Generalized Linear Models and Exponential Family

DeepEigen‖ Course: AI-1.0X: Machine Learning | Year: 2019 | Instructor: Sanjeev Sharma

1 Exponential Family

This section describes a family of probability distributions known as the exponential family of distributions. Many of the majorly used probability distribution functions belong to the exponential family. A probability density function (pdf) or probability mass function (pmf) $p(y;\eta)$, for $y \in \mathbb{R}^l$ and $\eta \in \mathbb{R}^n$, is in the exponential family if it can be written in the form

$$
\begin{aligned}
p(y;\eta) &= \frac{1}{Z(\eta)}\lambda(y)\exp\{\eta^\intercal S(y)\}, && (1)\\
&= \exp\{-\underbrace{\log(Z(\eta))}_{A(\eta)}\}\lambda(y)\exp\left\{\eta^\intercal S(y)\right\}, \\
&= \lambda(y)\exp\{\eta^\intercal S(y) - A(\eta)\}, && (2)
\end{aligned}
$$

where

$$
\begin{aligned}
Z(\eta) &= \int_{y\in\mathbb{R}^l} \lambda(y)\exp\{\eta^\intercal S(y)\}\,dy && (3)\\
A(\eta) &= \log(Z(\eta)) && (4)
\end{aligned}
$$

Here $Z(\eta)$ is known as the **partition function** and $A(\eta)$ is known as the **log-particicition function** (also called **cumulant function**). Partition function, $Z(\eta)$ (consequently, $\exp(-A(\eta))$), essentially plays the role of normalization constant that makes sure that $p(y;\eta)$ integrates or sums over y to 1. Here η is called the **natural parameter** (it is also called the **canonical parameter**); $S(y)$ is a vector of **sufficinet statistic**; and $\lambda(y)$ is scaling constant.

A fixed choice of $S(y), A(\eta)$, and $\lambda(y)$ defines a family of distributions, that is parametrized by η. By varying η we get different distributions within the family. The next sections show that the Bernoulli and the Gaussian distributions are examples of exponential family distributions.

1.1 Bernoulli and Gaussian Distributions as Exponential Family

Earlier we covered linear regression and logistic regression (classification). In the linear regression, we had $p(y|x;w) \sim \mathcal{N}(\mu,\sigma^2)$, with $\mu = w^\intercal\phi(x)$. In the classification (logistic regression), we considered $p(y|x;w) \sim$ Bernoulli(μ), with $\mu = 1/\{1+e^{-w^\intercal\phi(x)}\} = Sigmoid(w^\intercal\phi(x))$. In this section, we will now show that the Bernoulli and the Gaussiation distributions are examples of exponential family distributions.

1.1.1 Bernoulli

The Bernoulli distribution ($\mathcal{B}$) for $y \in \{0,1\}$, with mean μ is: $\mathcal{B}(y;\mu) = \mu^y(1-\mu)^{1-y}$. Here $p(y=1;\mu) = \mu$, and $p(y=0;\mu) = 1-\mu$. By varying μ, we obtain Bernoulli distributions with different means. We now show that the Bernoulli distribution can be written as an exponential family distribution. We have,

$$
\begin{aligned}
p(y;\mu) &= \mu^y(1-\mu)^{1-y} \\
&= \exp\{y\log(\mu) + (1-y)\log(1-\mu)\} \\
&= \exp\left\{y\log\left(\frac{\mu}{1-\mu}\right) + \log(1-\mu)\right\} && (5)\\
&= \lambda(y)\exp\left\{\eta^\intercal S(y) - A(\eta)\right\}. && (6)
\end{aligned}
$$

By comparing eq.(5) and eq.(6) we obtain:

$$\begin{aligned}\lambda(y) &= 1 \\ \eta &= \log\left(\frac{\mu}{1-\mu}\right) \implies \mu = \frac{1}{1+e^{-\eta}} = \text{Sigmoid}(\eta) \\ S(y) &= y \\ A(\eta) &= -\log(1-\mu) = \log\left(\frac{1}{1-\mu}\right) = \log(1+e^{\eta}).\end{aligned} \tag{7}$$

Thus, Bernoulli distribution is an example of the exponential family distributions.

1.1.2 Gaussian

The univariate Gaussian (Normal) distribution, of $y \in \mathbb{R}$ with mean μ and variance σ^2, i.e., $y \sim \mathcal{N}(\mu, \sigma^2)$, is:

$$\begin{aligned}p(y;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\} \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}y^2 + \frac{\mu}{\sigma^2}y - \frac{1}{2\sigma^2}\mu^2\right\} \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{\begin{bmatrix}\frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2}\end{bmatrix}^\intercal \begin{bmatrix} y \\ y^2\end{bmatrix} - \frac{\mu^2}{2\sigma^2}\right\} &(8) \\ &= \underbrace{\frac{1}{\sqrt{2\pi}\sigma\exp\left(\frac{\mu^2}{2\sigma^2}\right)}\exp\left\{\begin{bmatrix}\frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2}\end{bmatrix}^\intercal \begin{bmatrix} y \\ y^2\end{bmatrix}\right\}}_{\frac{1}{Z(\eta)}\lambda(y)\exp(\eta^\intercal S(y))} &(9)\end{aligned}$$

which gives us that

$$\eta = \begin{bmatrix}\frac{\mu}{\sigma^2} \\ \frac{-1}{2\sigma^2}\end{bmatrix} \tag{10}$$

$$S(y) = \begin{bmatrix} y \\ y^2\end{bmatrix} \tag{11}$$

$$\lambda(y) = 1 \tag{12}$$

$$Z(\eta) = \sqrt{2\pi}\sigma\exp\left\{\frac{\mu^2}{2\sigma^2}\right\} \tag{13}$$

$$\begin{aligned}A(\eta) &= \log(Z(\eta)) \\ &= \frac{1}{2}\log(2\pi) + \log(\sigma) + \frac{\mu^2}{2\sigma^2} \\ &= \frac{1}{2}\log(2\pi) + \log\left(\sqrt{(-2)\frac{-1}{2\sigma^2}}\right) + \frac{\mu^2}{2\sigma^2}\end{aligned}$$

$$A(\eta) = \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(-2\eta_2) - \frac{\eta_1^2}{4\eta_2} \tag{14}$$

Recall that in linear regression, we saw that the variance σ^2 played no role in the maximum likelihood estimate of the learnable parameters (weight vector) w, and also it had no effect on the mapping function $\mathcal{F}_w(\phi(x))$. Thus, we

