A Novel Approach for Efficient SVM Classification with Histogram Intersection Kernel

Gaurav Sharma, Frédéric Jurie

A Novel Approach for Efficient SVM Classification with Histogram Intersection Kernel

Gaurav Sharma[1]
gaurav.sharma@unicaen.fr
Frederic Jurie
frederic.jurie@unicaen.fr

GREYC CNRS UMR6072
University of Caen
Boulevard Marechal Juin
14032 Caen, France

Abstract

The kernel trick – commonly used in machine learning and computer vision – enables learning of non-linear decision functions without having to explicitly map the original data to a high dimensional space. However, at test time, it requires evaluating the kernel with each one of the support vectors, which is time consuming. In this paper, we propose a novel approach for learning non-linear SVM corresponding to the histogram intersection kernel without using the kernel trick. We formulate the exact non-linear problem in the original space and show how to perform classification directly in this space. The learnt classifier incorporates non-linearity while maintaining O(d) testing complexity (for d-dimensional input space), compared to O($d \times N_{sv}$) when using the kernel trick. We show that the SVM problem with histogram intersection kernel is quasi-convex in input space and outline an iterative algorithm to solve it. The proposed approach has been validated in experiments where it is compared with other linear SVM-based methods, showing that the proposed method achieves similar or better performance at lower computational and memory costs.

1 Introduction and related work

The two main ingredients of most of the successful approaches for visual recognition are (i) the representation of images by distributions (*i.e*. histograms) of visual features *e.g*. bag-of-features [1] and HOG [2] and (ii) the use of margin maximizing classifiers such as support vector machines (SVM) [15]. Systems built on them have led to state-of-the-art performance on image classification [8, 9, 16] and object detection [5, 7, 21].

The standard formulation for learning classifiers is the SVM primal formulation (Eq. 1, see [15] for more details) which allows the learning of a linear classification boundary in the space of (images represented as) distributions. However, general visual tasks *e.g*. scene or object based classification of unconstrained images, are very challenging due to the presence of high variability due to viewpoint, lighting, pose *etc*. and linear decision boundaries are not sufficient. Many competitive methods in image classification [3, 9] and object detection [7, 21], thus, use non linear classifiers. Such non linear classifier are obtained by using the

[1]GS is currently with Technicolor

kernel trick with the dual formulation (Eq. 2) of the SVM. The SVM dual formulation only requires the dot products between the vectors and so a nonlinear *kernel* function $K(\mathbf{x}_1, \mathbf{x}_2)$ is used which implicitly defines a (non linear) mapping $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$ of input vectors to a high (potentially infinite) dimensional *feature* space with $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle$. With the kernel trick a linear decision boundary in the feature space is learned which corresponds to a non linear decision boundary in the input space. Such kernel based SVMs have been shown to improve the performance of linear SVMs in many visual tasks (*e.g.* classification and detection) by a significant margin *e.g.* [7, 9, 21].

While the dual formulation allows the learning of non linear decision boundaries, the computation of classifier decision for a test vector $\mathbf{x}$, $f(\mathbf{x}) \propto \sum_i c_i K(\mathbf{x}, \mathbf{x}_i)$ (c_i being the model parameters), depends on kernel computation with *all support vectors* $\{\mathbf{x}_i \in \mathbb{R}^d | i = 1 \ldots N_{sv}\}$. Hence, the test time and space complexities becomes $O(d \times N_{sv})$ vs. $O(d)$ for the linear case *i.e.* $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$. In practice, N_{sv} is of the order of number of training examples, and this leads to significant cost in terms of time and space. Such high cost makes it impractical for kernel based classifiers to be used for large scale tasks *e.g.* object detection, in which the classifier has to be applied to more than 100,000 windows per image [7, 21] or large scale image classification [13] with thousands of classes. Similarly, it makes them impractical to use with limited capability mobile devices in consumer applications *e.g.* smart-phone/tablet applications for object or landmark recognition or for real time object based video editing.

