

You may use your class notes, the text, or any calculus books — please use no other references (including internet or other statistics texts). If you use Mathematica to derive results, you must include the notebook with your solution so I can see what you did.

1. Maximum-likelihood Cost Function for Multiclass Problems

This problem extends the cross-entropy error function to multiple classes. Suppose we have L classes $\omega_1, \dots, \omega_L$ and each example feature vector x is from an object that belongs to *one and only one* class. Suppose further that the class labeling scheme assigns a binary vector y with L components to each example with

$$y_i(x) = 1 \text{ if } x \in \omega_i \text{ and } y_j(x) = 0 \text{ for all } j \neq i .$$

That is, each vector y has *exactly one element* set equal to 1, and the other elements set equal to 0. We can then write the probability of the class label vector y for a sample with features x as a *multinomial* distribution

$$p(y|x) = \prod_{i=1}^L \alpha_i(x)^{y_i} \quad (1)$$

with $0 \leq \alpha_i(x)$ <https://eduassistpro.github.io/>

(a) We want to be sure that $p(y|x)$ is properly normalized

$$\sum_{\{y\}} p(y|x) = 1 \quad (2)$$

where the sum is over *the set of all allowed* vectors y . Show that this normalization condition requires that

$$\sum_{i=1}^L \alpha_i(x) = 1 \quad \forall x . \quad (3)$$

(To be clear, you should probably explicate the sum over the allowed label vectors by giving the first several terms in the sum in (2).)

- (b) Suppose we have a collection of N statistically independent samples with feature vectors x^a and label vectors y^a , $a = 1, 2, \dots, N$ (the superscript denotes the sample number). Write the likelihood of the data set

$$p(\{y^{(1)}, y^{(2)}, \dots, y^{(N)}\} | \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}, \alpha_1(x), \dots, \alpha_L(x)) \quad (4)$$

that follows from the likelihood for each data sample from equation (1).

- (c) Show that maximizing the log-likelihood of the entire data set is equivalent to minimizing the cost function

$$\mathcal{E} = - \sum_{a=1}^N \sum_{i=1}^L y_i^a \log \alpha_i(x^a) . \quad (5)$$

2. Extension of Logistic Regression to Multi-Class Problems

In logistic regression, we have a two classes and we model the posterior for the *single* class label $y \in \{0, 1\}$ as

$$\alpha(x) \equiv p(y = 1|x) = \frac{1}{1 + \exp(V^T x + \nu)} \quad (6)$$

where V and ν are the (vector and scalar respectively) parameters in the model. We fit V and ν to data by minimizing the cross-entropy error function

$$\mathcal{E} = -\log p(\{y\}|\{x\}) = \sum_{a=1}^N y^a \log \alpha(x^a) + (1 - y^a) \log(1 - \alpha(x^a)) \quad (7)$$

We can fit the two-class problem into the framework in Problem 1. We use two class labels y_i , $i = 1, 2$ with $y_1 = 1, y_2 = 0$ if the example is in class ω_1 , and $y_1 = 0, y_2 = 1$ if the example is in class ω_2 .

- (a) A natural model for the class posteriors is the *soft-max* function

$$\alpha_i(x) = \frac{\exp g_i(x)}{\sum_{j=1}^2 \exp g_j(x)} \quad (8)$$

where $-\infty < g_i(x) < \infty$. Show that the softmax function guarantees that

and <https://eduassistpro.github.io/>

- (b) Show that for our two-class case the soft-max forms of the posteriors are

$$\alpha_1(x) = \frac{1}{1 + \exp(g_2 - g_1)}$$

and

$$\alpha_2(x) = \frac{1}{1 + \exp -(g_2 - g_1)} \quad .$$

Thus we really need only one $g(x)$, and a familiar candidate is the logistic regression choice $g_2 - g_1 = V^T x + \nu$.

- (c) We fit the parameters V, ν by minimizing the cost function derived in problem 1 (Eqn.5)

$$\mathcal{E} = -\sum_{a=1}^N \sum_{i=1}^2 y_i^a \log \alpha_i(x^a) \quad (9)$$

Show that for the two-class case (with our choice of class labels) this error function reduces to the cross-entropy function Eqn. (7).

This extends to the general L-class case. The $g_i(x)$ functions can be linear functions of x as in logistic regression. They can also be realized by more complicated functions — for example, the outputs of a neural network.