Cite it as: *Sanjeev Sharma, "Lecture Notes on Generalized Linear Models and Exponential Family". Swaayatt Robots, 2019.*

can ignore σ^2, or assume $\sigma^2 = 1$ when representing the univariate Gaussian as an exponential family distribution. Thus, we will now represent univariate Gaussian as an exponential family, with $\sigma^2 = 1$. With $\sigma^2 = 1$, we have

$$\begin{aligned} p(y;\eta) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(y-\mu)^2\right) \\ &= \underbrace{\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-y^2}{2}\right)\exp\left(\mu y - \frac{\mu^2}{2}\right)}_{\lambda(y)\exp(\eta^\intercal S(y) - A(\eta))} \qquad (15) \end{aligned}$$

which gives us

$$\lambda(y) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right) \qquad (16)$$

$$\eta = \mu \qquad (17)$$

$$S(y) = y \qquad (18)$$

$$A(\eta) = \frac{\mu^2}{2} = \frac{\eta^2}{2} \implies Z(\eta) = \exp\left(\frac{\eta^2}{2}\right) \qquad (19)$$

1.2 Log Partition Function

Note that in Section-(1) we mentioned that the *log-partition function* $A(\eta)$, is also known as the **cumulant function**. This is because the log-partition function can be used to generate the cumulants of the sufficient statistics, thereby justifying the name cumulant function.

Here we briefly discuss what moments (of a random variable with some distribution) and cumulants are. For a 1-dimensional random variable x, with some distribution $p(x)$, the k^{th} moment, represented as m_k is $\mathbb{E}_{x\sim p}[x^k]$.

Cumulants are certain non-linear combinations of moments, and they arise naturally when analyzing sums of independent random varilable. Here we will ignore the discussion of cumulant generation functions. Here we only discuss the first 4 cumulants of a random variable x. We will represent the k^{th} cumulant as c_k, for a 1-dimensional random varilable x with m_k as its k^{th} moment.

$$c_1 = m_1 = \mathbb{E}[x] \qquad -\text{mean}$$

$$c_2 = m_2 - m_1^2 = \mathbb{E}[x^2] - \mathbb{E}[x]^2 \qquad -\text{variance}$$

$$c_3 = m_3 - 3m_1m_2 + 2m_1^3 \qquad -\text{skeweness}$$

$$c_4 = m_4 - 3m_2^2 - 4m_1m_3 + 12m_1^2m_2 - 6m_1^4 \qquad -\text{kurtosis}$$

In these notes, we refrain from discussing any additional properties of cumulants in general, and interested readers can find more details in some good statistics text-book. We will now see how the derivatives of $A(\eta)$ provide us cumulants. We will only compute the first two cumulants. We have $A(\eta) = \log(Z(\eta))$, which gives us $A(\eta) = \log\{\int \lambda(y)\exp[\eta^\intercal S(y)]dy\}$. We will assume that η is scalar – this was the case for Bernoulli distribution (Section-1.1.1) and univariate Gaussian (Section-1.1.2) with $\sigma^2 = 1$. Thus, $A(\eta) = \log\int \lambda(y)\exp(\eta S(y))dy$. The first derivative of $A(\eta)$ is:

$$\begin{aligned} \frac{dA(\eta)}{d\eta} &= \frac{d}{d\eta}\left\{\log\int \lambda(y)\exp\left(\eta S(y)\right)dy\right\} \\ &= \frac{1}{\int \lambda(y)\exp\left(\eta S(y)\right)dy}\frac{1}{d\eta}\int \lambda(y)\exp\left(\eta S(y)\right)dy \\ &= \frac{\int S(y)\lambda(y)\exp\left(\eta S(y)\right)dy}{\exp(A(\eta))} \qquad (20) \end{aligned}$$

in eq.(20) we just used the definition of $A(\eta)$. Proceeding further, we get:

$$\begin{align}
\frac{dA(\eta)}{d\eta} &= \int \frac{S(y)\lambda(y)\exp\left(\eta S(y)\right)dy}{\exp A(\eta)} \nonumber \\
&= \int S(y)\lambda(y)\exp\left(\eta S(y) - A(\eta)\right)dy \nonumber \\
&= \int S(y)\underbrace{\lambda(y)\exp\left(\eta S(y) - A(\eta)\right)}_{p(y;\eta)}dy \nonumber \\
&= \int S(y)p(y;\eta)dy \tag{21} \\
&= \mathbb{E}[S(y)] \tag{22}
\end{align}$$

where we have used the definition of expectation of a function F of some random variable $x \sim p(x)$, which is given by $\mathbb{E}_{x\sim p(x)}[F(x)] = \int F(x)p(x)dx$, in eq.(21) to obtain eq.(22). Thus, the first derivate of the log-partition function $A(\eta)$ provides us the mean (expected value) of the sufficient statistic. Now we will compute the second derivative, making use of the first derivative computed above:

$$\begin{align}
\frac{d^2A(\eta)}{d\eta^2} &= \frac{d}{d\eta}\int S(y)\lambda(y)\exp(\eta S(y) - A(\eta))dy \nonumber \\
&= \int S(y)\lambda(y)\exp[\eta S(y) - A(\eta)](S(y) - A'(\eta))dy \nonumber \\
&= \int S(y)^2\lambda(y)\exp[\eta S(y) - A(\eta)]dy - A'(\eta)\int S(y)\lambda(y)\exp[\eta S(y) - A(\eta)]dy \nonumber \\
&= \int S(y)^2 p(y;\eta)dy - A'(\eta)\int S(y)p(y;\eta)dy \nonumber \\
&= \int S(y)^2\lambda(y)\exp[\eta S(y) - A(\eta)]dy - \underbrace{A'(\eta)}_{\mathbb{E}[S(y)]}\int S(y)\lambda(y)\exp[\eta S(y) - A(\eta)]dy \nonumber \\
&= \mathbb{E}[S(y)^2] - \mathbb{E}[S(y)]\,\mathbb{E}[S(y)] \nonumber \\
&= \mathbb{E}[S(y)^2] - \mathbb{E}[S(y)]^2 \tag{23} \\
&= \mathrm{var}[S(y)] \qquad\qquad \text{definition of variance} \tag{24}
\end{align}$$

Thus, the second derivative of the log-partition function is the variance of the sufficient statistic $S(y)$. It can be easily verified, that for the multivariate case, we will have

$$\frac{\partial^2 A(\eta)}{\partial\eta_i\partial\eta_j} = \mathbb{E}[S_i(y)S_j(y)] - \mathbb{E}[S_i(y)]\,\mathbb{E}[S_j(y)] \tag{25}$$

where $S_k(y)$ denotes the k^{th} component of the vector of sufficient statistic $S(y)$. We thus get:

$$\nabla^2 A(\eta) = \mathrm{cov}[S(y)]. \tag{26}$$

Note that earlier we had discussed that a twice differentiable function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex iff $\nabla^2 f(x) \succeq 0$, i.e. the Hessian is a positive semidefinite matrix. From eq.(26), we have that $\nabla^2 A(\eta) = \mathrm{cov}[S(y)]$, and thus the log-partition function is a convex-function. Where we have used the property of covariance matrices, i.e., they are positive (semi-) definite matrices.