The second component of successful approaches for visual recognition, relies on the use of distributions of features for representing the images. The two of the most popular/successful of such features being the Histogram of Oriented Gradients (HOG) [2, 5] capturing shape information and bag-of-features [1, 17] capturing the appearance. The related histogram intersection kernel [18] $\sum_d \frac{x_d y_d}{|x_d||y_d|} \min(|x_d|, |y_d|) \ \forall \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ has thus been a natural choice, in visual tasks, for measuring the similarity between image representations $\mathbf{x}$ and $\mathbf{y}$ (*e.g.* [6, 9]). Being positive definite, it can be used directly as a kernel with SVM. However, recall that using kernel SVM leads to high *i.e.* $O(d \times N_{sv})$, time and memory requirement as a result of using the kernel trick. This has led to many recent works proposing faster ways to compute the classifier and/or the decision function corresponding to the histogram intersection kernel *e.g.* [11, 12, 20].

Traditionally, the kernel based classifiers having led to the best results on image classification problems with the standard bag-of-features [1, 17] image representation (*e.g.* the high ranking entries of the PASCAL VOC 2007 competition [3]) used classifiers based on non-linear kernels with support vector machine (SVM) classifiers [15]. While the kernel-SVMs perform well they incur a very high testing cost, as explained above. To overcome this limitation, Maji et al. [12] showed that SVM classifier decision corresponding to the histogram intersection kernel can be computed in logarithmic (wrt N_{sv}) time and also proposed a constant (wrt N_{sv}) time and space approximation for the same. More recently progress has been made in approximating the mapping to high dimensional feature spaces corresponding to commonly used kernels *e.g.* Maji and Berg [11] approximate the feature map corresponding to the intersection kernel and Vedaldi and Zisserman [20] approximate general additive kernels. The advantage of doing such mappings is that they allow the use of linear classification methods with the feature mapped vectors, but the drawback is that each data point has to be explicitly mapped, which has a cost.

In this paper, we take an (as far we know) unexplored route and show that it is possible to learn a nonlinear classifier implementing the histogram intersection kernel directly in the input space without using the dual formulation and the kernel trick, while achieving similar

classification performance. We work with the primal optimization problem (Eq. 1) and view the kernel as a parametrized scoring function inducing a nonlinear boundary in the input space. As a result, the problem we solve becomes non convex, but remains *quasi* convex. To solve the problem, we outline an approximate optimization algorithm which starts by solving a, highly smoothed, convex approximation of the objective and then successively solves less smoothed objectives, initialized with the solution of the previous one, converging to the current objective. In practice, however, we find that a simple stochastic sub-gradient descent solver, working with the sub-gradient of the non convex objective, with very small steps and multiple passes over the data achieves competitive performance. While with the current approaches [12, 20] two costs are incurred at test time, (i) mapping the test vector and (ii) computing $O(d')$ dot product, in higher d'-dimensional feature space, with the proposed method we only need to perform $O(d)$, with $d < d'$, operations in original space. Along with being conceptually interesting, our method thus leads to reduced space and time complexities.

2 Approach

A standard formulation for learning linear classifier is the SVM primal optimization,

$$\min_{w} \left\{ \frac{\lambda}{2}||\mathbf{w}||_2^2 + \frac{1}{|\mathcal{I}_t|}\sum_i \xi_i \right\}, \qquad \text{s.t. } y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \tag{1}$$

where $\mathbf{w}$ is the normal to the linear decision hyperplane and $\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{I}_t, y_i \in \{-1, +1\}\}$ are the training vector and label pairs. The optimization problem is convex and well studied, and many standard libraries (*e.g.* liblinear [4]) exist for solving it. However, only a linear decision boundary (*i.e.* a hyperplane parametrized by $\mathbf{w}$) can be learnt with this formulation. In general the classes may not be separable by linear boundaries and to allow learning more complex nonlinear decision boundaries, the dual formulation of the problem, given by

