LETTER

Improving the Accuracy of Least-Squares Probabilistic Classifiers

Makoto YAMADA[†a)], **Masashi SUGIYAMA**[†,††], ***Members***, **Gordon WICHERN**[†††], ***and*** **Jaak SIMM**[†], ***Nonmembers***

SUMMARY The *least-squares probabilistic classifier* (LSPC) is a computationally-efficient alternative to kernel logistic regression. However, to assure its learned probabilities to be non-negative, LSPC involves a post-processing step of rounding up negative parameters to zero, which can unexpectedly influence classification performance. In order to mitigate this problem, we propose a simple alternative scheme that directly rounds up the classifier's negative outputs, not negative parameters. Through extensive experiments including real-world image classification and audio tagging tasks, we demonstrate that the proposed modification significantly improves classification accuracy, while the computational advantage of the original LSPC remains unchanged.

key words: *least-squares probabilistic classifier, kernel logistic regression, density ratio, PASCAL VOC 2010, freesound*

1. Introduction

The *least-squares probabilistic classifier* (LSPC) [1] is an efficient non-linear probabilistic classification method that learns class-posterior probabilities. In LSPC, a linear combination of kernels centered at training points is employed as a model of class-posterior probabilities. Then the LSPC model is trained so that the squared difference to the true class-posterior probability is minimized. An advantage of this linear least-squares formulation is that the global optimal solution can be obtained *analytically* (cf. kernel logistic regression [2], [3], which can be used for similar purposes, but requires iterative optimization such as Newton's method). However, since LSPC involves a post-processing step of rounding up negative model parameters to zero for assuring the non-negativity of probability estimates, learned class-posterior probabilities can change unexpectedly.

In order to mitigate this problem, we propose a simple alternative scheme that directly rounds up the classifier's negative outputs, as opposed to its negative parameters. While the original parameter-rounding scheme influences learned class-posterior probabilities *globally* through the change of parameters for basis functions, the proposed output-rounding scheme confines the influence to only those points where probabilities are negative (see Fig. 1, which will be explained in detail later). This localization effect is expected to prevent the degradation of classification performance. Through extensive experiments on real-world image classification and automatic audio tagging tasks, we demonstrate that the proposed modification to LSPC significantly contributes to improving classification accuracy, while maintaining the computational advantage of the original LSPC algorithm.

Manuscript received October 20, 2010.
Manuscript revised January 28, 2011.

[†]The authors are with Tokyo Institute of Technology, Tokyo, 152–8552 Japan.
[††]The author is with JST PRESTO, Tokyo, 152–8552 Japan.
[†††]The author is with Arizona State University, Tempe, AZ 85287–8709, U.S.A.
a) E-mail: yamada@sg.cs.titech.ac.jp
DOI: 10.1587/transinf.E94.D.1337

2. Least-Squares Approach to Probabilistic Classification

In this section, we review the *least-squares probabilistic classifier*[*] (LSPC) [1].

2.1 Problem Formulation

Suppose we are given n paired samples of input $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ and its label $y \in \{1, \ldots, c\}$ (c denotes the number of classes):

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n,$$

which are independently drawn from a joint probability distribution with density $p(\boldsymbol{x}, y)$. Our goal is to estimate the class-posterior probability $p(y|\boldsymbol{x})$ from the samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The class-posterior probability allows us to classify a test sample $\boldsymbol{x}$ to the class $\widehat{y}$ with confidence $p(\widehat{y}|\boldsymbol{x})$:

