Approximation Algorithms for Minimizing Empirical Error by Axis-Parallel Hyperplanes

Tapio Elomaa[1], Jussi Kujala[1], and Juho Rousu[2]

[1] Institute of Software Systems, Tampere University of Technology
[2] Department of Computer Science, Royal Holloway University of London
elomaa@cs.tut.fi, jussi.kujala@tut.fi, juho@cs.rhul.ac.uk

Abstract. Many learning situations involve separation of labeled training instances by hyperplanes. Consistent separation is of theoretical interest, but the real goal is rather to minimize the number of errors using a bounded number of hyperplanes. Exact minimization of empirical error in a high-dimensional grid induced into the feature space by axis-parallel hyperplanes is NP-hard. We develop two approximation schemes with performance guarantees, a greedy set covering scheme for producing a consistently labeled grid, and integer programming rounding scheme for finding the minimum error grid with bounded number of hyperplanes.

1 Introduction

In supervised learning a training sample $S = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \}$ of n labeled instances is given. The instance vectors x_i are composed of the values of d attributes and the class labels y_i usually come from a relatively small set C. The objective of a learning algorithm is to categorize the examples given to reflect the true classification of all instances. However, since it can only be observed through the training sample, a common subtask in learning algorithms is to fit a hypothesis closely to the training examples.

In real-world domains fully consistent separation of instances is mostly impossible due to noise and other inherent complications. Instead, one needs to solve an optimization problem of *empirical (or training) error minimization*, finding the hypothesis that errs in the classification of minimum number of training examples. Indeed, Vapnik's [1] *empirical risk minimization* principle suggests to choose the hypothesis with minimal training error. Fitting the hypothesis too closely to the training sample is, though, seen to lead to *overfitting* and, therefore, some form of *regularization* is required to guide error minimization.

By generalization error bounding techniques the error minimizing hypothesis does not have true error far from optimal. Unfortunately, in many classes finding the minimum error hypothesis is computationally intractable. It is, e.g., NP-hard to solve for the class of monomials, i.e. hyperplanes, in arbitrary dimension [2]. We will develop approximation algorithms for one such intractable problem, separating points by a restricted number of axis-parallel hyperplanes. This problem is related to practice, e.g., through the problem of naïve Bayesian

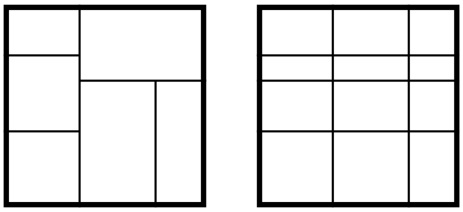

Fig. 1. Two ways of separating the plane by the same set of hyperplanes: nested half-spaces and the grid defined by hyperplanes penetrating each other.

classification in which the small number of decision boundaries per attribute taken together divides the input space into hyper-rectangular cells each of which gets an assigned class according to the relevant marginal distributions.

The axis-parallel separation problem that we study is, though, not quite the same problem because it does not take marginal distributions into account. However, this is a necessary step on the road to developing an optimal discretization algorithm for Naïve Bayes and general Bayesian networks. Discretization of numerical attributes is a central problem in learning network structure [3].

We consider the d-dimensional Euclidean instance space $\mathbb{R}^d$. For the ease of illustration, we will be mainly dealing with the two-dimensional case, $d = 2$. Sometimes we also set $|C| = 2$ for the sake of clarity.

2 Problem Definition and Prior Work

Axis-parallel hyperplanes arise in classifiers that compose their hypothesis from value tests for single attributes. Denote the value of an attribute A for the instance vector x by $\mathrm{val}_A(x)$. For a numerical attribute A the value test is of the form $\mathrm{val}_A(x) \le t$, where t is a threshold value. Now, $\mathrm{val}_A(x) = t$ defines an axis-parallel hyperplane that divides the input space in two half-spaces.

We are interested in the situation where the number of hyperplanes is restricted. If we are at liberty to choose the number of hyperplanes at will, we can always guarantee zero error for consistent data by separating each point to its own subset. Minimizing the number of hyperplanes needed to obtain a consistent partitioning is NP-complete, but can be approximated with ratio d in $\mathbb{R}^d$ [4].

At least two natural ways exist to partition a plane using axis-aligned straight lines (Fig. 1): They can define a hierarchy of nested half-spaces or a grid on the whole input space. An archetype example of the former method is top-down induction of decision trees. The root attribute first halves the whole instance space, its children the resulting subspaces, and so forth. The final subspaces (the leaves of a decision tree) are eventually assigned with a class label prediction, e.g., by choosing the majority label of the examples in the subspace. In our example five lines lead to six nested half-spaces. In the alternative division of the input space the lines always span through the whole input space and penetrate each other. Using the corresponding five lines leads to a grid of twelve cells.

