Discriminative Prior Bias Learning for Pattern Classification

Takumi Kobayashi and Kenji Nishida
National Institute of Advanced Industrial Science and Technology,
1-1-1 Umezono, Tsukuba, Japan

Keywords: Pattern Classification, Discriminative Learning, Bias, SVM.

Abstract: Prior information has been effectively exploited mainly using probabilistic models. In this paper, by focusing on the *bias* embedded in the classifier, we propose a novel method to discriminatively learn the *prior bias* based on the extra prior information assigned to the samples other than the class category, *e.g.*, the 2-D position where the local image feature is extracted. The proposed method is formulated in the framework of maximum margin to adaptively optimize the biases, improving the classification performance. We also present the computationally efficient optimization approach that makes the method even faster than the standard SVM of the same size. The experimental results on patch labeling in the on-board camera images demonstrate the favorable performance of the proposed method in terms of both classification accuracy and computation time.

1 INTRODUCTION

Prior information has been effectively exploited in the fields of computer vision and machine learning, such as for shape matching (Jiang et al., 2009), image segmentation (El-Baz and Gimel'farb, 2009), graph inference (Cremers and Grady, 2006), transfer learning (Jie et al., 2011) and multi-task learning (Yuan et al., 2013). Learning prior has so far been addressed mainly in the probabilistic framework on the assumption that the prior is defined by a certain type of generative probabilistic model (Wang et al., 2010; Kapoor et al., 2009); especially, non-parametric Bayesian approach further considers the hyper priors of the probabilistic models (Ghosh and Ramamoorthi, 2003).

In this paper, we focus on the classifier, $y = \boldsymbol{w}^\top \boldsymbol{x} + b$, and especially on the bias term, so called 'b' term (Poggio et al., 2001)[1], while some transfer learning methods are differently built upon the prior of the weight $\boldsymbol{w}$ for effectively transferring the knowledge into the novel class categories (Jie et al., 2011; Gao et al., 2012) and the prior of $\boldsymbol{w}$ also induces a regularization on $\boldsymbol{w}$. The bias is regarded as rendering the prior information on the class probabilities (Bishop, 1995; Van Gestel et al., 2002) and we aim to learn the *unstructured* prior bias b without assuming any specific models. While the bias b is generally set as a constant across samples depending only on the class category, in this study we define it adaptively based on the extra prior information other than the class category, as follows.

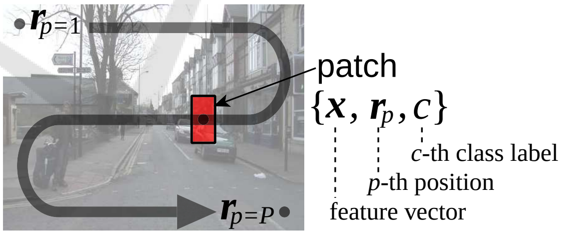


Figure 1: Patch labeling. The task is to predict the class labels c of the patches, each which consists of the appearance feature vector $\boldsymbol{x}$ and the prior position p. Note that there are P positions in total.

Suppose samples are associated with the extra prior information $p \in \{1,..,P\}$ as well as the class category $c \in \{1,..,C\}$, where P and C indicate the total number of the prior types and the class categories, respectively. For instance, in the task of labeling patches on the on-board camera images, each patch (sample) is assigned with the appearance feature $\boldsymbol{x}$, the class category c and the position (extra prior information) p, as shown in Fig. 1. Not only the feature $\boldsymbol{x}$ but also the prior position p where the feature is extracted is useful to predict the class category of the patch; the patches on an upper region

[1] In this paper, we describe the classifier in such a linear form for simplicity, but our proposed method also works on the kernel-based classifier by simply replacing the feature $\boldsymbol{x}$ with the kernel feature $\boldsymbol{\phi_x}$ in the reproducing kernel Hilbert space.

Kobayashi T. and Nishida K..
Discriminative Prior Bias Learning for Pattern Classification.
DOI: 10.5220/0004813600670075
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 67-75
ISBN: 978-989-758-018-5

probably belong to *sky* and the lower region would be *road*, even though the patches extracted from those two regions are both less textured, resulting in similar features.

The probabilistic structure that we assume in this study is shown in Fig. 2b with comparison to the `simple` model in Fig. 2a. By using generalized linear model (Bishop, 2006), the standard classifier (Fig. 2a) is formulated to estimate the posterior on the class category c as[2]

$$\log \mathsf{p}(c|\boldsymbol{x}) \sim \log \mathsf{p}(\boldsymbol{x}|c) + \log \mathsf{p}(c) = \boldsymbol{w}_c^\top \boldsymbol{x} + b_c, \quad (1)$$

where $b_c = \log \mathsf{p}(c)$ indicates the class-dependent bias. On the other hand, the `proposed` model (Fig. 2b) using the prior p induces the following classifier;

$$\begin{aligned}\log \mathsf{p}(c|\boldsymbol{x}, p) &\sim \log \mathsf{p}(\boldsymbol{x}|c) + \log \mathsf{p}(p|c) + \log \mathsf{p}(c)\\ &= \boldsymbol{w}_c^\top \boldsymbol{x} + b_c^{[p]}, \quad (2)\end{aligned}$$

