TV-SVM: SUPPORT VECTOR MACHINE WITH TOTAL VARIATIONAL REGULARIZATION

Zhendong Zhang and Cheolkon Jung

School of Electronic Engineering, Xidian University, Xian, Shaanxi 710071, China
zhengzk@xidian.edu.cn

ABSTRACT

It is required that input features are represented as vectors or scalars in machine learning for classification, e.g. support vector machine (SVM). However, real world data such as 2D images is naturally represented as matrices or tensors with higher dimensions. Thus, structural information of the data whose dimensions are more than two is not successfully considered. One typical structural information which is useful for the classification task is the spatial relationship of the nearby data points. In this paper, to leverage this kind of structural information, we propose a novel classification method which combines total variational (TV) regularization with SVM, called TV-SVM. Since TV achieves a local smoothing property by penalizing the local discontinuity of data, TV-SVM preserves better local structure than the original SVM due to TV regularization. We solve the objective function of TV-SVM via the alternating direction method of multipliers (ADMM) algorithm. Experimental results on image classification show that TV-SVM is competitive to the state-of-the-art learning method in both classification accuracy and computational complexity.

Index Terms— Classification, machine learning, support vector machine, structural information, total variation

1. INTRODUCTION

Data classification is a basic problem in the field of machine learning. A large amount of classification methods have been developed for this problem. Most of the classification methods, such as support vector machines [1] (SVM) and decision trees [2], have been proposed in the case that the inputs are represented as vectors. However, many real world data is naturally represented as matrices or even tensors with higher dimensions. One typical example is image data. Gray images are represented as matrices, while color images can be regarded as three-dimensional tensors. When we deal with the classification problem in gray images via classical methods such as SVM, we reshape the input vectors with a matrix form. However, the structural information, especially the spatial relations of nearby data points in images [3], would be ignored during this process. Several methods have been proposed to deal with this issue over the past decade. In this work, we focus on the classification methods based on classical SVM. Roughly speaking, these methods regard the regression coefficients $\mathbf{W}$ as a matrix, and introduce new constraints or regularization of $\mathbf{W}$ based on some matrix properties to leverage the spatial corrections of input matrices. Wolf et al.[3] proposed rank-*k* SVM whose regression matrix is modeled as the sum of *k* rank-one matrices. Pirsiavash et al. [4] proposed a bilinear SVM (B-SVM) whose regression matrix is factorized into two low rank matrices. Recently, Luo et al. [5] proposed support matrix machine (SMM) which adds the nuclear norm of the regression matrix as a regularization term for SVM since the nuclear norm is a good approximation of matrix rank[6]. They are based on a similar assumption that the input matrices are low rank matrices, i.e. their rows or columns are highly correlated. However, the direct use of the low rank assumption in 2D data, especially natural images, may suffer problems [7]. Image data is correlated in arbitrary angles including the horizontal or vertical direction. A horizontal straight line is one rank, but when we rotate it into some other angles, e.g. $45°$, its rank would be increased. Thus, the rank of matrix is sensitive to rotations while image classification is required to be rotation invariant. To handle this problem, we propose a novel classification method for 2D data, called TV-SVM. We add total variation (TV) of the regression matrix into SVM as a regularization term based on the local smoothness assumption. TV regularization term penalizes local discontinuity of a matrix [8]. TV can also be regarded as L_1 norm of the regression matrix in the transformed domain (See Section 2.2). Thus, the proposed TV-SVM achieves sparseness of the regression matrix in the gradient domain since L_1 norm is a commonly used regularization term for sparse representation. Since the objective function of TV-SVM is convex but not smooth, it is optimized via an alternating direction method of multipliers (ADMM) [9]. Specifically, we develop an iteration algorithm for its optimization based on a fast version of ADMM [10]. Experimental results on three image classification data sets (INRIA Annotations for Graz-02 (IG02), CIFAR-10 and INRIA person) demonstrate that TV-SVM is competitive to SMM, i.e. the state-of-the-art method,

This work was supported by the National Natural Science Foundation of China (No. 61271298) and the International S&T Cooperation Program of China (No. 2014DFG12780).

in both classification accuracy and computational complexity.

2. RELATED WORK

2.1. Support Vector Machine

We give the notation and formulation of Support vector machine (SVM) which contanis a hinge loss function and a L_2 regularization. Given a set of training data $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \boldsymbol{R}^d$ is the *i*th input and $y_i \in \{-1, 1\}$ is its class label. If the input data is represented as matrices, we need to convert the input data into vectors to use SVM. SVM is formulated as follows:

$$\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{n} [1 - y_i(w^T x_i + b)]_+ \tag{1}$$

where $\mathbf{w} \in \boldsymbol{R}^d$ and b are regression coefficients, C is a hyper parameter of the regularization term, and $[x]_+$is the hinge loss function.