2.1 Connection to Naïve Bayesian Discretization

The grid defined by hyperplanes penetrating each other is a result of executing several attribute value tests simultaneously. Our interest in this problem comes through Naïve Bayes, which predicts for each instance x the most probable class $\arg\max_{c \in C} \mathbf{P}(c \mid x)$ as determined by the training data. Probability estimation is based on the *Bayes rule* $\mathbf{P}(c \mid x) = \mathbf{P}(x \mid c)\ \mathbf{P}(c) / \mathbf{P}(x)$.

In determining the most probable class the values of all attributes of the instance are all looked at simultaneously because of the (naïve) assumption that the attributes $A_1, \ldots, A_d$ are independent of each other given the class, which indicates that $\mathbf{P}(x \mid c) = \prod_{i=1}^{d} \mathbf{P}(\mathrm{val}_{A_i}(x) \mid c)$. The probabilities are estimated from the marginal distributions of the training sample S. A common way of handling a numerical attribute A is to discretize its value range into successive half-open intervals $t_{j-1} < \mathrm{val}_A(x_i) \leq t_j$ using threshold values $t_1, t_2, \ldots$ after which the numerical attribute can be handled similarly as a nominal one.

In decision tree learning the value ranges of numerical attributes can often be discretized to a small number of intervals without a loss in training accuracy [5]. The same is true for optimal discretization of Naïve Bayes: its *decision boundaries* for each dimension can be recovered without loss of accuracy after discretization [6]. Together all decision boundaries of the d dimensions divide the input space into a hyper-grid in which each cell gets labeled by the class that is most probable according to the evidence contained in the training set.

Let R be one of the hyper-rectangles induced by the chosen axis-parallel hyperplanes minimizing training error. In other words, R is a convex region defined by the value of each attribute A_i restricted to some interval R_i contained in R. As empirical error is minimized in R, it must be that $\mathbf{P}(R, c)$ is maximized within R for (one of) the majority class(es) c. When the instances come from a product distribution, we can apply the naïve Bayesian assumption to get

$$\mathbf{P}(R, c) = \mathbf{P}(R \mid c)\ \mathbf{P}(c) = \mathbf{P}(c) \prod_{i=1}^{d} \mathbf{P}(R_i \mid c)\,.$$

This product is the numerator in the equation determining the prediction of Naïve Bayes and actually chooses the same class as Naïve Bayes. Since c has maximum probability within R, it must also be the choice of Naïve Bayes within this bin in a discretization. Hence, the Naïve Bayes optimal discretization is attained on the axis-parallel hyperplanes that minimize empirical error. However, this does not hold for all possible data distributions.

2.2 Related Work

The simplest linear separator class, single unrestricted hyperplanes, is usually considered to be a too restricted class of hypotheses for practical purposes because of the restrictions of the *perceptron* algorithm. There have, though, been many successful applications of even such simple hypotheses and *kernel methods* can take advantage of linear machines combined with other techniques [7].

Minimizing empirical error has been studied extensively in connection of decision trees. Optimal decision tree construction is NP-complete in general settings

[8] and in arbitrary dimensions [9, 10]. Furthermore, optimal decision tree learning is highly inapproximable [11]. In fixed dimensions Das and Goodrich [12] have shown that it is NP-complete to decide whether points of $\mathbb{R}^3$ that come from two classes have a consistent linear decision tree of at most k nodes.

The problem of separating two point sets with k unconstrained hyperplanes is NP complete in general dimension and solvable in polynomial time in fixed dimension [2, 9]. Grigni et al. [11] have shown that, unless NP=ZPP, the number of nodes containing linear decision functions (hyperplanes) in a decision tree cannot be approximated within any fixed polynomial. Moreover, the depth of such a classifier cannot be approximated within any fixed constant.

Auer et al. [13] devised an algorithm that minimizes empirical error in the class of two-level decision trees. Dobkin and Gunopulos [14] further consider learning restricted decision trees and studied the learnability of piecewise linear and convex concepts in low dimensions defined as the intersection of a constant number of half-spaces [15]. Dobkin et al. [16] also show that minimizing empirical error (in binary classification) is equivalent to computing the maximum bi-chromatic discrepancy. They are thus able to devise algorithms for minimizing error for axis-aligned boxes and hyperplanes.