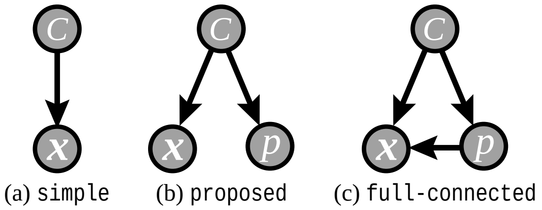
where the bias $b_c^{[p]} = \log \mathsf{p}(p|c) + \log \mathsf{p}(c)$ is dependent on both the class category c and the prior information p. Thus, if the bias could be properly determined, the classification performance would be improved compared to the standard classification model (1). One might also consider the `full-connected` model shown in Fig. 2c whose classifier is formulated by

$$\begin{aligned}\log p(c|\boldsymbol{x}, p) &\sim \log p(\boldsymbol{x}|c, p) + \log p(p|c) + \log p(c)\\ &= \boldsymbol{w}_c^{[p]\top} \boldsymbol{x} + b_c^{[p]}, \quad (3)\end{aligned}$$

where the classifier weight $\boldsymbol{w}_c^{[p]}$ relies on the prior p as the bias $b_c^{[p]}$ does. This model is more complicated and consumes large memory storage since the classifier model $\{\boldsymbol{w}_c^{[p]}, b_c^{[p]}\}$ is prepared for respective priors $p = 1, .., P$. And, due to the high degree of freedom (D.O.F) of this model, it would be vulnerable to over-learning. These models are summarized in Table 1 and will be again discussed later.

In this paper, we propose a novel method for discriminatively learning the prior biases $b_c^{[p]}$ in (2) to improve the classification performance. The proposed method is formulated in the optimization problem of the maximum margin criterion (Smola et al., 2000). We also propose the computationally efficient approach for the optimization which contains large amount of samples drawn from all the priors $p \in \{1, .., P\}$. Thereby, the proposed method is even faster than the standard SVM (Vapnik, 1998) of the same size, while providing the high-performance classifier that exploits the prior information.

[2] '$\sim$' in (1) means the equality in disregard of the irrelevant constant term $\log \mathsf{p}(\boldsymbol{x})$ or $\log \mathsf{p}(\boldsymbol{x}, p)$ in (2) and (3).



(a) `simple` (b) `proposed` (c) `full-connected`

Figure 2: Graphical models to depict the probabilistic dependencies. The notations c, $\boldsymbol{x}$ and p denote the class category, the (appearance) feature vector and the extra prior information, respectively. The arrows show the probabilistic dependencies. (a) The feature $\boldsymbol{x}$ is simply drawn from the class category c in the `simple` model. (b) The `proposed` model incorporates the extra prior information p which is connected to $\boldsymbol{x}$ via c. (c) Those three variables are fully connected in the `full-connected` model.

Table 1: Classification methods for c-th class category. The dimensionality of the feature vector is denoted by D, $\boldsymbol{x} \in \Re^D$, and the number of prior types is P.

Method	Model	D.O.F
`simple`	$y_c = \boldsymbol{w}_c^\top \boldsymbol{x} + b_c$	$D+1$
`proposed`	$y_c = \boldsymbol{w}_c^\top \boldsymbol{x} + b_c^{[p]}$	$D+P$
`full-connected`	$y_c = \boldsymbol{w}_c^{[p]\top} \boldsymbol{x} + b_c^{[p]}$	$PD+P$

2 BIAS LEARNING

We detail the proposed method by first defining the formulation for learning the biases and then presenting the computationally efficient approach to optimize them. As we proceed to describe a general form regarding the prior biases, it might be helpful for understanding to refer to the task of labeling patches in on-board camera images as shown in Fig. 1; the sample is represented by the appearance feature $\boldsymbol{x}$ and the prior position $p \in \{1, .., P\}$.

2.1 Formulation

We consider a binary class problem for simplicity and take a one-vs-rest approach for multi-class tasks. Suppose we have P types of prior information, and let $\boldsymbol{x}_i^{[p]} \in \Re^D$ denote the D-dimensional feature vector of the i-th sample ($i = 1, .., n^{[p]}$) drawn from the p-th type of prior. As described in Sec.1, we deal with the classification defined by

$$y = \boldsymbol{w}^\top \boldsymbol{x}^{[p]} + b^{[p]}, \quad (4)$$

where y denotes the classifier output which is subsequently thresholded by zero for performing binary classification, and $\boldsymbol{w}$ and $b^{[p]}$ are the classifier weight vector and the bias, respectively. Note again that the bias $b^{[p]}$ depends on the p-th type of prior, $p \in \{1,..,P\}$. The classifier (4) can be learned via the following optimization formulation in the framework of maximum margin (Smola et al., 2000);

$$\min_{\boldsymbol{w},\{b^{[p]}\}_p} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_p^P\sum_i^{n^{[p]}}\xi_i^{[p]} \tag{5}$$

$$s.t.\ \forall p \in \{1,..,P\},\ \forall i \in \{1,..,n^{[p]}\},$$
$$y_i^{[p]}(\boldsymbol{w}^\top\boldsymbol{x}_i^{[p]} + b^{[p]}) \geq 1 - \xi_i^{[p]},\ \xi_i^{[p]} \geq 0,$$

where C is the cost parameter. This is obviously convex and its Lagrangian is written by

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_p^P\sum_i^{n^{[p]}}\xi_i^{[p]} - \sum_p^P\sum_i^{n^{[p]}}\beta_i^{[p]}\xi_i^{[p]} \tag{6}$$
$$-\sum_p^P\sum_i^{n^{[p]}}\alpha_i^{[p]}\{y_i^{[p]}(\boldsymbol{w}^\top\boldsymbol{x}_i^{[p]} + b^{[p]}) - 1 + \xi_i^{[p]}\},$$

where we introduce the Lagrange multipliers $\alpha_i^{[p]} \geq 0$, $\beta_i^{[p]} \geq 0$. The derivatives of the Lagrangian are