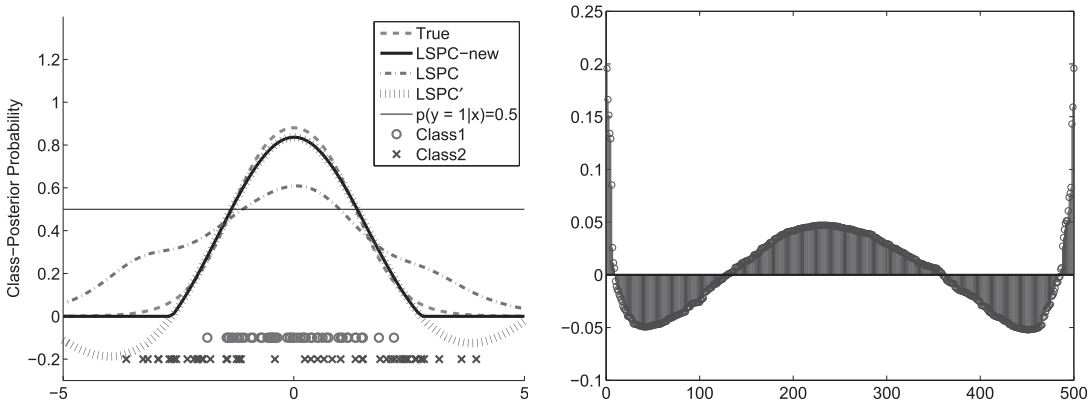
$$\widehat{y} := \operatorname*{argmax}_y p(y|\boldsymbol{x}).$$

Let us denote the marginal density of $\boldsymbol{x}$ by $p(\boldsymbol{x})$. Then the class-posterior probability can be expressed as

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})}, \tag{1}$$

where we assume $p(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{X}$. This density-ratio expression is utilized in the derivation of LSPC.

[*]More precisely, the LSPC method we are reviewing here was referred to as 'LSPC (full)' in the original LSPC paper [1], where 'full' means that all kernels are used for learning (cf. Eq. (2)). On the other hand, a computationally more efficient variant of LSPC where irrelevant kernels are removed was also proposed in the original paper. However, since the range of application of this model simplification idea is limited to localized kernels such as Gaussian kernels, we decided to adopt the more general 'LSPC (full)' method in this paper. We note that the results we present in this paper can also be applied to the simplified LSPC method.



(a) $p(y = 1|x)$ as a function of x. (b) $\widetilde{\alpha}_1$ as a function of sample index l.

Fig. 1 (a): True class-posterior probability, and class-posterior probabilities estimated by LSPC-new (negative outputs are rounded up to zero), LSPC (negative parameters are rounded up to zero), and LSPC' (negative parameters are used as they are). (b): The parameter values of $\widetilde{\alpha}_1$, where the training samples were sorted as $x_1 \leq x_2 \leq \cdots \leq x_{500}$.

2.2 Least-Squares Probabilistic Classifier

Let us model the class-posterior probability $p(y|\boldsymbol{x})$ for class y by the following linear model:

$$\sum_{l=1}^{n} \alpha_{y,l}\phi_l(\boldsymbol{x}) = \boldsymbol{\alpha}_y^\top \boldsymbol{\phi}(\boldsymbol{x}),$$

where $^\top$ denotes the transpose of a matrix or a vector and $\boldsymbol{\alpha}_y = (\alpha_{y,1}, \ldots, \alpha_{y,n})^\top$ are parameters to be learned from training samples. The basis function is written as

$$\boldsymbol{\phi}(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n))^\top, \tag{2}$$

where

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right)$$

is the Gaussian kernel with width σ.

Then, an empirical and regularized solution of LSPC is given as

$$\widetilde{\boldsymbol{\alpha}}_y := \operatorname*{argmin}_{\boldsymbol{\alpha}_y}\left[\frac{1}{2}\boldsymbol{\alpha}_y^\top \widehat{\boldsymbol{H}}\boldsymbol{\alpha}_y - \widehat{\boldsymbol{h}}_y^\top \boldsymbol{\alpha}_y + \lambda \boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_y\right], \tag{3}$$

where $\lambda \boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_y$ for $\lambda > 0$ is a regularizer, and

$$\widehat{\boldsymbol{H}} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^\top, \quad \widehat{\boldsymbol{h}}_y := \frac{1}{n}\sum_{i:y_i=y} \boldsymbol{\phi}(\boldsymbol{x}_i).$$

Since Eq. (3) is an unconstrained quadratic minimization problem, the global optimal solution $\widetilde{\boldsymbol{\alpha}}_y$ can be obtained as

$$\widetilde{\boldsymbol{\alpha}}_y = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_n)^{-1}\widehat{\boldsymbol{h}}_y,$$

where $\boldsymbol{I}_n$ denotes the n-dimensional identity matrix.

