8. Supervised Learning: Softmax Regression

Softmax Regression: Model

Setting

We consider a K-class classification problem where the the output value $y \in \{1,2,\dots,K\}$ and the input variable $\boldsymbol{x} \in \mathbb{R}^{m+1}$.

We also consider that we have access to labelled data $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right), i = 1,\dots,n$, with $y^{(i)} \in \{1,2,\dots,K\}$ and $\boldsymbol{x}^{(i)} = \begin{bmatrix} 1 & x_1^{(i)} & x_2^{(i)} & \cdots & x_m^{(i)} \end{bmatrix}$

We want to estimate $Pr(y|\boldsymbol{x})$ for $y \in \{1,2,\dots,K\}$ given $\boldsymbol{x} = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{m+1}$

Model

The hypothesis is such that:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \begin{bmatrix} Pr(y=1|\boldsymbol{x};\boldsymbol{\theta}_1) \\ \cdots \\ Pr(y=K|\boldsymbol{x};\boldsymbol{\theta}_K) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\theta_k^t \cdot x}} \cdot \begin{bmatrix} e^{\boldsymbol{\theta}_1^t \cdot \boldsymbol{x}} \\ \cdots \\ e^{\boldsymbol{\theta}_K^t \cdot \boldsymbol{x}} \end{bmatrix}$$

where $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K \in \mathbb{R}^{m+1}$ are the parameters of the model

Softmax Regression: Cost Function

How does the learning algorithm select the linear hypothesis / model? We need a cost function…

Cost Function

The softmax cost function is given by:

$$J(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} 1\{y^{(i)} = k\} \cdot \log_2 \frac{e^{\boldsymbol{\theta}_k^t \cdot x^{(i)}}}{\sum_{j=1}^{K} e^{\theta_j^t \cdot x^{(i)}}}$$

where $1\{\cdot\}$ is the indicator function given by: $1\{\text{true statement}\} = 1$ and $1\{\text{false statement}\} = 0$.

Why this cost function?

This cost function generalizes the logistic regression cost function.

Softmax Regression: Learning Algorithm

Learning Algorithm

We should select the logistic regression model parameters that minimize the cost function as follows:

$$\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_K^* = \underset{\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_K}{\operatorname{argmin}} J(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$$

New Predictions

The new predictions are then given by:

$$\begin{bmatrix} Pr(y = 1|\boldsymbol{x}; \boldsymbol{\theta}_1^*) \\ \dots \\ Pr(y = K|\boldsymbol{x}; \boldsymbol{\theta}_K^*) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\boldsymbol{\theta}_k^{*t} \cdot \boldsymbol{x}}} \cdot \begin{bmatrix} e^{\boldsymbol{\theta}_1^{*t} \cdot \boldsymbol{x}} \\ \dots \\ e^{\boldsymbol{\theta}_K^{*t} \cdot \boldsymbol{x}} \end{bmatrix}$$

There is also no closed-form solution for the optimal parameters

⇩

Gradient descent algorithm

Softmax Regression: Learning Algorithm

Considerations

- The softmax regression has a redundant set of parameters because the hypothesis is not affected by subtracting some fixed vector $\boldsymbol{\psi}$ to each of our parameter vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ i.e.

$$Pr(y = k|\boldsymbol{x}) = \frac{e^{(\boldsymbol{\theta}_k^t - \boldsymbol{\psi})\cdot \boldsymbol{x}}}{\sum_{j=1}^{K} e^{(\boldsymbol{\theta}_j^t - \boldsymbol{\psi})\cdot \boldsymbol{x}}} = \frac{e^{\psi \cdot \boldsymbol{x}} \cdot e^{\boldsymbol{\theta}_k^t \cdot \boldsymbol{x}}}{e^{\psi \cdot \boldsymbol{x}} \cdot \sum_{j=1}^{K} e^{\boldsymbol{\theta}_j^t \cdot \boldsymbol{x}}} = \frac{e^{\boldsymbol{\theta}_k^t \cdot \boldsymbol{x}}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_j^t \cdot \boldsymbol{x}}}$$

- Therefore, the softmax regression model is overparameterized in the sense that there are multiple parameter settings that give rise to exactly the same hypothesis function.

- The cost function is convex so gradient descent will not run into local optima problem

- The cost function Hessian is non-invertible so Newton's methods will run into numerical problems