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_p^P\sum_i^{n^{[p]}}\alpha_i^{[p]}y_i^{[p]}\boldsymbol{x}_i^{[p]} = \boldsymbol{0}$$
$$\Rightarrow \boldsymbol{w} = \sum_p^P\sum_i^{n^{[p]}}\alpha_i^{[p]}y_i^{[p]}\boldsymbol{x}_i^{[p]} \tag{7}$$

$$\frac{\partial L}{\partial \xi_i^{[p]}} = C - \alpha_i^{[p]} - \beta_i^{[p]} = 0 \Rightarrow 0 \leq \alpha_i^{[p]} \leq C \tag{8}$$

$$\frac{\partial L}{\partial b^{[p]}} = \sum_i^{n^{[p]}}\alpha_i^{[p]}y_i^{[p]} = 0. \tag{9}$$

Thereby, the dual is finally obtained as

$$\min_{\{\alpha_i^{[p]}\}_{i,p}} \frac{1}{2}\sum_{p,q}^P\sum_i^{n^{[p]}}\sum_j^{n^{[q]}}\alpha_i^{[p]}\alpha_j^{[q]}y_i^{[p]}y_j^{[q]}\boldsymbol{x}_i^{[p]\top}\boldsymbol{x}_j^{[q]} - \sum_p^P\sum_i^{n^{[p]}}\alpha_i^{[p]} \tag{10}$$

$$s.t.\ \forall p,\ \sum_i^{n^{[p]}}\alpha_i^{[p]}y_i^{[p]} = 0,\ \forall i, \forall p,\ 0 \leq \alpha_i^{[p]} \leq C.$$

This is a quadratic programming (QP) analogous to the dual of SVM (Vapnik, 1998) except that there exist P linear equality constraints with respect to $\boldsymbol{\alpha}^{[p]}$. The standard QP solver is applicable to optimize (10), though requiring substantial computation cost. For optimizing QP of the SVM dual, the method of SMO (Platt, 1999) is successfully applied, but in this case, we can not employ it directly due to the multiple equality constraints. In what follows, we present the computationally efficient approach to optimize (10).

2.2 Optimization

A large number of variables $\{\alpha_i^{[p]}\}_{i,p}$ in the QP (10) are inherently partitioned into block-wise variables regarding the prior p; we obtain P blocks of $\boldsymbol{\alpha}^{[p]} = \{\alpha_i^{[p]}\}_{i=1,..,n^{[p]}} \in \Re^{n^{[p]}}$, $p = 1,..,P$. According to those block-wise variables, (10) is decomposed into the following sub-problem as well:

$$\min_{\boldsymbol{\alpha}_i^{[p]}} \frac{1}{2}\sum_{i,j}^{n^{[p]}}\alpha_i^{[p]}\alpha_j^{[p]}y_i^{[p]}y_j^{[p]}\boldsymbol{x}_i^{[p]\top}\boldsymbol{x}_j^{[p]}$$
$$-\sum_i^{n^{[p]}}\alpha_i^{[p]}\left\{1 - y_i^{[p]}\sum_{q\neq p}^P\sum_j^{n^{[q]}}\alpha_j^{[q]}y_j^{[q]}\boldsymbol{x}_i^{[p]\top}\boldsymbol{x}_j^{[q]}\right\} \tag{11}$$
$$s.t.\ \sum_i^{n^{[p]}}\alpha_i^{[p]}y_i^{[p]} = 0,\ \forall i,\ 0 \leq \alpha_i^{[p]} \leq C.$$

This is again a quadratic programming which resembles the SVM dual except for the linear term with respect to $\boldsymbol{\alpha}^{[p]}$ and thus is effectively optimized by using the SMO (Platt, 1999). Therefore, the whole procedure for optimizing (10) consists of iteratively optimizing the sub-problem (11) with respect to the prior p by means of SMO as shown in Algorithm 1.

In order to discuss the convergence of the iterative optimization, we mention the KKT condition of (10) (Fan et al., 2005). The optimizer $\alpha_i^{[p]}$ satisfies the following condition:

$$G_{i,p}(\boldsymbol{\alpha}) + b_i^{[p]}y_i^{[p]} = \lambda_i^{[p]} - \mu_i^{[p]}, \tag{12}$$
$$\lambda_i^{[p]}\alpha_i^{[p]} = 0,\ \mu_i^{[p]}(C - \alpha_i^{[p]}) = 0,\ \lambda_i^{[p]} \geq 0,\ \mu_i^{[p]} \geq 0,$$

where $G_{i,p}(\boldsymbol{\alpha}) = y_i^{[p]}\boldsymbol{x}_i^{[p]\top}\sum_q^P\sum_j^{n^{[q]}}\alpha_j^{[q]}y_j^{[q]}\boldsymbol{x}_j^{[q]} - 1$ is the derivative of the objective function in (10) with respect to $\alpha_i^{[p]}$. This is rewritten into

$$\alpha_i^{[p]} < C:\ G_{i,p}(\boldsymbol{\alpha}) + b_i^{[p]}y_i^{[p]} \geq 0, \tag{13}$$
$$\alpha_i^{[p]} > 0:\ G_{i,p}(\boldsymbol{\alpha}) + b_i^{[p]}y_i^{[p]} \leq 0, \tag{14}$$

