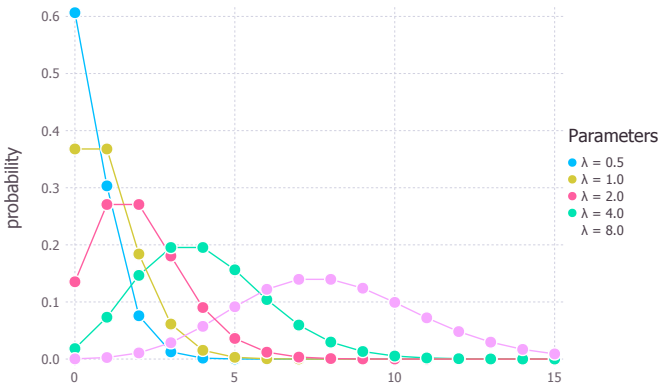


7.4 Poisson

$$\boldsymbol{y \mid x; w \sim \text{Poisson}(\lambda)}$$

Attributes:

Support:	$y \in \{0, 1, 2, \dots\}$
Parameter:	$\lambda \in \{0, 1, 2, \dots\}$ (mean & variance)
Mean:	$\mathbb{E}(y \mid \lambda) = \lambda$
Variance:	$\mathbb{V}\text{ar}(y \mid \lambda) = \lambda$



Mapping probability mass function (pmf) to exponential family general form:

$$\begin{aligned} p(y;\lambda) &= \frac{\lambda^y}{y!}\mathrm{e}^{-\lambda} \\ &= \underbrace{\frac{1}{y!}}_{b(y,\tau)} \exp(\underbrace{\log(\lambda)}_{\eta}\ \underbrace{y}_{T(y)}\ -\ \underbrace{\lambda}_{a(\eta)}\ ) \end{aligned}$$

$$T(y) = y \quad , \quad \eta = \log(\lambda) \quad , \quad a(\eta) = \lambda \quad , \quad b(y,\tau) = \frac{1}{y!} \quad , \quad c(\tau) = 1$$

$$= \mathrm{e}^{\eta}$$

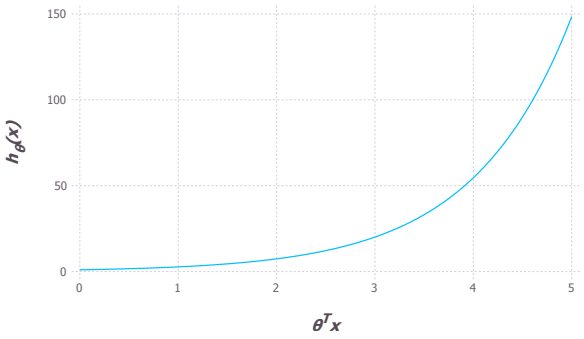
Link: $g^{-1}(\lambda) = \eta = \log \lambda$ (i.e. link function is the log function)

Expected Value: $\mathbb{E}(y \mid x; w) = \lambda = a'(\eta) = g(\eta) = \mathrm{e}^{\eta} = \mathrm{e}^{w^T x}$

Variance: $\mathbb{V}\text{ar}(y \mid \mu, \sigma^2) = a''(\eta)\, c(\tau) = \mathrm{e}^{\eta} = \lambda$

So the hypothesis is the exponential function:

$$\boldsymbol{h_w(x) = \mathrm{e}^{w^T x}}$$



7.5 Exponential

$$\boldsymbol{y \mid x; w \sim \text{Exponential}(\lambda)}$$

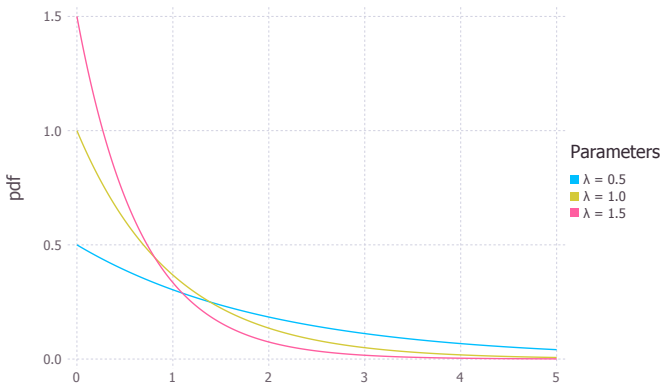
Attributes:

Support: $y \in [0, \infty)$

Parameter: $\lambda > 0$ (rate)

