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

International Development Engineering
Tokyo Institute of Technology

Nopriadi
nopriadi@yahoo.com

Yukihiko Yamashita
yamasita@ide.titech.ac.jp

1. Introduction

Maximum *a Posteriori* (MAP) has been adopted and studied in pattern recognition for the purpose of classification. In MAP classifier the information of *a posteriori* probability $P(y|x)$ is essential, but calculation for estimating $P(y|x)$ is not easy. Beside the cost function should be *Strict Sense Bayesian* (SSB), it also should be solved by non linear optimization. We propose a new approach for classification problem based on MAP than can be solved by linear programming.

> We do not estimate $P(y|x)$ directly for classification, but we estimate a surrogate function $w(x,y)$ that satisfies
> $$\arg\max_y w(x,y) = \arg\max_y P(y|x).$$

2. Maximum *a Posteriori* (MAP) Estimation

Let y be a category to be estimated from a data x,
$P(x)$ is a prior probability density function (p.d.f) of x,
$P(y)$ is a prior probability of y,
$P(x|y)$ is a conditional p.d.f of x given y, and
$P(y|x)$ is *a posteriori* of y.

Bayes theorem can be derived from the joint probability of x and y is

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x).$$

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

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

A classifier system is designed to estimate a category $\hat{y}$ for an unlearned pattern x. The MAP estimates category $\hat{y}$ that is defined as the mode of *a posterior* probability as follows:

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

3. Problem Estimating *P*(*y*|*x*)

Suerri et al. proposed a cost function C(h,d) to estimate $P(y|x)$ that can be reformulated for 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 a Kronecker delta. The function h(x) becomes *a posteriori* if and only if C(h,d) is SSB and can be expressed as follows:

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

where $g_i(\alpha)$ is any positive function $(g_i(\alpha) > 0, 0 \le \alpha \le 1)$ that doesn't depend on h.

> The problem of providing cost function to estimate *a posteriori* is that it should be SSB (it makes the freedom to choose a cost function to be restricted) and it couldn't be solved by using linear optimization.

4. Model Formalization

Our new approach is not to estimate $P(y|x)$ directly for classification, but use a surrogate function $w(x,y)$ that satisfies:

$$\arg\max_y w(x,y) = \arg\max_y P(y|x).$$

Consider a set of samples $\{(x_i, y_i)\}_{i=1}^{N}$ and restrict a problem to binary classification, i.e. $y \in \{-1,+1\}$.

The criterion of $w(x,y)$ is written as:

$$\begin{aligned}&\max && \sum_{y\in\{-1,+1\}} E_x P(y|x) \min(w(x,y),1)\\ &\text{subject to} && \sum_{y\in\{-1,+1\}} E_x w(x,y) = 1,\\ &&& w(x,y) \ge 0.\end{aligned}$$

The solution of the optimization problem above shows $w(x,y)$ behaves in similar way as MAP, where

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

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

We assume that the function $w(x,y)$ is defined by using a kernel function as follows

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

where $k(x,z)$ is Gaussian kernel, i.e.

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

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

By exchanging the ensemble mean by sample mean the criterion becomes

$$\begin{aligned}&\max && \frac{1}{N}\sum_{i=1}^{N} \min\left(\alpha_{y_i,j} k(x_i, x_j), 1\right)\\ &\text{subject to} && \sum_{y\in\{-1,+1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{y,j} k(x_i, x_j) = 1,\\ &&& \sum_{j=1}^{N} \alpha_{y,j} k(x_i, x_j) \ge 0,\ \forall i\ .\end{aligned}$$

To simplify the calculation in linear programming problem, the condition $w(x_i, y) \ge 0$ could be approximated by $\alpha_{y,j} \ge 0$ if $k(x,y) > 0$.