Next we will see the Bernoulli and Gaussian distributions examples to show how the first derivate of the log-partition function computes the mean.

1.2.1 Examples: Bernoulli and Gaussian

In Bernoulli distribution, we have $A(\eta) = \log(1 + e^{\eta})$. The first derivative gives us

$$\begin{aligned}\frac{dA(\eta)}{d\eta} &= \frac{d}{d\eta}\log(1+e^{\eta}) \\ &= \frac{e^{\eta}}{1+e^{\eta}} \\ &= \frac{1}{1+e^{-\eta}} \qquad (27) \\ &= \mu \qquad (28)\end{aligned}$$

where eq.(28) follows from eq.(7) and eq.(27). Note that sufficient statistic $S(y) = y$, and μ is the expected value of y, a Bernoulli random variable ($\mathbb{E}[y] = p(y = 1; \mu) \times 1 + p(y = 0; \mu) \times 0 = \mu$). Furterhmore, the second derivative of $A(\eta)$ is

$$\begin{aligned}\frac{d^2A(\eta)}{d\eta^2} &= \frac{d}{d\eta}\frac{1}{1+e^{-\eta}} = \frac{d}{d\eta}\text{sigmoid}(\eta) \\ &= \mathit{sigmoid}(\eta)(1-\text{sigmoid}(\eta)) \\ &= \mu(1-\mu) \qquad (29) \\ &= \text{var}[S(y)] \qquad (= \text{var}[y]; \quad y \sim \mathcal{B}(\mu)) \qquad (30)\end{aligned}$$

In case of univariate Gaussian distribution, with $\sigma^2 = 1$, we hvae $A(\eta) = \frac{\eta^2}{2}$. Thus,

$$\begin{aligned}\frac{dA(\eta)}{d\eta} &= \eta = \mu \qquad (31) \\ \frac{d^2A(\eta)}{d\eta^2} &= 1 \qquad (32)\end{aligned}$$

Now let us calculate the derivate for univariate Gaussian with variance σ^2. In this case, $A(\eta) = \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(-2\eta_2) - \frac{\eta_1^2}{4\eta_2}$. Note that the sufficient statistic, $S(y)$ is a vector, $[y, y^2]^\intercal$. By taking the derivate with respect to η_1 gives:

$$\begin{aligned}\frac{\partial A(\eta)}{\partial \eta_1} &= \frac{\partial}{\partial \eta_1}\left\{\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(-2\eta_2) - \frac{\eta_1^2}{4\eta_2}\right\} \\ &= \frac{-2\eta_1}{4\eta_2} = \frac{-2\frac{\mu}{\sigma^2}}{\frac{-4}{2\sigma^2}} \\ &= \mu\end{aligned}$$

which is the mean of the first component of the sufficient statistic, i.e., of y. Furthermore, the second derivate with respect to η_1 is:

$$\begin{aligned}\frac{\partial^2 A(\eta)}{\partial \eta_1^2} &= \frac{\partial}{\partial \eta_1}\frac{-2\eta_1}{4\eta_2} \qquad (33) \\ &= \frac{-1}{2\eta_2} = \frac{-1}{2.\frac{-1}{2\sigma^2}} \qquad (34) \\ &= \sigma^2 \qquad (35)\end{aligned}$$

which is the variance of the first compoment, y, of the sufficient statistic.

Cite it as: *Sanjeev Sharma, "Lecture Notes on Generalized Linear Models and Exponential Family". Swaayatt Robots, 2019.*

1.3 Sufficient Statistic

In this section we only briefly discuss what is Sufficient Statistic. We do not cover the factorization theorem extensively and entirely omit the discussion on minimal sufficient statistic. Interested readers can read more about sufficient statistic and factor theorem in a good statistic textbook.

Let $\{y^1, ..., y^n\}; y^i \in \mathbb{R}^N$, is generated from some distribution $p(y;\theta)$, i.e., $\{y^1, ..., y^n\} \sim p(y;\theta)$, where θ is some unknown parameter of the distribution, which may **not** necessarily be a random variable (in Bayesian setting, we typically assume a distribution on the parameters, in which case $\theta \sim p(\theta)$). Furthermore, θ might be a parameter that perhaps we would like to learn. For example, if y is a Bernoulli random variable with mean μ, i.e., $y \sim \mathcal{B}(y;\mu)$, with $p(y = 1;\mu) = \mu$. Given the sample $\{y^1, ..., y^n\}$, perhaps we would like to estimate μ.

What is a statistic?
A statistic is any real valued function, $T = r(y^1, ..., y^n)$, of the observation $\{y^1, ..., y^n\}$. T should not have any unknown parameters. For example:

$$\begin{aligned}
T_1 &= \text{mean}\{y^1, ..., y^n\} \\
T_2 &= \max\{y^1, ..., y^n\} \\
T_3 &= 100 \\
T_4 &= y^1 + \gamma
\end{aligned}$$

Here the first 3 examples, T_1, T_2, T_3, are all valid statistics. However, in T_4, if γ is unknown, then it is not a statistic. The third example is a bit strange example, as it ignored the random sample, but it still is a statistic.

1.3.1 Sufficiency

Informally, we say T is sufficient statistic (for parameter θ) if knowing the value of T is as good as knowing the entire random sample, for estimating the value of θ. That is, the random sample contains no information about the parameter θ beyond what is contained in T. We can put this definition more formally: $T(y)$ is a sufficient statistic for θ, if the conditional distribution of y given $T(y)$ **does not** depend on θ.

Putting if mathematically:

Definition 1 *Suppose $\{y^1, ..., y^n\} \sim p(y;\theta)$. T is sufficient statistic for θ if the conditional distribution of $\{y^1, ..., y^n\}$ given $T = t$ is independent of θ for each t:*

$$p(y^1, ..., y^n | T = t; \theta) = p(y^1, ..., y^n | T = t) \implies T \textit{ sufficient for } \theta.$$

To see this, we can consider an example of a Bernoulli random variable $y \in \{0, 1\} \sim \mathcal{B}(\mu)$.

Example 1 Let $\{y^1, ..., y^n\}$ be IID Bernoulli trials, with $p(y^i = 1;\mu) = \mu, i = 1, ..., n$. Let $T = \sum_{i=1}^n y^i = t$. We will see that T is sufficient statistic for μ.