Mean: $\mathbb{E}(y \mid \lambda) = \lambda^{-1}$

Variance: $\mathbb{V}\text{ar}(y \mid \lambda) = \lambda^{-2}$



Mapping probability mass function (pmf) to exponential family general form:

$$\begin{aligned} p(y;\lambda) &= \lambda \mathrm{e}^{-\lambda y} \\ &= \exp(\log\lambda - \lambda y) \\ &= \exp(\underbrace{-\lambda}_{\eta}\ \underbrace{y}_{T(y)}\ +\ \underbrace{\log\lambda}_{a(\eta)}) \end{aligned}$$

$$T(y) = y \quad , \quad \eta = -\lambda \quad , \quad a(\eta) = -\log\lambda = -\log(-\eta) \quad , \quad b(y,\tau) = 1 \quad , \quad c(\tau) = 1$$

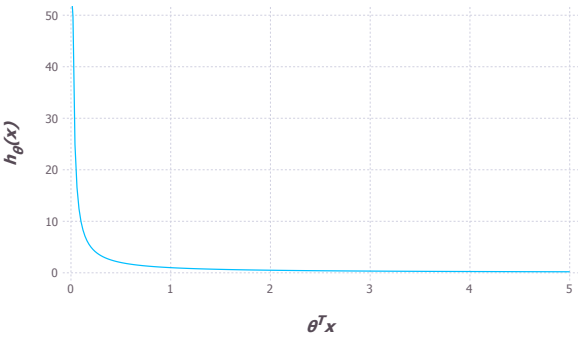
Link: $g^{-1}(\lambda^{-1}) = \eta = -\lambda = -\dfrac{1}{\lambda^{-1}}$ (i.e. link function is the negative inverse function)

Expected Value: $\mathbb{E}(y \mid x; w) = \lambda^{-1} = a'(\eta) = g(\eta) = -\eta^{-1} = -\dfrac{1}{w^T x}$

Variance: $\mathbb{V}\text{ar}(y \mid \mu, \sigma^2) = a''(\eta)\, c(\tau) = \eta^{-2} = \lambda^{-2}$

So the hypothesis is the inverse function (we can drop the negative by setting $\hat{w} = -w$):

$$\boldsymbol{h_w(x) = \frac{1}{w^T x}}$$

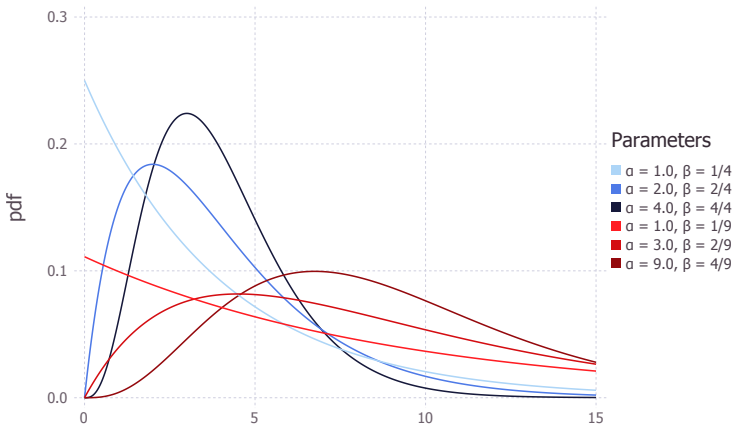


7.6 Gamma

$$\boldsymbol{y \mid x; w \sim \Gamma(\alpha, \beta)}$$

Attributes:

Support:	$y \in (0, \infty)$
Parameters:	$\alpha > 0$ (shape)
	$\beta > 0$ (rate)
Mean:	$\mathbb{E}(y \mid \alpha, \beta) = \dfrac{\alpha}{\beta}$
Variance:	$\mathbb{V}\text{ar}(y \mid \alpha, \beta) = \dfrac{\alpha}{\beta^2}$



There are multiple ways to parametrise the Gamma Distribution. What makes this challenging is that the mean depends on both parameters (α&β) and it is not straightforward to re-parametrise to create a mean parameter and a dispersion parameter.

