Maximum *a Posteriori* Based Kernel Classifier Trained by Linear Programming

Nopriadi and Yukihiko Yamashita

Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo 152-8552, Japan
nopriadi@yahoo.com, yamasita@ide.titech.ac.jp
http://www.titech.ac.jp/

Abstract. We propose a new approach for classification problem based on the maximum a posteriori (MAP) estimation. The necessary and sufficient condition for the cost function to estimate a posteriori probability was obtained. It was clarified by the condition that a posteriori probability cannot be estimated by using linear programming. In this paper, a kernelized function of which result is the same as that of the MAP classifier is estimated. By relieving the problem from to estimate a posteriori probability to such a function, the freedom of cost function becomes wider. We propose a new cost function for such a function that can be solved by using linear programming. We conducted binary classification experiment by using 13 datasets from the UCI repository and compared the results to the well known methods. The proposed method outperforms the other methods for several datasets. We also explain the relation and the similarity between the proposed method and the support vector machine (SVM). Furthermore, the proposed method has other advantages for classification. Besides it can be solved by linear programming which has many excellent solvers, it does not have regularization parameter such as C in the cost function in SVM and its cost function is so simple that we can consider its various extensions for future work.

Keywords: Maximum a posteriori, Kernel Function, Linear Programming, Cost Function.

1 Introduction

In statistics, the method of maximum a posteriori (MAP) estimation can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. The MAP based method is adopted and studied in machine learning and pattern recognition. Many methods are then developed and implemented based on this method for the purpose of classification, detection, decision, and also estimation. We can explore some of them in references [1], [2], [3], [4], [5]. Many patterns (data sets) such as speech recognition [14], DNA sequences classification [18], image watermark identification [6], face image recognition [1], breast-cancer detection [15] have been used on the methods and achieve promising performance.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 493–502, 2010.

It is important to note that designing a classifier based on MAP depends on the available information about *a posteriori* probabilities. Otherwise, based on Bayes' theorem, the information about *a prior* probabilities and the *likelihood* are imperative. In fact, *a posteriori* probabilities is difficult to be determined directly from data, as well as the *likelihood* is. Even some MAP based method needs other parameters such as a mean vector and covariance matrix for each class [1]. However, the methods to estimate *a posteriori* probability are available and in particular neural network can be used to estimate *a posteriori* probabilities [7], [8], [10], [19].

The papers [7], [8] adress their discussion on the problem of designing cost functions to estimate *a posteriori* probabilities. Any cost function which provides *a posteriori* class probabilities is called *Strict Sense Bayesian* (SSB)[7]. General conditions for the SSB cost function can be found in [7], [9].

In this paper we propose a new approach for the classification problem. It is based on the maximum *a posteriori* (MAP) estimation. A kernelized function $w(x, y)$ that provides the same result as the MAP classifier is estimated. We do not estimate directly *a posteriori* probability $P(y|x)$. By relieving the problem from to estimate *a posteriori* probability to such a function, the freedom to choose the cost functions becomes wider. In other words, we do not need to consider if the cost function is SSB or not.

Beside that, the SSB cost function in [7] is nonlinear, then it can be solved by nonlinear optimization. We can not use linear programming to solve the optimization problem if the cost function is nonlinear function [15], [16]. In this paper we provide a cost function that can be solved by linear programming, which has many excellent solvers.

In order to evaluate the perpormance of our proposed method we conducted binary classification experiment by using 13 datasets. Based on the results we compare the performance of the proposed method to widely known methods. The conclusion is that our proposed method is competitive to the others. We also discuss the relation and the similarity between the proposed method and SVM. Finally, we explain that the cost function of our proposed method is so simple and there will be much room to explore its extension.

2 Maximum a Posteriori (MAP) Estimation

In this section we start explaining some important probability formulas and also the definition of maximum *a posteriori* classification, then reviewing a former method to estimate *a posteriori* probability and finally proposing our new approach.