Proof: *Using Bayes theorem, we have*

$$\begin{aligned}
p(x^1, ..., x^n | T = t) &= \frac{p(x^1, ..., x^n)}{p(T = t)} \\
&= \frac{\mu^t(1-\mu)^{1-t}}{\binom{n}{t}\mu^t(1-\mu)^{1-t}} = \frac{1}{\binom{n}{t}}
\end{aligned}$$

The conditional distribution does not involve μ, which proves $\sum_{i=1}^n y^i$ is sufficient statistic for θ.

It is difficult, however, to use this defition-1 for general cases to check if a statistic is a sufficient statistic, or to find a sufficient statistic. There is a theorem that enables us to find sufficient statistics.

Theorem 1 Factorization Theorem: *Let $\{x^1, ..., x^n\}$ be a set of random sample with joint density $p(x^1, ..., x^n; \theta)$. A statistic $T(x) = r(x^1, ..., x^n)$ is sufficient if an only if the joint density can be factored as follows:*

$$p(x^1, ..., x^n; \theta) = f(x^1, ..., x^n) g(r(x^1, ..., x^n); \theta) \tag{36}$$

where the functions f and g are non-negative functions. Function f may depend on x but does not depend of θ. The function g depends on θ and depends on observed values of the samples $x^i; i = 1, ..., n$ only through the value of sufficient statistic $T(x)$.

Lets consider an example to see the factorization theorem.

Example 2 Let $x^1, ..., x^n \sim \mathcal{N}(\mu, \sigma^2)$. Let $\mathbf{x}^n$ represent the random sample set, i.e, $\mathbf{x}^n = \{x^1, ..., x^n\}$. Let $\hat{x} = \frac{1}{n}\sum_i x^i$. Further, assume that σ^2 is known. We have

$$\begin{aligned}
p(\mathbf{x}^n; \mu, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \mu)^2}{2\sigma^2}\right\} = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x} + \hat{x} - \mu)^2}{2\sigma^2}\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x})^2 + n(\hat{x} - \mu)^2 - 2(\hat{x} - \mu)\sum_i (x^i - \hat{x})}{2\sigma^2}\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x})^2 + n(\hat{x} - \mu)^2 - 2(\hat{x} - \mu)(n\hat{x} - n\hat{x})}{2\sigma^2}\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x})^2 + n(\hat{x} - \mu)^2}{2\sigma^2}\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x})^2}{2\sigma^2}\right\} \times \exp\left\{-\frac{n(\hat{x} - \mu)^2}{2\sigma^2}\right\} \\
&= \underbrace{\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \hat{x})^2}{2\sigma^2}\right\}}_{f(\mathbf{x}^n)} \times \underbrace{\exp\left\{-\frac{n(\hat{x} - \mu)^2}{2\sigma^2}\right\}}_{g(T(\mathbf{x}^n);\mu)}
\end{aligned}$$

Thus, $\hat{x}$ is a sufficient statistic for μ, when σ^2 is known.

For general case, we have:

$$\begin{aligned}
p(\mathbf{x}^n; \mu, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i - \mu)^2}{2\sigma^2}\right\} = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{\sum_i (x^i)^2 + \mu^2 - 2\mu x^i}{2\sigma^2}\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\sum_i (x^i)^2 + \frac{\mu}{\sigma^2}\sum_i x^i - \frac{n}{2\sigma^2}\mu^2\right\}
\end{aligned} \tag{37}$$

Thus, by factorization theorem, we have that $T = \left(\sum_i x^2, \sum_i (x^i)^2\right)$ is sufficient statistic for $\theta = (\mu, \sigma^2)$.

We omit the detailed discussion on factorization theorem, and minimal sufficient statistic. Interested readers can find more information in a good statistical notebook.

2 Generalized Linear Models

Consider the linear regression example presented in earlier notes. There, we assumed that we have a dataset $\mathcal{D}\{x^i, y^i\}; i = 1, ..., m; x^i \in \mathbb{R}^k, y^i \in \mathbb{R}, \phi(x) \in \mathbb{R}^N$. The goal is then to learn a mapping function $\mathcal{F}_w(\phi(x)) : \mathbb{R}^N \rightarrow \mathbb{R}$, using the data which can be used to predict the target values $\hat{y}$ given a new input data point $\hat{x}$. Furthermore, we assumed that the dataset has some noise ϵ^i, i.e., each of the input data-points in $\mathcal{D}$ follows the relation:

$$y^i = \mathcal{F}_w(\phi(x^i)) + \epsilon^i.$$

Furthermore, we assumed that $\mathcal{F}_w(\phi(x))$ is linear in its input, which gave us linear (in features) predictor:

$$y^i = \phi(x^i)^\intercal w + \epsilon^i$$

where each of the ϵ is IID. If we assume that $\epsilon^i \ \mathcal{N}(0, \sigma^2)$, then it results in least squares regression, or $\mathbb{L}_2^2$ error minimization over the dataset $\mathcal{D}$ with respect to the parameters w, with optimal (MLE) solution

$$w_{\text{mle}} = \underset{w}{\operatorname{argmin}} \| \mathbf{y} - \Phi w\|_2^2 \tag{38}$$

Where $\Phi \in \mathbb{R}^m \times \mathbb{R}^N$ is the feature matrix, $\mathbf{y} \in \mathbb{R}^m$ is a vector of target values, as discussed in the earlier lectures. The conditional distribution of y given x is thus a Normal distribution with mean $\mathcal{F}_w(\phi(x))$, and variance σ^2, i.e., $y|x; w \sim \mathcal{N}(\mathcal{F}_w(\phi(x)), \sigma^2)$. If we assume that the noises are IID and have Lapcacian distribution with mean 0 and diversion b, i.e., $\epsilon^i \sim \mathcal{L}(0, b)$, then the maximum likelihood estimate will result in minimization of the $\mathbb{L}_1$ error on the dataset, i.e, following optimization problem:

$$w_{\text{mle}} = \underset{w}{\operatorname{argmin}} \| \mathbf{y} - \Phi w\|_1 \tag{39}$$

We have seen earlier in regularization discussion that assuming Laplacian prior on the parameter vector w results in $\mathbb{L}_1$ regularized regression. Following similar steps, by computing the likelihood of the parameters and the maximimizing the log-likelihood or minimizing the negative log-likelihood will result in (39). As we saw in the regularized regression example, there is no closed form solution for (39) and the solution to the optimization problem is obatained via sub-gradient descent. In the regression model, we assumed that the noise is additive, Laplacian or Gaussian.

In the classification example, using logistic regression, we considered that $y \in \{0, 1\}$ is a Bernoulli random variable, with mean μ, i.e., $y \sim \mathcal{B}(\mu)$. The conditional distribution of y given x is is a Bernoulli distribution, with $p(y = 1|x; \mu) = \mu$. Generalized Linear Models (GLMs) are a statistical framework, a way, to unify regression and classification. Furthermore, GLMs allow us to easily consider other probability models, and not just Laplacian, Gaussian or Bernoulli.