A slack variables $\xi_i \ge 0$ is introduced, then we have

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

Finally we have a linear programming problem of 3N variables and that can be expressed as follows:

$$\max$$

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

subject to :

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

$$\alpha_{y_i,j} \ge 0 \quad (y \in \{-1,+1\},\ j = 1,2,\ldots,N),$$

$$\sum_{j=1}^{N} \alpha_{y,j} k(x_i, x_j) - \xi_i \le 1 \quad (i = 1,2,\ldots,N),$$

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

5. Experiment and Result

1. In the experiment we used an open source package GNU Linear Programming Kit (GLPK) for optimization problem.
2. We conducted experiment by using 13 data sets from the UCI repository.
3. To estimate parameter γ for each data set we used 5 fold cross validation. We used the first 5 realizations of train data for validation. For each realization we performed a cross validation and then we chose a median of the best values of parameter (with least error).
4. We compare our proposed method (MAPLP) with other methods: Support Vector Machine (SVM), Kernelized Maximum *a Posteriori* (KMAP), a single RBF classier, AdaBoost (AB), regularized AdaBoost (ABR), and Kernel Fisher Discriminant (KFD).

Experimental Results (Error) and comparison to other methods

Data Set		MAPLP	SVM	KMAP	RBF	AB	AB$_R$	KFD
Banana	4.217	**10.7 ± 0.6**	11.5 ± 0.7	10.9 ± 0.5	10.8 ± 0.6	12.3 ± 0.7	10.9 ± 0.4	10.8 ± 0.5
Breast Cancer	0.316	**25.8 ± 4.0**	26.0 ± 4.7	28.1 ± 4.4	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	25.6 ± 2.1	24.3 ± 1.9	26.5 ± 2.3	23.8 ± 1.8	**23.2 + 1.6**
Flare Solar	1.778	33.0 ± 7.8	**32.4 + 1.8**	32.8 ± 1.7	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**	26.6 ± 2.3	24.7 ± 2.4	27.5 ± 2.5	24.3 ± 2.1	23.7 ± 2.2
Heart	0.237	17.8 ± 3.2	**16.0 + 3.3**	16.4 ± 3.8	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	7.4 ± 1.5	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	1.6 ± 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	12.5 ± 0.7	10.0 ± 1.0	10.1 ± 0.5	**9.5 + 0.7**	10.5 ± 0.6
Thyroid	1.685	5.0 ± 2.3	4.8 ± 2.2	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	24.2 ± 4.6	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	**2.3 +0.1**	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	9.7 ± 0.4	10.7 ± 1.1	10.8 ± 0.6	**9.8 + 0.8**	9.9 ± 0.4

Computational Time of Learning & Classification Process (in seconds)

Dataset	MAPLP	SVM	KMAP
Banana	135.3	4.030	1366
Breast cancer	20.6	0.700	779.2
Diabetes	216	2.332	536.2
Flare Solar	498.2	4.536	1666
German	765.7	5.734	1873
Heart	19.9	0.596	47.9
Image	12520	1.938	2135
Ringnorm	164.6	14.104	2432
Splice	365.4	8.402	1432
Thyroid	13.9	0.308	151.7
Titanic	13.4	1.170	487
Twonorm	181.4	9.800	941.5
waveform	200.9	10.444	1666

6. Discussion

1. The experimental results show that our proposed method has promising performance. It is competitive to the others and superior on some data sets (banana, breast cancer and titanic).
2. The computational complexity shows that MAPLP is slower than LIBSVM (library for support vector machines). However, LIBSVM is a specialized program for SVM and on the other hand we used GLPK that is a general purpose library. If we compare to KMAP (the method is also based on MAP and we used GLPK) MAPLP is faster.
3. The cost function of SVM to obtain the optimal separating hyperplane is

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

The slack variables ζ_i expresses the hinge loss, where

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

The concept of hinge loss in SVM is quite similar to our criterion. 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 the equation below is considered as a hinge loss.

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

7. Conclusion

1. We proposed a new approach for classification problem based on MAP. Instead of estimating P(y|x), we use a surrogate function $w(x,y)$ that behaves in a similar way to MAP Classier.
2. The advantage of this approach is the cost function can be directly optimized with linear programming and we have only one parameter to adjust the system.
3. The experiment using 13 data sets shows that the proposed method has promising performance and it is competitive enough to the other state-of-the-art classification methods.

8. References

1. 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, vol. 10, pp. 645{656 (1999)
2. Xu, Z., Huang, K., Zhu, J., King, I., Lyu, M.R.: A Novel Kernel-Based Maximum a Posteriori Classication Method. J. Neural Networks, vol. 22, pp. 121-146 (2009)