Let y be the category to be estimated from a data x. $P(x)$, $P(y)$, $P(x|y)$, and $P(y|x)$ denote respectively as a prior probability density function (p.d.f) of x, a prior probability of y, a conditional p.d.f of x given y, and *a posteriori* of y. Bayes theorem can be derived from the joint probability of x and y (i.e. $P(x, y)$) as follows:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x) \ . \tag{1}$$

The expectation value of a function $f(x)$ in a data x is written as:

$$E_x\{f(x)\} = \int f(x)P(x)dx \ . \tag{2}$$

Then the MAP estimates category $\hat{y}$ that is defined as the mode of the posterior probability as follows:

$$\hat{y} = \arg\max_y \ P(y|x) \ . \tag{3}$$

A classifier system is designed to estimate category $\hat{y}$ for an unlearned pattern x. As shown in eq.(3) we need information about $P(y|x)$, even in neural networks the estimation of $P(y|x)$ is imperative in making decision.

For instance, in [7] Suerri *et al.* proposed a cost function $C(h, d)$ to estimate *a posteriori* probabilities. We substitute a vector of functions that should be *a posteriori* into h and a vector that expresses a category into d. We describe the criterion using $C(h, d)$ for a binary classification problem as follows:

$$\sum_{y\in\{+1,-1\}} E_x P(y|x) C\left((h(x), \begin{pmatrix}\delta_{y,+1}\\ \delta_{y,-1}\end{pmatrix}\right)$$

where $h(x)$ is a 2-dimensional vector of functions of x to be optimized and δ is the Kronecker delta. If $C(h, d)$ is SSB, the criterion above, the function h becomes *a posteriori* probability, i.e.

$$h(x) = \begin{pmatrix}P(+1|x)\\ P(-1|x)\end{pmatrix}.$$

They also found the necessary and sufficient condition for a symmetric and separable SSB cost function, that is $C(h, d)$ is expressed in the following form

$$C(h, d) = \sum_{i=1}^{2} \int_{d_i}^{h_i} g_i(\alpha)(\alpha - d_i)\, d\alpha + r(d)$$

where $g_i(\alpha)$ is any positive function ($g_i(\alpha) > 0, 0 \leq \alpha \leq 1$) which does not depend on d_i and $r(d)$ is an arbitrary function which does not depend on h. We can see that $\int_{d_i}^{h_i} g_i(\alpha)(\alpha - d_i)\, d\alpha + r(d)$ is a nonlinear function, then it is to be solved by nonlinear optimization.

In this research we do not estimate $P(y|x)$ directly for classification, but estimating a function $w(x, y)$. We could use $w(x, y)$ if it satisfies:

$$\arg\max_y \ w(x, y) = \arg\max_y \ P(y|x) \ . \tag{4}$$

Furthermore in the next section we will explain the advantage of our new approach that the cost function can be directly optimized with linear programming.

3 Model Formalization

Now we consider a set of pattern samples $\{(x_i, y_i)\}_{i=1}^{N}$ and we restrict a problem to binary classification, i.e. $y \in \{-1, +1\}$. We need a criterion to choose the function $w(x, y)$ in order to satisfies (4). The criterion of $w(x, y)$ proposed in this paper is written as follows:

maximize

$$\sum_{y \in \{-1,+1\}} E_x P(y|x)\ \min(w(x, y), 1)\ . \tag{5}$$

subject to:

$$\sum_{y \in \{-1,+1\}} E_x w(x, y) = 1\ , \tag{6}$$

$$w(x, y) \geq 0\ . \tag{7}$$

To achieve optimum $w(x, y)$ we maximize the expectation function of $P(y|x)$ times $\min(w(x, y), 1)$. In this cost function we evaluate the value of $w(x, y)$ up to one. Beside that the constraint (6) and (7) are also consistent with probability laws.

