Credal Classification of Automobile Risk

Ben Willis

February 5, 2017

Abstract

This project investigates how the credal classifier can be applied to the problem of determining auto mobile insurance risk. The goal is to show how the credal classifier can model uncertainty better than other classifiers and hence return more accurate classifications. This uncertainty may arise from small sample sets or missing data which are normally difficult to deal with.

Contents

Chapter 1

Introduction

1.1 Classification

Classification is the problem of identifying which class an object belongs to. Each object can be distinguished by a set of properties know as features and each object belongs to a single class. A classifier is an algorithm which, given previous observations and their classes, can determine which class a new observation belongs to [5]. There are many applications of classifiers including image recognition, sentiment analysis and medical diagnosis.

There are two types of classifiers, supervised and unsupervised. Unsupervised classifiers infer classes from the data. Supervised classifiers are constructed from a set of data for which the true classes are known and this is the type of classifier we will be exploring [3].

1.2 Auto mobile Insurance

Classifiers have many applications in the finance industry ranging from financial trading [1] to credit card fraud detection [2]. We will study the problem of classifying the risk to an insurer of a car and comparing this solution to the classification of an expert. We will then examine how both classifications compare to the normalised loss to the insurer.

The data set we will be analysing contains vehicular information from 205 auto mobiles. Its features include dimensions, engine specifications and vehicle characteristics. It also contains an expert's assessed risk to the insurer of the vehicle on an integer scale of -2 to 3 with 3 being most risky and -2 being least risky. In addition to the technical information and the experts assessment, the data set also contains the normalized loss to the insurer. This ranges from 65 to 256 and is normalized for all vehicles within a particular size classification (two-door small, station wagons, etc.) and represents the average loss per car per year [6].

Chapter 2

Naive Bayes Classifier

2.1 Theory

To demonstrate the workings of the naive Bayes classifier will introduce a new data set from a remote sensing study. The study measured spectral information in the green, red and infrared wavelengths on three separate dates of 523 different areas of forest in Japan. In total we have nine continuous attributes and four possible classes: Sugi forest, Hinoki forest, Mixed deciduous forest and other non-forest land. This data set was chosen as it contained a large number of observations and demonstrates a situation where this type of classifier works well.

Formally, let us denote the class variable by C, taking values in the set $\{0, 1, 2, 3, 4\}$ for each of the four types of forest. Also we measure 9 features $A_1, \ldots, A_9$. We discretize these continuous features so that they all take values in from the set $\{0, \ldots, 9\}$. We denote observations of the observed values as c and $a_1, \ldots, a_9$ respectively.

We are interested in the probability of a forest being of type c given sensor readings $\mathbf{a}$ i.e. $P(c \mid \mathbf{a})$. Using Bayes theorem we can rewrite this as:

$$P(c \mid \mathbf{a}) = \frac{P(\mathbf{a} \mid c)P(c)}{P(\mathbf{a})} \tag{2.1}$$

Moreover we can make use of the naivety assumption. The naivety assumptions states that each attribute is conditionally independent of one another. In the context of this data set we are assuming the same sensor readings for the same forest are conditionally independent which indicates the possible issues with this strong assumption. We can now write the probability of an object being in class c with attributes $a_1, \ldots, a_k$ as:

$$P(c \mid \mathbf{a}) = \frac{P(c) \prod_{i=1}^{k} P(a_i \mid c)}{P(\mathbf{a})} \tag{2.2}$$

To turn this into a classifier we need a way to make a decision for which class an object falls into based on the estimated probabilities. A common method is choosing the class that maximises $P(c \mid \mathbf{a})$. This is known as the maximum a posteriori (MAP) estimate. We also note that $P(\mathbf{a})$ is not dependent on C hence we can write our estimate as:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c) \prod_{i=1}^{k} P(a_i \mid c) \tag{2.3}$$

Now that we have our method for making our decision we need to estimate the required probabilities.

Firstly we parametrise these probabilities. We denote the unknown chances of observing an object in class c by θ_c and the chance of observing an object in class c with attributes $\mathbf{a}$ by $\theta_{\mathbf{a},c}$. Similarly we denote the conditional chances of observing an indiviual attribute a_i and a set of attributes $\mathbf{a}$ given $C = c$ by $\theta_{a_i|c}$ and $\theta_{\mathbf{a}|c}$ respectively.

Now we have parametrised the probabilities we wish to estimate we can consider the likelihood function for θ, the vector whose elements are the chances $\theta_{\mathbf{a},c}$. Using our data we denote the frequencies of objects in each class c by $n(c)$ and the number of objects in class c with attribute a_i by $n(a_i, c)$. For example the number of observations of class 0 is 158 so $n(0) = 158$ We then consider the vector $\mathbf{n}$ which contains these frequencies.

The likelihood function can be expressed as:

$$l(\theta \mid \mathbf{n}) \propto \prod_{c \in \mathcal{C}} \left[\theta_c^{n(c)} \prod_{i=1}^{k} \prod_{a_i \in \mathcal{A}_i} \theta_{a_i|c}^{n(a_i,c)} \right] \tag{2.4}$$

A simple estimate for these parameters is the maximum likelihood estimate (MLE). To find the MLE first we take the log likelihood:

$$L(\theta \mid \mathbf{n}) \propto \sum_{c \in \mathcal{C}} n(c) log(\theta_c) + \sum_{c \in \mathcal{C}} \sum_{i=1}^{k} \sum_{a_i \in \mathcal{A}_i} n(a_i, c) log(\theta_{a_i|c}) \tag{2.5}$$

So to maximise the likelihood function we need to maximise the each part of the log likelihood function.

To do so we use the method of Lagrange multipliers. This is a strategy for finding local maxima and minima of a function subject to constraints.