We have already seen that Gaussian and Bernoulli both can be represented as Exponential Family distributions. We will now see that Both the linear regession and logistic regression are special cases of GLMs. Furthermore, we will use GLMs to derive a learning algorithm known as Softmax Regression.

2.1 Constructing GLMs

Note that, in both the classification and regression problem, the goal was to predict the conditional distribution of y given x, where the distribution is parametrized by some learnable parameters w. Again, consider a problem (classification or regression), where again we would like to predict y given x. Also, recall that instead of learning the mapping directly on x, we learn a mapping on $\phi(x)$. We will represent an exponential family distribution with natural parameter η as $\xi(\eta)$. Driving GLM for this problem will require following three assumptions, or more accurately, design choices:

- $p(y|x; w) \sim \xi(\eta)$; i.e., the distribution of y given x, parametrized by w, follows or can be represented as an exponential family distribution with natural parameters η, with conditional mean μ, $\mathbb{E}[y|x] = \mu$, where $\eta = \Psi^{-1}(\mu)$, where $\Psi(.)$ is some real-function.

Cite it as: *Sanjeev Sharma, "Lecture Notes on Generalized Linear Models and Exponential Family". Swaayatt Robots, 2019.*

- $\eta = w^{\intercal}\phi(x)$, where $\phi(x)$ is the non-linear feature function mapping $\phi(x) : x \in \mathbb{R}^k \rightarrow \mathbb{R}^N$. This is a design choise, in Generalized Linear Models (deriving their name). If η is a vector, then $\eta_i = W_i^{\intercal}\phi(x)$.
- $\mu = \Psi(\eta) = \Psi(w^{\intercal}\Phi(x))$, where Ψ is some real function; given x, our goal is to predict expected value of the sufficient statistic $S(y)|x$. In case of Bernoulli and Gaussian (with known σ^2) distributions, $S(y) = y$. Thus, in the cases we consider, the goal would be to predict $\mathbb{E}[y|x] = \mu$. We further assume that $\mu = \mathcal{F}_w(\phi(x))$. The goal, eventually, is then to predict the mapping function $\mathcal{F}_w(\phi(x))$, that predicts the $\mathbb{E}[y|x] = \mu$.

2.1.1 GLMs: Linear Regression with L_2^2 Loss

In the linear regression with sum of suqared errors, or $\mathbb{L}_2$-Norm Squared ($\mathbb{L}_2^2$) loss function, we learned earlier that we assume an additive Gaussian noise in the dataset. Furthermore, the $\mathbb{E}[y|x]$ is a Gaussian distribution.

To show that Linear Regression with $\mathbb{L}_2^2$ error can be expressed as GLMs, we will assume that y is continuous, $y \in \mathbb{R}$, and we assume that that $p(y|x)$ is a Normal distribution, $\mathcal{N}(\mu, \sigma^2)$ (μ may depend on x). Thus, we will let $\xi(\eta)$ be the Gaussian distribution. Furthermore, we earlier saw that (with known σ^2), $S(y) = y$. Also, we saw that for Gaussian distribution, $\eta = \mu$. Thus, we have:

$$\begin{aligned}
\mathbb{E}[S(y)|x] &= \mathbb{E}[y|x] \\
&= \mu = \mathcal{F}_w(\phi(x)) \\
&= \eta \\
&= w^{\intercal}\phi(x). \qquad (40)
\end{aligned}$$

Thus, we get that $\mathcal{F}_w(\phi(x)) = w^{\intercal}\phi(x) = \mu$, which was exactly the case in Linear Regression example. This proves that the Linear Regression with $\mathbb{L}_2^2$ loss is a special case of GLMs.

2.1.2 GLMs: Logistic Regression

We now show that Logistic Regression is a special case of GLMs. We assume that $y \in \{0, 1\}$. Thus it is natural to assum a Bernoulli distribution, with some mean μ, i.e., $y \sim \mathcal{B}(\mu)$. We thus model $y|x \sim \mathcal{B}(\mu)$, i.e., the conditional distribution of $y|x$ is a Bernoulli distribution with some mean μ. Also, recall that if $y|x; w \sim \mathcal{B}(\mu)$, then $\mathbb{E}[y|x] = \mu$ $\{\mathbb{E}[y|x; w] = 1 \times p(y = 1|x; w) + 0 \times p(y = 0|x; w) = \mu\}$. Furthermore, as we saw in section-1.1.1, that in case of Bernoulli distribution, $S(y) = y$, and $\mu = \mathit{Sigmoid}(\eta)$. Again, we assume that $\mu = \mathcal{F}_w(\phi(x)) = \mathbb{E}[y|x; w]$. Thus, we have:

$$\begin{aligned}
\mathbb{E}[S(y)|x] &= \mathbb{E}[y|x] \\
&= \mu = \mathcal{F}_w(\phi(x)) \\
&= \mathit{Sigmoid}(\eta) \\
&= \frac{1}{1 + e^{-\eta}} \\
&= \frac{1}{1 + e^{-w^{\intercal}\phi(x)}} \qquad (41) \\
&= \mathit{Sigmoid}(w^{\intercal}\phi(x)). \qquad (42)
\end{aligned}$$

Thus, we get that $\mathcal{F}_w(\phi(x)) = \mathit{Sigmoid}(w^{\intercal}\phi(x))$, which was exactly the case in Logistic Regression example. This proves that Logistic Regression is an special case of GLMs.

2.2 GLMs: Softmax Regression

We will end the topic of Exponential Family distribution and Generalized Linear Models by deriving a learning algorithm known as Softmax Regression. Softmax Regression generalizes Logistic Regression for multi-class classification tasks, i.e., when y is non-binary. Lets consider an example where $y \in \{1, 2, ..., q\}$. For example, this is useful when the input variable x can have more than two possible labels/classes, when classifying it.

Cite it as: Sanjeev Sharma, "Lecture Notes on Generalized Linear Models and Exponential Family". Swaayatt Robots, 2019.