It is clear that the solution of eqs.(5), (6), and (7) is given as

$$w(x, +1) = \begin{cases} 1 & \text{if } P(+1|x) > P(-1|x) \\ 0 & \text{if } P(+1|x) < P(-1|x) \\ \beta_x & \text{if } P(+1|x) = P(-1|x) \end{cases}$$

$$w(x, -1) = \begin{cases} 0 & \text{if } P(+1|x) > P(-1|x) \\ 1 & \text{if } P(+1|x) < P(-1|x) \\ 1 - \beta_x & \text{if } P(+1|x) = P(-1|x) \end{cases} \tag{8}$$

where β_x is an arbitrary number ($0 \leq \beta_x \leq 1$). We can see $w(x, y)$ provides the same results of the MAP classifier.

We define $w(x, y)$ by using a kernel function $k(x, z)$, that can be written in the form

$$w(x, y) = \sum_{j=1}^{N} \alpha_{y,j} k(x, x_j) \tag{9}$$

and we use Gaussian kernel function which can be expressed by

$$k\,(x, z) = \exp\left(-\gamma\ \|x - z\|^2\right)\ . \tag{10}$$

The parammeter γ determines the width of the Gaussian kernel and in the training mode we adjust it for each pattern samples.

By exchanging the ensemble mean by sample mean and subtituting eq.(9) to eq.(5), we have a cost function in the form

$$\sum_{y \in \{-1,+1\}} E_x P(y|x)\ \min(w(x, y), 1) \backsimeq \frac{1}{N} \sum_{i=1}^{N}\ \min\left(\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i, x_j), 1\right)\ . \tag{11}$$

The constraints (6) and (7) respectively become:

$$\sum_{y\in\{-1,+1\}} E_x w(x,y) \backsimeq \frac{1}{N} \sum_{y\in\{-1,+1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) = 1 \ . \tag{12}$$

$$w(x_i,y) = \sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) \geq 0 \ , \quad \forall i \ . \tag{13}$$

Now, we have an optimization problem to be solved. It consist of a cost function (11), of which the maximum value we want to find, along with a set of constraints (12) and (13). To simplify the calculation in linear programming problem, the condition (13) could be changed with

$$\alpha_{y,j} \geq 0 \ , \tag{14}$$

if $k(x,y) \geq 0$. However, even if we use condition (13), the optimization problem still could be solved by linear programming. In many cases if $k(x,y) \geq 0$, the adoption (14) allows us to reduce the number of variables in a linear programming problem and the experiment shows better results.

In order to realize eq.(11), we introduce a slack variables $\xi_i \geq 0$, then we write it in the form

$$\min\left(\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j), 1\right) = \sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j) - \xi_i.$$

By introducing the above form to eq.(11), we have a linear programming problem of $3N$ variables $\alpha_{y,j}$ and ξ_i $(y \in \{-1,+1\},\ i,j = 1,2,...,N)$ that can be expressed in the form as follows:

maximize

$$\sum_{y\in\{-1,+1\}} \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \delta_{y,y_i} k(x_i,x_j)\right) \alpha_{y,j} - \sum_{i=1}^{N} \xi_i \ ,$$

In other word $w(x,y)$ is a surrogate function that behaves in as similar way to *a posteriori* probability. subject to:

$$\sum_{y\in\{-1,+1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) = N,$$

$$\alpha_{y,j} \geq 0 \qquad (y \in \{-1,+1\}, j = 1,2,...,N),$$

$$\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j) - \xi_i \leq 1 \qquad (i = 1,2,...,N),$$

$$\xi_i \geq 0 \qquad (i = 1,2,...,N).$$

Now we have a cost function that can be solved by using a linear programming.

4 Experiment

In the experiment we used an open source package GNU Linear Programming Kit (GLPK) to carry out optimization problem. The GLPK is a set of routines designed to solve large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. Then, in order to evaluate the performance of our proposed method we conducted experiment with two-class classification problem, using 13 data sets from the UCI repository. The properties of the data sets we used are shown in Table 1.