Mapping probability density function (pdf) to exponential family general form (method 1):

$$\begin{aligned}
p(y; \alpha, \beta) &= \frac{\beta^\alpha \, y^{\alpha-1} \, \mathrm{e}^{-y\beta}}{\Gamma(\alpha)} \\
&= \exp(\alpha \, \log \, \beta + (\alpha - 1) \, \log \, y - y\beta - \log \Gamma(\alpha)) \\
&= \frac{1}{y} \exp(-\beta \, y + \alpha \, \log \, y + \alpha \, \log \, \beta - \log \Gamma(\alpha)) \\
&= \underbrace{\frac{1}{y}}_{b(y,\tau)} \exp\Bigg(\underbrace{[-\beta \;\; \alpha]}_{\eta^T} \underbrace{\begin{bmatrix} y \\ \log \, y \end{bmatrix}}_{T(y)} + \underbrace{\alpha \, \log \, \beta - \log \Gamma(\alpha)}_{a(\eta)} \Bigg)
\end{aligned}$$

$$T(y) = \begin{bmatrix} y \\ \log \, y \end{bmatrix} \quad , \quad \eta = \begin{bmatrix} -\beta \\ \alpha \end{bmatrix} \quad , \quad a(\eta) = \log \Gamma(\alpha) - \alpha \, \log \, \beta \quad , \quad b(y, \tau) = \frac{1}{y} \quad , \quad c(\tau) = 1$$

Expected Value: $\begin{bmatrix} \mathbb{E}(y \mid x; w) \\ \mathbb{E}(\log \, y \mid x; w) \end{bmatrix} = \nabla_\eta \, a(\eta) = \begin{bmatrix} \dfrac{\alpha}{\beta} \\ \psi(\alpha) - \log \beta \end{bmatrix} = \begin{bmatrix} \dfrac{\eta_2}{-\eta_1} \\ \psi(\eta_2) - \log(-\eta_1) \end{bmatrix}$

where ψ (the digamma function) is the logarithmic derivative of the gamma function

i.e. $\psi(\alpha) = \dfrac{d}{d\alpha} \log \Gamma(\alpha) = \dfrac{\Gamma'(\alpha)}{\Gamma(\alpha)}$

Variance:
$$\begin{bmatrix} \mathbb{V}\text{ar}(y \mid x; w) \\ \mathbb{V}\text{ar}(\log\, y \mid x; w) \end{bmatrix} = \nabla^2_\eta\, a(\eta)\, c(\tau) = \begin{bmatrix} \frac{\alpha}{\beta^2} \\ \psi_1(\alpha) \end{bmatrix} = \begin{bmatrix} \frac{\eta_2}{\eta_1^2} \\ \psi_1(\eta_2) \end{bmatrix}$$

where ψ_1 (the trigamma function) is the second polygamma function

i.e. $\psi_1(\alpha) = \dfrac{d^2}{d\alpha^2} \log \Gamma(\alpha) = \dfrac{d}{d\alpha}\psi(\alpha)$

The difficulty with method 1 is that there are two parameters to learn if we want to predict the mean. An alternative is to break the two parameter dependancy on the mean by creating a location parameter and dispersion parameter. This exponential family mapping is a bit more complicated but we end up with a single parameter to learn.

Mapping probability density function (pdf) to exponential family general form (method 2):

$$\begin{aligned} p(y; \alpha, \beta) &= \frac{\beta^\alpha\, y^{\alpha-1}\, \mathrm{e}^{-y\beta}}{\Gamma(\alpha)} = \frac{y^{\alpha-1}}{\Gamma(\alpha)} \exp(\alpha \log \beta - \beta y) \\ &= \frac{y^{\alpha-1}}{\Gamma(\alpha)} \exp\left(\frac{\frac{\beta}{\alpha} y - \log \beta}{-\frac{1}{\alpha}} \right) \\ &= \frac{y^{\alpha-1}}{\Gamma(\alpha)} \exp\left(\frac{\frac{\beta}{\alpha} y - \log \frac{\beta}{\alpha}}{-\frac{1}{\alpha}} + \frac{\log \frac{1}{\alpha}}{\frac{1}{\alpha}} \right) \qquad (\text{using } \log \beta = \log \frac{\beta}{\alpha} - \log \frac{1}{\alpha}) \\ &= \underbrace{\frac{y^{\alpha-1}(\frac{1}{\alpha})^\alpha}{\Gamma(\alpha)}}_{b(y,\tau)} \exp\left(\frac{\overbrace{-\frac{\beta}{\alpha}}^{\eta}\ \overbrace{y}^{T(y)} + \overbrace{\log \frac{\beta}{\alpha}}^{a(\eta)}}{\underbrace{\frac{1}{\alpha}}_{c(\tau)}} \right) \end{aligned}$$

$$T(y) = y \quad , \quad \eta = -\frac{\beta}{\alpha} \quad , \quad a(\eta) = -\log \frac{\beta}{\alpha} = -\log(-\eta) \quad , \quad b(y,\tau) = \frac{y^{\alpha-1}(\frac{1}{\alpha})^\alpha}{\Gamma(\alpha)} \quad , \quad c(\tau) = \frac{1}{\alpha}$$