By definition, the class-posterior probability should be non-negative. However, $\widetilde{\boldsymbol{\alpha}}_y$ obtained above can take negative values. In order to assure the learned probabilities to be non-negative, the negative elements in $\widetilde{\boldsymbol{\alpha}}_y$ are rounded up to zero as follows:

$$\widehat{\boldsymbol{\alpha}}_y := \max(\boldsymbol{0}_n, \widetilde{\boldsymbol{\alpha}}_y),$$

where $\boldsymbol{0}_n$ denotes the n-dimensional vector with all zeros, and the 'max' operation for vectors is applied in an element-wise manner.

Finally, given a test input point, an estimator of the class-posterior probability is obtained via normalization as

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\widehat{\boldsymbol{\alpha}}_y^\top \boldsymbol{\phi}(\boldsymbol{x})}{\sum_{y'=1}^{c} \widehat{\boldsymbol{\alpha}}_{y'}^\top \boldsymbol{\phi}(\boldsymbol{x})}. \tag{4}$$

3. Improving Accuracy of LSPC

In the original LSPC paper, it was claimed that the learned parameters $\widetilde{\boldsymbol{\alpha}}_y$ are usually non-negative when the basis functions and the regularization parameter value are chosen appropriately, e.g., by cross-validation. In this section, we point out that the above claim is not always true through a numerical example, and illustrate that rounding up negative parameters can actually have strong influence on the learned class-posterior probabilities. Then we describe a simple alternative scheme for assuring the learned probabilities to be non-negative, and illustrate its usefulness.

3.1 Influence of Rounding-Up Negative Parameters

First, we illustrate how the class-posterior probabilities are learned by LSPC, and investigate the influence of rounding-up negative model parameters.

Let us consider a one-dimensional binary classification problem (i.e., $d = 1$ and $c = 2$). We independently draw samples in each class from the following class-conditional densities:

$$p(x|y = 1) = N(x; 0, 1),$$

$$p(x|y=2) = \frac{1}{2}N(x;-2,1) + \frac{1}{2}N(x;2,1),$$

where $N(x;\mu,\tau^2)$ denotes the Gaussian density with mean μ and variance τ^2. We used 250 training samples per class (500 samples in total), and 250 test samples per class (500 samples in total). The kernel width σ and regularization parameter λ in LSPC were chosen based on 2-fold cross-validation.

Let $\widetilde{p}(y|\boldsymbol{x})$ be an estimator of the class-posterior probability for $\widetilde{\boldsymbol{\alpha}}_y$ (i.e., without the rounding-up operation), which we refer to as LSPC':

$$\widetilde{p}(y|\boldsymbol{x}) = \frac{\widetilde{\boldsymbol{\alpha}}_y^\top \boldsymbol{\phi}(\boldsymbol{x})}{\sum_{y'=1}^{c} \widetilde{\boldsymbol{\alpha}}_{y'}^\top \boldsymbol{\phi}(\boldsymbol{x})}. \tag{5}$$

Note that the output of LSPC' is not necessarily a probability since it can be negative or larger than one. Figure 1 (a) shows the true class-posterior probability and the class-posterior probabilities estimated by LSPC' (4) and LSPC' (5). The graph shows that, without the rounding-up operation, the class-posterior probability estimates take negative values around $x \in (-5,-3)$ and $x \in (3,5)$ in Fig. 1 (a). On the other hand, the LSPC solution always takes non-negative values thanks to the rounding-up operation. However, the LSPC estimate of the class-posterior probability are significantly different from the true class-posterior probability.

In order to investigate the effect of rounding-up negative parameters to zero in more detail, we plotted the values of $\widetilde{\boldsymbol{\alpha}}_1$ in Fig. 1 (b), where the training samples were sorted as $x_1 \le x_2 \le \cdots \le x_{500}$. The graph shows that many parameters actually took negative values, even though the estimate of class-posterior probabilities take negative values only locally. This shows that, rounding-up negative parameters to zero can actually have a strong influence on the learned class-posterior probability even when negative values are taken only locally.

