Machine Learning

Lecture 3: Logistic Regression

Feng Li

fli@sdu.edu.cn

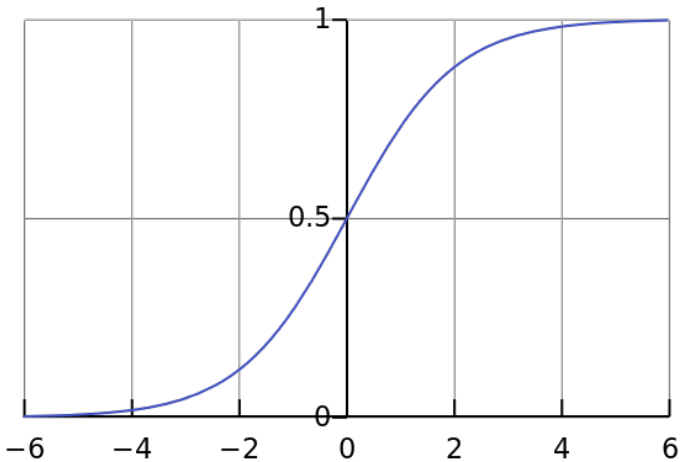
https://funglee.github.io

School of Computer Science and Technology
Shandong University

Fall 2018

Logistic Regression

- Classification problem
  - Similar to regression problem, but we would like to predict only a small number of discrete values (instead of continuous values)
  - Binary classification problem: $y \in \{0, 1\}$ where 0 represents negative class, while 1 denotes positive class
  - $y^{(i)} \in \{0, 1\}$ is also called the **label** for the training example
- Logistic regression
  - Use a logistic function (or sigmoid function) $g(z) = 1/(1 + e^{-z})$ to continuously approximate discrete classification



Logistic Regression (Contd.)

- Properties of the sigmoid function
  - Bound
    $$g(z) \in (0, 1)$$
  - Symmetric
    $$1 - g(z) = g(-z)$$
  - Gradient
    $$g'(z) = g(z)(1 - g(z))$$

Logistic Regression (Contd.)

- Logistic regression defines $h_\theta(x)$ using the sigmoid function

$$h_\theta(x) = g(\theta^T x) = 1/(1 + e^{-\theta^T x})$$

- First compute a real-valued "score" $(\theta^T x)$ for input x and then "squash" it between $(0, 1)$ to turn this score into a probability (of x's label being 1)
- Thus, we have

$$\Pr(y = 1 \mid x; \theta) = h_\theta(x) = 1/(1 + \exp(-\theta^T x))$$
$$\Pr(y = 0 \mid x; \theta) = 1 - h_\theta(x) = 1/(1 + \exp(\theta^T x))$$

Logistic Regression: A Closer Look ...

- What's the underlying decision rule in logistic regression?
- At the decision boundary, both classes are equiprobable; thus, we have

$$\begin{aligned} & \Pr(y = 1 \mid x; \theta) = \Pr(y = 0 \mid x; \theta) \\ \Rightarrow \quad & \frac{1}{1 + \exp(-\theta^T x)} = \frac{1}{1 + \exp(\theta^T x)} \\ \Rightarrow \quad & \exp(\theta^T x) = 1 \\ \Rightarrow \quad & \theta^T x = 0 \end{aligned}$$

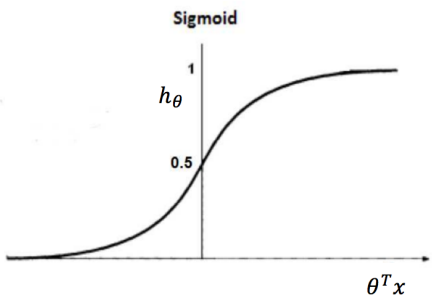
- Therefore, the decision boundary of logistic regression is nothing but a linear hyperplane
- Hence, $y = 1$ if $\theta^T x \geq 0$; otherwise, $y = 0$

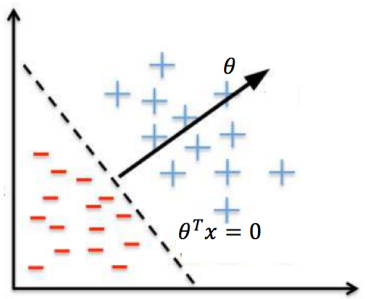
Interpreting The Probabilities ...