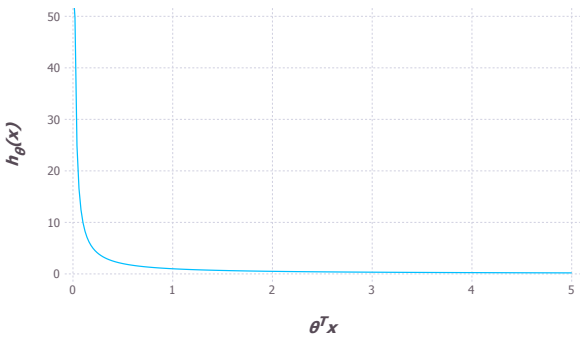
Link: $g^{-1}(\frac{\alpha}{\beta}) = \eta = -\frac{\beta}{\alpha} = -\frac{1}{\frac{\alpha}{\beta}}$ (i.e. link function is the inverse function)

Expected Value: $\mathbb{E}(y \mid x; w) = \frac{\alpha}{\beta} = a'(\eta) = g(\eta) = -\frac{1}{\eta} = -\frac{1}{w^T x}$

Variance: $\mathbb{V}\text{ar}(y \mid \mu, \sigma^2) = a''(\eta)\, c(\tau) = \frac{1}{\eta^2}.\frac{1}{\alpha} = \frac{\alpha^2}{\beta^2}\frac{1}{\alpha} = \frac{\alpha}{\beta^2}$

So the hypothesis is the inverse function (we can drop the negative by setting $\hat{w} = -w$):

$$\boldsymbol{h_w(x) = \frac{1}{w^T x}}$$



7.7 Negative Binomial

TODO - INCOMPLETE

7.8 Softmax

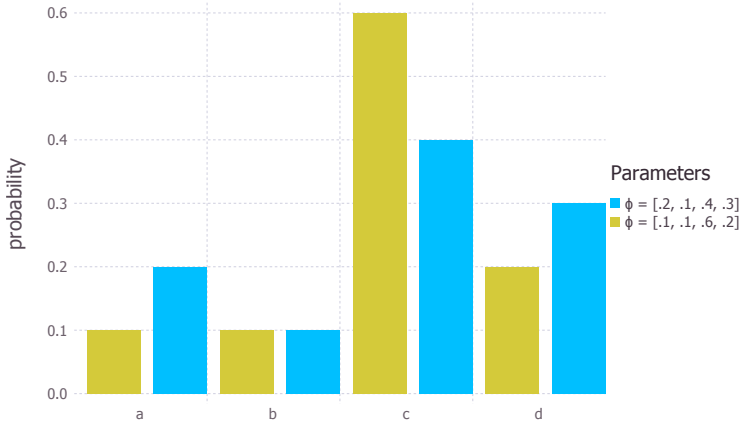
Softmax regression is used for a classification problem where $y \in \{1, 2, \ldots, K\}$. The underlying distribution can be either Categorical (an extension of Bernoulli) or Multinomial (an extension of Binomial).

An example is an image recognition problem where we need to classify handwritten digits as one of $\{0, 1, 2, 3, \ldots, 8, 9\}$ e.g. the MNIST dataset.

The distribution is parametrised with $K - 1$ parameters $\phi = [\phi_1, \phi_2, \ldots, \phi_{K-1}]$ where $\phi_k = P(y = k; \phi)$, and $\phi_K = P(y = K; \phi) = 1 - \sum_{k=1}^{K-1} \phi_k$.

Attributes:

Support:	$y \in \{1, 2, \ldots, K\}$
Parameter:	$\phi_k \in [0, 1]$, $k \in \{1, 2, \ldots, K\}$ (mean for each k)
Mean:	$\mathbb{E}[y = k] = \phi_k$
Variance:	$\mathbb{V}\text{ar}[y = k] = \phi_k(1 - \phi_k)$



Each of the parameter vectors sum to 1.

Notation:

Define $T(y) \in \mathbb{R}^{K-1}$ as a vector with 1 on the element that is equal to y and 0 on every other element e.g.

$$T(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad T(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad T(K-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad T(K) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$T(y)_k$ denotes the k-th element of vector $T(y)$. E.g. if $y = 2$ then $T(y)_2 = 1$.