Chlebus and Nguyen [17] showed the NP-completeness of consistent partitioning of the real plane using minimum number of axis-parallel lines that penetrate each other by reducing the minimum set cover problem in polynomial time to it. Hence, we cannot expect to find an efficient algorithm to solve the problem of our interest exactly (unless P=NP). Based on this result one can also prove Naïve Bayes optimal discretization to be NP-hard [6].

Călinescu et al. [4] dealt also with the problem that we consider here. However, they were interested in the case of consistent partitioning and used as many hyperplanes as needed to obtain complete separation of different colored points. We, on the other hand, are interested in the more realistic problem of restricted number of hyperplanes and inconsistent data. Nevertheless, we are able to take advantage of the proof techniques of Călinescu et al. [4].

3 Minimum Set Cover Approximation

As a reduction from minimum set cover to the consistent partitioning of the real plane has been used [17], it seems natural also to try to approximate empirical error minimization through that problem. Given a set U of n items, a collection $\mathcal{S}$ of m subsets of U, and a natural number k, the set covering problem is [18]:

> SET COVER($U, \mathcal{S}, k$): Does there exist a collection of at most k subsets $\{ S_{r_1}, \ldots, S_{r_k} \} \subset \mathcal{S}$ such that every item of U is contained in at least one of the sets in the collection?

SET COVER is approximable within $\ln n + 1$, but not within $(1 - \varepsilon) \log n$ for any $\varepsilon > 0$. The algorithm attaining the logarithmic approximation ratio is the straightforward greedy covering method, which chooses to the evolving cover the subset that contains the largest number of yet uncovered elements [18].

The problem of consistent partitioning of $\mathbb{R}^d$ with axis-parallel hyperplanes is:

> ConsAxis(S, n): Given a set S of n points $\{ x_1, \ldots, x_n \} \subset \mathbb{R}^d$ each labeled either positive or negative, find a consistent partitioning of S with axis-parallel hyperplanes using as few hyperplanes as possible.

We reduce ConsAxis to Set Cover, which allows us to use the greedy set covering algorithm to solve the ConsAxis problem. Given an instance S of the ConsAxis problem, we generate a new element $u_{x,x'}$ corresponding to each conflicting pair of labeled points (x, y) and (x', y'), where $y \neq y'$, in S. Let U be the set of all such generated elements. In the worst case $|U| = \Omega(n^2)$.

It is sufficient to restrict the axis-parallel hyperplanes to a set $\mathcal{H}$ of at most $n-1$ *canonical hyperplanes* per dimension. There is a canonical hyperplane per consecutive points with respect to a dimension (say, at the average coordinate between the two). In terms of cut point analysis [5], they correspond to the *bin borders* of the axes. In empirical error minimization one can further reduce the number of intervals [5]. For each $h \in \mathcal{H}$ we create a set that has a representative for each pair of conflicting examples that can be separated by h ($\equiv \mathrm{val}_{A_i}(x) = t$):

$$H(X,t) = \{ u_{x,x'} \in U \mid \mathrm{val}_{A_i}(x) \leq t \leq \mathrm{val}_{A_i}(x') \vee \mathrm{val}_{A_i}(x') \leq t \leq \mathrm{val}_{A_i}(x) \}$$

Let H denote the collection of all such sets. Now, H is an instance of the Set Cover problem such that its solution defines a set of hyperplanes which, by construction, separate all conflicting example pairs from each other.

Applying the greedy set covering algorithm to the collection of sets H constructed above, gives an approximation algorithm for ConsAxis with approximation quality $O(k^*(1 + \ln n^2)) = O(k^*(1 + 2 \ln n))$, where k^* is the minimum number of axis-parallel hyperplanes needed to consistently partition the set S.

Inconsistent data is also easy to handle. Determine the majority class within a set of all examples with the same instance vector and delete all members of the minority classes before converting the problem.

In practice one is often allowed or wants to use only k hyperplanes for the partition. The bounded number of hyperplanes now at our disposal does not necessarily suffice to reach the lowest error, and the goal becomes to attain as low error as possible using them.