- Recall that

$$\Pr(y = 1 \mid x; \theta) = \frac{1}{1 + \exp(-\theta^T x)}$$

- The "score" $\theta^T x$ is also a measure of distance of x from the hyperplane (the score is positive for pos. examples, and negative for neg. examples)





- High positive score: High probability of label 1
- High negative score: Low probability of label 1 (high prob. of label 0)

Logistic Regression Formulation

- Logistic regression model $h_\theta(x) = g(\theta^T x) = 1/(1 + e^{-\theta^T x})$
- Assume $\Pr(y = 1 \mid x; \theta) = h_\theta(x)$ and $\Pr(y = 0 \mid x; \theta) = 1 - h_\theta(x)$, then we have

$$\Pr(y \mid x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

- If we assume $y \in \{-1, 1\}$ instead of $y \in \{0, 1\}$, then

$$\Pr(y \mid x; \theta) = \frac{1}{1 + \exp(-y\theta^T x)}$$

- Assuming the training examples were generated independently, we define the likelihood of the parameters as

$$\begin{aligned} L(\theta) &= \Pr(Y \mid X; \theta) \\ &= \prod_i \Pr(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_i (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}} \end{aligned}$$

Logistic Regression Formulation (Contd.)

- Maximize the log likelihood

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \left(y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}) \right)$$

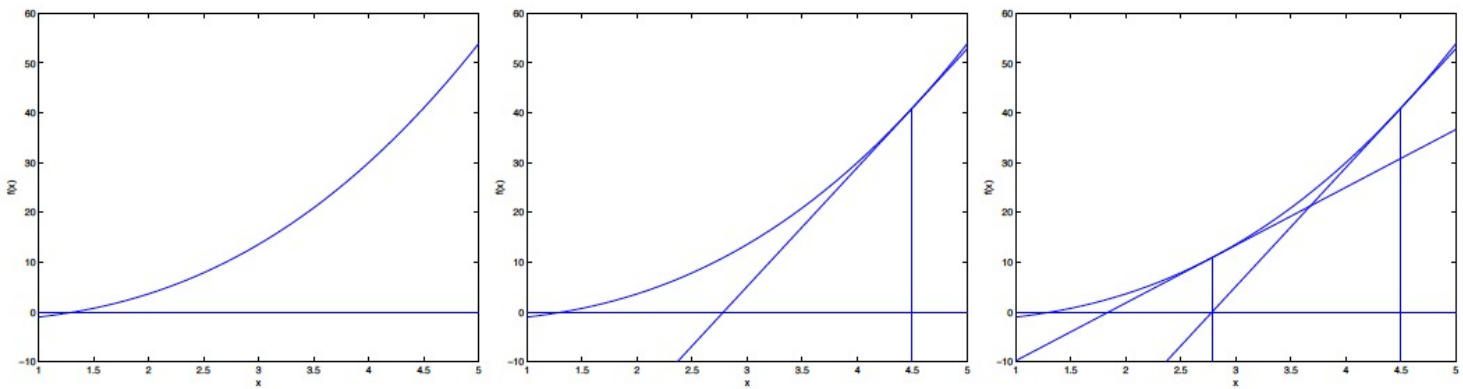
- Gradient ascent algorithm
  - $\theta_j \leftarrow \theta_j + \alpha \bigtriangledown_{\theta_j} \ell(\theta)$ for $\forall j$, where

$$\frac{\partial}{\partial \theta_j} \ell(\theta) = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

Newton's Method

- Given a differentiable real-valued $f : \mathbb{R} \to \mathbb{R}$, how can we find x such that $f(x) = 0$?
- The following iteration is performed until a sufficiently accurate value is achieved

$$x \leftarrow x - \frac{f(x)}{f'(x)}$$



Newton's Method (Contd.)