Indicator function is defined as $\mathbf{1}\{\text{True}\} = 1$, $\mathbf{1}\{\text{False}\} = 0$. E.g. $T(y)_k = \mathbf{1}\{y = k\}$ i.e. $T(y)_k = 1$ if $y = k$.

Mapping probability mass function (pmf) to exponential family general form:

$$\begin{aligned}
p(y; \phi) &= \phi_1^{\mathbf{1}\{y=1\}} \phi_2^{\mathbf{1}\{y=2\}} \ldots \phi_{K-1}^{\mathbf{1}\{y=K-1\}} \phi_K^{\mathbf{1}\{y=K\}} \\
&= \phi_1^{\mathbf{1}\{y=1\}} \phi_2^{\mathbf{1}\{y=2\}} \ldots \phi_{K-1}^{\mathbf{1}\{y=K-1\}} \phi_K^{1-\sum_{k=1}^{K-1} \mathbf{1}\{y=k\}} \\
&= \phi_1^{T(y)_1} \phi_2^{T(y)_2} \ldots \phi_{K-1}^{T(y)_{K-1}} \phi_K^{1-\sum_{k=1}^{K-1} T(y)_k} \\
&= \exp(\, T(y)_1 \log \phi_1 + T(y)_2 \log \phi_2 + \cdots + T(y)_{K-1} \log \phi_{K-1} + 1 - \textstyle\sum_{k=1}^{K-1} T(y)_k \log \phi_K \,) \\
&= \exp(\, T(y)_1 \log \frac{\phi_1}{\phi_K} + T(y)_2 \log \frac{\phi_2}{\phi_K} + \cdots + T(y)_{K-1} \log \frac{\phi_{K-1}}{\phi_K} + \log \phi_K \,) \\
&= \exp(\eta^T T(y) + \log \phi_K)
\end{aligned}$$

$$T(y) = \begin{bmatrix} T(y)_1 \\ T(y)_2 \\ \vdots \\ T(y)_{K-1} \end{bmatrix} \quad , \quad \eta = \begin{bmatrix} \log \frac{\phi_1}{\phi_K} \\ \log \frac{\phi_2}{\phi_K} \\ \vdots \\ \log \frac{\phi_{K-1}}{\phi_K} \end{bmatrix} \quad , \quad a(\eta) = -\log \phi_K \qquad , \quad b(y,\tau) = 1 \quad , \quad c(\tau) = 1$$

$$= -\log(1 - \textstyle\sum_{k=1}^{K-1} \phi_k)$$

Invert the natural parameter to find the response function:

$$\begin{aligned}
& e^{\eta_k} && = && \frac{\phi_k}{\phi_K} \\
\Rightarrow \quad & \phi_K e^{\eta_k} && = && \phi_k \\
\Rightarrow \quad & \phi_K \textstyle\sum_{k=1}^{K} e^{\eta_k} && = && \textstyle\sum_{k=1}^{K} \phi_k \;=\; 1 \qquad \text{(Above line is true for each } k \Rightarrow \text{true for the sum of all } k\text{'s.)} \\
\Rightarrow \quad & \phi_K && = && \frac{1}{\sum_{k=1}^{K} e^{\eta_k}} \\
\Rightarrow \quad & \phi_k && = && \frac{e^{\eta_k}}{\sum_{k=1}^{K} e^{\eta_k}} \qquad \text{(This is the softmax function.)}
\end{aligned}$$

Link: for $k = 1, \ldots, K$ $\quad \eta_k = \log \frac{\phi_k}{\phi_K}$

Expected Value: $\mathbb{E}[T(y) \,|\, x; w] = \phi = a'(\eta) = g(\eta) = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{1+e^{-\eta}} = \frac{1}{1+e^{-w^T x}}$

Variance: $\mathbb{V}\mathrm{ar}(y \,|\, \mu, \sigma^2) = a''(\eta)\, c(\tau) = \frac{e^{\eta}}{(1+e^{\eta})^2} = \phi(1-\phi)$

8 Constructing a GLM

The general method for constructing a GLM is as follows:

1. Choose an appropriate distribution.

   I.e. Let the exponential family (η) distribution be a relevant distribution to the particular problem.

2. Find ws by applying principle of maximum likelihood.

   I.e. Find w to maximise the likelihood function $L(w) = p(y|X; w)$ or the log likelihood function $\ell(w) = \log p(y|X; w)$.