and since $y_i^{[p]} \in \{+1,-1\}$, the above conditions result in

$$-y_i^{[p]}G_{i,p}(\boldsymbol{\alpha})\begin{cases} \leq b_i^{[p]} & i \in \mathbb{I}_+^{[p]} \\ \geq b_i^{[p]} & i \in \mathbb{I}_-^{[p]} \end{cases}, \tag{15}$$

where

$$\mathbb{I}_+^{[p]} = \{i|(\alpha_i^{[p]} < C \wedge y_i^{[p]} = 1) \vee (\alpha_i^{[p]} > 0 \wedge y_i^{[p]} = -1)\}, \tag{16}$$

$$\mathbb{I}_-^{[p]} = \{i|(\alpha_i^{[p]} < C \wedge y_i^{[p]} = -1) \vee (\alpha_i^{[p]} > 0 \wedge y_i^{[p]} = 1)\}. \tag{17}$$

Therefore, we can conclude that $\alpha_i^{[p]}$ is a stationary point if and only if

$$\delta^{[p]} \triangleq \left[\max_{i\in\mathbb{I}_+^{[p]}} -y_i^{[p]} G_{i,p}(\boldsymbol{\alpha})\right] - \left[\min_{i\in\mathbb{I}_-^{[p]}} -y_i^{[p]} G_{i,p}(\boldsymbol{\alpha})\right] \le 0. \tag{18}$$

On the basis of this measure, we can stop the iteration when $\max_p \delta^{[p]} < \varepsilon$ with a small tolerance $\varepsilon > 0$. The measure $\delta^{[p]}$ also provides a clue for effectively selecting the prior p to be optimized via (11). That is, we perform the (sub-)optimization (11) at $p^* = \arg\max_p \delta^{[p]}$ so as to effectively minimize $\max_p \delta^{[p]}$. This approach will be empirically validated in the experiment. At the optimum, the bias $b^{[p]}$ is retrieved by

$$b^{[p]} = \frac{1}{|\mathbb{I}^{[p]}|} \sum_{i\in\mathbb{I}^{[p]}} -y_i^{[p]} G_{i,p}(\boldsymbol{\alpha}), \tag{19}$$

$$\text{where } \mathbb{I}^{[p]} = \{i | 0 < \alpha_i^{[p]} < C\}, \tag{20}$$

since the right hand side in (12) equals zero for $i \in \mathbb{I}^{[p]}$.

Finally, we describe the technical tip for further reducing the computational cost in the optimization. From a practical viewpoint, the samples of the two class categories are not equally distributed across the priors $p = 1,..,P$ but are localized in limited number of priors. For instance, in the case of on-board camera images, the *road* never appears in upper regions where the *sky* usually dominates. That is, we occasionally encounter the following sub-problem;

$$\min_{\boldsymbol{\alpha}_i^{[p]}} \frac{1}{2} \sum_{i,j}^{n^{[p]}} \alpha_i^{[p]} \alpha_j^{[p]} y_i^{[p]} y_j^{[p]} \boldsymbol{x}_i^{[p]\top} \boldsymbol{x}_j^{[p]} - \sum_{i}^{n^{[p]}} \alpha_i^{[p]} \left\{ 1 - y_i^{[p]} \sum_{q\neq p}^{P} \sum_{j}^{n^{[q]}} \alpha_j^{[q]} y_j^{[q]} \boldsymbol{x}_i^{[p]\top} \boldsymbol{x}_j^{[q]} \right\} \tag{21}$$

$$s.t. \sum_{i}^{n^{[p]}} \alpha_i^{[p]} y_i^{[p]} = 0,\ \forall i,\ 0 \le \alpha_i^{[p]} \le C, \tag{22}$$

$$\underline{\forall i,\ y_i^{[p]} = 1} \text{ (or } \underline{\forall i,\ y_i^{[p]} = -1}\text{)}. \tag{23}$$

The above QP is trivially optimized by $\boldsymbol{\alpha}^{[p]} = \mathbf{0}$ due to the constraint (22), and the bias $b^{[p]}$ can be determined as

$$b^{[p]} = \begin{cases} +\infty & \forall i,\ y_i^{[p]} = 1 \\ -\infty & \forall i,\ y_i^{[p]} = -1 \end{cases}, \tag{24}$$

which means that the samples from such a prior are definitely classified as positive (or negative) no matter how the appearance features of the samples are. In this case, the class category is solely dependent on the prior information via the bias $b^{[p]} \in \{+\infty, -\infty\}$. This setting (24) might be too excessive and more mild one would be preferable for the classification; this is our future work. By eliminating such trivial types of prior, we can reduce the computational burden of the whole procedure to optimize (10). As a result, the proposed optimization procedure is shown in Algorithm 1.