$$\max_{\alpha} \left\{ \sum_i \alpha_i + \left(\frac{1}{2} - \frac{1}{\lambda}\right) \sum_{i,j} \alpha_i \alpha_j y_i y_j \, \mathbf{x}_i^T \mathbf{x}_j \right\}, \qquad \text{s.t. } 0 \leq \alpha_i \leq \frac{1}{|\mathcal{I}_t|}. \tag{2}$$

is used. The dual formulation allows the use of the *kernel trick*, where a kernel function is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ with $\phi : \mathbb{R}^d \to \mathcal{F}$ being a feature map, mapping the vector $\mathbf{x} \in \mathbb{R}^d$ in the *input* space to a high (potentially infinite) dimensional *feature* space $\mathcal{F}$, where the classes are hoped to be linearly separable. Since in the dual formulation (Eq. 2) only the dot products (*i.e.* the kernel functions) are required, ϕ is only implicitly defined via the kernel function (see [15] for detailed discussion).

Towards the goal of learning a nonlinear classifier in input space, we start with the SVM problem (unconstrained formulation equivalent to Eq. 1) in feature space, obtained by mapping the input space vectors using the feature map $\phi : \mathbb{R}^d \to \mathcal{F}$,

$$L_\phi(\mathbf{w}_\phi) = \frac{\lambda}{2}||\mathbf{w}_\phi||_2^2 + \frac{1}{|\mathcal{I}_t|}\sum_i \max(0, 1 - y_i \langle \mathbf{w}_\phi, \phi(\mathbf{x}) \rangle) \tag{3}$$

where $\mathbf{w}_\phi \in \mathcal{F}$ denotes the (parameters of the) linear decision boundary in the feature space. While noting that exact pre-images of vectors in feature space might not exist in general [15], we assume that the pre-image of the vector $\mathbf{w}_\phi$ exists *i.e.* $\exists \mathbf{w} \in \mathbb{R}^d$ such that $\mathbf{w}_\phi = \phi(\mathbf{w})$ (in

case an exact pre-image does not exist, consider $\mathbf{w}$ to be the best approximation of the pre-image of $\mathbf{w}_\phi$). We can now write the objective function as,

$$L(\mathbf{w}) = L_\phi(\mathbf{w}_\phi) = \frac{\lambda}{2}||\phi(\mathbf{w})||_2^2 + \frac{1}{|\mathcal{I}_t|}\sum_i \max(0, 1 - y_i\langle\phi(\mathbf{w})\phi(\mathbf{x})\rangle). \tag{4}$$

Although we are working with non-negative bag-of-features histograms $\{\mathbf{x} \in \mathbb{R}_+^d\}$ in input space, we expect the vector $\mathbf{w}$ to be negative as well in general. We could see the $\mathbf{w}$ vector as a combination of two non-negative vectors $\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_- \forall \mathbf{w}_+, \mathbf{w}_- \in \mathbb{R}_+^d$ where the $\mathbf{w}_+$ ($\mathbf{w}_-$) capture the discriminative information supporting the positive (negative) class.

We now focus our interest on the feature map corresponding to the generalized intersection kernel *i.e.*

$$\langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle = K(\mathbf{x}, \mathbf{y}) = \sum_d \frac{x_d y_d}{|x_d||y_d|} \min(|x_d|, |y_d|) \ \forall \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{5}$$

With this map, the regularization term in Eq. 4 becomes

$$||\phi(\mathbf{w})||_2^2 = \langle\phi(\mathbf{w}), \phi(\mathbf{w})\rangle = K(\mathbf{w}, \mathbf{w}) = \sum_d \frac{w_d^2}{|w_d|^2} \min(|w_d|, |w_d|) = \sum_d |w_d| = ||\mathbf{w}||_1, \tag{6}$$

and part of the second term can be written as $y_i\langle\phi(\mathbf{w}), \phi(\mathbf{x})\rangle = y_i f(\mathbf{w}, \mathbf{x}_i)$ with $f(\mathbf{w}, \mathbf{x}) = K(\mathbf{w}, \mathbf{x})$ acting like a scoring function inducing a non-linear decision boundary in the input space.

