Optimum reject options for multiclass classification

Johannes Kummert

October 19, 2015

Bachelor Thesis

Optimum reject options for multiclass classification

Johannes Kummert

1. Reviewer **Prof. Dr. Barbara Hammer**

Theoretical Computer Science
Bielefeld University

2. Reviewer **Benjamin Paaßen**

Theoretical Computer Science
Bielefeld University

Supervisors Prof. Dr. Barbara Hammer and Benjamin Paaßen

October 19, 2015

Johannes Kummert

Optimum reject options for multiclass classification

Bachelor Thesis, October 19, 2015

Reviewers: Prof. Dr. Barbara Hammer and Benjamin PaaÃ§en

Supervisors: Prof. Dr. Barbara Hammer and Benjamin PaaÃ§en

Abstract

Contents

Introduction

Reject Options

2.1 Two Classes

To make our way towards optimal rejects for multi class classification, we start of small by looking at a general two class classifier f that divides the space via a decision boundary.

$$f : \mathbb{R}^n \to \{1, 2\}$$

Let r be a measure of confidence that a point is part of its respective class, e.g. distance to the decision boundary. If $r(\bar{x})$ is large it means that $\bar{x}$ is likely in the class it was assigned to.

2.1.1 Reject Strategy

We look for a threshold θ that defines for a given point $\bar{x}$ whether it is rejected or not:

$$r(\bar{x}) < \theta : \bar{x} \ rejected$$

2.1.2 Optimal θ

In order to find an optimal threshold we need to decide how to evaluate it. Optimally only wrongly classified data points are rejected (now referred to as true rejects). But generally there will be rejected although correctly classified points (false rejects). A given labeled (by y) data set X is divided into the two sets L and E by applying f as a classifier with:

$$L = \{\forall \bar{x} \in X \mid f(\bar{x}) = y(\bar{x})\} : correctly\ classified\ points \tag{2.1}$$

$$E = \{\forall \bar{x} \in X \mid f(\bar{x}) \neq y(\bar{x})\} : incorrectly\ classified\ points \tag{2.2}$$

And by applying our reject strategy with the threshold θ, X is divided into A and R with:

$$A_\theta = \{\forall \bar{x} \in X \mid r(\bar{x}) \geq \theta\} : accepted\ points \tag{2.3}$$

$$R_\theta = \{\forall \bar{x} \in X \mid r(\bar{x}) < \theta\} : rejected\ points \tag{2.4}$$

The set T contains all true rejects and F all false rejects.

$$T_\theta = R \cap E \tag{2.5}$$

$$F_\theta = R \cap L \tag{2.6}$$

Naturally we want $|T_\theta|$ to be large and $|F_\theta|$ to be small. Since these two goals often contradict each other, e.g. more true rejects most times bring more false ones (TODO: graph), there is no single optimal choice of θ in general. Still there is a set of values that we consider optimal. As shown above each θ corresponds to a tuple $(|T_\theta|, |F_\theta|)$. θ is optimal if

$$\nexists \theta' : |F_\theta'| \leq |F_\theta|, |T_\theta'| \geq |T_\theta|$$

and at least one term is unequal (TODO: wording?).

2.1.3 Finding θ

Now that we know how to evaluate our thresholds we still need to know where to look for them. Obviously θ needs to be in the range of r so

$$\Theta := \{\theta \in \mathbb{R} \mid \theta < \max_{\bar{x}} r(\bar{x})\}$$

is a set of possible thresholds. But it would leave us with an infinite search area (TODO: wording?) but this is easily reduced when recognized that it is unnecessary to consider multiple thresholds in between two points that are next to each other (when sorted according to r). So with $\Theta := \{r(\bar{x}) | \forall \bar{x} \in X\}$ we have a feasible set of possible thresholds. It can be refined by looking again at our criteria for an optimal θ. We want $|T_\theta|$, the amount fo true rejects, to be large so we should skip cluster of true rejects since choosing a point in the middle of such cluster can not be better than the point at the end of it. Additionally we want $|F_\theta|$, the amount fo false rejects, to be small, so we should skip cluster of false rejects since a point in the middle of such cluster can not be better than a point at the beginning of it. In conclusion this means that it is sufficient to consider only points at the beginning of clusters of false rejects, so

$$\Theta := \{r(\bar{x}_i) \mid \bar{x}_i \in L, \ \bar{x}_{i-1} \in E, \ \bar{x}_{i+1} \in L\}$$

where i is the index of the points in X when sorted according to r. This gives us an easily computed set of thresholds to consider to be optimal. (TODO: example graphic)