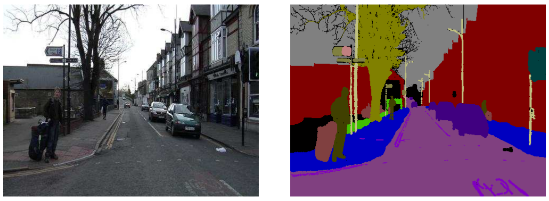
Algorithm 1: Bias Learning.

```
Input: $\{\boldsymbol{x}_i^{[p]}, y_i^{[p]}\}$: feature vector and its class label
    of the $i$-th training sample from the $p$-th type of
    prior, $p = 1,..,P$, $i = 1,..,n^{[p]}$.
    $\varepsilon > 0$: small tolerance for terminating the iteration.
1: $\mathbb{P} = \{p | \exists i, y_i^{[p]} = 1 \wedge \exists i, y_i^{[p]} = -1\}$
2: Initialization: $\forall p \in \{1,..,P\}$, $\boldsymbol{\alpha}^{[p]} = \mathbf{0}$
3: Randomly pick up $p \in \mathbb{P}$
4: repeat
5:     Set $\boldsymbol{\alpha}^{[p]}$ as the optimizer of (11)
6:     Compute $\delta^{[p]}$ in (18), $\forall p \in \mathbb{P}$
7:     $p \leftarrow \arg\max_{p\in\mathbb{P}} \delta^{[p]}$
8: until $\max_{p\in\mathbb{P}} \delta^{[p]} < \varepsilon$
Output: $\boldsymbol{w}$ computed by (7) and $\{b^{[p]}\}_{p=1,..,P}$ computed by (19) for $p \in \mathbb{P}$ and (24) for $p \notin \mathbb{P}$, using
    the optimizers $\{\boldsymbol{\alpha}^{[p]}\}_p$.
```

2.3 Discussion

In the proposed method, all samples across all types of priors are leveraged to train the classifier, improving the generalization performance. In contrast, the `full-connected` method (Table 1) treats the samples separately regarding the priors, and thus the p-th classifier is learnt by using only a small amount of samples belonging to the p-th type of prior, which might degrade the performance. On the other hand, the `simple` method learning the classifier from the whole set of samples is less discriminative without utilizing the prior information associated with the samples. The proposed method effectively introduces the priors into the classifiers via the biases which are discriminatively optimized.

The proposed method is slightly close to the cross-modal learning (Kan et al., 2012; Sharma and Jacobs, 2011). The samples belonging to different priors are separated as if they are in different modalities, though the feature representations are the same in this case. The proposed method deals with them in a unified manner via the adaptive prior biases. Actually, the proposed method is applicable to the samples that are distributed differently across the priors; the sample distribution is shifted (translated) as $\boldsymbol{x}^{[q]} = \boldsymbol{x}^{[p]} + \boldsymbol{e}$ and the prior bias can adapt to it by $b^{[q]} = b^{[p]} - \boldsymbol{w}^\top \boldsymbol{e}$ since $y^{[p]} = \boldsymbol{w}^\top \boldsymbol{x}^{[p]} + b^{[p]}$, $y^{[q]} = \boldsymbol{w}^\top \boldsymbol{x}^{[q]} + b^{[q]} = \boldsymbol{w}^\top \boldsymbol{x}^{[p]} +$



On-board image　Label image

Figure 3: CamVid dataset (Brostow et al., 2008).

$(b^{[q]}+\boldsymbol{w}^{\top}\boldsymbol{e}) = y^{[p]}$. Therefore, the samples of the different priors are effectively transferred into the optimization to improve the classification performance.

3 EXPERIMENTAL RESULTS

We evaluated the proposed method on patch labeling in the on-board camera images by using CamVid dataset (Brostow et al., 2008). This patch labeling contributes to understand the scene surrounding the car.

3.1 Setting

The CamVid dataset (Brostow et al., 2008) contains several sequences composed of *fully* labeled image frames as shown in Fig. 3: each pixel is assigned with one of 32 class labels including 'void'. Those labeled images are captured at 10 Hz. In this experiment, we employ the major 11 labels frequently seen in the image frames, *road*, *building*, *sky*, *tree*, *sidewalk*, *car*, *column pole*, *sign symbol*, *fence*, *pedestrian* and *bicyclist*, to form the 11-class classification task.

We extracted the GLAC image feature (Kobayashi and Otsu, 2008) from a local image patch of 20 × 40 pixels which slides at every 10 pixels over the resized image of 480 × 360. In this case, the feature vector $\boldsymbol{x} \in \Re^{2112}$ is associated with the 2D position of the patch as the extra prior information; the total number of prior types (grid points) is $P = 1551$. Thus, the task is to categorize the patch feature vectors extracted at 1511 positions into the above-mentioned 11 classes.

We used the three sequences in the CamVid dataset, and partitioned each sequence into three subsequences along the time, one of which was used for training and the others were for test. This cross validation was repeated three times and the averaged classification accuracy is reported.

For comparison, we applied the methods mentioned in Sec.1; `simple` and `full-connected` methods as listed in Table 1. The `simple` method is a standard classification using the weight $\boldsymbol{w}$ with the bias b without relying on the prior information p. The `full-connected` method applies classifiers comprising $\boldsymbol{w}^{[p]}$ and $b^{[p]}$ at respective priors $p = 1,..,P$. This method requires tremendous memory storage for those P classifiers; in this experiment, 2112-dimensional weight vectors $\boldsymbol{w}$ in 11 class categories are stored at each of 1511 positions. On the other hand, in the `proposed` method, the feature vectors are classified by using the identical weight $\boldsymbol{w}$ across the priors together with the adaptively optimized bias $b^{[p]}$ depending on the prior p.