- To maximize $f(x)$, we have to find the stationary point of $f(x)$ such that $f'(x) = 0$.
- According to Newton's method, we have the following update

$$x \leftarrow x - \frac{f'(x)}{f''(x)}$$

- Newton-Raphson method:
  For $\ell : \mathbb{R}^n \to \mathbb{R}$, we generalization Newton's method to the multidimensional setting

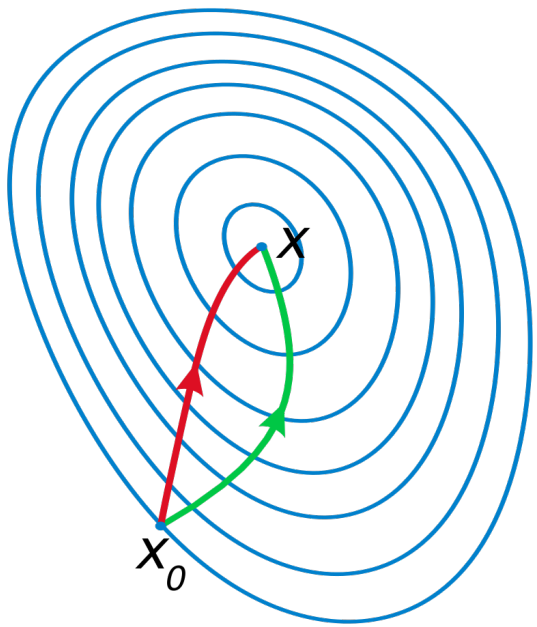
$$\theta \leftarrow \theta - H^{-1} \bigtriangledown_\theta \ell(\theta)$$

  where H is the Hessian matrix

$$H_{i,j} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j}$$

Newton's Method (Contd.)

- Higher convergence speed than (batch) gradient descent
- Fewer iterations to approach the minimum
- However, each iteration is more expensive than the one of gradient descent
  - Finding and inverting an $n \times n$ Hessian



More details about Newton's method can be found at `https://en.wikipedia.org/wiki/Newton%27s_method`

Generalized Linear Models (GLMs)

- So far,
  - Regression problem $y \mid x; \theta \sim \mathcal{N}(\mu, \sigma^2)$
  - Classification problem $y \mid x; \theta \sim Bernoulli(\phi)$
- In fact, both of these methods are special cases of a broader family of models, so-called generalized linear models (GLMs)

The Exponential Family

- An exponential family is a set of probability distributions of the following form

$$p(y;\eta) = b(y)\exp(\eta^T \Phi(y) - a(\eta))$$

  - η is called the natural parameter (also called the canonical parameter)
  - $\Phi(y)$ is the sufficient statistic (we usually take $\Phi(y) = y$)
  - $b(y)$ is the underlying measure (e.g., counting measure or Lebesgue measure)
  - $a(\eta)$ is the log partition (or cumulant) function

The Exponential Family (Contd.)

- An exponential family is a set of probability distributions of the following form

$$\begin{aligned} p(y;\eta) &= b(y)\exp(\eta^T\Phi(y) - a(\eta)) \\ &= \frac{b(y)\exp(\eta^T\Phi(y))}{\exp(a(\eta))} \end{aligned}$$

- $\exp\left(a(\eta)\right)$ is a constant for normalization purpose, such that $a(\eta)$ is not a degree of freedom in the specification of an exponential family density

$$\text{Lebesgue integral} : \exp\left(a(\eta)\right) = \int_y b(y)\exp\left(\eta^T\Phi(y)\right)dy$$

- $\Phi(\cdot)$, $a(\cdot)$ and $b(\cdot)$ define a family of distributions parameterized by η

The Exponential Family (Contd.)