8.1 Choosing an Appropriate Distribution

Note that the key distribution is $P(y|X)$. This is the distribution we are making an assumption about i.e. that $P(y|X) = \text{Exponential Family}(\eta)$. And we need to choose, from this family of distributions, the most appropriate for the given problem.

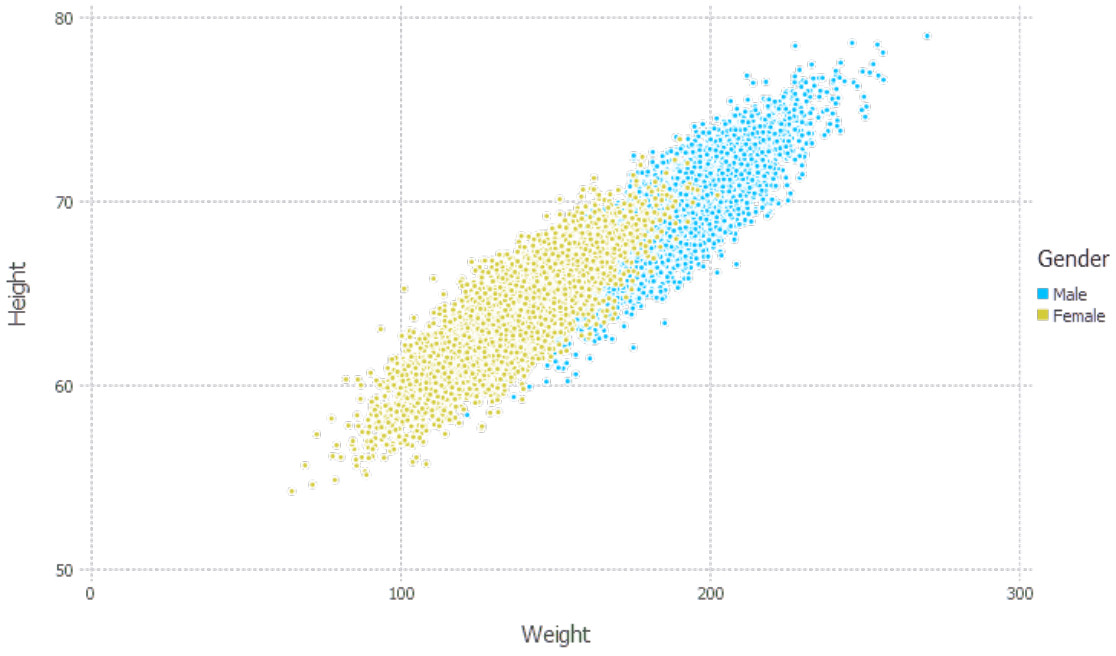
Below are a few of the probability distributions present:

$P(X)$	distribution of the input features
$P(y)$	distribution of the target
$P(y\|X)$	distribution of the target y given the input features x
$P(y\|X)P(X)$	joint distribution, produces the pairs of training examples $x^{(m)}, y^{(m)}$

What is the conditional probability $P(y|x)$. And how does it compare to $P(y)$ the distribution of y. Let's demonstrate with examples.

8.1.1 Example - Heights, Weights and Gender

To explore, we're going to use this data from John Myles White's superb book Machine Learning for Hackers. It is a set of heights, weights and genders of $10,000$ adults and looks like this:



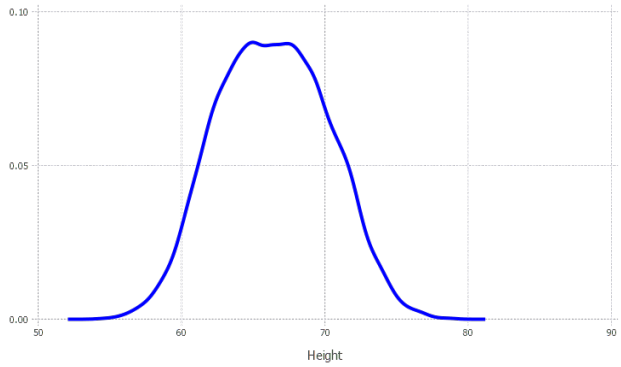
Example 1 - Real valued target, categorical feature variable

Say we want to predict a person's height, given their gender i.e. given that they are either male or female. In this setup we'd have y is height and x is gender.

The figure on the left below shows a density plot of all the heights. This is the distribution $P(y)$. It also looks approximately normally distributed with an odd looking peak. So if we only had this information we would be able to make predictions of someone's height. But can we do better by looking at other variables in the data set?

