Applied Machine Learning: Tutorial Number 2

September 2022

1. Consider using logistic regression for a two-class classification problem in two dimensions:

$$p(y = 1|\boldsymbol{x}) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

Here σ denotes the logistic (or sigmoid) function $\sigma(z) = 1/(1 + \exp(-z))$, y is the target which takes on values of 0 or 1, $\boldsymbol{x} = (x_1, x_2)$ is a vector in the two-dimensional input space, and $\boldsymbol{w} = (w_0,\ w_1, w_2)$ are the parameters of the logistic regressor.

(a) Consider a weight vector $\boldsymbol{w}_A = (-1, 1, 0)$. Sketch the decision boundary in $\boldsymbol{x}$ space corresponding to this weight vector, and mark which regions are classified with labels 0 and 1.

(b) Consider a second weight vector $\boldsymbol{w}_B = (5, -5, 0)$. Again sketch the decision boundary in $\boldsymbol{x}$ space corresponding to this weight vector, and mark which regions are classified with labels 0 and 1.

(c) Plot $p(y = 1|\boldsymbol{x})$ as a function of x_1 for both $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$, and comment on any differences between the two.

2. Consider the logistic regression setup in the previous questions, but with a new weight vector $\boldsymbol{w} = (0, -1, 1)$. Consider the following data set:

Instance	x_1	x_2	Class
0	0.5	-0.35	$-$
1	-0.1	0.1	$-$
2	-1.2	1.0	$+$

(a) Compute the gradient of the *negative log likelihood* of the logistic regression model for this data set.

(b) Suppose that we take a single step of gradient descent with step size $\eta = 3.0$. What are the updated values for the model weights?

(c) Do the new weights do a better job of classifying the three training instances above?

It will help you to remember the following facts:

- The negative log-likelihood in logistic regression is

$$\begin{aligned}\mathrm{NLL}(\boldsymbol{w}) &= -\frac{1}{N}\sum_{i=1}^{N} \log p(y = y_i|\boldsymbol{x}_i; \boldsymbol{w}) \\ &= -\frac{1}{N}\sum_{i=1}^{N} [y_i \log p(y = 1|\boldsymbol{x}_i; \boldsymbol{w}) + (1 - y_i) \log p(y = 0|\boldsymbol{x}_i; \boldsymbol{w})]\end{aligned}$$

- The partial derivative of the negative log-likelihood with respect to a parameter w_d is

$$\frac{\partial \mathrm{NLL}}{\partial w_d} = \frac{1}{N}\sum_{i=1}^{N}(\sigma(\boldsymbol{w}^\top \boldsymbol{x}_i) - y_i)x_{id}$$

- To minimize a function $\mathrm{NLL}(\boldsymbol{w})$, we use the gradient *descent* rule, which is

$$\boldsymbol{w}' \leftarrow \boldsymbol{w} - \eta \nabla \mathrm{NLL}$$

3. You have a collection of 1000 nature photographs which were taken under many different conditions. All of the images are of size 300×300 pixels. You wish to develop a binary classifier that labels a photograph as to whether or not it depicts a sunny day on a beach. The images have been pre-processed in the following manner:

- Each image $i \in \{1 \ldots 1000\}$ is partitioned nine regions $R_{i,1} \ldots R_{i,9}$. Each region is 100×100 pixels. The regions are arranged in a 3×3 grid, so that the region R_{i1} is the top-left corner of image i, the region R_{i2} is the top middle portion of the image, and so on.
- For each region $R_{i,j}$, we compute the average *hue*[1] of pixels within the region $R_{i,j}$. The hue value is quantised into 7 discrete bins: "red", "orange", "yellow", "green", "blue", "indigo" and "violet".

(a) What features would you use to describe the data given the description above?

(b) How many features are there? Are they categorical, ordinal or numeric?

(c) What values can they take on?

[1]The *hue* is a scalar representation of color. It ranges from $0°$ to $360°$. For example, colors with hues around $0°$ look red, hues around $120°$ look blue, and hues around $240°$ look green.