Pattern Recognition Letters 42 (2014) 40–46

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Exploring some practical issues of SVM+: Is really *privileged* information that helps? ☆

Carlos Serra-Toro, V. Javier Traver *, Filiberto Pla

Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castelló de la Plana, Spain
Department of Computer Languages and Systems, Universitat Jaume I, 12071 Castelló de la Plana, Spain

ARTICLE INFO

Article history:
Received 2 August 2013
Available online 31 January 2014

Keywords:
Privileged information
Learning using privileged information (LUPI)
Random features
Support Vector Machine (SVM)
SVM+

ABSTRACT

Learning using privileged information (LUPI) is a machine learning paradigm which aims at improving classification by taking advantage of information that is *only* available at training time —*not* at test time. SVM+ is an SVM-based implementation of LUPI. Despite this paradigm has potential interest for many applications, both LUPI and SVM+ have been scarcely explored up to date. In this work we report our effort in reproducing some results in the SVM+ literature and explore some practical issues of SVM+. The main finding is that just using randomly generated features as privileged information may perform similarly to using sensible (i.e. meaningful a priori) privileged information, at least in some problems.

1. Introduction

Learning using privileged information (LUPI) [1,2] is a recently proposed machine learning paradigm that draws inspiration from human teaching–learning. The paradigm builds on the observation that good human teachers, besides examples, provide students with other relevant information. This information is not available to the students when they face novel, real-world situations, but assists them in building better models during their training. LUPI aims at mimicking this behaviour in the computational world by extending the traditional setting of supervised learning.

Under the conventional supervised machine learning framework, a set of m examples $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ and their class labels $y_i \in \mathcal{Y} \subset \mathbb{Z}$, $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, are provided at training stage from which a model is built for predicting the class label $y \in \mathcal{Y}$ for a new input $\mathbf{x} \in \mathcal{X}$. LUPI extends this paradigm by considering that additional privileged information $\mathbf{z}_i \in \mathcal{Z} \subset \mathbb{R}^{n_\mathcal{Z}}$ will be available for each training example i. That is, the training set will be $(\mathbf{X}, \mathbf{Z}, \mathbf{y}) = \{(\mathbf{x}_i, \mathbf{z}_i, y_i)\}_{i=1}^m$. The goal is to learn classification schemes $h : \mathcal{X} \to \mathcal{Y}$ that utilise all the available information (i.e. both privileged $\mathcal{Z}$ and regular $\mathcal{X}$) during the training stage, and can perform classification during the test stage using only the regular data $\mathcal{X}$.

Very little work has been performed within the LUPI paradigm. After its formulation for Support Vector Machines (SVM) [1,2], namely SVM+, LUPI has been considered in the setting of unsupervised learning [3], and some benefits have been reported in its application in the financial field [4]. The relation between SVM+ and multi-task learning has been studied [5]. Better optimisation methods for SVM+ have been explored [6,7], as well as theoretical analysis about the conditions for faster learning rates in privileged empirical risk minimisation (ERM) with respect to regular ERM [8]. However, more research is required both in the theoretical and practical sides of LUPI for a better understanding of its nature as well as its possibilities and limitations. This work intends to take a step forward in this direction, by focusing on the practical side of SVM+.

Conceptually, the main idea behind LUPI is that leveraging the *privileged* information can boost the performance earlier (i.e. with fewer training instances). Examples of this privileged information are, e.g., [2]: 3D structures of proteins, which is an advanced technical information which is hard and time consuming to obtain; future information which can be available at training, but obviously not at test time; and human-derived poetic description of digits. In all these cases, the privileged information comes only as the result of costly human or computational efforts, and one cannot (easily) afford to have this privileged information for all but a few training instances, or it is completely impossible to have it for actual test examples (e.g. future information). However, during the course of our study of the LUPI paradigm, the following question arose: what if random features are used instead of sensible (i.e. meaningful a

☆ This paper has been recommended for acceptance by A. Marcelli.

* Corresponding author at: Department of Computer Languages and Systems, Universitat Jaume I, 12071 Castelló de la Plana, Spain. Fax: +34 964 728435.
E-mail addresses: cserra@uji.es (C. Serra-Toro), vtraver@uji.es (V.J. Traver), pla@uji.es (F. Pla).

http://dx.doi.org/10.1016/j.patrec.2014.01.013
0167-8655/

priori) privileged information? This is the main issue discussed in this work. Other practical aspects are explored such as the relative performances of SVM and SVM+ with different trade-offs of the sizes of the validation and training sets.