Hence, the complete feature space optimization can be written in input space as

$$L(\mathbf{w}) = \frac{\lambda}{2}||\mathbf{w}||_1 + \frac{1}{|\mathcal{I}_t|}\sum_i \max(0, m - y_i f(\mathbf{w}, \mathbf{x})) \tag{7a}$$

$$f(\mathbf{w}, \mathbf{x}) = \sum_d \frac{w_d}{|w_d|} min(x_d, |w_d|) \ \ (\text{as } \mathbf{x} \in \mathbb{R}_+^d) \tag{7b}$$

where we have replaced the constant 1 (in the hinge loss term) with a free variable m as a unit distance in feature space will, in general, does not correspond to a unit distance in the input space.

Minimizing L_ϕ *w.r.t.* $\mathbf{w}_\phi$ in the feature space is a convex problem. It is solved using the kernel trick which evades the need of explicitly specifying ϕ. However, at test time, to compute the prediction for a test image, computing kernels with all the support vectors (which are of the order of number of training images) is required.

In this paper, instead of minimizing the convex objective L_ϕ (Eq. 4) in feature space, we propose to directly minimize the non linear and non-convex objective L (Eq. 7a). Towards this goal we now analyze the nature of the objective function L.

2.1 Quasi convexity of the objective function

The objective function we propose to solve *i.e.* $L(\mathbf{w}), \forall \mathbf{w} \in \mathbb{R}^d$ (Eq. 7a) is quasi-convex. Quasi-convexity requires that the function is only locally flat, more formally all the level sets of the function $S_\alpha = \{x | f(x) \leq \alpha\} \forall \alpha \in \mathbb{R}$ are convex sets. The ℓ_1 regularization term is convex and hence quasi convex (QC). To see the QC of the second term we note that it is a max over per example loss. $m - f(\mathbf{w}, \mathbf{x})$ is QC (as a function of $\mathbf{w}$ with fixed $\mathbf{x}$) as it

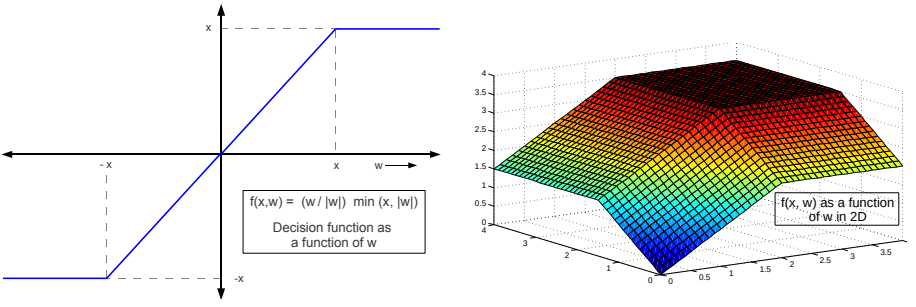

Figure 1: The scoring function for one dimension (left) and in the positive quadrant in two dimensions (right).

is piecewise affine and monotonic (albeit not strictly). Fig. 1 shows the decision function $f(\mathbf{w},\mathbf{x})$ for one and two dimensional spaces as a function of $\mathbf{w}$. Constant function is also QC and hence the per example loss, being a max over QC functions is also QC. As weighted sum of QC functions with positive weights is also QC, the loss over all examples is thus QC. Hence, we have to minimize a QC objective L (Eq. 7a) in input space.

2.2 Linear decision function as a convex relaxation

A standard approach to deal with non-convex optimization is to resort to approximation and solve a convex relaxation of the objective instead. Fig. 2 shows how a series of relaxations of the intersection based decision function (Eq. 7b) can be constructed, with final convergence to linear, and hence, convex decision function (convex optimization problem). An algorithm could, thus, be designed to solve the nonlinear optimization by successive smoothing. The algorithm would proceed by first solving a highly smoothed convex problem (corresponding to linear decision function) and then iteratively solving less smoothed versions, of the objective, initialized with the solution of previous ones.