- For exponential family, we have

$$\int_y p(y;\eta)dy = \int_y b(y)\exp(\eta^T\Phi(y) - a(\eta))dy = 1$$

- Calculate derivative w.r.t. η

$$a'(\eta) = \int_y b(y)\exp(\eta^T\Phi(y) - a(\eta))\Phi(y)dy$$

The Exponential Family (Contd.)

- The log-likelihood function

$$\begin{aligned}
\ell(\eta) &= \log \prod_{i=1}^{m} p(y^{(i)}; \eta) \\
&= \sum_{i=1}^{m} \log p(y^{(i)}; \eta) \\
&= \sum_{i=1}^{m} \log b(y^{(i)}) - m a(\eta) + \eta^T \sum_{i=1}^{m} \Phi(y^{(i)})
\end{aligned}$$

- Maximizing $\ell(\eta)$

$$a'(\eta) = \frac{1}{m} \sum_{i=1}^{m} \Phi(y^{(i)})$$

- Sufficiency: The likelihood for η only depends on y through $\Phi(y)$
- When $m \to \infty$, $a'(\eta) \to E(\Phi(y))$

The Exponential Family (Contd.)

- Bernoulli distribution

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

  - $\eta = \log(\psi/(1-\psi))$ (and thus $\psi = 1/(1+e^{-\eta})$)
  - $\Phi(y) = y$
  - $a(\eta) = -\log(1-\psi) = \log(1+e^{\eta})$
  - $b(y) = 1$

The Exponential Family (Contd.)

- Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$

$$\begin{aligned} p(y; \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(\frac{\mu}{\sigma^2}y - \frac{1}{2\sigma^2}y^2 - \frac{1}{2\sigma^2}\mu^2 - \log\sigma\right) \end{aligned}$$

  - $\eta = \begin{bmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{bmatrix}$
  - $\Phi(y) = \begin{bmatrix} y \\ y^2 \end{bmatrix}$
  - $a(\eta) = \frac{\mu^2}{2\sigma^2} + \log\sigma = -\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2)$
  - $b(y) = 1/\sqrt{2\pi}$

The Exponential Family (Contd.)

- Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where σ^2 is a constant (e.g. $\sigma = 1$)

$$\begin{aligned} p(y; \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y-\mu)^2\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \cdot \exp\left(\mu y - \frac{1}{2}\mu^2\right) \end{aligned}$$

  - $\eta = \mu$
  - $\Phi(y) = y$
  - $a(\eta) = \mu^2/2 = \eta^2/2$
  - $b(y) = 1/\sqrt{2\pi}\exp\left(-1y^2 2\right)$

The Exponential Family (Contd.)

- Piosson distribution

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

  - $\eta = \log\lambda$
  - $\Phi(y) = y$
  - $a(\eta) = \lambda = e^{\eta}$
  - $b(y) = 1/y!$

The Exponential Family (Contd.)

- Multinomial distribution ($\sum_{i=1}^{k} y_i = 1$)

$$
\begin{aligned}
p(y;\psi) &= \psi_1^{y_1}\psi_2^{y_2}\cdots\psi_k^{y_k} \\
&= \exp\left(\sum_{i=1}^{k} y_i \log\psi_i\right) \\
&= \exp\left(\sum_{i=1}^{k-1} y_i \log\psi_i + (1-\sum_{i=1}^{k-1} y_i)\log(1-\sum_{i=1}^{k-1}\psi_i)\right) \\
&= \exp\left(\sum_{i=1}^{k-1} \log(\frac{\psi_i}{1-\sum_{i=1}^{k-1}\psi_i})y_i + \log(1-\sum_{i=1}^{k-1}\psi_i)\right)
\end{aligned}
$$

The Exponential Family (Contd.)