2. SVM+ formulation

Briefly, for regular SVM, the optimisation problem in its dual form is defined as

$$\max_{\alpha}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j),\quad \text{s.t.}\sum_{i=1}^{m}y_i\alpha_i=0,\quad 0\leqslant\alpha_i\leqslant C, \tag{1}$$

where α_i are Lagrange multipliers, $K(\cdot,\cdot)$ is the kernel function, and C is the regularisation parameter. For SVM+, the problem is formulated as [2]:

$$\max_{\alpha,\beta}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)-\frac{1}{2\eta}\sum_{i,j=1}^{m}(\alpha_i+\beta_i-C)(\alpha_j+\beta_j-C)K_z(\mathbf{z}_i,\mathbf{z}_j), \tag{2}$$

subject to $\sum_{i=1}^{m}(\alpha_i+\beta_i-C)=0, \sum_{i=1}^{m}y_i\alpha_i=0$, and $\alpha_i,\beta_i\geqslant 0$. In this case, $K_z(\cdot,\cdot)$ is the kernel in the $\mathcal{Z}$ space, β_i are Lagrange multipliers, and η is an additional regularisation parameter. In both cases, the decision function $f(\mathbf{x})$ takes place in the $\mathcal{X}$ space, $f(\mathbf{x})=\sum_{i=1}^{m}y_i\alpha_i K(\mathbf{x}_i,\mathbf{x})$, and a correcting function is also required in the SVM+. Two kernels, $K(\cdot,\cdot)$ and $K_z(\cdot,\cdot)$, are used in SVM+, each measuring the similarity in different spaces. Although only $K(\cdot,\cdot)$ takes part in the decision function, both kernels are coupled through the α's, since these coefficients are in all the terms in (2) and in the decision function as well. Since this formulation includes the SVM solution as a particular case, SVM+ can either use the privileged information when found helpful through $K_z(\cdot,\cdot)$, or resort to the SVM solution otherwise [2].

3. Experimental work

We first indicate in Section 3.1 the experimental methodology common to all the experiments performed. The following sections report the details and results with SVM and SVM+ for three problems related to computer vision: two of them (Sections 3.2 and 3.3) are taken from Vapnik et al.'s work [2,7] and another one (Section 3.4) is proposed here. A final toy synthetic example (Section 3.5) is used for subsequent discussion (Section 4).

3.1. Experimental methodology

Features

Experiments were performed with two types of privileged information: the proposed *genuine* privileged information on the one hand, and synthetically generated random features on the other. We tested with both, class separable, and non-separable random features. In the class-separable case, features were uniformly generated by taking the feature values in disjoint ranges for each class, i.e. $\mathbf{z}_{ij}\in[y_i-\delta,y_i+\delta]$, with $\mathbf{z}_{ij}$ denoting the j-th feature of vector $\mathbf{z}_i$, and δ was chosen to allow class separability per feature (in all the experiments below, we set $\delta=0.4$). The number of random features used was the same as in the genuine case. In the non-separable case, the features were uniformly taken from a normalised range, i.e. $\mathbf{z}_i\in[0,1]$. The three SVM+ versions using these three different sources of privileged information are referred to as SVM+ (with genuine privileged information), SVM+R$^{\text{sep}}$ (with random but separable features) and SVM+R$^{\text{non-sep}}$ (with fully random, non separable features). We may use SVM+ to refer to any of these versions and SVM+R to any of the two versions with random features.

Classifiers and evaluation protocol

We used the efficient implementation of SVM provided by the LIBSVM 3.16 [9], and an SVM+ implementation built on LIBSVM. Two models, X*SVM+ and dSVM+, have been proposed [2]. X*SVM+ applies SVM+ directly on the available privileged information and regular data, whereas dSVM+consists of two stages: first, SVM is applied on the regular information alone; second, SVM+ is applied using as privileged information the deviation values of the SVM trained in the first stage. Since the interest of our study is exploring the role played by random features as opposed to genuine privileged information, any of the two models can be used. We chose X*SVM+ since it is somehow simpler and easier to use.

Following Vapnik and Vashist [2], we chose to use a radial-basis function (RBF) as the kernel function for both the regular and the privileged information spaces, defined as $K(\mathbf{x},\mathbf{x}')=\exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)$. This leads to four parameters to be tuned when executing the code to train the learner: the penalty parameters C and C_z, and γ and γ_z for the RBF kernels in the $\mathcal{X}$ and $\mathcal{Z}$ spaces, respectively. Notice that only C and γ are required for SVM whereas all the four are required for SVM+. Although C_z does not appear in the general formulation (2), it is a required parameter for the optimisation functions used by the particular SVM+ implementation used. To choose their optimal values, a coarse-to-fine grid search was performed over a validation set. First, a coarse grid search was made over the series $C,C_z\in\left\{2^i:i\in\{-5,-3,\ldots,11\}\right\}$, and $\gamma,\gamma_z\in\left\{2^j:j\in\{-15,-13,\ldots,3\}\right\}$. Once the optimal values for i and j were found, $\hat{i},\hat{j}$, a fine search focused on the range $\{\hat{i}-1.0,\hat{i}-0.8,\ldots,\hat{i}+1.0\}$ for $\hat{i}$ and equivalently for $\hat{j}$.

