Classification Based on Multiple-Resolution Data View

Mateusz Kobos and Jacek Mańdziuk

Warsaw University of Technology,
Faculty of Mathematics and Information Science,
Plac Politechniki 1, 00-661 Warsaw, Poland
{M.Kobos,J.Mandziuk}@mini.pw.edu.pl

Abstract. We examine efficacy of a classifier based on average of kernel density estimators; each estimator corresponds to a different data "resolution". Parameters of the estimators are adjusted to minimize the classification error. We propose properties of the data for which our algorithm should yield better results than the basic version of the method. Next, we generate data with postulated properties and conduct numerical experiments. Analysis of the results shows potential advantage of the new algorithm when compared with the baseline classifier.

Keywords: kernel density estimation, classification based on density estimation, average of density estimators.

1 Introduction

The main idea that inspired the work on the topic presented in this paper was that combining views on the data with different "resolutions" should produce an insight into the structure of the data. This insight, in turn, should aid solving problems related to data analysis such as classification. To implement the idea of data "resolution", we have decided to use Kernel Density Estimator (KDE) with its bandwidth parameter interpreted as the "resolution". In density estimation function generated by KDE, it can generally be observed that the larger (smaller) the bandwidth, the more similar (dissimilar) values at distant points. This phenomenon can be interpreted as manipulating the resolution. To combine different resolutions, we simply average output of KDEs with different bandwidths. Note that classifier based on KDE that we use was noticed by Specht in [1] to have the form of a neural network.

In practice, the above idea underlying the algorithm is implemented as follows. For a given test point and for each class, we use KDEs with different bandwidths to independently estimate density at the point. Then, for each class, the estimates are averaged to form final density estimate. Next, these estimates are inserted into the Bayes formula to produce probability estimate for each class. The bandwidths used in this process are selected beforehand to minimize the cross-validation classification error on the training set during the training

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part III, LNCS 6354, pp. 124–129, 2010.

process. The minimization is carried out by the L-BFGS-B quasi-Newton optimization algorithm introduced in [2]. For a detailed description of the training and classification phases as well as an explanation of some of the design decisions made while developing the algorithm (e.g. why we are using a single bandwidth parameter per KDE instead of a matrix of parameters) see [3, Sect. 2].

The methods appearing in the literature that seem to be the most similar to the proposed algorithm build upon an idea of combining different density estimators. These approaches can be mostly divided into two groups. The first one embraces methods that use a combination of density estimators, and their main goal is to optimize quality of density estimation. Even if they are used in a classification task, the classification error is not optimized directly what generally should result in obtaining suboptimal classification outcome. This group includes the algorithm that uses a linear combination of Gaussian Mixture Models (GMMs) and KDEs with predefined bandwidths to estimate density [4]. Another example is the method in which the boosting algorithm is used on KDEs with fixed bandwidths [5]. Yet another approach is developed in [6] where the EM algorithm is used on an average of GMMs to optimize density estimation quality. In the other group, there are methods where a combination of classifiers based on density estimators is used. Classification error is optimized directly in these approaches; however, the algorithms are combined on classifiers level instead of being combined on a deeper level of density estimators. An example of the algorithm in this group is the BoostKDC binary classifier, proposed in [7], which uses Real AdaBoost method with classifiers based on KDEs.

We propose an algorithm that is situated between the above-mentioned groups; namely, it combines different density estimators, but the parameters of the estimators are selected directly to optimize the classification error (and not the quality of density estimation). Such an approach is quite novel. To authors' knowledge, the only other algorithm that belongs to this category is a binary classifier introduced in [8]. Our approach is significantly different from the mentioned algorithm and much simpler.

The efficacy of our algorithm was tested using 19 popular benchmark data sets. The goal of the experiments was to test whether using as little as two different KDEs improves real-world results of KDEs-based classifier and gives an algorithm that is competitive when compared with the methods from the literature. A detailed description of these experiments can be found in [3, Sect. 3], but the results can be recapitulated as follows. The introduced algorithm yielded statistically significantly better result that the baseline version. What is more, the comparison between the results obtained and the literature results indicates that the algorithm is competitive when compared with other classification methods.

In this article, we examine properties of the algorithm introduced in [9] and [3]. In Sect. 2, we hypothesize about properties of data sets for which our algorithm should produce superior results when compared with a baseline KDE-based classifier. To confirm our hypotheses, we generate an artificial dataset with postulated properties and test our algorithm on it in Sect. 3.

2 Data Characteristics

Let us consider a classification problem described by densities of two classes, each density defined by a four-element, two-dimensional Gaussian Mixture Model. We define two structures separated by a long distance. The first one is a large low-density structure while the second one is a small high-density structure. Both of them generate non-trivial optimal decision boundaries (see Fig. 1). In the case of the low-density structure, a KDE with a large bandwidth would model well the decision boundary, and in the case of high-density structure, a small bandwidth would be appropriate. An average of these two KDEs should also give good results since the KDE with the small bandwidth will dominate in the high-density region while the KDE with the large bandwidth will dominate in the low-density region.