Table 1. Overview of the 13 data sets used in the experiment

Data set	# training samples	# test samples	# realization	Dimension
Banana	400	4900	100	2
Breast cancer	200	77	100	9
Diabetis	468	300	100	8
Flare-Solar	666	400	100	9
German	700	300	100	20
Heart	170	100	100	13
Image	1300	1010	20	18
Ringnorm	400	7000	100	20
Splice	1000	2175	20	60
Thyroid	140	75	100	5
Titanic	150	2051	100	3
Twonorm	400	7000	100	20
Waveform	400	4600	100	21

In order to have a more sophisticated model selection we considered the generalization performance of the model. For this purpose we ran 5-fold cross validation to estimate the parameter γ. We treated each pattern (from banana to waveform) separately by the cross validation. Our goal is to have an appropriate parameter for each pattern. We used the first 5 realizations of train data for validation. For each realization we performed a cross validation in the following manner: We split a training sample set into 5 equally sized and disjoint subsamples. Of the 5 subsamples, a single subsample was retained as test data and the remaining 4 subsamples were combined to form a training data for cross validation. We performed validation with the new train and test data then calculated the error. The cross validation process was repeated 5 times and each of the 5 subsamples was used exactly once as the test data. The 5 results of the folds then was averaged to produce a single estimation error. Since we used 5 realizations, then we chose a median of the best values of parameter (with minimum error).

Based on the parameter γ we performed experiment by using test data sets as shown in Table 1. The result of experiment is summarized in Table 2.

Table 2. Result of Experiment. Comparison with other methods. The best result is in bold face.

Data set	γ	Proposed	SVM	RBF	AB	AB_R	KFD
Banana	4.217	**10.7 ± 0.6**	11.5 ± 0.7	10.8 ± 0.6	12.3±0.7	10.9±0.4	10.8±0.5
B.Cancer	0.316	**25.8 ± 4.0**	26.0 ± 4.7	27.6 ± 4.7	30.4±4.7	26.5 ± 4.5	**25.8 ± 4.6**
Diabetis	0.649	25.0 ± 1.9	23.5 ± 1.7	24.3 ± 1.9	26.5±2.3	23.8±1.8	**23.2±1.6**
F.Solar	1.778	33.0 ± 7.8	**32.4 ± 1.8**	34.4± 2.0	35.7±1.8	34.2±2.2	33.2±1.7
German	0.270	25.3 ± 2.3	**23.6 ± 2.1**	24.7±2.4	27.5±2.5	24.33±2.1	23.7±2.2
Heart	0.237	17.8 ± 3.2	**16.0 ± 3.3**	17.6±3.3	20.3±3.4	16.5±3.5	16.1±3.4
Image	17.783	4.0 ± 2.7	3.0 ± 0.6	3.3±0.6	**2.7±0.7**	**2.7±0.6**	4.8±0.6
Ringnorm	0.090	2.5 ± 1.0	1.7 ± 0.1	1.7±0.2	1.9±0.3	1.6±0.1	**1.5±0.1**
Splice	0.129	24.2 ± 2.4	10.9 ± 0.7	10.0±1.0	10.1±0.5	**9.5±0.7**	10.5±0.6
Thyroid	1.685	5.00 ± 2.3	4.8 ± 2.2	4.5±2.1	4.4±2.2	4.6 ±2.2	**4.2±2.1**
Titanic	0.562	**21.6 ± 5.0**	22.4 ± 1.0	23.3±1.3	22.6±1.2	22.6±1.2	23.2 ±2.0
Twonorm	0.140	**2.5 ± 0.2**	3.0 ± 0.2	2.9±0.3	3.0±0.3	2.7 ±0.2	2.6 ±0.2
Waveform	0.225	11.0 ± 1.8	9.9 ± 0.4	10.7±1.1	10.8 ±0.6	**9.8±0.8**	9.9±0.4

Table 3. Computational time of learning and classification process for all realizations

Data set	# realization	Proposed (in seconds)	SVM (in seconds)
Banana	100	135.3	4.030
B.Cancer	100	20.6	0.700
Diabetis	100	216	2.332
F.Solar	100	498.2	4.536
German	100	765.7	5.734
Heart	100	19.9	0.596
Image	20	1.252×10^4	1.938
Ringnorm	100	164.6	14.104
Splice	20	365.4	8.402
Thyroid	100	13.9	0.308
Titanic	100	13.4	1.170
Twonorm	100	181.4	9.800
Waveform	100	200.9	10.444

In the experiment we also measure the computational time which is needed in learning and classification process for all realizations of each data set. Table 3 shows the computational time.