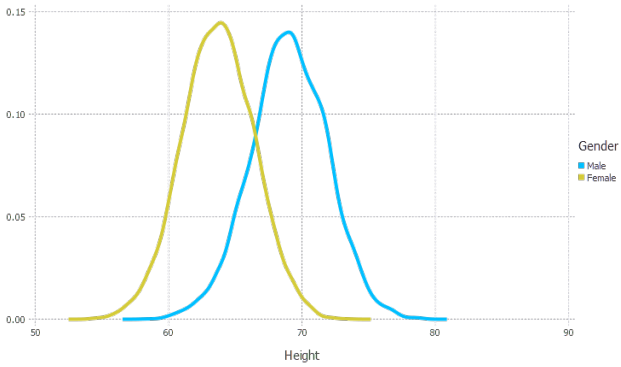
If we split the data by the qualitative variable (gender) we see a hidden pattern that was missing in the population view i.e. that the population is made up of two overlapping bell curves. And each of these bell curves looks approximately normally distributed. I.e. $height \,|\, male \sim \mathcal{N}(\mu_m, \sigma_m^2)$ and $height \,|\, female \sim \mathcal{N}(\mu_f, \sigma_f^2)$.

So it looks like, if we want to use gender as an input feature to predict a person's height, the assumption that $y \,|\, x \sim \mathcal{N}(\mu, \sigma^2)$ is a reasonable one.



Population

$y \sim \mathcal{N}$



Show Gender

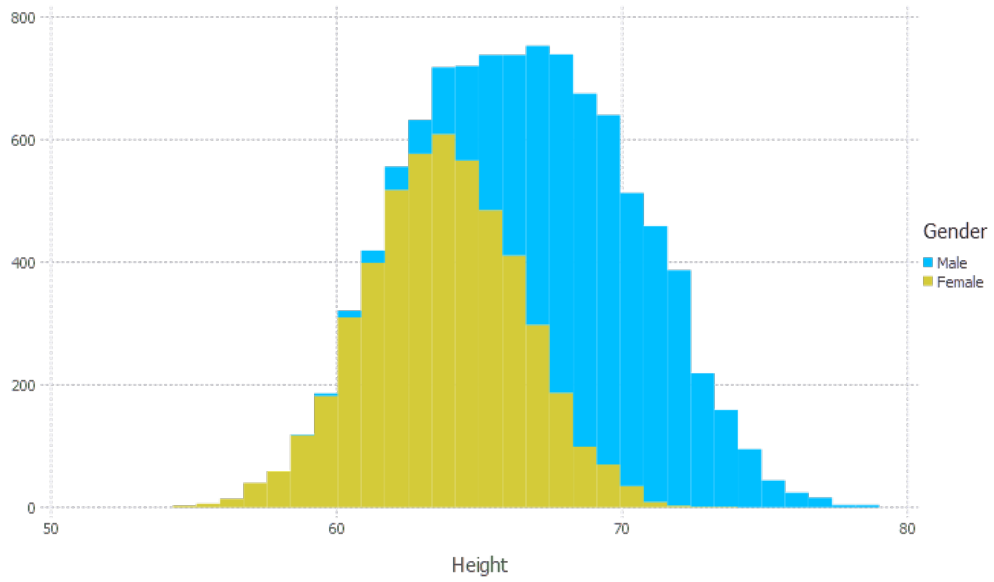
$y \,|\, gender \sim \mathcal{N}$

Example 2 - Categorical target, real valued feature variable

If we'd like to predict a person's gender using their height we have y is gender and x is height.

The figure below shows a stacked histogram of the heights with the gender proportions shown. It's clear that for lower heights it's more likely that a person is a woman, and for higher heights it's more likely that a person is a man. Taking the frequencies as probabilities we'd have:
$P(y = female \,|\, height < 60) = 95\%$,
$P(y = male \,|\, height > 70) = 85\%$,
$P(y = female \,|\, height = 67) = 50\%$, i.e. more broadly
$P(y \,|\, gender)$ is bernoulli distributed.

So it looks like, if we want to use height as an input feature to predict a person's gender, the assumption that $y \,|\, x \sim \mathcal{B}ernoulli(\phi)$ is a reasonable one.