3.2. Hand-written digits classification

Description

The popular MNIST dataset [10] consists of grey-scaled, scanned images of hand-written digits (from 0 to 9). Although each digit in this dataset is 28×28 pixels, Vapnik and Vashist [2] resized them to 10×10 pixels to make the problem harder. They defined a binary classification problem by considering only the digits 5 and 8, $\mathcal{Y}=\{5,8\}$, and considered three disjoint sets: training, validation, and test, with 100, 4002, and 1866 examples each, respectively. As privileged information, a 28-dimensional feature vector obtained from a poetic description [1,2] of each of the 100 training digits was used. We used the data files as available at [11]. The training and validation sets provided were balanced while the test set was slightly biased towards digit 8 with a proportion 0.52 : 0.48.

Experimentation

Since no indication is provided in [2] about how data was processed, several approaches were tested: performing no processing at all, scaling the data so that the features were in a given range, normalising the features individually to zero mean and unit variance, and normalising each feature vector so that each one had unit length. The results with some of these approaches were far from those reported in [2], while the latter was the option which was closer and yielded the best ones, and it is thus the one used to report our results. Notice that this normalisation does not assure complete separability of the originally random separable data. Following [2], twelve repetitions were made over each training size, each repetition using a different random sample from the provided training set, and we made sure each sample was class-balanced.

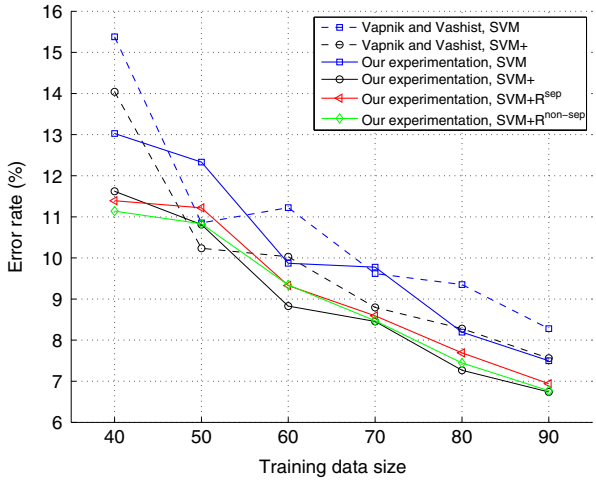


Fig. 1. Error rates of SVM, SVM+, and SVM+R for the problem of hand-written digits recognition (Section 3.2).

Results

Fig. 1 shows the averaged error rates obtained when reproducing the experiments (solid lines), as well as the results reported by Vapnik and Vashist [2] (dashed lines), traced from Fig. 5(a) in their paper. It can be observed that SVM is outperformed by every SVM+ version, even by the ones using random features as privileged information. The statistical significance of the differences in error rates is assessed with paired t-tests (right-tailed to check for the error of the compared algorithm being lower than the error for the baseline algorithm). Results, given in Table 1, confirm this observation and also reveal that, except for one training size ($m = 60$), SVM+ does not perform significantly better than SVM+R.

We wondered whether the large amount of examples used for validation was a requirement for SVM+ to perform well, and thus we repeated the experiments with a validation set a quarter this size (i.e. $m_v = 1002$ instead of $m_v = 4002$). The error rate increased in an absolute difference of about 0.5 percentage units for both SVM and SVM+, but SVM+ still outperformed SVM. This suggests that smaller validation sets are possible to keep enjoying the benefits of SVM+.

After this observation, a natural and interesting question was whether part of the available validation set could be used for training SVM and whether this could outperform SVM+. Notice that increasing the training set for SVM+ is not always so easy: in this problem, for instance, additional human expert effort would be required to produce the poetic description of every new digit. In order to explore a variety of trade-offs between the sizes of the training and validation sets, we tested with an increasing number of training instances so that $m = 90 + \{10, 20, \ldots, 100\}$, and a varying size m_v of the validation set, with $m_v \in \{50, 100, 200, \ldots, 900, 1000, 1500, 2000\}$. The extra examples used for training were taken from the first half of the original validation set, while the second half was retained to extract the validation subsets used. Results, averaged over 12 repetitions, are shown in Fig. 2 (for the sake of clarity, only results for a subset of the sizes of the validation sets are shown). As expected, performance improves with increasing size of the training and the validation sets. As a relevant example (highlighted in Fig. 2), just adding 50 training examples ($m = 90 + 50 = 140$) suffices for SVM to outperform SVM+ with a validation set as small as 400 examples. It is worth stressing that we are using as a baseline/reference the SVM+ trained with 90 examples and tuned with the original *full* validation set with 4002 instances. If the smaller validation sets used for SVM are also used for SVM+, its performance degrades (e.g. see SVM+ with $m_v = 1002$ in Fig. 2) and and therefore SVM outperforms SVM+ even earlier (if using a smaller training set) or faster (if using a smaller validation set). Furthermore, even if random features can be obtained for free and used as "privileged" information, the significantly higher cost of training and validating with SVM+ (e.g. up to four parameters have to be tuned, as discussed in Section 3.1) may not compensate with respect to the computationally lighter SVM.

3.3. Visual object classification

To further assess the effect of random features in SVM+, we also tried to reproduce the results reported in [7] over a dataset generated from the ESP on-line game [12].

Table 1
Significance results of paired t-tests for different SVM and SVM+ comparisons for different training sizes m. The p-values have been rounded to three decimal places.

