Machine Learning, Lecture 11: Multiclass classification

S. Nõmm

[1]Department of Computer Science, Tallinn University of Technology

16.04.2015

Multiclass classification

- One versus all.
- All versus one.
- Classification tree.
- Naïve Bayes
- Maximum entropy model (multiclass logistic regression)

One versus All

Supervised learning

One has to assign one of K labels to a given vector

- Train K models (binary classifiers) such that:
- Training data: for the classifier k:
  - let the positive examples be all the points in class k,
  - let the negative examples be all the points not in a class k
- Predicting a class of a new element:
  - Make prediction with each model.
  - Add the results (-1 or 1) to the respective component of the score vector.

$$f(x) = \arg\max_i f_i(x).$$

All versus all

Supervised learning

One has to assign one of K labels to a given vector

- Train $K(K-1)/2$ models (binary classifiers), such that for each i-th and j-th class pairs:
  - let the positive examples be all the points in class i
  - let the negative examples be all the points in class j
- Predicting a class of a new element:
  - Make prediction with each model

$$f(x) = \arg\max_i \Big(\sum_j f_{i,j}(x) \Big).$$

Classification tree

- Build binary tree of binary classifiers
- With K classes $K - 1$ classifiers are necessary
- At the root, half of the classes are considered positive and the other half negative
- Knowledge of the data structure is necessary.

Bayes theorem

- Let us suppose that there k classes are given.
- The *posterior probability* of a class C_k for an input x is:

$$p(C_k \mid x) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(x)}$$

- $p(\boldsymbol{x} \mid C_k)$ is the likelihood, $p(C_k)$ is the *prior probability*, $p(x)$ is the *marginal data likelihood*.
- $p(C_k)$ is the probability of a class $p(C_k)$ *a priori*, before getting about any knowledge about the data.
- $p(C_k \mid \boldsymbol{x})$ is the class probability *a posteriori*, after getting knowledge about the data.
- Bayes theorem updates prior distribution into posterior on the basis of empiric information.

Conditional and unconditional independence

- If X and Y are *unconditionally independent* then their joint distribution is the product of the marginal distributions:

$$X \perp Y \Leftrightarrow p(X, Y) = p(X)p(Y)$$

- If the influence is mediated through a third variable Z, then X and Y are said to be *conditionally independent*

$$X \perp Y \mid Z \Leftrightarrow p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z)$$

- Conditional independence does not imply unconditional independence and vice versa:

$$X \perp Y \mid Z \not\Leftrightarrow X \perp Y$$

Example: Spam detection

- Inputs $\boldsymbol{x}$ are the e-mail messages (text documents)
- m labeled training pairs (x_i, y_i), where $y_i \in \{0, 1\}$. 0 - indicates "clear" message and 1 - spam
- Task is to classify a new e-mail spam/not a spam
- According to Bayes theorem

$$p(y \mid x) = \frac{p(\boldsymbol{x} \mid y)p(y)}{p(\boldsymbol{x})} \propto p(\boldsymbol{x} \mid y)$$

- The demoniator may be computed as

$$p(\boldsymbol{x}) = \sum_{y'} p(\boldsymbol{x} \mid y')p(y')$$

Feature representation

- Amount of the training data may pose a problem in computing likelihood $p(\boldsymbol{x} \mid y)$. (Low amout of training data may prevent reliable computation of the likelihood).
- Consider the document as the set of words
- for the given vocabulary V present each document as a binary vector.
- If word belong to the vocabulary corresponding element take the value 1 and 0 otherwise.
- This approach will lead to the following likelihood function

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{|V|} p(x_j \mid y)$$

Naïve Bayes assumption

- Likelihood is computed as:

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{n} p(x_j \mid y)$$

- *Naïve Bayes assumption:* the features are conditionally independent given the class label.
- the word *naïve* reveres to the fact that actually features are not expected to be independent or conditionally independent.
- Model has relatively few parameters and therefore immune to overfilling.

Naïve Bayes model

- Parameters of the model

$$\begin{aligned}\theta_{j|y=1} &= p(x_1 = 1 \mid y = 1)\\ \theta_{j|y=0} &= p(x_1 = 1 \mid y = 0)\\ \theta_y &= p(y = 1)\end{aligned}$$

- The MLE estiamtes of the parameters are:

$$\begin{aligned}\theta_{j|y=1} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 1)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}\\ \theta_{j|y=0} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 0)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 0)}\\ \theta_y &= \frac{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}{m}\end{aligned}$$

Prediction with naïve Bayes model

- the goal is to find wether a new element is of class 1 or 0 (in the example of spam filtering wether given e-mail message is spam or not).
- According to Bayes theorem.

$$p(y = 1 \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta}) = p(y = 1 \mid \theta) \prod_{j=1}^{n} p(x_{i,j} \mid y = 1, \boldsymbol{\theta})$$

$$p(y = 0 \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta}) = p(y = 0 \mid \theta) \prod_{j=1}^{n} p(x_{i,j} \mid y = 0, \boldsymbol{\theta})$$

- Predict the class with highest posterior probability:

$$y^* = \arg \max_{y \in \{0,1\}} p(y \mid \boldsymbol{x}, \boldsymbol{\theta})$$

Drawbacks related to MLE estimates

In the context of spam example

- Let us suppose that an e-mail contains a word with the index w which is in vocabulary but was never observed during the training.
- This will lead

$$
\begin{aligned}
p(x_w \mid y = 1) &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,w} = 1, y_i = 1)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)} = 0 \\
p(x_w \mid y = 0) &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,w} = 1, y_i = 0)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 0)} = 0
\end{aligned}
$$

- In this case posterior probabilities of predicting class are 0. *Explain why ?*

Smoothing

- If training set does not contain "something" does not necessarily mea that the probability of this "something" is 0. ("Black swan" case).
- *Smoothing* is used to overcome the problem. Basic idea is is to take away some probability mass from the observed values and to preserve it to the unobserved values.
- *Add - one smoothing* is one of the simplest techniques.
- $$\begin{aligned}\theta_{j|y=1} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j}=1, y_i=1)+1}{\sum_{i=1}^{m} \mathbb{I}(y_i=1)+2} \\ \theta_{j|y=0} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j}=1, y_i=0)+1}{\sum_{i=1}^{m} \mathbb{I}(y_i=0)+2}\end{aligned}$$