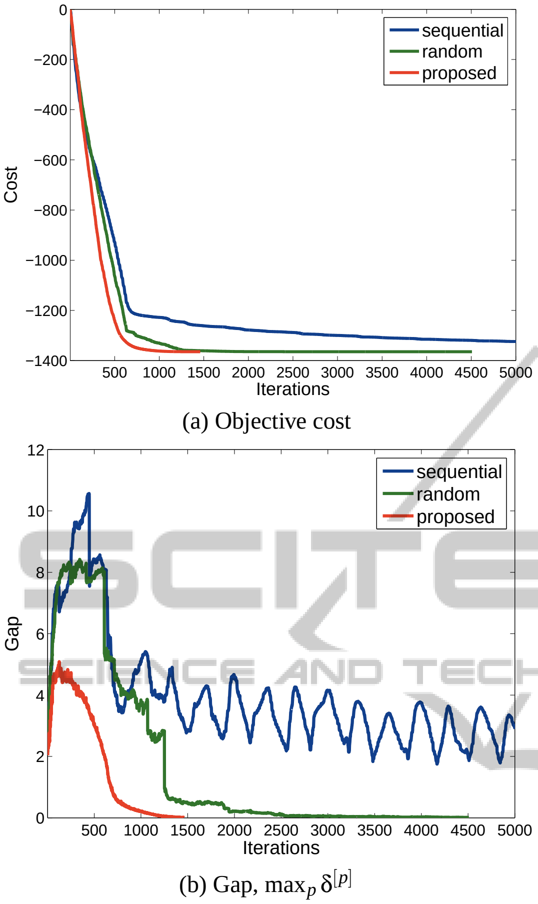
3.2 Computation Cost

We evaluated the proposed method in terms of computation cost.

The first issue is related to the way of selecting p in the iterative optimization; the proposed procedure selects p deterministically by $p^* = \arg\max_p \delta^{[p]}$ (the 7-th line in Algorithm 1). As the alternative for the proposed selection, the other two ways are conceivable, *sequential* and *random* selections. In the sequential selection, the target prior p is simply selected as in raster scan over the image frame. The random selection means that the target p is randomly picked up from the whole set $\{1,..,P\}$. Fig. 4 shows the comparison results with respect to the objective cost in (10) and the gap $\max_p \delta^{[p]}$ in (18) measuring violation of the KKT condition, both of which should be decreased toward convergence. The optimization is fast converged via the proposed method, while in the other methods the optimization takes a larger number of iterations until convergence; in particular, the sequential method requires more than 10,000 iterations. These results reveal the importance of selecting p to be optimized and show that the proposed method quickly decreases the cost as well as the gap, leading to fast convergence.

The second issue is about scalability of the proposed method. The method trains the classifier by using all the samples across the priors, scale of which is as large as in the `simple` method. Fig. 5a shows the computation time with comparison to the `simple` method on various sizes of training samples. These methods are implemented by MATLAB using libsvm (Chang and Lin, 2001) on Xeon 3.33GHz PC[3]. The proposed method is significantly faster than the `simple` method. The time complexity of `simple` method which solves the standard SVM dual has been empirically shown to be $O(n^{2.1})$ (Joachims, 1999). The proposed optimization approach iteratively works

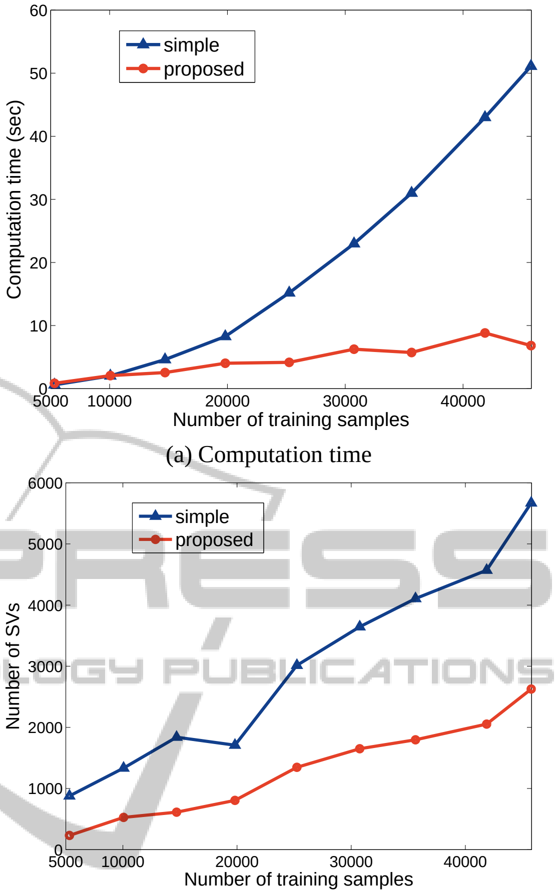
[3] In this experiment, the feature vectors are actually converted into the form of the kernel Gram matrix to which the QP solver in libsvm is directly applied, for fair comparison of the QP problems in the `proposed` and `simple` methods.



(a) Objective cost

(b) Gap, $\max_p \delta^{[p]}$

Figure 4: Comparison for the ways of selecting the target prior p in terms of (a) the objective cost in (10) and (b) the gap $\max_p \delta^{[p]}$ in (18) which measures violation of KKT condition.