2.2 Multi class Classification

Let us now expand the problem to a classifier for N multiple classes. By using a one vs all strategy, we get N binary classifiers f_i like in chapter 2.1 and an according measure of confidence r_i. Points are now classified in the class where confidence is maximal among all classes. This gives us a multi class classifier f with

$$f : \mathbb{R}^n \to \{1, ..., i\} \tag{2.7}$$

$$\bar{x} \mapsto i \mid \arg\max_i r_i(\bar{x}) \tag{2.8}$$

2.2.1 Global Reject

To adapt our reject strategy from before, we search for a threshold θ and reject according to our measures of confidence:

$$\max_i r_i(\bar{x}) < \theta \; : \; \bar{x} \; rejected$$

We look again to choose θ so that it meets our requirements for an optimal threshold (see chapter 2.1.2). This strategy comes with the problem that it relies on all measures r_i to be scaled the same way (TODO: exmaple?). While there might be a workaround for this a global reject is still imprecise when the internal structures of the classes differ a lot. A reasonable threshold for a class with most points and also most classification errors near the decision plane is probably not well suited for a class with a wider (TODO: wording?) set of data points (TODO: example). This leads to the idea of having individual thresholds for each class.

2.2.2 Local Reject

To account for differences in class structure each one now has a local reject threshold θ_i. A point is rejected if

$$\max_i r_i(\bar{x}) < \theta_i \mid \forall \bar{x} \text{ where } f(\bar{x}) = i$$

(TODO: need max?) This gives us an i-dimensional threshold vector $\bar{\theta} = (\theta_1, ..., \theta_i)$. Each θ_i regulates the reject practice only in their respective class. (TODO: example)

2.2.3 Optimal Local Reject

For each threshold θ_i the same criteria apply to determine if it is optimal as for a global threshold (see chapter 2.1.2). With the difference that we look at each class individually. So the true and false rejects are now given by

$$T_{\theta_i}^i = R_{\theta_i}^i \cap E_i \tag{2.9}$$
$$F_{\theta_i}^i = R_{\theta_i}^i \cap L_i \tag{2.10}$$

where $R_{\theta_i}^i$ is the amount of rejected points in class i given the threshold θ_i, E_i the amount of falsely classified points in class i and L_i the amount of correctly classified points in i. As before we consider a certain set Θ_i of possible thresholds for each class (see chapter 2.1.3). We conclude that the threshold vector is $\bar{\theta}$ is optimal if each θ_i is optimal. To now find the optimal local reject vector $\bar{\theta}$ we need to consider every combination of thresholds in the sets Θ_i of possible optimal thresholds. Brute forcing this would mean that finding $\bar{\theta}$ has an exponential complexity. However the problem is equivalent to it a multiple choice knapsack problem (MCKP) where thresholds within a class correspond to items, false rejects correspond to costs, and true rejects correspond to their value. (TODO: link)

2.2.4 Computation by Dynamic Programming

TODO: how do it get to the bellman-equation? The Bellmann equation to compute opt results in

$opt(n, j, i) =$

$$\begin{aligned}
&\text{if } n = 0 &&: \sum_{k=1}^{N} |T_{\Theta_k(0)}^k| &&(2.11)\\
&\text{if } n > 0,\, j = 0 &&: \sum_{k=1}^{N} |T_{\Theta_k(0)}^k| &&(2.12)\\
&\text{if } n > 0,\, j > 0,\, i = 0 &&: opt(n, j-1, |\Theta_{j-1}| - 1) &&(2.13)\\
&\text{if } 0 < n < |F_{\Theta_j(i)}^j|,\, j > 0 &&: opt(n, j, i-1) &&(2.14)\\
&\text{if } n \geq |F_{\Theta_j(i)}^j| > 0,\, j > 0 &&: max\{opt(n, j, i-1),\\
&&&\quad opt(n - |F_{\Theta_j(i)}^j|, j-1, |\Theta_{j-1}| - 1) + |F_{\Theta_j(i)}^j| - |F_{\Theta_j(0)}^j|\} &&(2.15)
\end{aligned}$$

Each case is explained as follows:

- Case 2.11: $n = 0$ means that no false rejects are allowed so the first threshold (index 0) in each class is chosen. The resulting amount of true rejects is summed up over all classes.

- Case 2.12: $j = 0$ means that no class is under consideration for a threshold. So we stay with the first threshold in each class (see above).

- Case 2.13: $i = 0$ means that no threshold is under consideration in class j. So we look in the previous cell with the strictest threshold possible.

- Case 2.14: The chosen threshold i in class j exceeds the allowed amount of false rejects, so the next less strict threshold is considered.

- Case 2.15: Here the ith threshold in j is a possible threshold but it is not clear whether it is optimal. We consider both cases. If it is not the optimal threshold, we take the next less strict one. If it is optimal, we continue our search in the previous class but with $|F^j_{\Theta_j(i)}|$ less allowed false rejects in consequence to choosing the threshold. The other consequence is that the threshold results in a number of gained true rejects compared to the least strict threshold and this gain is added.

2.2.5 Greedy Computation

2.2.6 Evaluation

Application for SVM

3

Conclusions

4

List of Figures

List of Tables