In logistic regression, $y \in \{0,1\}$, and the (stochastic-) functional mapping $\mathcal{F}_w(\phi(x))$ gives us the conditional probability $p(y=1|x;w)=\mu, p(y=0|x;0)=1-\mu$. We assumed that there are only two classes, 0 and 1. Softmax Regression generalizes it to q-classes, $y \in \{1,...,q\}$. To summarize Softmax Regression, we compute the conditional probability $y|x$ using a **Softmax Function**, or also known as the **normalized exponential function**. The conditional probabilitities $y|x$ (parametrized by $W \in \mathbb{R}^{q\times N}$) are given by the functional mapping $(\mathcal{F}_W(\phi(x)) = p(y|x;W))$:

$$W = \begin{bmatrix} —(w^1)^\intercal— \\ —(w^2)^\intercal— \\ \vdots \\ —(w^q)^\intercal— \end{bmatrix} \tag{43}$$

$$p(y=j|x;W) = \frac{\exp(\phi(x)^\intercal w^j)}{\sum_i \exp(\phi(x)^\intercal w^i)} \tag{44}$$

$$= \frac{\exp(W_j\phi(x))}{\sum_i \exp(W_i\phi(x))} \tag{45}$$

where $(w^j)^\intercal = W_j \in \mathbb{R}^N$ is the j^{th} row of the matrix W. We can now use the gradient descent or *Newton's Method* to maximize the likelihood of the parameters, to compute $W^\star = W_{\text{MLE}}$. Furthermore, $w^q = \mathbf{0}$, which we will see later. We will now construct a GLM to derive the formulations for Softmax Regression learning algorithm.

2.2.1 Softmax Regression as GLM

y is discrete and can have multiple values, and thus we will model it as a Multinomial distribution. The multinomial distribution over q possible values of y can be parametrized using q parameters, $\mu_1, \mu_2, ..., \mu_q$, where $\mu_i = p(y = i|x;W)$. Also $\mu_i; i = 1,...,q$, satsify:

$$\sum_{j=1}^{q} \mu_j = 1. \tag{46}$$

Using constraint-46, the number of parameters can be reduced to $q-1$, as $\mu_q = 1 - \sum_{j=1}^{q-1} \mu_j$. For notational convenience in the derivation below, we will assume that $\mu_q = 1 - \sum_{j=1}^{q-1} \mu_j$, but note that μ_q is not a parameter, and it can computed by $\mu_1, ..., \mu_{q-1}$. In case of Logistic regression, the sufficient statistic $S(y) = y$. To express a multinomial distribution as an exponential family distribution we define $S(y) \in \mathbb{R}^{q-1}$ as follows:

$$S(y=1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; S(2) = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, S(q-1) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, S(q) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{47}$$

We can further represent $S(y)$ more effectively using an Indicator function. An indicator function $\mathcal{I}\{\mathcal{E}\}$, is equal to 1 if the event $\mathcal{E}$ is true, and is equal to 0 if the event $\mathcal{E}$ is false. We can thus write the sufficient statistic $S(y)$ more compactly as:

$$S(y) = \begin{bmatrix} \mathcal{I}\{y=1\} \\ \mathcal{I}\{y=2\} \\ \vdots \\ \mathcal{I}\{y=q-1\} \end{bmatrix}. \tag{48}$$

Our goals is now to compute the conditional expectation of the Sufficient Statistic $S(y)$ given x. We can see that the expected value of the sufficient statistic, which is a $q-1$ dimensional vector, is given by the $\mu_i, ; i = 1, ..., q-1$.

We can show this mathematically, for the i^{th} component of the sufficient statistic $S(y)_i$, as:

$$\mathbb{E}[S(y)_i|x] = \mathbb{E}[\mathcal{I}\{y = i\}|x] = p(y = i|x) = \mu_i \tag{49}$$

Thus, our goal is to learn the stochastic functional mapping $\mathcal{F}_W(\phi(x))$, which computes the vector of conditional probabilities $\mu_i = p(y = i|x; W); i = 1, ..., q$. Note that we will only compute $\mu_i; i = 1, ..., q - 1$, and then $\mu_q = 1 - \sum_{i=1}^{q-1} \mu_i$. This mapping function will be the Softmax function or the normalized exponential function. We will now express the multinomial distribution as an Exponential Family Distribution $(\xi(\eta))$ with natural parameter $\eta \in \mathbb{R}^{q-1}$. Hereafter $\mu = [\mu_1, ..., \mu_q]^\intercal$, where μ_q can be computed using $\mu_1, ..., \mu_{q-1}$ as discussed earlier. We have

$$\begin{aligned}
p(y;\mu) &= \prod_{i=1}^{q} p(y = 1; \mu)^{\mathcal{I}\{y=i\}} && (50)\\
&= \prod_{i=1}^{q} \mu_i^{\mathcal{I}\{y=i\}} && (51)\\
&= \left\{\prod_{i=1}^{q-1} \mu_i^{\mathcal{I}\{y=i\}}\right\} \mu_q^{\mathcal{I}\{y=q\}} && (52)\\
&= \left\{\prod_{i=1}^{q-1} \mu_i^{S(y)_i}\right\} \mu_q^{1-\sum_{j=1}^{q-1} S(y)_j} && (53)\\
&= \exp\left(\sum_{i=1}^{q-1} S(y)_i \log(\mu_i) + \left(1 - \sum_{j=1}^{q-1} S(y)_j\right) \log(\mu_q)\right) && (54)\\
&= \exp\left(\sum_{i=1}^{q-1} S(y)_i \log\left(\frac{\mu_i}{\mu_q}\right) + \log(\mu_q)\right) && (55)\\
&= \lambda(y) \exp\{\eta^\intercal S(y) - A(\eta)\} && (56)
\end{aligned}$$

which gives us

$$\begin{aligned}
\eta &= \begin{bmatrix} \log\left(\frac{\mu_1}{\mu_q}\right) \\ \log\left(\frac{\mu_1}{\mu_q}\right) \\ \vdots \\ \log\left(\frac{\mu_{q-1}}{\mu_q}\right) \end{bmatrix} && (57)\\
A(\eta) &= -\log(\mu_q) && (58)\\
\lambda(y) &= 1. && (59)
\end{aligned}$$

Thus, the multinomial distribution belongs the the exponential family distribution with above parameters. For convenience, we can define $\eta \in \mathbb{R}^q$ with $\eta_q = \log \frac{\mu_q}{\mu_q} = 0$. From the equations above, we can now obtain μ as:

$$\eta_i = \log\left(\frac{\mu_i}{\mu_q}\right) \implies \mu_q e^{\eta_i} = \mu_i. \tag{60}$$

By summing over all the $i = 1, ..., q$, we get:

$$\begin{aligned}
\sum_{i=1}^{q} \mu_q e^{\eta_i} &= \sum_{i=1}^{q} \mu_i = 1 && (61)\\
\implies \mu_q &= \frac{1}{\sum_{i=1}^{q} e^{\eta_i}} && (62)
\end{aligned}$$

Substituting the result (eq-62) back into eq-60 gives us:

$$\mu_i = \mu_q e^{\eta_i} \implies \mu_i = \frac{e^{\eta_i}}{\sum_{j=1}^{q} e^{\eta_j}} \tag{63}$$

This functional mapping, $\mu = \Psi(\eta)$, i.e., $\mu_i = \frac{e^{\eta_i}}{\sum_{j=1}^{q} e^{\eta_j}}$, from η to μ is known as Softmax Function, as discussed in earlier section, see eq-44 and eq-45. To obtain the results of eq-45, we use the assumption number 2 in GLMs, discussed in section-2.1, i.e., η is a linear function of the input x or equivalently of the feature function representation of x, $\phi(x)$. Thus, we have that $\eta_i = \phi(x)^\intercal w^i, i = 1, ..., q-1$, where w^i are the parameters for the i^{th} class/label are the parameters of the model. We also define $w^q = \mathbf{0}$ a vector of 0s, which gives us $\eta_q = w^{q\intercal}\phi(x) = 0$, as discussed earlier.