2.2. Total Variation

The concept of total variation (TV) has a very long history, and it had been first proposed as a norm for 2D signal [8] in 1992. In this work, we use one of variations for a 2D discrete signal of the original *TV*-norm as follows:

$$L_{TV}(X) = \sum_{i,j} |X_{i+1,j} - X_{i,j}| + |X_{i,j+1} - X_{i,j}| \tag{2}$$

where $\mathbf{X} \in \boldsymbol{R}^{m \times n}$ is a matrix. L_{TV} can be also represented as linear operators combined with standard L_1 norm. Specifically, we stack the matrix $\mathbf{X}$ into its corresponding vector $\mathbf{x} \in \mathbf{R}^N$ where N equals to $m \times n$ and construct two special sparse matrices $\mathbf{D}_x, \mathbf{D}_y \in \boldsymbol{R}^{N \times N}$, called 2D forward differentiation operators in the x- and y-directions, respectively, i.e. $\mathbf{D}_x \in \{0, 1, -1\}^{N \times N}$. There are up to 2 non-zero elements on each row of $\mathbf{D}_x$ (the one is 1, while the other is -1). The positions of the non-zero elements depend on the pairs of neighbors in the x-direction of the input matrix $\mathbf{X}$. $\mathbf{D}_y$ is constructed in the same way except that the positions of its non-zero elements depend on the pairs of neighbors in the y-direction [1]. In this way, the spatial information of $\mathbf{X}$ is preserved in $\mathbf{D}_x$ and $\mathbf{D}_y$ even when $\mathbf{X}$ is reshaped into a vector. Then, (2) is represented as follows:

$$L_{TV}(X) = \|D_x x\|_1 + \|D_y x\|_1 \tag{3}$$

For simplicity, let $D = [D_x; D_y]$, i.e. the concatenation of rows, then (3) is rewritten as follows:

$$L_{TV}(X) = \|Dx\|_1 \tag{4}$$

[1] Our MATLAB code is available at https://github.com/zzd1992/Differentiation-Operators

2.3. Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM) blends the decomposability of dual ascent with the convergence property of the method of multipliers [9] as follows:

$$\min \ f(x) + g(z) \quad st. \ Ax + Bz = c \tag{5}$$

with $\mathbf{x} \in \boldsymbol{R}^n, \mathbf{z} \in \boldsymbol{R}^m, \mathbf{A} \in \boldsymbol{R}^{p \times n}, \mathbf{B} \in \boldsymbol{R}^{p \times m}$and $\mathbf{c} \in \boldsymbol{R}^p$. The objective function is separated into two parts, called f and g, and both of them are required to be convex. As in the method of multipliers, the scaled augmented Lagrangian is represented as follows:

$$L_\rho(x, z, u) = f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c + u\|_2^2 \tag{6}$$

The general iterations of ADMM are as follows:

$$x^{k+1} = \arg\min_x L_\rho(x, z^k, u^k) \tag{7}$$

$$z^{k+1} = \arg\min_z L_\rho(x^{k+1}, z, u^k) \tag{8}$$

$$u^{k+1} = u^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{9}$$

3. SUPPORT VECTOR MACHINE WITH TOTAL VARIATIONAL REGULARIZATION

3.1. Problem Formulation

Given a set of training data $\{\mathbf{X}_i, y_i\}_{i=1}^n$, where $\mathbf{X}_i \in \mathbf{R}^{m \times n}$ is the *i*th sample formed as a matrix and $y_i \in \{-1, 1\}$ is its corresponding label. $\mathbf{W} \in [\mathbf{R}]^{m \times n}$ and $b \in \mathbf{R}$ are the regression matrix and bias, respectively, which are to be learned. Without reshaping the inputs into vectors, the original formulation of SVM is rewritten as follows:

$$\min_{\mathbf{W},b} \ \frac{1}{2} tr(\mathbf{W}^T \mathbf{W}) + C \sum_{i=1}^{n} [1 - y_i(tr(\mathbf{W}^T \mathbf{X}_i) + b)]_+ \tag{10}$$

It is obvious that (10) is equivalent to (1). To take the advantage of the local smooth assumption and restrain the regression matrix $\mathbf{W}$ to be sparse in the gradient domain, we add TV regularization of $\mathbf{W}$ into SVM. Thus, TV-SVM is represented as follows:

$$\begin{aligned} \min_{\mathbf{W},b} \quad & \frac{1}{2} tr(\mathbf{W}^T \mathbf{W}) + \tau L_{TV}(\mathbf{W}) \\ & + C \sum_{i=1}^{n} [1 - y_i(tr(\mathbf{W}^T \mathbf{X}_i) + b)]_+ \end{aligned} \tag{11}$$

where L_{TV} is TV norm defined in (2) and τ is a hyper parameter which determines the degree of TV regularization. Note that when $\tau = 0$, TV-SVM is equivalent to SVM defined in (10). Recall that L_{TV} is represented as a vectorized form of the input matrix by introducing forward differentiation operators (See (3) and (4)). Thus, TV-SVM can also be represented

as a vectorized form. Stack $\mathbf{X}_i$ and $\mathbf{W}$ into vectors $\mathbf{x}_i$ and $\mathbf{w}$ respectively. Eq. (11) is rewritten as follows:

$$\min_{w,b} \frac{1}{2}w^T w + \tau|Dw|_1 + C\sum_{i=1}^{n}[1 - y_i(w^T x_i + b)]_+ \quad (12)$$