The intersection decision function (for single dimensional feature $x \in \mathbb{R}$) could be relaxed with the following function, parametrized by t,

$$f_t(w,x) = \begin{cases} wx/t & \text{if} \quad w \leq t, \\ w\tan\left\{\tan^{-1}(x).(t-x)/(1-x)\right\} & \text{otherwise,} \end{cases} \tag{8}$$

for $x \in \mathbb{R}_+$ (the function is odd *i.e.* $f(w) = -f(-w)\ \forall w \in \mathbb{R}_-$). The parameter $t \in [x,1]$ ($x \leq 1$ as it is a component of ℓ_1 normalized BoF histogram) controls the amount of smoothness on the objective function, Fig. 2 illustrates the smoothed versions of the function for one dimensional w. We have no smoothing when $t = x$, while $t = 1$ leads to heavily smoothed linear decision function. When the decision function is linear *i.e.* $t = 1$, the optimization problem becomes convex.

When using this function for the sum of hinge losses for all d dimensional examples (*i.e.* many different $\mathbf{x} \in \mathbb{R}^d$) t is replaced with $\max(t, x_d)$ for each dimension of each example. The discussion above, pertaining to the convex relaxation of the objective, changes accordingly.

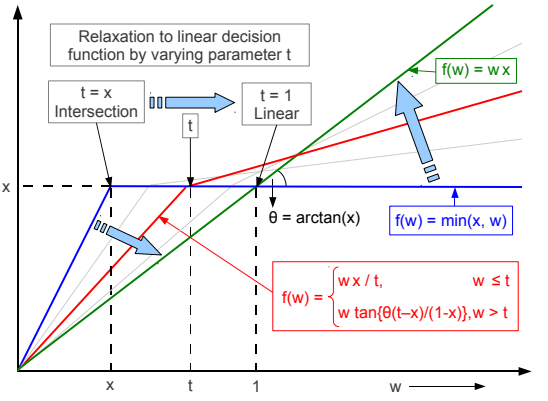

Figure 2: The linear decision function (green) can be seen as a convex relaxation of the histogram intersection like decision function (blue). If we introduce a parameter $t \in [x, 1]$, we can construct series of relaxations of the intersection like decision function converging to the convex linear decision function as t varies from x to 1, shown in red in here. Only the first quadrant is shown, the graphs in the third quadrant are obtained by mirroring about the two axes i.e. all functions are odd functions with $f(-x) = -f(x)$.

Algorithm 1 Stochastic gradient descent for learning nSVM

1: Initialize: $\mathbf{w} = \mathbf{0}$, r, λ and m
2: **for** iter $= 1, \ldots, 100$ **do**
3: $S \leftarrow$ random_shuffle($\mathcal{I}_t$)
4: **for all** $i \in S$ **do**
5: $w_d \leftarrow w_d + r y_i \; \mathbb{1}(y_i f(\mathbf{x}_i, \mathbf{w}) < m) \; \mathbb{1}(\text{abs}(w_d) < x_{id}) \;\; \forall \; d$ dimensions
6: $w_d \leftarrow \text{sign}(w_d) \; \max(0, \text{abs}(w_d) - r\lambda) \; \forall \; d$ dimensions
7: **end for**
8: **if** iter $= 50$ **do** $r \leftarrow r/10$ **end if**
9: **end for**