5 Discussion

To evaluate the performance of our proposed method we compare our experiment result to other state-of-the-art methods, as shown in Table 2. Here we choose Support Vector Machine (SVM), a single RBF classifier, AdaBoost (AB),

regularized AdaBoost (ABR), and Kernel Fisher Discriminant (KFD) as comparators. The experiment data in [11] is used for comparison. We are interested in comparing our new approach to the other methods because we know that the SVM and the boosting (in general the margin-based classifiers) have demonstrated their excellent performances in binary classification. Meanwhile, KFD is very competitive and in some cases even superior to the other algorithms on almost all data sets [11].

The result in Table 2 shows that our proposed method has promising performance. We can say that it is competitive to the others and superior on some data sets (banana, breast cancer, titanic and twonorm).

Regarding with the computational complexity we can see in Table 3 that the proposed method is slower than LIBSVM (library for support vector machines) [20]. However, LIBSVM is a specialized program for SVM. On the other hand we used GLPK that is a general purpose library. Therefore, it is difficult to compare the computational complexity from these results. Furthermore, the computational time of the proposed method is enough fast to apply it to many problems.

The next part of this section we discuss in particular relation and similarity between our proposed method and SVM. Maximization of eq.(11) can be translated to minimization of a loss function. Many loss functions are proposed such as the hinge loss and the fisher consistent loss [12]. Originally the SVM is designed for the binary classification problem and its paradigm has a nice geometrical interpretation of discriminating one class from another by a hyperplane with the maximum margin [13]. The cost function of SVM to obtain the optimal separating hyperplane is written as

$$\min \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_i,x_j) + C\sum_{i=1}^{N}\zeta_i \ , \tag{15}$$

where α_i is Lagrange multiplier which is optimized, C is the tuning parameter, ζ_i is the non negative slack variables. In the SVM the extra term $C\sum_{i=1}^{N}\zeta_i$ in the cost function is to accommodate some data which is not linearly separable by the hyperplane. Then the slack variables ζ_i express the hinge loss. If we neglect the treshold, the ζ_i can be expressed as follows:

$$\zeta_i = \max\left(0,\ 1 - \sum_{j=1}^{N}\alpha_j y_j k(x_i,x_j)\right) \ .$$

The parameter C is also called as the margin parameter that determines the tradeoff between the maximization of the margin and the minimization of the classification error. This term is to balance the goals of maximum margin separation and the correctness of training set classification.

The concept of hinge loss in SVM is quite similar to our criterion (11). We have the following arithmetic relation:

$$\min(a,1) = 1 - \max(0,(1-a)).$$

In both criteria, the classification functions are substituted into a. Therefore eq.(11) is considered as a hinge loss. Instead of the regularization term (the first term in eq.(15)), the eq.(12) is introduced. The advantage of our proposed method is that we do not have a regularization parameter such as C in the cost function of SVM. We only need one parameter, that is γ.

6 Conclusion

In this paper we proposed a new approach for classification problem based on maximum *a posteriori* probability. We do not estimate directly $P(y|x)$, but we use a kernelized function $w(x, y)$ that can be regarded as a surrogate function that behaves in a similar way to MAP Classifier. The advantage of this approach is the cost function can be directly optimized with linear programming. The experiment using 13 data sets from the UCI repository shows that our proposed method has promising performance and it is competitive enough to the other state-of-the-art classification methods. We also explained the relation between the proposed method and the support vector machine (SVM). The similarity between the hinge loss and the proposed criterion was discussed. Furthermore, we can consider its various extensions, similar to the extensions of SVM, to improve the classifier performance in the future work. It is very possible because our proposed cost function is so simple.