We can summarize it as follows:

$$\mu_i = \frac{e^{\phi(x)^\intercal w^i}}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \tag{64}$$

$$\mu_q = \frac{1}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \tag{65}$$

where $w^i \in \mathbb{R}^N; i = 1, ..., q-1$ are the model parameters, and $w^q = \mathbf{0}$ (not a model parameter, mentioned only for convenience) is an N-dimensional vector with all its components being 0. We can define an $q \times N$ matrix W (last row is not a model parameter, represented only for convenience), which contains all these model parameters as:

$$W = \begin{bmatrix} —(w^1)^\intercal— \\ —(w^2)^\intercal— \\ \vdots \\ —\mathbf{0}^\intercal— \end{bmatrix}_{q \times N} \tag{66}$$

Furthermore, our functional mapping $\mathcal{F}_W(\phi(x))$ is given by:

$$\begin{aligned} \mathcal{F}_W(\phi(x)) &= \mathbb{E}[S(y)|x] \\ &= \mathbb{E}\left[\left.\begin{bmatrix} \mathcal{I}\{y=1\} \\ \mathcal{I}\{y=2\} \\ \mathcal{I}\{y=3\} \\ \vdots \\ \mathcal{I}\{y=q-1\} \end{bmatrix}\right| x\right] & (67) \\ &= \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \\ \mu_{q-1} \end{bmatrix} & (68) \\ &= \begin{bmatrix} \frac{e^{\phi(x)^\intercal w^1}}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \\ \frac{e^{\phi(x)^\intercal w^2}}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \\ \frac{e^{\phi(x)^\intercal w^3}}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \\ \vdots \\ \frac{e^{\phi(x)^\intercal w^{q-1}}}{\sum_{j=1}^{q} e^{\phi(x)^\intercal w^j}} \end{bmatrix} & (69) \end{aligned}$$

Cite it as: *Sanjeev Sharma, "Lecture Notes on Generalized Linear Models and Exponential Family". Swaayatt Robots, 2019.*

Thus, the functional mapping $\mathcal{F}_W(\phi(x)) : \mathbb{R}^N \rightarrow [0,1]^{q-1}$ gives us the class probabilities. It is thus a stochastic function, which generates the conditional probabilities $y|x$. Althoug it doesn't directly provide μ_q but it can be computed simply as $\mu_q = 1 - \sum_{j=1}^{q-1} \mu_j$. Furthermore, with our notation of W, whose last row is a vector of 0s, we can assume that the $\mathcal{F}_W(\phi(x))$ produces q dimensional output as, $\mathcal{F}_W(\phi(x)) : \mathbb{R}^N \rightarrow [0,1]^q$:

$$\mathcal{F}_W(\phi(x)) = \begin{bmatrix} \frac{e^{\phi(x)^\intercal w^1}}{\sum_{j=1}^q e^{\phi(x)^\intercal w^j}} \\ \frac{e^{\phi(x)^\intercal w^2}}{\sum_{j=1}^q e^{\phi(x)^\intercal w^j}} \\ \frac{e^{\phi(x)^\intercal w^3}}{\sum_{j=1}^q e^{\phi(x)^\intercal w^j}} \\ \vdots \\ \frac{e^{\phi(x)^\intercal w^{q-1}}}{\sum_{j=1}^q e^{\phi(x)^\intercal w^j}} \\ \frac{1}{\sum_{j=1}^q e^{\phi(x)^\intercal w^j}} \end{bmatrix} \tag{70}$$

We thus obtain the same result, after deriving the Softmax Regression using GLMs, as mentioned in eq-45. This finishes our discussion on Generalized Linear Models.

3 Softmax Regression

Having presented the Softmax Regression learning algorithm, which can be used for multi-class classification tasks, we will now see how we can learn the parameters using the gradient descent algorithm. Assume that we have a dataset $\mathcal{D}\{x^i, y^i\}; i = 1, ..., m; x \in \mathbb{R}^k, y \in \{1, 2, 3, ..., \mathcal{K}\}$. Given the feature function $\phi(x) : \mathbb{R}^k \rightarrow \mathbb{R}^N$, the goal is to learn $\mathcal{F}_W(\phi(x)) : \mathbb{R}^N \rightarrow [0,1]^{\mathcal{K}}$, where $[0,1]^{\mathcal{K}}$ represents a $\mathcal{K}$-dimensional vector, with each component of the vector beglonging to set $[0,1]$. The conditional probabilities are $p(y|x;W) = \mu_i = \frac{e^{W_i\phi(x)}}{\sum_{j=1}^{\mathcal{K}} e^{W_j\phi(x)}}$, where W_i is the i^{th} row of the matrix $W \in \mathbb{R}^{\mathcal{K} \times N}$.

We can compute the maximum likelihood estimate (MLE) of the parameters W by maximizing the likelihood function $\mathcal{L}(W)$, as we did in Logistic Regression:

$$\begin{aligned} \mathcal{L}(W) &= \prod_{i=1}^m p(y^i|x^i;W) && (71) \\ \mathcal{LL}(W) = \log \mathcal{L}(W) &= \sum_{i=1}^m \log p(y^i|x^i;W) && (72) \\ &= \sum_{i=1}^m \log \prod_{j=1}^{\mathcal{K}} \mu_j^{i\,\mathcal{I}\{y^i=j\}} && (73) \\ &= \sum_{i=1}^m \log \prod_{j=1}^{\mathcal{K}} \left(\frac{e^{W_j\phi(x^i)}}{\sum_{s=1}^{\mathcal{K}} e^{W_s\phi(x^i)}} \right)^{\mathcal{I}\{y^i=j\}} && (74) \end{aligned}$$

where W_l is a row vector (l^{th}-row of matrix W), which is $(w^l)^\intercal, l = 1, ..., \mathcal{K}$. We can maximize the log-likelihood function, $\mathcal{LL}(W)$, to obtain the MLE estimate of the parameters W. This can be done using algorithms like Gradient Descent or Netwon's Method.

3.1 Maximizing Log Likelihood: Gradient Descent

We have the log-likelihood function $\mathcal{LL}(W)$ given in eq-74. To compute the update rule for gradient descent algorithm, we have to compute the gradients of the log-likelihood funtions, $\nabla_W \mathcal{LL}(W)$, with respect to the parameters $w^i; i = 1, ..., \mathcal{K}$. We can represent $\phi(x^i)^\intercal w^j$ as z_i^j to simplify the notations.