Problem (Section)	m (or $\overline{m}$)	SVM+ *vs.* SVM	SVM+R^{sep} *vs.* SVM	SVM+R$^{non-sep}$ *vs.* SVM	SVM+ *vs.* SVM+R$^{non-sep}$
Digits (3.2)	40	0.000***	0.000***	0.000***	0.934
	50	0.002***	0.023**	0.000***	0.472
	60	0.005***	0.048**	0.043**	0.005***
	70	0.001***	0.002***	0.001***	0.473
	80	0.001***	0.036**	0.003***	0.114
	90	0.000***	0.005***	0.002***	0.398
Objects (3.3)	50	0.067*	0.029**	0.182	0.500
	100	0.077*	0.517	0.812	0.010***
	150	0.791	0.945	0.656	0.766
	200	0.999	0.989	0.960	0.966
Actions (3.4)	26.8	0.068*	0.154	0.164	0.368
	54.2	0.130	0.065*	0.188	0.408
	163.8	0.062*	0.093*	0.190	0.231
	274.0	0.870	0.594	0.389	0.914
	383.6	0.620	0.793	0.767	0.361
	493.2	0.751	0.606	0.263	0.924
Bananas (3.5)	50	0.000***	0.000***	0.000***	0.000***
	70	0.000***	0.000***	0.000***	0.085*
	90	0.002***	0.017**	0.088*	0.085*
	100	0.000***	0.012**	0.000***	0.070*
	150	0.000***	0.074*	0.003***	0.051*
	200	0.302	0.511	0.053*	0.799

* p-value < 0.1.
** p-value < 0.05.
*** p-value < 0.01.

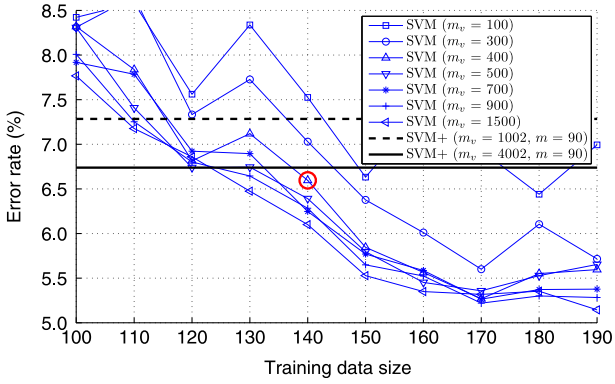


Fig. 2. Performances of SVM and SVM+ for the digits recognition problem (Section 3.2). SVM is tested with varying sizes of the training and validation sets, whereas SVM+ (horizontal lines) are taken as baseline.

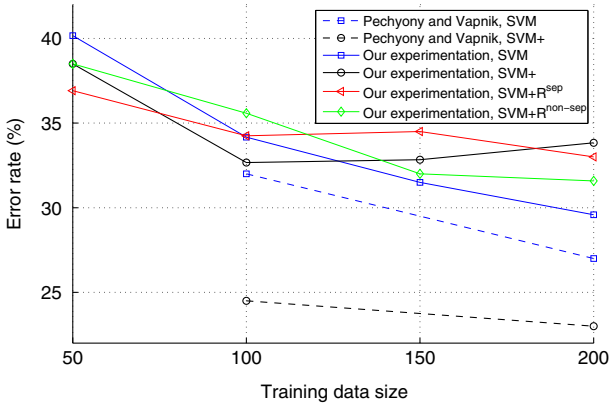


Fig. 3. Error rates of SVM, SVM+, and SVM+R for the problem of tagged images classification using the ESP2 experiment (Section 3.3).

Description

The ESP dataset consists of real-world images with descriptive textual tags manually associated to them through the aforementioned game. It was intended for its use in an auto-annotation problem [13], but in [7] it is used as a binary classification task. Images in the dataset are tagged with a subset of a tag set $\mathcal{T}$. The classification goal is to discriminate images tagged with $t_1 \in \mathcal{T}$ from those tagged with $t_2 \in \mathcal{T}, \mathcal{Y} = \{0, 1\}$. Thus, tags are only used to create the classes and as privileged information for LUPI. We used the features and tags used in [13], available at [14]. As $\mathcal{X}$ features [7], the concatenation of DenseHue, DenseSift, HarrisHue, and HarrisSift, extracted from the raw images, were used, resulting in $n = 2200$ features. The privileged information associated to each training image was a binary vector in which each feature indicated whether a certain tag was used to describe the image, i.e. $\mathbf{z}_i \in \{0, 1\}^{n_\mathcal{Z}}$, with $n_\mathcal{Z} = |\mathcal{T} \setminus \{t_1, t_2\}| = 266$.