2.3 Stochastic subgradient based solver

In practice, we find that a simple stochastic solver using the subgradient of the objective function, Eq. 7a, performs well in practice. Eq. 7a can be seen as a ℓ_1 regularized hinge loss minimization, albeit with non-convex scoring function. ℓ_1 regularization with convex losses is a well studied topic (*e.g.* [14]). While the problem is non-convex, systems learned with ℓ_1 regularization perform well in practice (and are specially interesting for their sparsity inducing effects). We tried a stochastic (sub) gradient based algorithm, outlined in Alg. 1 ($\mathbb{1}(c) = 1$ if condition c is true, 0 otherwise), to solve Eq. 7a. We formulated the update steps analogous to other recent work in computer vision using stochastic gradients [13, 16]. There are two update steps, first (line 5, Alg. 1) updates $\mathbf{w}$ based on subgradient of the loss (denoted L_s) of the current example $\mathbf{x}$, given by (for each coordinate)

$$\nabla L_s(x_d) = \begin{cases} -y^i & \text{if } \; y^i f(\mathbf{w}, \mathbf{x}^i) < m \;\; \text{and} \;\; abs(w_d) < x_d \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and the second update (line 6, Alg. 1) is based on the sub gradient of the regularization term, simply the sign of $\mathbf{w}$ for each dimension. We clip the update step due to the second term to avoid zero crossing. The two step update in Alg. 1 does not correspond to the accurate sub gradient of the objective in Eq. 7a (see §2.1 in [14] for full subgradient). We assume $\lambda \ll 1$, which is true in practice, and thus ignore some effects of λ in the subgradient based updates. We find that the proposed algorithm performs well in practice. We choose to use this algorithm for its simple implementation and report results using it in the experimental section.

3 Experiments

We now report results for the nonlinear SVM learned in the original space directly using the Alg. 1. In the following we call the proposed method *n*on-linear SVM as nSVM. We report results on experiments designed to study the parameter sensitivity, speed and memory gains and the ability to achieve competitive performance.

3.1 Experimental setup

We report results on two standard computer vision benchmark datasets, (i) PASCAL VOC 2007[1] object image classification dataset [3] with 20 classes and 9963 images and (ii) Scene-15 database[2] [9] with 15 scene classes with 4492 images. We report results with standard protocol *i.e.* `train+validation` sets for training and `test` set for reporting performance for PASCAL VOC 2007 dataset and 100 random images per class for training and rest for reporting performance for Scene-15 dataset, and evaluation metrics for the datasets *i.e.* per-class and mean average precision (AP) for PASCAL VOC 2007 dataset and per-class and mean (over classes) accuracy for Scene-15 dataset.

We fix the image representation and compare classifiers (after feature mapping) on top of the representation. The image representation used (for both the proposed and compared methods) is the bag-of-features representation. We extract dense SIFT [10] features at 4 scales with a step size of 3 pixels with grayscale images. We learn a visual word vocabulary of size 1024 using k-means algorithm with 500,000 randomly sampled SIFT features from the training images. We use standard spatial pyramid representation [9] with the vector quantized features. We use 1×1, 2×2 and 3×1 (4×4) spatial cells for PASCAL VOC 2007 (Scene-15) dataset.

We compare with a closely related, recently proposed method of explicit feature maps by Vedaldi and Zisserman [20] which computes finite dimensional mappings corresponding to the histogram intersection kernel. Such mappings thus enable us to compute linear classifiers in the mapped space. It was shown by Vedaldi and Zisserman [20] that this feature mapping obtains better results that the one by Maji and Berg [11]. We use 3D features for the explicit feature map as this gave the best result in [20].

We use VLFeat library [19] to (i) compute SIFT descriptors, (ii) do k-means and (iii) compute the feature map corresponding to Vedaldi and Zisserman's method [20]. We use liblinear [4] (with ℓ_2 regularized ℓ_1 loss option) to learn SVM with the feature mapped vectors.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/
[2] http://www.featurespace.org/data.htm