When the number of hyperplanes is not restricted, no polynomial-time algorithm has approximation guarantee $1 + c$, where $c > 0$ is a constant [4]. I.e., by using a constant factor c more hyperplanes than in the optimal solution, one cannot guarantee to attain zero error. This result also implies a limitation to the situation where the number of hyperplanes is restricted: Assume that there were a polynomial-time algorithm with a constant approximation guarantee (to the number of erroneously labeled instances). More specifically, assume that

$$\mathrm{APP}_{(1+c)k} \leq \delta\, \mathrm{OPT}_k,$$

where APP_k denotes the error of the approximate solution using k hyperplanes and OPT_k stands for the minimum error using k hyperplanes. Now, setting

$\text{OPT}_k = 0$ yields an exact polynomial-time algorithm for the unrestricted case, which contradicts the fact that no such algorithm exists. Thus, no polynomial-time algorithm can guarantee an error at most a constant δ times that of the optimal algorithm using only a constant factor c more hyperplanes.

Set covering of conflicting pairs does not give an approximation algorithm in this case, because even though the evolving set cover (partition) reduces the number of conflicts of the sample, it does not guarantee diminishing error. Consider, e.g., four examples divided by one hyperplane into two subsets both containing one positive and one negative example; two classification conflicts have been removed by the hyperplane, but the error of this partition has not reduced.

4 Linear Programming Approximation

We will first formulate the axis-parallel separation problem as a zero-one integer program, and then give its linear program (LP) relaxation. The general problem of zero-one integer programming is NP-hard. However, for the LP relaxation, in which the integral constraints are replaced by ones that allow the variables to assume real values in $[0, 1]$, many efficient methods for solving are known.

We use two sets of binary variables. A variable w_i, $1 \le i \le n$, has value 1 if point p_i is not separated by the chosen hyperplanes from all points of different class, otherwise $w_i = 0$. The second set of variables z_j represents the axis-parallel hyperplanes. There are at most $d(n-1)$ of them. If a hyperplane $h_j \in \mathcal{H}$ is included in the set of solution lines, then $z_j = 1$ and otherwise $z_j = 0$.

Because each point that is not separated by the chosen lines from all points of different class will unavoidably lead to a misclassification, our objective is to minimize the number of such points. I.e., we want to optimize:

$$\min \textstyle\sum_{i=1}^{n} w_i$$

with constraints

$$\textstyle\sum_{j=1}^{d(n-1)} z_j \le k \text{ and } w_u + w_v + \sum z_j \ge 1.$$

The first constraint ensures that at most k hyperplanes are chosen. The second one is called the *separation condition*, and there is one for each pair of points p_u and p_v that have different class. The sum is taken over all those lines that separate the two points. The separation condition is sufficient because in each such pair at least one of the following holds:

- Either p_u or p_v, or both, is destined to be an error (in conflict with the label of the cell). In this case the separation condition is fulfilled by $w_u + w_v \ge 1$.
- A hyperplane h_j has been chosen that separates p_u and p_v. In this case the separation condition is fulfilled by $z_j = 1$.

Note that any value assignment for the w_i and z_j variables that satisfies the separation condition corresponds to a grid, where a point p_u that is labeled correct ($w_u = 0$) will only share its cell with points with the same label and points destined to be errors ($w_v = 1$).

Let us now turn to the LP relaxation, where the variables $\hat{w}_i$ and $\hat{z}_j$ take real values in $[0, 1]$. Obviously, the value of the solution to the relaxed problem using k lines, LP_k, is at most that of the integer program, OPT_k; $LP_k \leq \text{OPT}_k$. Let us consider all pairs of points p_u and p_v for which it holds $\hat{w}_u + \hat{w}_v \geq C$ for some constant C, $0 < C \leq 1$. We now round the values $\hat{w}$ so that in each such pair at least one variable gets value 1. A straightforward way is to round up all those variables that have $\hat{w}_i \geq C/2$. The remaining $\hat{w}$ variables are rounded down to 0. In the worst case we have to round up $n-1$ variables. The number of these variables determines an approximation to the solution of the optimization problem. Hence, this approach can guarantee an approximation ratio of

$$(2/C)\sum_{i=1}^{n} \hat{w}_i = (2/C)LP_k \leq 2\,\text{OPT}_k/C.$$

It remains to round the values $\hat{z}_j$. Here we adapt the counting based rounding procedure of Călinescu et al. [4]. Any two points p_u and p_v that need to be separated by a hyperplane after rounding of $\hat{w}$ values have $\hat{w}_u + \hat{w}_v < C$. By the separation condition, in the solution of the LP relaxation the sum of variables corresponding to hyperplanes in between the points is $\sum \hat{z}_j > 1 - C$. In order to include one of those to our rounded solution, we systematically go through the hyperplanes by dimensions and cumulate the sum of their $\hat{z}$ values. In the plane as long as the sum is below $(1 - C)/2$ we round the $\hat{z}_j$ variables down to 0. We choose all those lines that make the total sum reach or exceed $(1 - C)/2$. The sum is then reset to 0 and we continue to go through the lines.