(a) Computation time

(b) Number of support vectors (SVs)

Figure 5: Comparison of the `simple` and `proopsed` methods in terms of (a) computation time as well as (b) number of support vectors (SVs).

on the block-wise subset into which the whole training set is decomposed (Sec.2.2). The subset is regarded as the working set whose size is an important factor for fast computing QP (Fan et al., 2005). In the `proposed` method, it is advantageous to inherently define the subset, *i.e.*, the working set, of adequate size according to the prior. Thus, roughly speaking, the time complexity of the `proposed` method results in $O(M\frac{n^{2.1}}{M^{2.1}}) = O(\frac{n^{2.1}}{M^{1.1}})$. In particular, the computation time essentially depends on the (resultant) number of support vectors (SVs); Fig. 5b shows the number of support vectors produced by those two methods. The `proposed` method provides a smaller number of support vectors, which significantly contributes to reduce the computation time. As a result, the proposed optimization approach works quite well together with the working set (prior p) selection discussed in the previous experiment (Fig. 4). These results show the favorable scalability of the `proposed` method, especially compared to the standard `simple` method.

3.3 Classification Performance

We then compared the classification performance of the three methods, `simple`, `full-connected` and `proposed` (Table 1). Table 2 shows the overall performance, demonstrating that the `proposed` method outperforms the others. It should be noted that the `full-connected` method individually applies the classifier specific to the prior $p \in \{1, , P\}$, requiring a plenty of memory storage and consequently taking large classification time due to loading the enormous memory. The `proposed` method renders as fast classification as the `simple` method since it enlarges only the bias. By discriminatively optimizing the biases for respective priors, the performance is significantly improved in comparison to the `simple` method; the improvement is especially found at the categories of *car*, *pedestrian* and *bicyclist* that are composed of patch

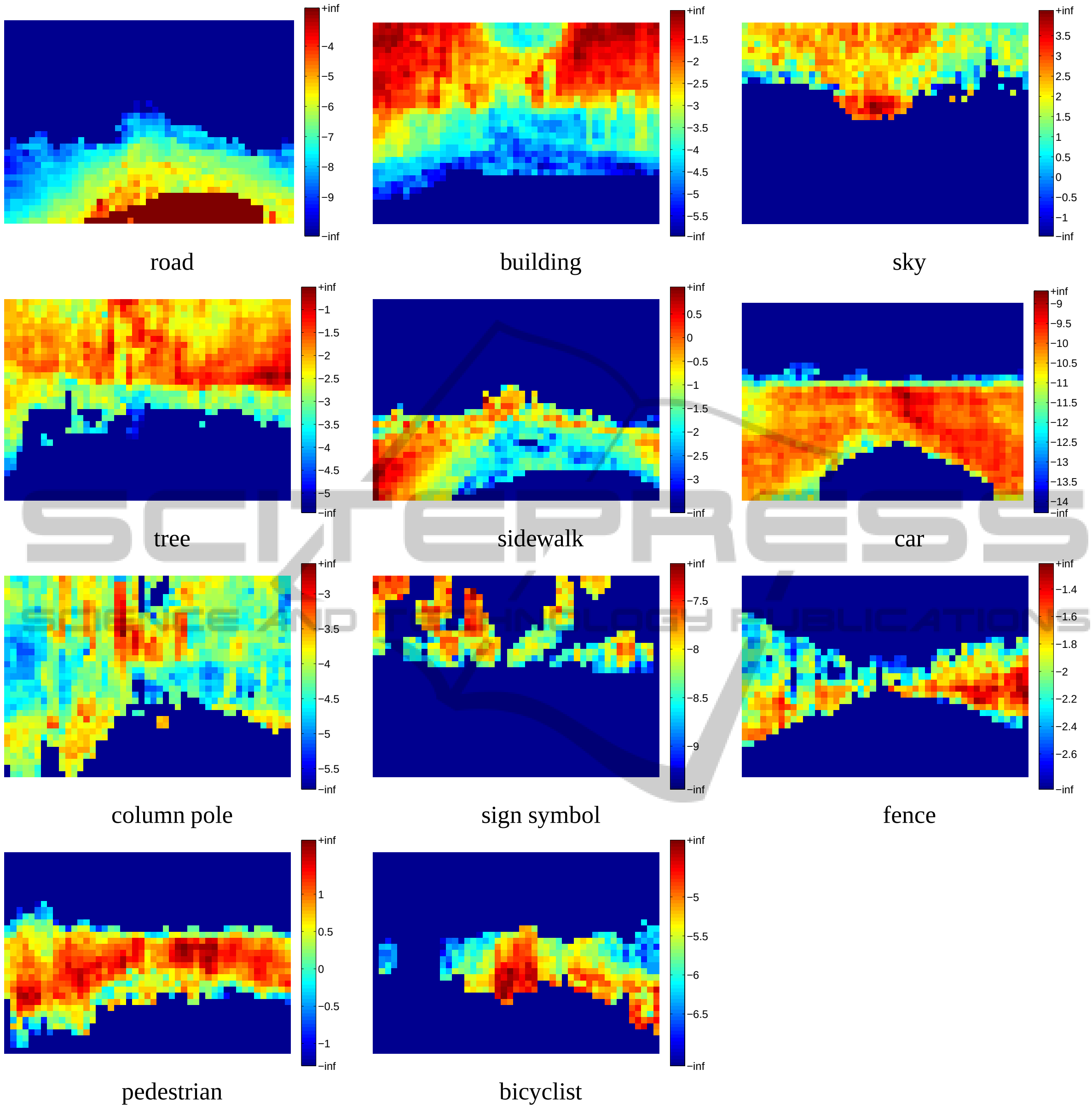


Figure 6: Maps of the biases learnt by the `proposed` method. The significance of the biases are shown by using pseudo colors from (dark) blue to (dark) red. This figure is best viewed in color.

parts similar to other categories but are associated with the distinct prior positions.