For the θc parameters we want to maximise:

$$f(\theta, \mathbf{n}) = \sum_{c \in \mathcal{C}} n(c) log(\theta_c) \tag{2.6}$$

under the constraint:

$$g(\theta, \mathbf{n}) = \sum_{c \in \mathcal{C}} \theta_c - 1 \tag{2.7}$$

This gives us our Lagrangian:

$$\mathcal{L}(\theta, \mathbf{n}, \lambda) = \sum_{c \in \mathcal{C}} n(c) log(\theta_c) - \lambda(\sum_{c \in \mathcal{C}} \theta_c - 1) \tag{2.8}$$

Differentiating with respect to θ_c we have:

$$\nabla_{\theta_c} \mathcal{L}(\theta, \mathbf{n}, \lambda) = \frac{n(c)}{\theta_c} - \lambda \tag{2.9}$$

Hence the maximum likelihood estimate is $\hat{\theta_c} = \frac{n(c)}{N}$. Intuitively this is just the relative frequency of observations that fall into that class. Returning to our example data set we know that $N = 523$ and $n(0) = 158$ so $\hat{\theta}_0 = \frac{158}{523} \approx 0.302$

We now have our naive Bayes classifier. We estimate $P(c)$ by $\frac{n(c)}{N}$ and $P(a_i \mid c)$ by $\frac{n(a_i,c)}{n(c)}$, the relative frequencies. Then we choose the class c which maximises $P(c) \prod_{i=1}^{k} P(a_i \mid c)$.

To measure how successful our classifier is we will initially use a technique known as k-fold cross validation to evaluate accuracy. In k-fold cross validation we split our dataset into k equally sized groups.11 Then for each group we train the classifier on all the other groups and test it on that group. We then average all these accuracy to return an estimate for the accuracy of our classifier.

The choice of k leads to different types of cross validation. A standard choice is $k = 10$. A special case of cross validation is when $k = n$ (the number of observations). This is knowns as *leave-one-out cross validation* [4].

The accuracy of the classifier is 82.37% on this data set, using 10-fold cross validation.

2.2 Application to Automobile Data set

To make our data appropriate for this method we discretize the continuous variables into 10 bins with an equal frequency.

Unlike in the trees data set in this data set we have objects with missing values for attributes. We have no mechanism for considering these so we must discard these observations. This reduces our data set from 205 observations to 193 observations.

The accuracy of the classifier is 56.48% on this data set, using 10-fold cross validation. This is considerably worse than the example forest data set.

2.3 Conclusions

Clearly the classifier performs better on the forest type dataset than on the auto mobile data set. There are also general failings in our classifier we can fix to improve it form both data sets.

Firstly our classifier falls down if there are no observations with attribute a_j and class c in our training set. In these case the maximum likelihood estimate for $\theta_{a_j|c}$ is 0. This estimate leads to $P(c \mid \mathbf{a}) = 0$ and would rule out assigning any objects with the attribute a_j to class c. This is especially problematic for small sets of data. We can tackle this by introducing prior probabilities for the theta chances.

Another reason the classifier appears to perform worse on our auto mobile data set could be the nature of its categories. The accuracy metric does not take into account how close the classification is. For example if the true class is 2 an assigned class of 1 should be considered better than an assigned class of -2.

Chapter 3

Corrected NBC with Dirichlet Prior

3.1 Theory

We return to our likelihood function eq. (2.4) for our theta variables. We can introduce a prior distribution for these parameters and then consider the posterior distribution.

The Dirichlet distribution is the multinomial extension of the beta distribution for $x_1, \ldots, x_k$ where $x_i \in (0,1)$ and $\sum_{i=1}^{k} x_i = 1$ with probability density function:

$$f(x_1, \ldots, x_k \mid s, t(1), \ldots, t(k)) \propto \prod_{i=1}^{k} x_i^{st(i)-1} \tag{3.1}$$

where $s > 0$ and each $t(i) > 0$ such that $\sum_{i=1}^{k} t(i) = 1$.

We introduce a similar distribution as our prior density:

$$f(\theta \mid \mathbf{t}, s) \propto \prod_{x \in \mathcal{C}} \left[\theta_c^{st(c)-1} \prod_{i=1}^{k} \prod_{a_i \in \mathcal{A}_i} \theta_{a_i|c}^{st(c,a_i)-1} \right] \tag{3.2}$$

where $t(\cdot)$ corresponds to $n(\cdot)$. This prior Dirichlet distribution [7] has the following constraints:

$$\sum_{c \in \mathcal{C}} t(c) = 1 \tag{3.3}$$

$$\sum_{a_i \in \mathcal{A}_i} t(a_i, c) = t(c) \tag{3.4}$$

$$t(a_i, c) > 0 \tag{3.5}$$

For all (i, a_i, c).

When we multiply our likelihood by this prior density get a posterior in the same form. This prior distribute is same conjugate family as our likelihood function.

How do we choose prior parameters? s affects the speed at which our classifier learns and $t(c)$ kind of represents our beliefs for θ_c.

Bibliography

[1]

[2] PhD thesis.

[3] D. J. Spiegelhalter D. Michie and C. C. Taylor. *Machine Learning, Neural and Statistical Classification.* 1994.

[4] Paul E. Keller Kevin L. Priddy. *Artificial Neural Networks: An Introduction.* SPIE Press, 2005.

[5] Konstantinos Koutroumbas S. Theodoridis. *Pattern Recognition.* Elsevier Science, 2003.

[6] J. Schlimmer. *Automobile Data Set*, 1987 (accessed November 8, 2016). `https://archive.ics.uci.edu/ml/datasets/Automobile`.

[7] M. Zaffalon. Statistical inference of the naive credal classifier. 2001.