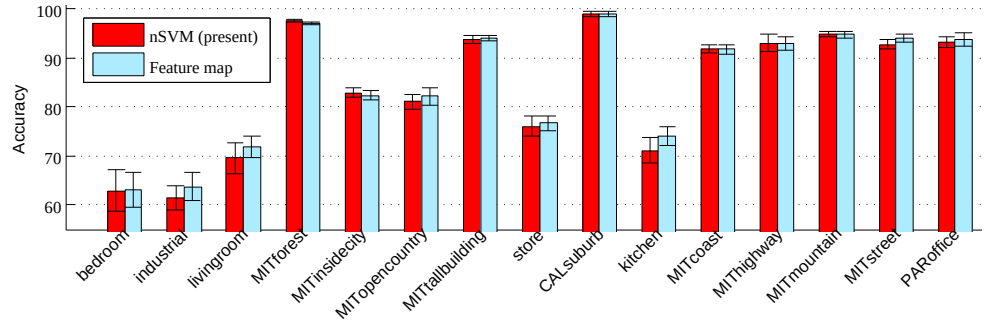

Figure 3: The accuracies for different classes (mean and std over 10 splits) of the Scene 15 dataset [9] for the proposed method (nSVM) and the explicit feature mapping of Vedaldi and Zisserman [20] (for histogram intersection kernel with linear SVM).

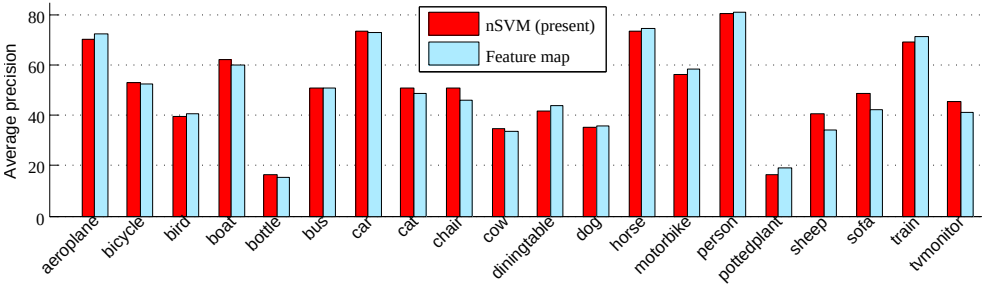

Figure 4: The average precisions for different classes of the Pascal VOC 2007 dataset [3] (image classification task) for the proposed method (nSVM) and the explicit feature mapping of Vedaldi and Zisserman [20] (for histogram intersection kernel with linear SVM).

3.2 Performance comparison

Fig. 4 and 3 show the per class performances of our method vs. explicit feature map [20] with linear SVM, on the two datasets. We get similar performance, on an average, compared to feature maps with mean average precision of 50.5 (present) vs. 49.7 (feature map) on PASCAL VOC 2007 dataset and mean class accuracy (over 10 random runs) of 84.0 ± 0.5 (present) vs. 84.7 ± 0.4 (feature map). We conclude that our method for learning a classifier directly in original space achieves competitive performance.

	Time		Memory	
	Secs	Speedup	Kbytes	Reduction
Vedaldi and Zisserman [20]	3.8	1 (ref)	448	1 (ref)
Nonlinear SVM (present)	0.2	19×	64	7×

Table 1: Testing time, for aeroplane class of Pascal VOC 2007 dataset [3] (on the full test set), averaged over 10 runs and memory usage (for keeping model in memory) comparison for the proposed model and the feature mapping method of Vedaldi and Zisserman [20].

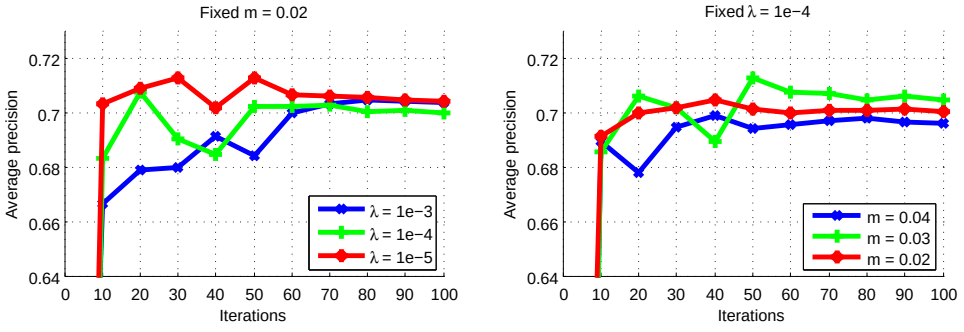

Figure 5: The average precisions for different values of (left) regularization parameter λ and (right) hinge loss parameter m, for a typical convergence of the proposed method.

