Classification Methods on Kaggle Titanic Data Set

Daniel Mukasa
Ian Hunt-Issak
Hillary Pan

Abstract

Well put our abstract here, however we plan to do that should involve some collaboration. The following sections will essentially be what is written on the instructions for the project

1. Introduction and Background

Kaggle is an online data science website devoted to the development of data science and machine learning methods. In order to stimulate this development, the Kaggle community has created a series of challenges for users to tackle. These challenges are then accompanied by larger real world data science problems that are posted by companies such as Google, Amazon and Netflix.

The purpose of the introductory challenges is to have users who have just completed a first course in machine learning implement what they know on this classification task. While the introductory problems alone do not solve many pertinent problems in the world around us, they set users up to handle more complex questions that machine learning can be used for. There is thus an overarching goal of showing deep trends that may only be seen with machine learning methods, and give users an understanding of the power of machine learning.

The Titanic disaster challenge is the first in this series of challenges posted on Kaggle. This dataset provides information of various passengers on the RMS Titanic and asks users to predict which passengers would have survived this disaster. This data set contain essential features such as gender, age, passenger class and more that are described in the experiments and results section. As one may expect, however, this data is filled with missing information, as not all information on any given passenger could be recorded. Such a classification task may easily be assessed via the Naive Bayes algorithm, Logistic regression, or Gaussian Process Classification, as all of these methods may be easily used for binary classification, and elegantly handle cases of missing data. We have thus implemented these models and compared there effectiveness of predicting which passengers would have survived this fateful day.

2. Models and Method

We have implemented three separate models on the Kaggle Titanic challenge data set in hopes of predicting who survived this disaster.

2.1. Naive Bayes

The Naive Bayes is one of the most simplistic, yet surprisingly effective, classification methods in machine learning. This method takes the form of a Bayes classifier, a model that is defined in concordance to Bayes rule. Given a classification c, a dataset of feature vectors $x_1, x_2, .., x_n$ complied vertically into a column vector X where

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \tag{1}$$

and corresponding classes in a column vector t, one defines a Bayes classifier as

$$P(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t})P(T_{new} = c|\mathbf{X}, \mathbf{t})}{p(\mathbf{x}_{new}|T_{new} = c', \mathbf{X}, \mathbf{t})P(T_{new} = c'|\mathbf{X}, \mathbf{t})} \tag{2}$$

where the terms in the numerator are defined as the class conditional distribution and the class prior respectively, as defined on page 169 of A First Course in Machine Learning.

Naive Bayes simplifies the structure of a Bayes classi-

Proceedings of the 1st Introduction to Probabilistic Models & Machine Learning Course, Oberlin, Ohio, USA, 2017.

fier by assuming each feature of the class conditional distribution is independent of one another. This simplification thus defined the first term in our numerator of equation 1 as

$$p(\mathbf{x}_n|t_n = c, \mathbf{X}, \mathbf{t}) = \prod_{d=1}^{D} p(x_{nd}|t_n = c, \mathbf{X}, \mathbf{t}) \qquad (3)$$

where D is the number of features in our feature vector. This allows for a very simple and quick calculation of the class conditional distribution, making this method much faster and easier to implement than most other machine learning methods. Along with these appealing features, this algorithm very easily handles missing data by simply assuming it does not occur. This therefore does not help or hurt the model, making this method especially appealing for the Titanic challenge.

The down falls of this method rest mainly in this naive assumption. Due to this assumption any information on the correlation between features is not captured in the model.This is of course extremely unrealistic to assume and is almost never the case. Along with this, this product presents a potential issue for zero counts in the data set. If a state of a feature vector never occurs then this class conditional likelihood will be zero. Therefore, the probability of a feature vector occurring with a feature state that has never been observed in a training set is zero. This however is fairly unlikely and over fits the model to the training set.

Despite these obvious downfalls of this method

This section is for Daniel

2.2. Logistic Regression

Binary logistic regression is another useful method we explored. This method measures the relationship between the categorical dependent variable, in this case whether the passenger survived, and the independent variables via the probability model

$$p(t|\mathbf{x}, \mathbf{w}) = \left(\frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}\right)^t \left(\frac{e^{-\mathbf{w}^T\mathbf{x}}}{1 + e^{-\mathbf{w}^T\mathbf{x}}}\right)^{1-t} \qquad (4)$$

where $\mathbf{x} = [x_1, ..., x_D]^T$ denotes the vector of input features, $\mathbf{w} = [w_1, ..., w_D]^T$ is the associated vector of model parameters, and $t \in \{0, 1\}$ denotes the output.

In order to determine the model parameter $\mathbf{w}$, an expression for the likelihood of the training data is necessary. Assuming all data points are i.i.d, the likelihood over the entire training data set can be written as

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P_n^{t_n}(1 - P_n)^{1-t_n} \qquad (5)$$

where notation in Eq. (3) is simplified such that

$$P_n = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}_n}} \qquad (6)$$

and $\mathbf{t} = [t_1, \ldots, t_N]^T$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$.