Experimentation

Three experiments using ESP are reported in [7]: ESP1 (t_1 ="fish", t_2 ="horse"), ESP2 (t_1 ="bird", t_2 ="horse"), and ESP3 (t_1 ="bird", t_2 ="fish"). A test set of 100 examples is reported in [7], but no explicit split for the training, validation and test sets are provided. Thus, we created a test set of 100 examples and a validation set of 250 examples for each subproblem (ESP1, ESP2, and ESP3), all of them class-balanced. A balanced training set of 250 examples was created, from which twelve balanced subsets were randomly sampled for sizes $m \in \{50, 100, 150, 200\}$. Although training sets of sizes $\{100, 200, 300, 400\}$ are reported in [7], we found impossible to create an appropriate balanced validation set when using training sets that large (e.g. for ESP2, when $m = 400$ then $m_v = 52$ only). Regarding data preprocessing, we tried several schemes, as we did in the hand-written recognition problem (Section 3.2), and, for the ESP problem, the use of the raw features without any normalisation performed the best.

Results

For the sake of brevity, we consider only the ESP2 experiment; SVM+ performed somehow better in ESP3, and worse in ESP1. Fig. 3 shows the averaged error rates obtained when reproducing the experiments (solid lines), and those reported in [7] (dashed lines). Despite our efforts, we could not reproduce the error rates reported in [7], neither for SVM nor for SVM+. However, some statistical difference in performance between SVM+ or SVM+R and SVM are found in a few cases when using a low number of training examples (Table 1).

3.4. Action recognition

We also explored SVM+ in the context of human action recognition. The main purpose here was to test the role of random

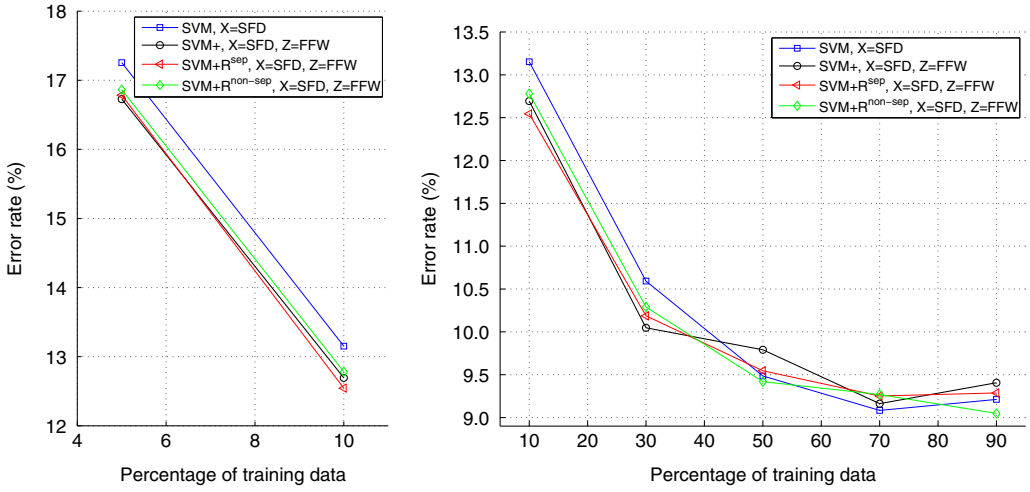


Fig. 4. Error rates of SVM, SVM+, and SVM+R for the problem of human action recognition using a subset of the Weizmann dataset (Section 3.4) and and features $\mathcal{X}$ = SFD, $\mathcal{Z}$ = FFW. The plot has been split into two with different scales for better visualisation.