Consider a conflicting pair of points which need to be separated by a line. If no vertical line was chosen to separate them, their cumulative sum must have been strictly below $(1 - C)/2$, and one of the horizontal lines in between the two is guaranteed to make the sum reach and exceed the threshold. As the sum of the fractional variables still obeys the upper bound of k by the first constraint, this way we may end up picking at most $2k/(1 - C)$ lines to our approximation.

Let APP_k denote the value of the above described rounding procedure and line selection using k lines. By the above computation, we have that

$$\text{APP}_{2k/(1-C)} \leq 2\,\text{OPT}_k/C.$$

Thus, we have demonstrated an approximation algorithm for the separation problem. As necessitated, the algorithm uses more lines and makes more false classifications than the optimal solution. For example, when $C = 1/2$, we have $\text{APP}_{4k} \leq 4\,\text{OPT}_k$. The general form of the above performance guarantee in d dimensions is $\text{APP}_{dk/(1-C)} \leq d \cdot \text{OPT}_k/C$.

5 Conclusion and Future Work

In this paper we studied two approaches for developing an approximation algorithm for separating classified points by axis-parallel hyperplanes. The first approach using the minimum set covering only works when the number of hyperplanes is not restricted, but a LP relaxation of an integer programming formulation of the problem yields an approximation algorithm also using only a bounded number of hyperplanes.

The LP approach can easily be extended to situations where the hyperplanes are not perpendicular to each other or are higher dimensional polynomials. The practicality of the approximation schemes remains to be studied. Even though, LP solvers are in principle efficient and sparse matrix techniques can in our case be used to further speed them up, the space complexity of the proposed approach defies the most straightforward implementation.

References

1. Vapnik, V.N.: Estimation of Dependencies Based on Empirical Data. Springer, New York (1982)
2. Kearns, M.J., Schapire, R.E., Sellie, L.M.: Toward efficient agnostic learning. Machine Learn. **17** (1994) 115–141
3. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learn. **29** (1997) 131–163
4. Călinescu, G., Dumitrescu, A., Wan, P.J.: Separating points by axis-parallel lines. In: Proc. Sixteenth Canadian Conference on Computational Geometry. (2004) 7–10
5. Elomaa, T., Rousu, J.: Efficient multisplitting revisited: Optima-preserving elimination of partition candidates. Data Mining and Knowl. Discovery **8** (2004) 97–126
6. Elomaa, T., Rousu, J.: On decision boundaries of naïve Bayes in continuous domains. In: Knowledge Discovery in Databases: PKDD 2003, Proc. Seventh European Conference. Volume 2838 of LNAI., Heidelberg, Springer (2003) 144–155
7. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
8. Hyafil, L., Rivest, R.R.: Constructing optimal binary decision trees is NP-complete. Inf. Process. Lett. **5** (1976) 15–17
9. Megiddo, N.: On the complexity of polyhedral separability. Discrete Comput. Geom. **3** (1988) 325–337
10. Blum, A., Rivest, R.R.: Training a 3-node neural net is NP-complete. Neural Networks **5** (1992) 117–127
11. Grigni, M., Mirelli, V., Papadimitriou, C.H.: On the difficulty of designing good classifiers. SIAM J. Comput. **30** (2000) 318–323
12. Das, G., Goodrich, M.: On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. Comput. Geom. **8** (1997) 123–137
13. Auer, P., Holte, R.C., Maass, W.: Theory and application of agnostic PAC-learning with small decision trees. In: Proc. Twelfth International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann (1995) 21–29
14. Dobkin, D., Gunopulos, D.: Geometric problems in machine learning. In: Applied Computational Geometry. Volume 1148 of LNCS., Heidelberg, Springer (1996) 121–132
15. Dobkin, D., Gunopulos, D.: Concept learning with geometric hypotheses. In: Proc. Eighth Annual Conference on Computational Learning Theory, New York, NY, ACM Press (1995) 329–336
16. Dobkin, D., Gunopulos, D., Maass, W.: Computing the maximum bichromatic discrepancy, with applications in computer graphics and machine learning. J. Comput. Syst. Sci. **52** (1996) 453–470
17. Chlebus, B.S., Nguyen, S.H.: On finding optimal discretizations for two attributes. In: Rough Sets and Current Trends in Computing, Proc. First International Conference. Volume 1424 of LNAI., Heidelberg, Springer (1998) 537–544
18. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2001)