References

1. Xu, Z., Huang, K., Zhu, J., King, I., Lyu, M.R.: A Novel Kernel-Based Maximum a Posteriori Classification Method. J. Neural Networks 22, 121–146 (2009)
2. Gauvain, J.L., Lee, C.H.: Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. J. IEEE Trans. Speech and Audio Processing 2, 291–298 (1994)
3. Chen, K., Wang, H.: Eigenspace-Based Maximum a Posteriori Linear Regression for Rapid Speaker Adaptation. In: Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing, Salt Lake City, USA, vol. 1, pp. 917–920 (2001)
4. Igual, J., Camachoa, A., Bernabeua, P., Vergarab, L.: A Maximum a Posteriori Estimate for the Source Separation Problem with Statistical Knowledge about the Mixing Matrix. J. Pattern Recognition Letters 24, 2519–2523 (2003)
5. Siohan, O., Myrvoll, T.A., Lee, C.H.: Structural Maximum a Posteriori Linear Regression for Fast HMM adaptation. J. Computer Speech & Language 16, 5–24 (2002)
6. Ng, T.M., Garg, H.K.: A Maximum a Posteriori Identification Criterion for Wavelet Domain Watermarking. International Journal of Wireless and Mobile Computing 3, 265–270 (2009)
7. Sueiro, J.C., Arribas, J.I., Munoz, S.E., Vidal, A.R.F.: Cost Functions to Estimate a Posteriori Probabilities in Multiclass Problems. J. IEEE Trans. Neural Networks 10, 645–656 (1999)
8. Arribas, J.I., Sueiro, J.C., Adali, T., Vidal, A.R.F.: Neural Architetures for Parametric Estimation of a Posteriori Probabilities by Constrained Conditional Density Functions. In: Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP), Madison, Wisconsin, USA, pp. 263–272 (1999)

9. Miller, J.W., Goodman, R., Smyth, P.: Objective Functions for Probability Estimation. In: International Joint Conference on Neural Networks, Seattle, USA, vol. 1, pp. 881–886 (1991)
10. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. J. IEEE Trans. Neural Network 1, 296–298 (1990)
11. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.R.: Fisher Discriminant Analysis with Kernels. In: IEEE Signal Processing Society Workshop In Neural Networks for Signal Processing IX, Madison, Wisconsin, USA, vol. 10, pp. 41–48 (1999)
12. Zou, H., Zhu, J., Hastie, T.: New Multicategory Boosting Algorithms Based on Multicategory Fisher-Consistent Losses. Annals of Applied Statistics 2, 1290–1306 (2008)
13. Lee, Y., Lin, Y., Wahba, G.: Multicategory Support Vector Machines Theory and Application to the Classification of Microarray Data and Satellite Radiance Data. Journal of the American Statistical Association 99, 67–81 (2004)
14. Huy, T., Takeda, K., Itakura, F.: Maximum a Posterior Probability and Cumulative Distribution Function Equalization Methods for Speech Spectral Estimation with Application in Noise Suppression Filtering. In: Faundez-Zanuy, M., Janer, L., Esposito, A., Satue-Villar, A., Roure, J., Espinosa-Duro, V. (eds.) NOLISP 2005. LNCS (LNAI), vol. 3817, pp. 328–337. Springer, Heidelberg (2006)
15. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (2004)
16. Kolman, B., Beck, R.E.: Elementary Linear Programming with Applications, 2nd edn. Academic Press, Elsevier Science & Technology Books (1995)
17. Arribas, J.I., Sueiro, J.C., Lopez, C.A.: Estimation of Posterior Probabilities with Neural Networks. In: Hanbook of Neural Engineering, IEEE Press, A John Wiley and Sons Inc. (2007)
18. Loewenster, D.M., Berman, H.M., Hirsh, H.: Maximum A Posteriori Classification of DNA Structure from Sequence Information. In: Proceedings of Pacific symposium on Biocomputing, Hawaii, USA, pp. 667–668 (1998)
19. Jaroudi, A.E., Makhoul, J.: A New Error Criterion for Posterior Probability Estimation with Neural Nets. In: International Joint Conference on Neural Networks, San Diego, CA, USA, vol. 90, pp. 185–192 (1990)
20. Chang, C.C., Lin, C.-J.: LIBSVM: a Library for Support Vector Machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`