3.3 Test time and space complexities

We compare the test time speedup and the memory usage reduction (Tab. 1) compared to the explicit feature map [20] which is already more than three orders faster than the kernel SVM (there are $O(10^3)$ support vectors and the scoring a new image/vector requires computing the histogram intersection kernel with each of them). We used the 3D features for explicit feature maps, as they were reported to give best results, which results in features of $7d$ dimensions for d-dimensional bag-of-features with spatial pyramid. While our method performs a linear scan on the d-dimensional features to calculate the test score (Eq. 7b), for explicit feature maps we have to, first, compute the mapping to $7d$ space and then compute a dot product in that space. Hence the model is $7\times$ bigger for explicit feature map compared to our method and (empirically) our method is about $19\times$ faster than explicit feature maps with linear SVM (the time is only due to classifier score computations and excludes the bag-of-features construction time for both methods). The training is also fast *e.g.* it takes about 45 secs to train a model for one class of PASCAL VOC 2007 dataset. We resorted to a conservative training strategy with multiple passes over the data and our training time can be arguably improved quite a bit.

3.4 Sensitivity to parameters

Fig. 5 shows the test average precision (AP) for the aeroplane class of the PASCAL VOC 2007 dataset with the iterations for the learning algorithm. We found that having a higher rate initially and then annealing by decreasing the learning rate midway was helpful for convergence, notice the convergence before and after iteration 50. The method converges for a range of λ and m parameters. While, the convergence happens for a range of values, cross validation is suggested for competitive performance.

4 Conclusion

Making non-linear classification efficient is advantageous for many applications specially with large number of images and categories *e.g.* large scale classification, and with limited computing resources *e.g.* in consumer devices like cameras or smart phones.

In the present paper we proposed a method for learning non-linear classifier corresponding to the histogram intersection kernel directly in the original space *i.e*. without using the kernel trick nor mapping the features explicitly to high dimensional space corresponding to the kernel. We formulated the non-linear optimization in the original space which corresponds to the linear optimization problem in the high dimensional feature space. We analyzed the objective function and proved that it is quasi-convex. We outlined an algorithm to optimize the quasi convex objective using successive relaxations, starting from a highly smoothed convex objective and iteratively solving less smoothed versions of the objective. We showed experimentally that a stochastic algorithm with subgradients works well in practice. Compared to a recent method for making non linear classification efficient, the proposed method is 19$\times$ faster and requires 7$\times$ less memory.

As future work, we plan to improve the training and use the system for applications mentioned above. Exploring similar learning strategies for kernels other than that based on the histogram intersection distance is also an interesting direction.

Aknowledgements

This research was supported by a grant from the *Conseil Régional de Basse-Normandie* (CRBN/11P01269/REPERE).

References

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Intl. Workshop on Stat. Learning in Comp. Vision*, 2004.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[4] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Dave Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[6] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007.

[7] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

[8] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with Fisher vectors for image categorization. In *ICCV*, 2011.

[9] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[10] David Lowe. Distinctive image features form scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004.

[11] Subhransu Maji and Alexander C Berg. Max-margin additive classifiers for detection. In *ICCV*, 2009.

[12] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[13] Florent Perronnin, Zeynep Akata, Zaid Harchaoui, and Cordelia Schmid. Towards good practice in large-scale learning for image classification. In *CVPR*, 2012.

[14] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for L1 regularization: A comparative study and two new approaches. In *ECML*, 2007.

[15] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.

[16] G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *CVPR*, 2012.

[17] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[18] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[19] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[20] A. Vedaldi and A. Zisserman. Efficient additive kernels using explicit feature maps. In *CVPR*, 2010.

[21] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.