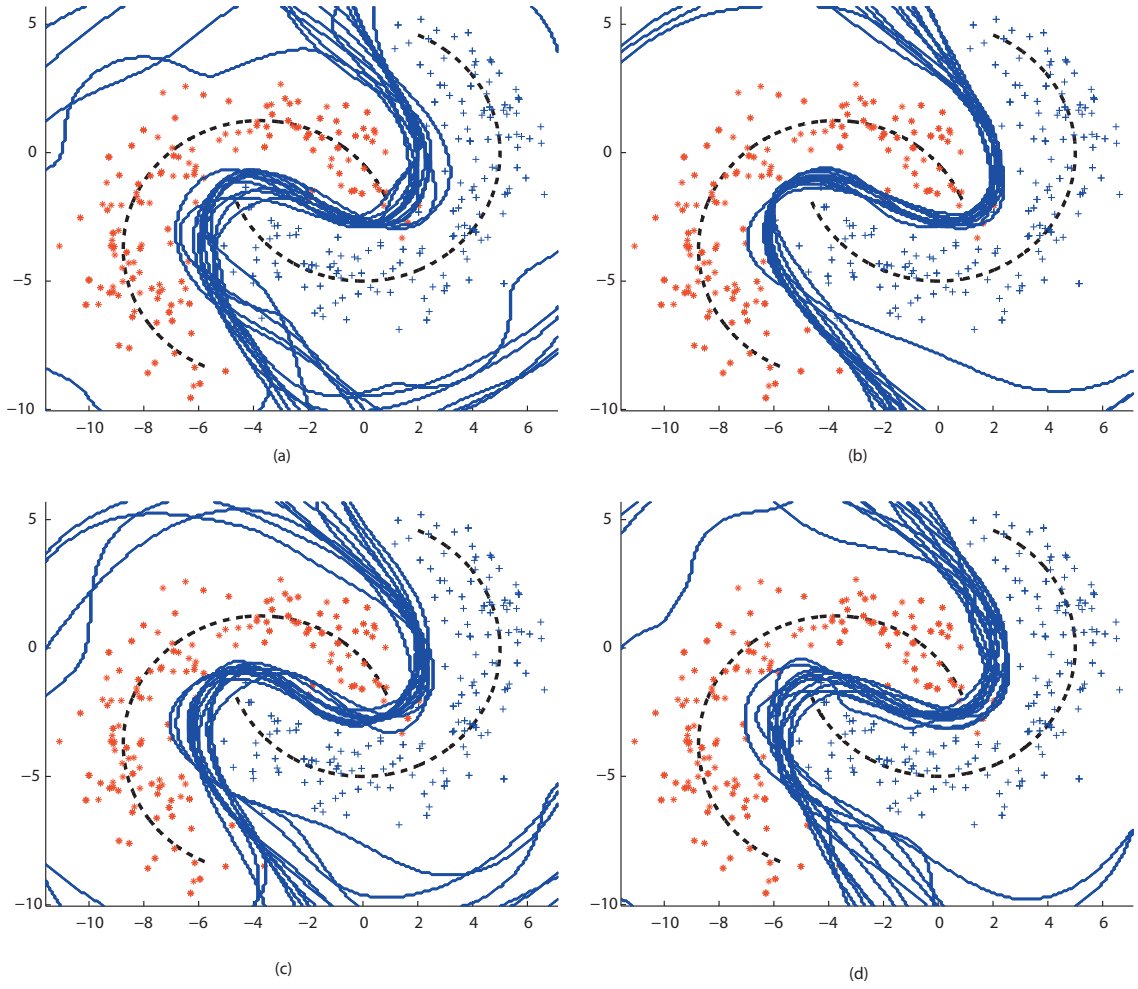


Fig. 5. Decision boundaries (in blue) found by (a) SVM, (b) SVM+, (c) SVM+R^{sep}, and (d) SVM with noise injection (with $k = 50$) on the synthetic bananas problem ($\sigma = 1.0, m = 50$ and $m_v = 1500$). Banana's generative spines, as computed by PRTools [18], are the dashed curves. Each decision boundary is obtained with just 25 examples per class, except in (d) where additional noised examples were added. The instances drawn correspond to all the examples used for the 12 repetitions. The noised examples generated for (d) are not shown. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

features, and how easy/difficult may be to define successful genuine (non-random) privileged information.

Description

We used the frame descriptor proposed in [15] since it includes different visual cues (shape and motion) as well as temporal summaries (past and future information). The richness of this descriptor lends itself to be used in the context of LUPI so that one feature subset can be regarded as regular data and other subset as privileged information.

We used the well-known Weizmann dataset [16], which consists of videos of 9 different subjects performing 10 different common actions each (e.g. running, walking, jumping, etc.). As a proof of concept, we chose to discriminate only between two "in place" actions (i.e. subjects do not translate horizontally): *jack* (jumping while waving both the arms and the legs) and *pjump* (jumping while keeping the arms and the legs vertical and close to the body and without moving them). Here, $\mathcal{Y} = \{4, 6\}$. We used the features already computed and available at [17] for the Weizmann and other datasets.

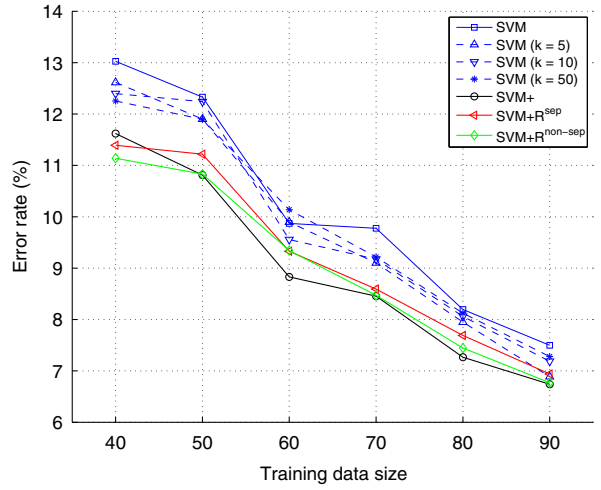
Experimentation

To generate the validation, training, and test sets for this dataset, we randomly assigned the subjects to disjoint sets of training (4 subjects), validation (2 subjects), and test (3 subjects). As in the previous experiments, the training set was used to sample a number of subsets with an increasing size of 5%, 10%, 30%, ..., 90% of the total number of training examples. For each size, 9 different subsets were randomly created. We repeated this procedure 5 times, yielding different random assignments of the subjects to each set. The validation and test sets were fixed for each repetition, having an averaged number of examples per assignment of $\overline{m_v} = 318.2$ and $\overline{m_t} = 381.4$ each, respectively. The sampling retained the original proportion of frames of each action. Action classification was performed on a per-frame basis.