- Multinomial distribution ($\sum_{i=1}^{k} y_i = 1$)
  - $\eta_k = \log(\frac{\psi_k}{1-\sum_{i=1}^{K-1}\psi_i}) = \log\frac{\pi_k}{\pi_K}$
  - $\Phi(y) = y$
  - $a(\eta) = -\log(1 - \sum_{i=1}^{K-1}\psi_i)$
  - $b(y) = 1$

Constructing GLMs

- Assumptions
  - $y \mid x; \theta \sim \text{ExponentialFamily}(\eta)$
  - $\Phi(y) = y$ which means $h_\theta(x) = \text{E}[y|x; \theta]$
  - $\eta = \theta^T x$

Constructing GLMs (Contd.)

- Ordinary least square
  - $y \mid x \sim \mathcal{N}(\mu, \sigma^2)$
  - Since $\eta = \mu$, we have

$$h_\theta(x) = E[y \mid x; \theta] = \mu = \eta = \theta^T x$$

Constructing GLMs (Contd.)

- Logistic regression
  - For Bernoulli distribution, $\psi = 1/(1 + e^{-\eta})$
  - Therefore, we have

$$h_\theta(x) = E[y \mid x; \theta] = \psi = 1/(1 + e^{-\eta}) = 1/(1 + e^{-\theta^T x})$$

Multiclass Classification

- Multiclass (or multinomial) classification is the problem of classifying instances into one of the more than two classes
- The existing multiclass classification techniques can be categorized into
  - Transformation to binary
  - Extension from binary
  - Hierarchical classification

Transformation to Binary

- One-vs.-rest (one-vs.-all, OvA or OvR, one-against-all, OAA) strategy is to train a single classifier per class, with the samples of that class as positive samples and all other samples as negative ones
  - Inputs: A learning algorithm L, training data $\{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,m}$ where $y^{(i)} \in \{1, ..., K\}$ is the label for the sample $x^{(i)}$
  - Output: A list of classifier f_k for $k \in \{1, \cdots, K\}$
  - Procedure: For $\forall k \in \{1, \cdots .K\}$, construct a new label $z^{(i)}$ for $x^{(i)}$ such that $z^{(i)} = 1$ if $y^{(i)} = k$ and $z^{(i)} = 0$ otherwise, and then apply L to $\{(x^{(i)}, z^{(i)})\}_{i=1,\cdots,m}$to obtain f_k. Higher $f_k(x)$ implies hight probability that x is in class k
  - Making decision: $y^* = \arg\min_k f_k(x)$
  - Example: Using SVM to train each binary classifier

Transformation to Binary

- One-vs.-One (OvO) reduction is to train $K(K-1)/2$ binary classifiers
  - For the (s,t)-th classifier:
    - Positive samples: all the points in class s
    - Negative samples: all the points in class t
    - $f_{s,t}(x)$ is the decision value for this classifier such that larger $f_{s,t}(x)$ implies that label s has higher probability than label t
    - $f_{s,t}(x) = -f_{t,s}(x)$
  - Prediction:

$$f(x) = \arg\max_s \left(\sum_t f_{s,t}(x) \right)$$

  - Example: using SVM to train each binary classifier

Softmax Regression (Contd.)

- A classification problem where $y \in \{1, 2, \cdots, k\}$
- Multinomial distribution
  - A generalization of the binomial distribution
  - A trial has k outcomes and we perform the trial n times independently, the multinomial distribution gives the probability of any particular combination of numbers of successes for the various categories
  - Assuming $\psi_i = p(y = i; \psi)$ $(i = 1, \cdots, k)$ denote the probability of i-th outcomes, since $\sum_i \psi_i = 1$, the multinomial distribution can be parameterized by only $k - 1$ parameters $\{\psi_1, \psi_2, \cdots, \psi_{k-1}\}$
- Question: Can we express the multinomial as an exponential family distribution?

Softmax Regression (Contd.)