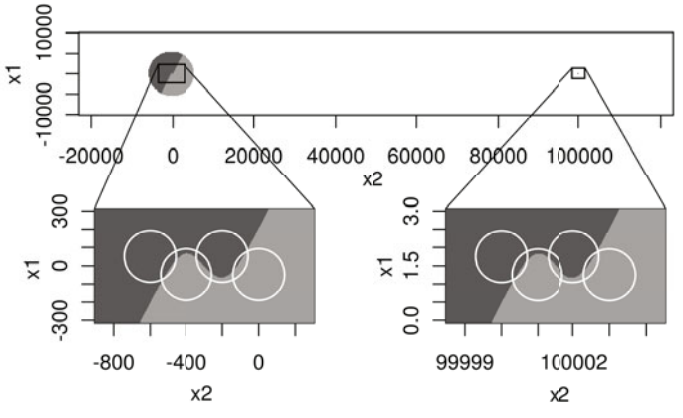

Fig. 1. Optimal decision regions (*dark gray* and *light gray* colors) in artificial dataset with a general view of the domain (*top*) and enlarged regions of low (*bottom left*) and high (*bottom right*) density structures. Note that the scale in depiction of both structures is different. Densities in each class are generated by a Gaussian Mixture Model (*white circles*). The white area in the general view of the domain corresponds to points where the density of both classes is negligibly small i.e. smaller than computer's machine precision.

The long distance that separates the structures is introduced for purely technical reasons. Our aim was to define a problem for which the optimal results are obtained in the case of equal bandwidths for both classes. This way we can narrow down the search space. Additionally, in the case of two bandwidths per class ($E = 2$), the misclassification error function depends on two parameters only; thus, it can be easily visualized. Both introduced structures are distant enough to be considered as separated because at every point, the density generated by at least one of the structures is negligibly small (smaller than the computer's machine precision). Therefore, they can be analyzed independently. Additionally, in each of the structures, the distribution of one class is a shifted version of

the other class's distribution. In this case, the use of a common bandwidth for both classes is justified (c.f. [10, p. 461]).

3 Experiments

The algorithm with two bandwidths ($E = 2$) per class was experimentally compared with the basic version of one bandwidth ($E = 1$) per class in the best-case scenario where the bandwidths were chosen optimally for the given classification problem. We generated a few instances of the problem, each having a different training set size. For each training set size, we generated independently 20 training sets, one testing set of size of 20 000 samples, and a set of bandwidth pairs. Each bandwidth came from the same range that starts at 0 and is long enough to contain the optimal bandwidth values for $E = 2$ and $E = 1$ cases (the length was selected manually). 100 equidistant bandwidth values were selected from this range to form a total number of 10 000 combinations of two bandwidths. A sample mean misclassification probability and a sample mean MSE were computed for the model based on each bandwidth pair. The mean values were computed over 20 training sets.

Figure 2 presents error functions computed for one of the training set sizes. Note that the values of the error function attainable for the $E = 1$ version belong to the half-line $(a, a),\ a \in [0, \infty)$. This half-line does not have to contain the global minimum of $E = 2$ version, thus using more than one KDE can potentially improve the results. Indeed, this is the case in the example presented in Fig. 2, where the global minima are located far away from the above-mentioned half-line. Another interesting observation is that in this case, the optimal solution consists of a small and a large bandwidth; this is consistent with our expectations. A surprising fact is that the minimum of the mean MSE function is located very close to the minimum of the mean misclassification probability; the situation is similar for other training set sizes. This property does not have to hold in a general case (see e.g. [10, p. 459]), but here it additionally justifies our approach to bandwidths selection where we minimize MSE instead of directly minimizing the misclassification probability (see [3, Sect. 2-E] for details).

The main objective of the experiment was to compare the best-case scenario results of $E = 1$ version of the algorithm with $E = 2$ version using training data sets of different sizes. Analysis of these results (see Fig. 3) leads to interesting observations. The most important one is that for each data set size except of the smallest one, the $E = 2$ version yields results that are statistically significantly better (paired t-test, $p \leq 1.56 \cdot 10^{-09}$) by a large margin than the results of the $E = 1$ version. For the larger data sets (800 samples and more), the $E = 2$ version yields misclassification error that is approximately two times smaller than the error of the $E = 1$ version when normalized by subtracting the Bayes risk. Moreover, in the case of the $E = 2$ version, minimizing the mean MSE gives results that are not statistically significantly different (paired t-test, $p \geq 0.142$) from minimizing the mean misclassification probability.