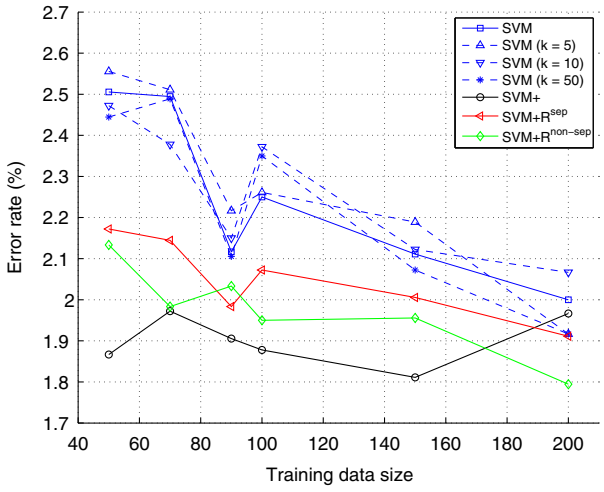
The regular features chosen were the single-frame descriptor (SFD), i.e. the concatenation of shape and optical flow (216 features altogether). As privileged information, we used the future-frame window (FFW) corresponding to the 10 principal components of the SFD of the 5 frames *after* the current 5-frame time window. We found that no normalisation was required for these features for SVM to perform well, hence no normalisation was either applied when using SVM+.

Results

Results (Fig. 4, Table 1) indicate that SVM is outperformed by SVM+ and SVM+R^{sep} only when using some of the smallest sizes for training. Therefore, although the use of future events could be



(a) Hand-written digits recognition problem



(b) Synthetic bananas problem

Fig. 6. Comparing SVM, SVM+, SVM+R, and noise injection on SVM for different number k of instances generated for each original training instance, on the problems of (a) the hand-written digits recognition and (b) the synthetic bananas ($\sigma = 1.0, m = 50$ and $m_v = 1500$, as in Fig. 5).

expected to be high quality privileged information, SVM+ seems not to generally benefit from this information in this case, and the use of random features provide similar or better performance. This experiment suggest that finding information to be successfully used as *privileged* is not straightforward; therefore, some uncommon and currently not well-defined desirable properties in the data are required.

3.5. Synthetic dataset

We tested with the "bananas" dataset generated using the PRTools [18] toolbox. Two pairs of bananas were generated, each pair with $\sigma = 1.0$ and $\sigma = 1.5$ as measures of class overlap (problem difficulty). From the data generated, 1500 examples were taken as a pool test set, 400 as training set, and 2000 as validation set in each case, all of them balanced. Tests were performed with $m \in \{50, 70, 90, 100, 150, 200\}$, and $m_v \in \{1000, 1500\}$, with no data preprocessing. In this case, $\mathcal{Y} = \{1, 2\}$. For each training size, 12 repetitions were performed.

Results

As privileged information for SVM+, our first and only choice turned out to work well: we used a 2-dimensional vector with the closest Euclidean distances from a given instance to each of the bananas' generative spines. As an illustration of the results (to be discussed in Section 4), the decision boundaries for the case in which $\sigma = 1$ and $m_v = 1500$ are given in Fig. 5, and the statistical differences are reported in Table 1.

4. Remarks

Improving generalisation

Previous results suggest that SVM+ not only is able to leverage the privileged information, but the proposed optimisation ends up "pushing" the usual capabilities of SVM a bit further and get a higher generalisation ability *even* when the "privileged" information is not necessarily such.

This effect on generalisation ability can be illustrated in a controlled setting such as the well-known 2-banana synthetic problem (Section 3.5). The decision boundaries found (Fig. 5(a)–(c)) by SVM+ are somehow better defined and more in the middle of the bananas' spines than in SVM case, suggesting a better generalisation ability of SVM+ even with random features as privileged information (SVM+R).

This phenomenon might be related to noise injection procedures which are known to be able to reduce overfitting in neural networks [19,20]. These techniques differ in many respects to LUPI and SVM+, and are therefore out of the scope of this paper. However, we still briefly explored this issue with a simple noise injection approach applied to training data with the conventional SVM, which was tested on the digits (Section 3.2) and bananas (Section 3.5) problems. To this end, noise was injected by generating k new instances for each original training instance and perturbing these new instances on a per-feature basis, with Gaussian noise with standard deviation $\sigma_i = 0.05 \cdot r_i$, with r_i being the range of values of the i-th feature in the training set. Results (Figs. 5(d) and 6) reveal that noise injection may improve the SVM performance, but lag behind SVM+, either with privileged information or random features. This suggests that noise injection with SVM has a similar regularisation effect, but is less powerful than SVM+ can be.

One possible direction for gaining insight into the role of random features within LUPI or SVM+ is by modelling the regular features, the privileged information, and the class labels as random variables within the information theory (IT) [21]. Since IT is classifier-independent, it may provide a nice framework for such a study; indeed, it has previously been used in the context of formally analysing decision and pattern recognition problems (e.g. [22–24]).