- Defining $\Phi(y) \in \mathbb{R}^{k-1}$

$$\Phi(1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Phi(2) = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \cdots \Phi(k-1) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \Phi(k) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- Assuming $\mathbf{1}\{True\} = 1$ and $\mathbf{1}\{False\} = 0$, we have

$$p(y;\psi) \quad = \quad \psi_1^{\mathbf{1}\{y=1\}} \psi_2^{\mathbf{1}\{y=2\}} \cdots \psi_k^{\mathbf{1}\{y=k\}}$$

Softmax Regression (Contd.)

$$
\begin{aligned}
p(y;\psi) &= \psi_1^{\mathbf{1}\{y=1\}}\psi_2^{\mathbf{1}\{y=2\}}\cdots\psi_k^{\mathbf{1}\{y=k\}} \\
&= \psi_1^{\mathbf{1}\{y=1\}}\psi_2^{\mathbf{1}\{y=2\}}\cdots\psi_k^{1-\sum_{i=1}^{k-1}\mathbf{1}\{y=i\}} \\
&= \psi_1^{(\Phi(y))_1}\psi_2^{(\Phi(y))_2}\cdots\psi_k^{1-\sum_{i=1}^{k-1}(\Phi(y))_i} \\
&= \exp\left[\log\left(\psi_1^{(\Phi(y))_1}\psi_2^{(\Phi(y))_2}\cdots\psi_k^{1-\sum_{i=1}^{k-1}(\Phi(y))_i}\right)\right] \\
&= \exp\left[\sum_{i=1}^{k-1}(\Phi(y))_i\log\psi_i + \left(1-\sum_{i=1}^{k-1}(\Phi(y))_i\right)\log\psi_k\right] \\
&= \exp\left[\sum_{i=1}^{k-1}(\Phi(y))_i\log(\psi_i/\psi_k) + \log\psi_k\right]
\end{aligned}
$$

Softmax Regression (Contd.)

- We have

$$\begin{aligned} p(y;\psi) &= \exp\left(\sum_{i=1}^{k-1}(\Phi(y))_i \log(\psi_i/\psi_k) + \log\psi_k\right) \\ &= b(y)\exp(\eta^T\Phi(y) - a(\eta)) \end{aligned}$$

  - $\eta = \begin{bmatrix} \log(\psi_1/\phi_k) \\ \log(\psi_2/\phi_k) \\ \cdots \\ \log(\psi_{k-1}/\phi_k) \end{bmatrix}$
  - $a(\eta) = -\log(\psi_k)$
  - $b(y) = 1$

Softmax Regression (Contd.)

- The link function is

$$\eta_i = \log \frac{\psi_i}{\psi_k}, \ \ for \ \forall i = 1, 2, \cdots, k$$

- Then, we have

$$e^{\eta_i} = \psi_i / \psi_k, \quad \psi_i = \psi_k e^{\eta_i}, \quad \psi_k \sum_{i=1}^{k} e^{\eta_i} = \sum_{i=1}^{k} \psi_i = 1$$

- Therefore, the response function is derived as follows

$$\psi_k = 1 / \sum_{i=1}^{k} e^{\eta_i}, \qquad \psi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$

- The function mapping from η's to the ψ's is so-called the softmax function

Softmax Regression (Contd.)

- Since $\eta_i = \theta_i^T x$ (for $i = 1, 2, \cdots, k-1$), we have the following softmax model

$$p(y = i|x, w) = \psi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}}$$

- Our hypothesis outputs

$$h_\theta(x) = \mathrm{E}[\Phi(y)|x; \theta] = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{k-1} \end{bmatrix} = \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \\ \frac{\exp(\theta_2^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_{k-1}^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \end{bmatrix}$$

Softmax Regression (Contd.)

- Training data set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})\}$
- Log-likelihood function

$$\ell(\theta) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}; \theta) = \sum_{i=1}^{m} \log \prod_{r=1}^{k} \left(\frac{e^{\theta_r^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \right)^{\mathbf{1}\{y^{(i)}=r\}}$$

- Maximizing $\ell(\theta)$ through gradient ascent or Newton's method

Thanks!

Q & A