3.2 Rounding-Up Negative Outputs

In order to overcome the above drawback, we propose to *locally* modify the solution as follows:

$$\overline{p}(y|\boldsymbol{x}=\widetilde{\boldsymbol{x}}) = \begin{cases} \frac{1}{Z}\max(0, \widetilde{\boldsymbol{\alpha}}_y^\top \boldsymbol{\phi}(\widetilde{\boldsymbol{x}})) & \text{if } Z > 0, \\ \frac{1}{c} & \text{otherwise,} \end{cases} \tag{6}$$

where $Z = \sum_{y'=1}^{c} \max(0, \widetilde{\boldsymbol{\alpha}}_{y'}^\top \boldsymbol{\phi}(\widetilde{\boldsymbol{x}}))$. Below, we refer to this method as 'LSPC-new'.

Figure 1 (a) also includes the class-posterior probabilities estimated by LSPC-new. As shown in the graph, the class-posterior probability estimated by LSPC-new and the true-class posterior probability have almost the same profile. On the other hand, the class-posterior probability obtained by the original LSPC has been strongly influenced by rounding-up negative parameters to zero.

4. Experiments

In this section, we compare the performance of LSPC-new, LSPC, and kernel logistic regression (KLR) on a real-world image classification task using the *PASCAL Visual Object Classes (VOC) 2010* dataset [4] and a real-world automatic audio-tagging task using te data collected by the *Freesound* project [5]. All tuning parameters (i.e., the kernel width σ and regularization parameter λ) were chosen based on 2-fold cross-validation. We used the MATLAB® implementation of KLR included in the 'minFunc' package [6]. Comparison is carried out in terms of classification performance and CPU computation time required for training each classifier after the Gaussian width and the regularization parameter are chosen by cross-validation.

4.1 PASCAL VOC 2010 Datasets

The VOC 2010 dataset consists of 20 binary classification tasks of identifying the existence of a person, aeroplane, etc. in each image. The total number of images in the dataset is 11319, and we used 1000 randomly chosen images for training and the rest for testing.

We first extracted visual features from each image with the *Speed Up Robust Features* (SURF) algorithm [7]. We then ran the *k-means* clustering algorithm [8] in the SURF space and obtained 500 cluster centers as *visual words*. Then, we computed a 500-dimensional *bag-of-feature* vector by counting the number of visual words in each image.

When evaluating the image classification performance, it is important to take into account both the false positive rate and true positive rate. Here we adopted the *area under the ROC curve* (AUC) as our error metric [9]. We randomly sampled the training and test data 50 times, and computed the means and standard deviations of the AUC.

Table 1 shows the mean AUC values (with standard deviations in brackets) over 50 trials. The best method in terms of the mean AUC and comparable methods according to the *t-test* at the significance level 5% are specified by bold face. The results showed that LSPC-new outperforms LSPC for all tasks and is slightly more accurate than KLR with much less computational cost.

4.2 Freesound Datasets

The *Freesound* dataset [5] consists of various audio files annotated with word tags such as 'people', 'noisy', and 'restaurant'. Note that such tags are not exclusive, meaning that each audio file can have multiple tags.

We extracted audio files from among all files in the dataset containing any of the 50 most used tags and between 3–60 seconds in length. We then used 180 randomly selected uncompressed audio files with a sampling rate greater than 44.1 kHz as our training set, and 1500 randomly selected audio files which were stored in a compressed format for testing. We used the *hidden Markov kernel* [10], instead of the simple Gaussian kernel due to the sequential nature of audio files. We repeat the audio-tagging experiment 50 times, by changing the random seed.

We computed the AUC value over all 1500 test samples

Table 1 Mean AUC values (with standard deviations in brackets) over 50 trials for the PASCAL VOC dataset. The best method in terms of the mean AUC and comparable methods according to the *t-test* at the significance level 5% are specified by bold face.