Work relevance

The findings of our study can be of relevance to practitioners and researchers in machine learning under any of the following profiles: (1) engineers wanting to apply SVM+ on particular real-world problems; (2) researchers interested in exploring the LUPI paradigm on classifier models other than SVM-based; and (3) theoreticians seeking formal and rigorous justification of these phenomena, and their extent and limitation. We believe all of these people can benefit from being aware of the results reported in this work.

5. Conclusions

LUPI and SVM+ are theoretically attractive and potentially useful in many problems; however, we have identified some issues

that might affect its applicability in practise. From our experiments, the following conclusions can be drawn:

- The performance of SVM+ seems to rely on a delicate relationship between the regular data and the privileged information. Additionally, it has been shown that just randomly generated features may play a key role as privileged information, at least in some problems. Thus, considering random features as privileged information seems to be not only a reasonable first choice, but also a baseline for genuine privileged information to be compared with.
- If the size of the validation set is traded off for a bigger training set, SVM is likely to be advantageous over SVM+ in terms of both computational and classification performances. This could be important in, for instance, problems where the privileged information is costly or difficult to obtain with respect to producing additional regular training examples.
- Whether SVM+ outperforms SVM may critically depend on experimental details such as data preprocessing, dataset splitting, validation protocol, parameter ranges and search procedure, etc.
- Some useful and clear design guidelines not existing yet would be much required, in particular regarding when and how can one envisage useful privileged information for a given problem.

Acknowledgements

The authors acknowledge Fundació Caixa-Castelló Bancaixa under project with code P1·1A2010–11, and Dr. Pechyony for having provided them with SVM+ code. Carlos Serra-Toro is funded by Generalitat Valenciana under the "VALi + d program for research personnel in training" with Grant code ACIF/2010/135.

References

[1] V. Vapnik, A. Vashist, N. Pavlovitch, Learning using hidden information: master-class learning, in: Proceedings of NATO Workshop on Mining Massive Data Sets for Security, IOS Press, 2008, pp. 3–14.
[2] V. Vapnik, A. Vashist, A new learning paradigm: learning using privileged information, Neural Networks 22 (5–6) (2009) 544–557.
[3] J. Feyereisl, U. Aickelin, Privileged information for data clustering, Inf. Sci. 194 (2012) 4–23.
[4] B. Ribeiro, C. Silva, A. Vieira, A. Gaspar-Cunha, J. das Neves, Financial distress model prediction using SVM+, in: International Joint Conference on Neural Networks, 2010, pp. 1–7.
[5] L. Liang, V. Cherkassky, Connection between SVM+ and multi-task learning, in: International Joint Conference on Neural Networks, 2008, pp. 2048–2054.
[6] D. Pechyony, R. Izmailov, A. Vashist, V. Vapnik, SMO-style algorithms for learning using privileged information, in: Proceedings of the 2010 International Conference on Data Mining, DMIN 2010, Las Vegas, Nevada, USA, 2010, pp. 235–241.
[7] D. Pechyony, V. Vapnik, Fast optimization algorithms for solving SVM+, in: Statistical Learning and Data Science, Chapman and Hall/CRC, 2011.
[8] D. Pechyony, V. Vapnik, On the theory of learning with privileged information, in: NIPS, 2010, pp. 1894–1902.
[9] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27. <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (accessed 19.07.13).
[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[11] Data for digits recognition with SVM+. http://ml.nec-labs.com/download/data/svm+/mnist.priviledged, (accessed 19.07.13).
[12] gwap, ESP game. <http://www.gwap.com/gwap/gamesPreview/espgame/>, (accessed 19.07.13).
[13] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: discriminative metric learning in nearest neighbor models for image auto-annotation, in: ICCV'09, 2009.
[14] M. Guillaumin, Features and tags for ESP game. <http://lear.inrialpes.fr/people/guillaumin/data.php>, (accessed 17.04.13).
[15] D. Tran, A. Sorokin, Human activity recognition with metric learning, in: European Conference on Computer Vision (ECCV), 2008, pp. 548–561.
[16] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 29 (12) (2007) 2247–2253. <http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html> (accessed 19.07.13).
[17] D. Tran, A. Sorokin, D. Forsyth, Human activity recognition with metric learning. <http://vision.cs.uiuc.edu/projects/activity>, (accessed 17.04.13).
[18] 37Steps, PRTools: pattern recognition tools. <http://www.37steps.com/software/>, (accessed 19.07.13).
[19] K. Matsuoka, Noise injection into inputs in back-propagation learning, IEEE Trans. Syst. Man Cybern. 22 (3) (1992) 436–440.
[20] I.B.V. da Silva, P.J.L. Adeodato, PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets, in: International Joint Conference on Neural Networks, 2011, pp. 2664–2669.
[21] R.W. Yeung, A First Course in Information Theory, Kluwer/Plenum, Norwell, MA/New York, 2002.
[22] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151.
[23] G. Brown, An information theoretic perspective on multiple classifier systems, in: Proceedings of the 8th International Workshop on Multiple Classifier Systems, Springer-Verlag, 2009, pp. 344–353.
[24] D. Pascual, F. Pla, J.S. Sánchez, Cluster validation using information stability measures, Pattern Recognit. Lett 31 (6) (2010) 454–461.