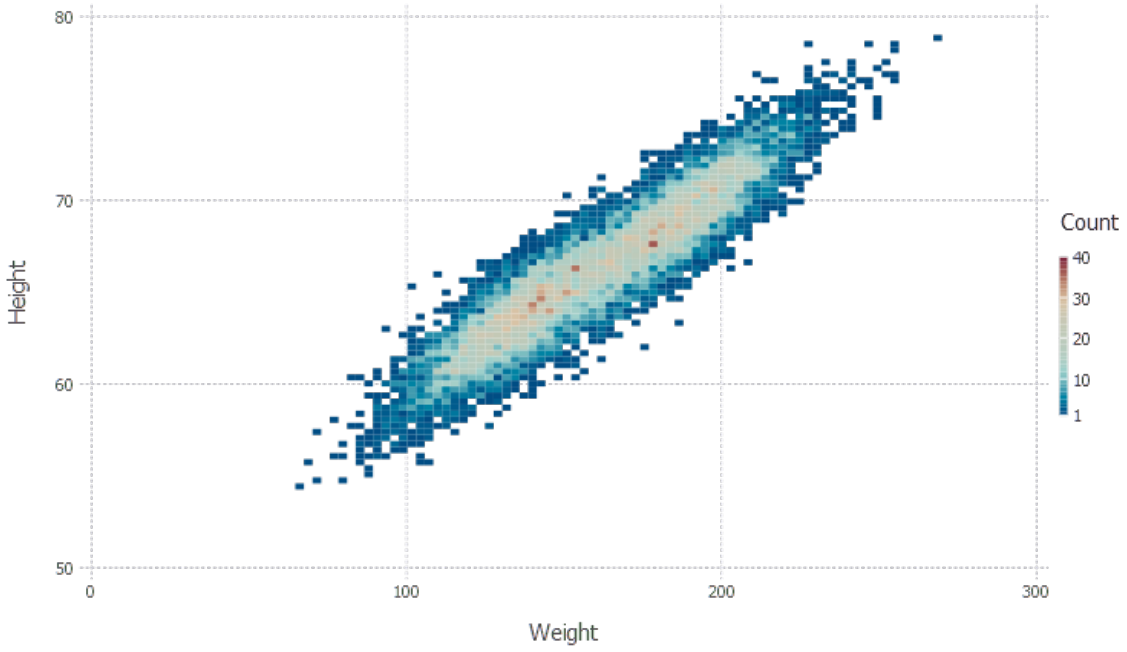


Example 3 - Real valued traget, real valued feature variable

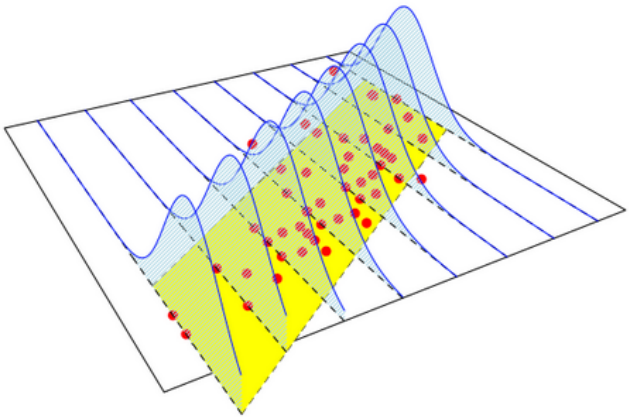
If we'd like to predict a person's height using their weight we have y is height and x is weight.

The figure below shows a 2D histogram of heights and weights. For any particular weight it looks like the height is symmetrically distributed.

So it looks like, if we want to use weight as an input feature to predict a person's height, the assumption that $y \mid x \sim \mathcal{N}(\mu, \sigma^2)$ is a reasonable one.



A nice way of picturing this is this image from the website freakonometrics. Take a cross section for a particular input variable value (or range of values) and this data is approximately normally distributed.



9 Table of Distributions and Uses

Distribution	Domain	Use	Link Function
Normal	real: $(-\infty, +\infty)$	linear-response data	identity
Exponential	real: $(0, +\infty)$	exponential response data, scale parameters - e.g. size of insurance claims, rainfall, service times	negative inverse
Gamma			
Inverse Gaussian	real: $(0, +\infty)$		inverse squared
Poisson	integer: $0, 1, 2, ...$	count of occurrences in fixed amount of time/space - e.g. count of insurance claims, count of arrivals	log
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	logit
Binomial	integer: $0, 1, 2, ..., N$	count of # of "yes" occurrences out of N yes/no occurrences	
Categorical	integer: $[0, K]$	outcome of single K-way occurrence	
	K-vector of integers: $[0, 1]$ where exactly one element in the vector has the value 1		
Multinomial	K-vector of integers: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences	