

All you need to know about logistic regression

Vidit Jain

February 24, 2006

1 Introduction

This document will be frequently updated as I learn more about logistic regression and related stuff. I am trying to put together all the properties of logistic regression and connections with common generative models at a single place so that people like me, can understand it better without spending too much time in hunting for references. Please email me any corrections or suggestions to make it better, useful and self-contained. Alternatively, post your comments on this topic on my blog <http://vimsu99.blogspot.com>.

2 Preliminaries

Let us assume that each observation consists of features $x \in \mathcal{X}$. Consider a random variable, $\xi : \mathcal{X} \rightarrow \mathcal{C}_n$. In a classification task, we are interested in learning $P(\xi(x) = c_i)$, where $c_i \in \mathcal{C}_n$ are the required class labels. For a concise representation of this probability term, we will use $p(c_i|x)$ in our discussion. Also we will refer to class conditional probabilities, $P(\{x : \xi(x) = c_i\})$ as $p(x|c_i)$.

3 Gaussian Mixture Models

Assuming m -dimensional gaussian distribution for each mixture component, $p(x|c_i)$ i.e.,

$$p(x|c_i) \sim \mathcal{N}(\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}.$$

The joint probability, $p(x, c_i)$, is given by

$$p(x, c_i) = p(c_i)p(x|c_i) = \exp(\alpha_i + \beta_i x + x^T \Omega_i x),$$

where

$$\begin{aligned} \alpha_i &= -\mu_i^T \Sigma_i^{-1} \mu_i - \frac{1}{2} \ln((2\pi)^m |\Sigma_i|) + \ln(p(c_i)) \\ \beta_i &= \mu_i^T \Sigma_i^{-1} + \Sigma_i^{-1} \mu_i \\ \Omega_i &= -\Sigma_i. \end{aligned}$$

Now, the quantity of interest,

$$\begin{aligned} p(c_i|x) &= \frac{p(x, c_i)}{p(x)} = \frac{p(x, c_i)}{\sum_j p(x, c_j)} \\ &= \frac{\exp(\alpha_i + \beta_i x + x^T \Omega_i x)}{\sum_j \exp(\alpha_j + \beta_j x + x^T \Omega_j x)}. \end{aligned} \quad (1)$$

If the covariance matrices, Σ_i , are same for all the classes then Ω_i are also same for all the classes. In this special case, Equation 1 reduces to the softmax functional form for the conditional (posterior) probability,

$$p(c_i|x) = \frac{\exp(\alpha_i + \beta_i x)}{\sum_j \exp(\alpha_j + \beta_j x)}. \quad (2)$$

3.1 Logistic Regression

When we have two mixture components, Equation 2 reduces to logistic regression as shown below.

$$p(c_1|x) = \frac{1}{1 + \exp(\alpha + \beta x)} = \frac{1}{1 + \exp(\boldsymbol{\theta}^T \mathbf{x})}, \quad (3)$$

where $\alpha = \alpha_0 - \alpha_1$, $\beta = \beta_0 - \beta_1$, $\mathbf{x} = [1 \ x]$ and $\boldsymbol{\theta}^T = [\alpha^T \ \beta^T]$.

Maximum likelihood estimation is often used to estimate the parameters of logistic regression on available pair of feature vector and class labels, (\mathbf{X}, \mathbf{C}) . For binary classification task, we have a binary response, $c \in \mathbf{C}$, associated with each observed feature vector, $\mathbf{x} \in \mathbf{X}$. Here, ξ is a bernoulli random variable with unknown parameter p . Each observation has a different parameter $p_k = p(c_1|\mathbf{x}_k)$ given by Equation 3 so they are not identically distributed but they are assumed to be independent.

The probability of the data \mathbf{X} given the parameter $\boldsymbol{\theta}$ (equivalently, the likelihood of the parameter $\boldsymbol{\theta}$ given the data \mathbf{x}) is given by

$$\begin{aligned} L(\boldsymbol{\theta}|\mathbf{x}_k, c_k) &= p(\mathbf{x}_k|\boldsymbol{\theta}) = p_k^{c_k} (1 - p_k)^{1-c_k} \\ L(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C}) &= \prod_{k=1}^N p(\mathbf{x}_k|\boldsymbol{\theta}) = \prod_{k=1}^N p_k^{c_k} (1 - p_k)^{1-c_k} \\ \mathit{l}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C}) &= \log L(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C}) = \sum_{k=1}^N c_k \log p_k + \sum_{k=1}^N (1 - c_k) \log(1 - p_k) \end{aligned}$$

Maximizing likelihood is equivalent to maximizing log-likelihood as log is a monotonic function.

$$\frac{\partial \mathit{l}}{\partial \boldsymbol{\theta}} = \sum_{k=1}^N \frac{c_k - p_k}{p_k(1 - p_k)} \frac{\partial p_k}{\partial \boldsymbol{\theta}}$$

$$= \sum_{k=1}^N (c_k - p_k) \mathbf{x}_k = 0, \quad (4)$$

which are $m + 1$ equations which are non-linear in θ . To solve equations (4), we use Newton-Raphson algorithm ¹, which requires the second derivative or Hessian matrix

$$\mathcal{H} = \frac{\partial^2 \mathbf{l}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = - \sum_{k=1}^N \mathbf{x}_k \mathbf{x}_k^T p_k (1 - p_k) \leq 0.$$

Thus, \mathcal{H} is negative semi-definite and thus log-likelihood, \mathbf{l} , is concave and we can use an iterative gradient ascent method (Newton-Raphson algorithm) for maximizing \mathbf{l} . Starting with some initial estimate for θ_0 , Newton-Raphson update is

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \left(\frac{\partial^2 \mathbf{l}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \right)^{-1} \frac{\partial \mathbf{l}}{\partial \boldsymbol{\theta}} \\ &= \boldsymbol{\theta}_t + (\mathbf{X}^T \mathbf{W} \mathbf{X}) \mathbf{X}^T (\mathbf{C} - \mathbf{P}), \end{aligned}$$

where \mathbf{W} is the diagonal matrix with $p_k(1 - p_k)$ as the k^{th} entry. By doing some algebra on the above expression, this can be seen as solving weighted least squares problem with weight matrix as \mathbf{W} and response $\mathbf{z} = \mathbf{X} \boldsymbol{\theta}_t + \mathbf{W}^{-1} (\mathbf{C} - \mathbf{P})$.

3.2 Discussion

Here we have shown that discriminative or conditional training of mixture of gaussians with equal variances is the same as optimizing the parameters of derived softmax regression (equations 2). However, the converse is not true. In other words, an infinite number of sets of parameters correspond to a single softmax formulation. Some prior distribution over these parameters is required to obtain a unique model, which thereby defines the generative training of generative models. Also, we will show that logistic regression can result from conditional training of other common generative models.

¹The Newton-Raphson algorithm minimizes f using updates, $\theta_{t+1} = \theta_t - \frac{f(\theta)}{f'(\theta)}$. In this case, we will minimize $\frac{\partial \mathbf{l}}{\partial \boldsymbol{\theta}}$.