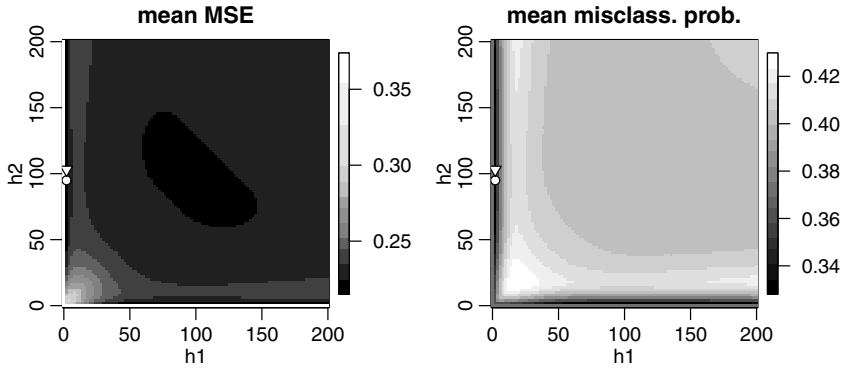

Fig. 2. Sample mean MSE (*left*) and sample mean misclassification probability (*right*) computed on the testing data from the artificial data set with the training set size equal 400. The darker the color of a point, the smaller the function value. The axes correspond to bandwidth values for each KDE. Note that in both plots, the points where one of the coordinates equals 0 correspond to high values of the function. The global minima of the mean misclassification probability and mean MSE are marked with a triangle and a circle respectively.

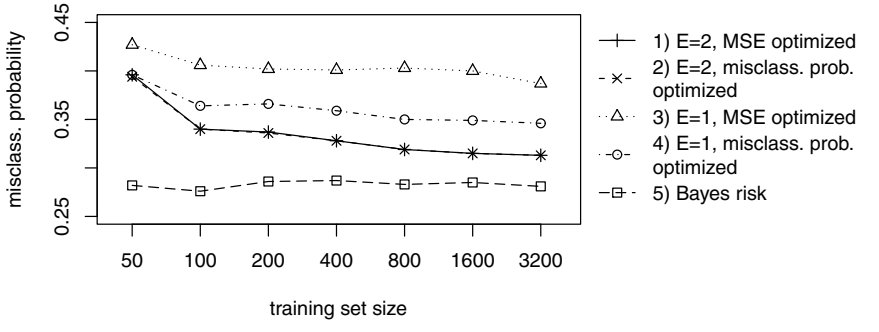

Fig. 3. Average misclassification probabilities on the artificial data set for different versions of the algorithm with optimally chosen bandwidths. Examined algorithm versions: 1) $E = 2$ with mean MSE optimized, 2) $E = 2$ with mean misclassification error optimized directly, 3) $E = 1$ with mean MSE optimized, 4) $E = 1$ with mean misclassification error optimized directly. Additionally, a sample optimal Bayes risk rate is also showed - line 5).

In summary, the proposed algorithm's version ($E = 2$) gives better best-case scenario results than the basic version ($E = 1$), which is consistent with our expectations. Additionally, for the data set considered and $E = 2$ version, minimizing the MSE gives results that are as good as the ones obtained by minimizing directly the misclassification probability.

4 Conclusions and Future Work

We showed that best-case scenario results on an artificial data set yielded by our algorithm are better than those yielded by the basic version. Currently, we are working on a version of the algorithm where the number of bandwidths is adjusted automatically to the data. This way no "expert knowledge" is required to choose the number of bandwidths per class E for a classification problem at hand. Another modification worth testing is selecting the starting point in a different way (e.g. by Scott's normal reference rule or Sheather-Jones method (see [11, Sect. 3])).

Acknowledgments. The authors would like to thank Prof. Jacek Koronacki for valuable suggestions concerning the direction of this research. This work has been supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme, by the European Social Fund and the National Budget in the framework of Integrated Operational Programme for Regional Development (ZPORR), Action 2.6: "Regional Innovation Strategies and Transfer of the Knowledge" through the Mazovian Voivodeship "Mazovian PhD Student Sholarship", and by the Warsaw University of Technology research grant.

References

1. Specht, D.: Probabilistic neural networks. Neural Networks 3, 109–118 (1990)
2. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Stat. Comput. 16, 1190–1208 (1995)
3. Kobos, M.: Combination of independent kernel density estimators in classification. In: Ganzha, M., Paprzycki, M. (eds.) International Multiconference on Computer Science and Information Technology, vol. 4, pp. 57–63 (2009)
4. Smyth, P., Wolpert, D.: Linearly combining density estimators via stacking. Mach. Learn. 36, 59–83 (1999)
5. Di Marzio, M., Taylor, C.C.: Boosting kernel density estimates: A bias reduction technique? Biometrika 91, 226–233 (2004)
6. Ormoneit, D., Tresp, V.: Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates. IEEE T. Neural. Networ. 9, 639–650 (1998)
7. Di Marzio, M., Taylor, C.C.: On boosting kernel density methods for multivariate data: density estimation and classification. Stat. Methods Appl. 14, 163–178 (2005)
8. Ghosh, A.K., Chaudhuri, P., Sengupta, D.: Classification using kernel density estimates: Multiscale analysis and visualization. Technometrics 48, 120–132 (2006)
9. Kobos, M., Mańdziuk, J.: Classification based on combination of kernel density estimators. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5769, pp. 125–134. Springer, Heidelberg (2009)
10. Ghosh, A.K., Chaudhuri, P.: Optimal smoothing in kernel discriminant analysis. Stat. Sinica 14, 457–483 (2004)
11. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall, London (1995)