Thus we get:

$$\mathcal{LL}(W) = \sum_{i=1}^{m} \log \prod_{j=1}^{\mathcal{K}} \left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right)^{\mathcal{I}\{y^i=j\}} \tag{75}$$

We can use the chain rule of differentiation to compute the $\nabla_{w^j} \mathcal{LL}(W)$ as:

$$\nabla_{w^j} \mathcal{LL}(W) = \frac{\partial \, \mathcal{LL}(W)}{\partial z^j} \times \frac{\partial z^j}{\partial w^j}. \tag{76}$$

Furthermore

$$\frac{\partial z^j}{\partial w^j} = \frac{\phi(x)^\intercal w^j}{\partial w^j} = \phi(x). \tag{77}$$

Having all the basic mathematical expressions in place, we can now compute the gradient $\nabla_{w^j} \mathcal{LL}(W)$. Also, there would be two cases: (i) $y^i = j$ and (ii) $y^i \neq j$.

Case-1: $y^i = j$

$$\begin{aligned}
\nabla_{w^j} \mathcal{LL}(W) &= \frac{\partial}{\partial z^j} \sum_{i=1}^{m} \left(\log \left[\prod_{j=1}^{\mathcal{K}} \left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right)^{\mathcal{I}\{y^i=j\}} \right] \times \overbrace{\frac{\partial z_i^j}{\partial w^j}}^{\phi(x^i)} \right) \qquad (78) \\
&= \sum_{i=1}^{m} \left(\frac{\partial}{\partial z^j} \log \left[\prod_{j=1}^{\mathcal{K}} \left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right)^{\mathcal{I}\{y^i=j\}} \right] \times \phi(x^i) \right) \qquad (79) \\
&= \sum_{i=1}^{m} \left(\frac{\partial}{\partial z^j} \log \left[\left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \right] \times \phi(x^i) \right) \qquad (80) \\
&= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^j}} \left(\frac{\partial}{\partial z^j} \frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \times \phi(x^i) \qquad (81) \\
&= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^j}} \left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} + e^{z_i^j} \frac{\partial}{\partial z^j} \frac{1}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \phi(x^i) \qquad (82) \\
&= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^j}} \left(\frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} - \frac{e^{z_i^j} e^{z_i^j}}{\left(\sum_{s=1}^{\mathcal{K}} e^{z_i^s} \right)^2} \right) \phi(x^i) \qquad (83) \\
&= \sum_{i=1}^{m} \left(1 - \frac{e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \phi(x^i) \qquad (84) \\
&= \sum_{i=1}^{m} \left(1 - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right) \phi(x^i) \qquad (85)
\end{aligned}$$

Case-2: $y^i = a \neq j$

$$\begin{aligned}
\nabla_{w^j} \mathcal{LL}(W) &= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^a}} \left(\frac{\partial}{\partial z^j} \frac{e^{z_i^a}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \times \phi(x^i) & (86)\\
&= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^a}} \left(e^{z_i^a} \frac{\partial}{\partial z^j} \frac{1}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \right) \times \phi(x^i) & (87)\\
&= \sum_{i=1}^{m} \frac{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}}{e^{z_i^a}} \left(- \frac{e^{z_i^a} e^{z_i^j}}{\left(\sum_{s=1}^{\mathcal{K}} e^{z_i^s}\right)^2} \right) \phi(x^i) & (88)\\
&= \sum_{i=1}^{m} \frac{-e^{z_i^j}}{\sum_{s=1}^{\mathcal{K}} e^{z_i^s}} \phi(x^i) & (89)\\
&= \sum_{i=1}^{m} \frac{-e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \phi(x^i) & (90)
\end{aligned}$$

Having derived the results for both the cases, we can now combine the results to obtain the gradient $\nabla_{w^j} \mathcal{LL}(W)$. By combining eq-85 and eq-90, we get:

$$\nabla_{w^j} \mathcal{LL}(W) = \left[\sum_{i=1}^{m} \left(1 - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right) \mathcal{I}\{y^i = j\} + \sum_{i=1}^{m} \frac{-e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \mathcal{I}\{y^i \neq j\} \right] \phi(x^i) \tag{91}$$

We can simplify it further as:

$$\begin{aligned}
\nabla_{w^j} \mathcal{LL}(W) &= \sum_{i=1}^{m} \left[\left(1 - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right) \mathcal{I}\{y^i = j\} - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \mathcal{I}\{y^i \neq j\} \right] \phi(x^i) & (92)\\
&= \sum_{i=1}^{m} \left[\mathcal{I}\{y^i = j\} - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \left(\underbrace{\mathcal{I}\{y^i \neq j\} + \mathcal{I}\{y^i = j\}}_{1} \right) \right] \phi(x^i) & (93)\\
&= \sum_{i=1}^{m} \left[\mathcal{I}\{y^i = j\} - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right] \phi(x^i) & (94)\\
&= \sum_{i=1}^{m} \left[\delta_{ij} - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right] \phi(x^i) & (95)
\end{aligned}$$

Where $\delta_{ij} = \mathcal{I}\{y^i = j\}$. Thus, we can compute the parameters, using the Batch-Gradient Descent algorithm, which uses all the training data at once to compute one update for the parameters. We can instead used mini-batch or stochastic gradient descent, which updates the parameters after observing every training data point as:

$$\nabla_{w^j} \mathcal{LL}(W; \mathcal{D}^i) = \left(\delta_{ij} - \frac{e^{\phi(x^i)^\intercal w^j}}{\sum_{s=1}^{\mathcal{K}} e^{\phi(x^i)^\intercal w^s}} \right) \phi(x^i) \tag{96}$$

Where $\nabla_{w^j} \mathcal{LL}(W; \mathcal{D}^i)$ is the gradient with respect to $w^j; j = 1, ..., \mathcal{K}$ for the i^{th} training data-point in the dataset $\mathcal{D}$. Eq-96 gives the update rule for the stochastic gradient descent algorith, for the parameters w^j.

Furthermore, as discussed earlier $w^{\mathcal{K}} = \mathbf{0}$. Thus, all the other weights can be initially initialized to random values, and $w^{\mathcal{K}}$ can be initialized to $\mathbf{0}$ and then after every stochastic gradient descent step, $w^{\mathcal{K}}$ will be shifted to some value $\neq 0$. We can thus, after every step of stochastic gradient descent, project $w^{\mathcal{K}}$ to $\mathbf{0}$, or simply reset them to $\mathbf{0}$. This completes our discussion of batch gradient descent, and stochastic gradient descent for the Softmax Regression.