Datasets	LSPC-new	LSPC	KLR
Aeroplane	**82.6** (1.0)	78.8 (1.3)	**83.0**(1.3)
Bicycle	**77.7** (1.7)	60.0(12.4)	76.6(3.4)
Bird	68.7 (2.0)	49.1 (5.7)	**70.8**(2.2)
Boat	**74.4** (2.0)	62.1 (4.1)	72.8(2.6)
Bottle	**65.4** (1.8)	63.0 (2.0)	62.1(4.3)
Bus	**85.4** (1.4)	80.3 (2.6)	**85.6**(1.4)
Car	**73.0** (0.8)	69.0 (2.6)	72.1(1.2)
Cat	**73.6** (1.4)	68.6 (5.0)	**74.1**(1.7)
Chair	**71.0** (1.0)	64.8 (2.0)	**70.5**(1.0)
Cow	**71.7** (3.2)	53.5 (9.0)	69.3(3.6)
Diningtable	**75.0** (1.6)	69.1 (2.4)	71.4(2.7)
Dog	**69.6** (1.0)	58.6 (6.0)	**69.4**(1.8)
Horse	**64.4** (2.5)	54.6 (4.4)	61.2(3.2)
Motorbike	**77.0** (1.7)	73.2 (2.5)	75.9(3.3)
Person	**67.6** (0.9)	65.9 (1.2)	67.0(0.8)
Pottedplant	**66.2** (2.6)	58.0 (4.6)	61.9(3.2)
Sheep	**77.8** (1.6)	57.6(12.9)	74.0(3.8)
Sofa	**67.4** (2.7)	66.1 (1.7)	65.4(4.6)
Train	**79.2** (1.3)	67.0 (7.7)	**78.4**(3.0)
Tvmonitor	**76.7** (2.2)	70.1 (2.4)	**76.6**(2.3)
Average	73.2 —	64.5 —	71.9 —
Train time [sec]	0.7 —	0.7 —	24.6 —

Table 2 Mean AUC values (with standard deviations in brackets) over all audio files for the Freesound dataset.

	LSPC-new	LSPC	KLR
AUC	70.1 (9.6)	64.4 (9.5)	66.7 (10.3)
Train time [sec]	0.005	0.005	0.612

for each tag, and this was averaged over all 50 trials and all 50 tags. Table 2 summarizes the performance of LSPC-new, LSPC, and KLR. The results show that LSPC-new is more accurate than LSPC with comparable computation time, and LSPC-new provides comparable classification performance to KLR with significantly less computation time.

5. Conclusions

Least-squares probabilistic classifier (LSPC) has been demonstrated to be a computationally-efficient alternative to kernel logistic regression (KLR). However, since LSPC involves a post-processing step of rounding-up negative parameters to zero, its performance can be degraded if many parameters take negative values. In this paper, we proposed not to round-up negative parameters, but to directly round-up negative outputs of LSPC. This localizes the influence of the rounding-up operation on the learned class-posterior probabilities. Through extensive experiments including real-world image classification and audio tagging tasks, we showed that the proposed modification significantly improves classification accuracy, while the computational advantage of LSPC remains unchanged.

Acknowledgement

MY acknowledges the JST PRESTO program for financial support, and MS acknowledges SCAT, AOARD, and the JST PRESTO program for financial support.

References

[1] M. Sugiyama, "Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting," IEICE Trans. Inf. & Syst., vol.E93-D, no.10, pp.2690–2701, Oct. 2010.
[2] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2001.
[3] T. Minka, "A comparison of numerical optimizers for logistic regression," Microsoft Research, Tech. Rep., 2007.
[4] M. Everingham, L.V. Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2010 (VOC2010) results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html
[5] "Freesound," http://www.freesound.org
[6] "minFunc," http://people.cs.ubc.ca/˜chmidtm/Software/minFunc.html
[7] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Surf: Speeded up robust features," Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.
[8] C.M. Bishop, Pattern recognition and machine learning, Springer-Verlag, New York, 2006.
[9] A.P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," Pattern Recognit., vol.30, pp.1145–1159, 1997.
[10] G. Wichern, J. Xue, H. Thornburg, B. Mechtley, and A. Spanias, "Segmentation, indexing and retrieval of environmental and natural sounds," IEEE Trans. Audio. Speech Lang. Process., vol.18, no.3, pp.688–707, 2010.