Finally, we show in Fig. 6 the biases learnt by the `proposed` method; the biases $\{b^{[p]}\}_p$ are folded into the form of image frame according to the x-y positions. These maps of the biases reflect the *prior* probability over the locations where the target category appears. These seem quite reasonable from the viewpoint of the traffic rules that the car obeys; since the CamVid dataset is collected at the Cambridge city (Brostow et al., 2008), in this case, the traffic rules are of the United Kingdom. The high biases for the *sky* are distributed above the horizontal line, while those of the *road* are high in the lower part. The *pedestrian* probably walks on the *sidewalk* mainly shown in the left side. The oncoming *car* runs on the right-hand road, and the row of the *building* is found on the roadside. These biases are adaptively learnt from the CamVid dataset and they would be different if we use other datasets collected under different traffic rules.

Table 2: Classification accuracy (%).

	simple	full-connected	proposed
road	93.10	93.80	**94.92**
building	75.90	72.96	**78.70**
sky	**90.52**	82.21	90.25
tree	70.49	77.59	**79.95**
sidewalk	77.06	78.43	**81.36**
car	53.84	58.64	**65.16**
column pole	9.53	**16.15**	12.85
sign symbol	**1.73**	1.62	1.70
fence	5.23	11.09	**13.48**
pedestrian	17.26	30.69	**31.52**
bicyclist	17.09	18.49	**24.88**
avg.	46.52	49.24	**52.25**

4 CONCLUSIONS

We have proposed a method to discriminatively learn the prior biases in the classification. In the proposed method, for improving the classification performance, all samples are utilized to train the classifier and the input sample is adequately classified based on the prior information via the learnt biases. The proposed method is formulated in the maximum-margin framework, resulting in the optimization problem of the QP form similarly to SVM. We also presented the computationally efficient approach to optimize the resultant QP along the line of SMO. The experimental results on the patch labeling in the on-board camera images demonstrated that the proposed method is superior in terms of classification accuracy and the computation cost. In particular, the proposed classifier operates as fast as the standard (linear) classifier, and besides the computation time for training the classifier is even faster than the SVM of the same size.

REFERENCES

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, New York, NY.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, Berlin, Germany.

Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV'08, the 10th European Conference on Computer Vision*, pages 44–57.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Cremers, D. and Grady, L. (2006). Statistical priors for efficient combinatorial optimization via graph cuts. In *ECCV'06, the 9th European Conference on Computer Vision*, pages 263–274.

El-Baz, A. and Gimel'farb, G. (2009). Robust image segmentation using learned priors. In *ICCV'09, the 12nd International Conference on Computer Vision*, pages 857–864.

Fan, R.-E., Chen, P.-H., and Lin, C.-J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918.

Gao, T., Stark, M., and Koller, D. (2012). What makes a good detector? - structured priors for learning from few examples. In *ECCV'12, the 12th International Conference on Computer Vision*, pages 354–367.

Ghosh, J. and Ramamoorthi, R. (2003). *Bayesian Nonparametrics*. Springer, Berlin, Germany.

Jiang, T., Jurie, F., and Schmid, C. (2009). Learning shape prior models for object matching. In *CVPR'09, the 22nd IEEE Conference on Computer Vision and Pattern Recognition*, pages 848–855.

Jie, L., Tommasi, T., and Caputo, B. (2011). Multiclass transfer learning from unconstrained priors. In *ICCV'11, the 13th International Conference on Computer Vision*, pages 1863–1870.

Joachims, T. (1999). Making large-scale svm learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Kan, M., Shan, S., Zhang, H., Lao, S., and Chen, X. (2012). Multi-view discriminant analysis. In *ECCV'12, the 12th International Conference on Computer Vision*, pages 808–821.

Kapoor, A., Hua, G., Akbarzadeh, A., and Baker, S. (2009). Which faces to tag: Adding prior constraints into active learning. In *ICCV'09, the 12nd International Conference on Computer Vision*, pages 1058–1065.

Kobayashi, T. and Otsu, N. (2008). Image feature extraction using gradient local auto-correlations. In *ECCV'08, the 10th European Conference on Computer Vision*, pages 346–358.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, USA.

Poggio, T., Mukherjee, S., Rifkin, R., Rakhlin, A., and Verri, A. (2001). b. Technical Report CBCL Paper #198/AI Memo #2001-011, Massachusetts Institute of Technology, Cambridge, MA, USA.

Sharma, A. and Jacobs, D. (2011). Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *CVPR'11, the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600.

Smola, A. J., Bartlett, P., Schölkopf, B., and Schuurmans, D. (2000). *Advances in Large-Margin Classifiers*. MIT Press, Cambridge, MA, USA.

Van Gestel, T., Suykens, J., Lanckriet, G., Lambrechts, A., De Moor, B., and Vandewalle, J. (2002). Bayesian framework for least squares support vector machine classifiers, gaussian processes and kernel fisher discriminant analysis. *Neural Computation*, 15(5):1115–1148.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York, NY, USA.

Wang, C., Liao, X., Carin, L., and Dunson, D. (2010). Classification with incomplete data using dirichlet process priors. *The Journal of Machine Learning Research*, 11:3269–3311.

Yuan, C., Hu, W., Tian, G., Yang, S., and Wang, H. (2013). Multi-task sparse learning with beta process prior for action recognition. In *CVPR'13, the 26th IEEE Conference on Computer Vision and Pattern Recognition*, pages 423–430.