2.2.1. Maximum Likelihood

The log likelihood of the training data is

$$\log L = \sum_{n=1}^{N} (t_n \log P_n + (1 - t_n) \log(1 - P_n)) \qquad (7)$$

To maximize the likelihood,

$$\frac{\partial L}{\partial \mathbf{w}} = \cdots = \sum_{n=1}^{N} \mathbf{x}_n(t_n - P_n) = \mathbf{0} \qquad (8)$$

However, Eq. 7 cannot be solved analytically and must be solved numerically. We therefore utilize the Newton-Raphson optimization method, in which the Hessian matrix with respect to $\mathbf{w}$ is written as

$$\frac{\partial^2 \log g}{\partial \mathbf{w} \partial \mathbf{w}^T} = \cdots = -\frac{1}{\sigma^2}\mathbf{I}_D - \sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^T P_n(1 - P_n) \quad (9)$$

The solution procedure starts with an initial guess of $\mathbf{w}$, in this case $\mathbf{w} = \mathbf{0}$. Then a loop is started. In each iteration, the gradient of the log likelihood in Eq. 7, called residual $\mathbf{r}$ of the current iteration, is first evaluated, followed by an update to $\mathbf{w}$ as $\mathbf{w} = \mathbf{w} - \mathbf{H}_{MLE}^{-1}\mathbf{r}$. Iteration continues until the residual is small enough. The converged solution is the MLE of $\mathbf{w}$.

2.2.2. Bayesian

In the maximum likelihood approach, $\mathbf{w}$ is treated as a regular variable. In Bayesian approach, $\mathbf{w}$ is treated as a random variable. Using a Gaussian prior on $\mathbf{w}$,

$$p(\mathbf{w}|\sigma^{\mathbf{2}}) = \mathcal{N}(0, \sigma^2\mathbf{I}) \qquad (10)$$

and the posterior is

$$\begin{aligned} p(\mathbf{w}|\mathbf{t},\mathbf{X},\sigma^2) &= \frac{p(\mathbf{w}|\sigma^2)p(\mathbf{t}|\mathbf{X},\mathbf{w})}{p(\mathbf{t}|\mathbf{X})} \\ &:= \frac{g(\mathbf{w};\mathbf{t},\mathbf{X},\sigma^2)}{\int g(\mathbf{w};\mathbf{t},\mathbf{X},\sigma^2)d\mathbf{w}} \end{aligned} \tag{11}$$

It is infeasible to obtain an analytical expression for the posterior because of the integral in the denominator. Numerically, there are three options.

2.3. Gaussian Process

And for Ian

3. Experiments and Results

Reference the project guide

3.1. Naive Bayes

The Titanic dataset, supplied by Kaggle, came originally with with ten features and corresponding classes. Two of these features, include the name and the ticket id, would be useless for Naive Bayes as they can not be represented as numbers so they were removed from the dataset. Of the remaining features, The passenger sex and embankment location were strings. A natural fix to this issue was to represent the sex as a binary variable, 0 representing male and 1 representing female. The three embarked location variables, Cherbourg, Queenstown, and Southamton, were changed to 1, 2, and 3 respectively. All missing data was denoted with a -1 for computational ease of excluding features.

This data was complied and placed in an array, represented the table 1 below, where **t** is an indicator variable, 1 implies survived, 0 implies not, and the features x_d represent Passenger class, sex, age, SbSp, Parch, Fare, and embarked location respectivley

Table 1. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

t	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	3	0	22	0	0	7.25	0
1	1	1	38	1	1	71.28	2
1	3	0	-1	4	3	7.93	0
0	2	0	54	1	5	30.07	0
0	2	1	-1	2	4	263	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	3	0	84	3	1	9	2

Using this classification method we tested two potential methods of Naive Bayes. Here we have a mixture of discrete and continuous variables, meaning we could either represent all variables as continuous, or recognize we have both discrete and continuous variables.

3.1.1. Implementation: Purely Continuous

The continuous representation of of our data assumes all features in our data set, refer to table 1, can be represented by a real number. This assumption allows for a marginally faster implementation of Naive Bayes as the class conditional distribution may easily be represented by a Gaussian distribution, where the mean is the average value of every respective feature and the variance is the variance of all of these features.

For our implementation of Naive Bayes, this leaves us the modified version of equation 2

$$p(\mathbf{x}_n|t_n = c,\mathbf{X},\mathbf{t}) = \prod_{d=1}^{D} \mathcal{N}(x_{nd}|\mu_d,\sigma_d) \tag{12}$$

which can easily be evaluated. This then leaves the class prior, our second term in the numerator of equation 1, which in practice is generally left as the MLE solution

$$P(T_n = c|\mathbf{X},\mathbf{t}) = \frac{n_c}{N} \tag{13}$$

where n_c is the number of elements in class c. The combination of equations 3 and 4 allow for the use of our Naive Bayes model defined in equation 1.

3.1.2. Results: Purely Continuous

The results of the MLE approximation for the parameters μ, σ, and $P(t_n = c|\mathbf{X},\mathbf{t}$ are described below in table 2

Table 2. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
μ_0	2.51	0.16	23.63	0.65	0.35	24.24	1.30
μ_1	2.09	0.74	21.50	0.49	0.45	45.04	1.49
σ_0	?	?	?	?	?	?	?
σ_1	?	?	?	?	?	?	?
$P(t_n = 0)$	3	0	-1	4	3	7.93	0
$P(t_n = 1)$	2	0	54	1	5	30.07	0

3.1.3. Implementation: Continuous and Discrete

3.2. Logistic regression

This is for Hillary

3.3. Gaussian Process

And for Ian

4. Discussion and Extensions

Reference the project guide

There is a pre-imported references section which we will need to change to what our references our (like the books we used to understand these algorithms or lectures)