3.2. Learning Algorithm

Since the objective function of TV-SMM is convex and non-smooth, we develop a learning algorithm based on a fast version of ADMM[10]. For simplicity, we derive the solution based on the vectorized form of TV-SMM, i.e. (12). Problem (12) is equivalent to the following equation:

$$\arg\min_{w,b,z} f(w,b) + g(z) \quad s.t.\ z - Dw = 0 \quad (13)$$

where $f(w,b) = \frac{1}{2}w^T w + C\sum_{i=1}^{n}[1 - y_i(w^T x_i + b)]_+$ and $g(z) = \tau|Dx|_1$. Then, the scaled augmented Lagrangian function of (13) is:

$$L_\rho(x,z,u) = f(w,b) + g(z) + \frac{\rho}{2}\|z - Dw + u\|_2^2 \quad (14)$$

In ADMM, variables are divided into two parts, i.e. (w,b) and z. Thus, problem (14) is divided into the following two subproblems:

$$\arg\min_{w,b} f(w,b) + \frac{\rho}{2}\|z - w + u\|_2^2 \quad (15)$$

$$\arg\min_{z} g(z) + \frac{\rho}{2}\|z - w + u\|_2^2 \quad (16)$$

The solution of subproblem (15) is:

$$w^* = \frac{1}{1+\rho}(\rho z + \rho u + \sum_{i=1}^{n}\alpha_i^* y_i x_i) \quad (17)$$
$$b^* = \frac{1}{|S|}\sum_{i\in S}\{y_i - (w^*)^T x_i\}$$

where $S = \{i : 0 < \alpha^* < C\}$ and α^* is the solution of the following constraint quadratic programming problem:

$$\arg\max_{\alpha} -\frac{1}{2}\alpha K\alpha + q^T\alpha \quad (18)$$
$$s.t. \quad 0 \preceq \alpha \preceq C,\ \sum_{i=1}^{n} y_i\alpha_i = 0$$

where $K \in \mathbf{R}^{n\times n}$ and $K_{ij} = \frac{y_i y_j x_i^T x_j}{1+\rho}$; $q \in \mathbf{R}^n$ and $q_i = 1 - \frac{\rho y_i (z+u)^T x_i}{1+\rho}$. The subproblem (16) is a standard 2D-TV problem. Many methods have been proposed for solving this kind of problem. In this work, we solve (16) via a an optimized taut-string method proposed by [11] which is the fastest method to solve Lasso. Its main idea for solving subproblem (16) is that TV of a 2D signal is decomposed into sum of TV of 1D signals. Specifically, TV of an $m \times n$ signal is decomposed into TV sum of *m* 1D signals with length *n* and *n* 1D signals with length *m*. Then, each 1D TV regularizer can be solved independently via the taut-string method. The computation complexity for solving each 1D TV regularizer is $O(N^2)$ in theory where *N* is the length of the input signal. However, in practice, its computation complexity is close to $O(N)$ [12], and thus the computation complexity for solving (16) via the optimized taut-string method is $O(mn^2)$ (Suppose $m \le n$).

Algorithm 1 Fast ADMM for TV-SVM

Initialize $z^{-1} = \widetilde{z}^0$, $u^{-1} = \widetilde{u}^0$, $\rho = 1$, $t^1 = 1$, $c^0 = 0$ and $\eta \in (0,1)$
for $k = 1, 2, 3...$ **do**
  $(w^k, b^k) = \arg\min_{w,k} f(w,b) + \frac{\rho}{2}\|\widetilde{z}^k - w + \widetilde{u}^k\|_2^2$
  $z^k = \arg\min_z g(z) + \frac{\rho}{2}\|z - w^k + \widetilde{u}^k\|_2^2$
  $u^k = u^k + \rho(z^k - w^k)$
  $c^k = \frac{1}{\rho}\|u^k - \widetilde{u}^k\|_2^2 + \rho\|z^k - \widetilde{z}^k\|_2^2$
  if $c^k < \eta c^{k-1}$ **then**
    $t^{k+1} = \frac{1 + \sqrt{1 + 4(t^k)^2}}{2}$
    $\widetilde{z}^{k+1} = z^k + \frac{t^k - 1}{t^{k+1}}(z^k - z^{k-1})$
    $\widetilde{u}^{k+1} = u^k + \frac{t^k - 1}{t^{k+1}}(u^k - u^{k-1})$
  else
    $t^{k+1} = 1$
    $\widetilde{z}^{k+1} = z^{k-1}$, $\widetilde{u}^{k+1} = u^{k-1}$
    $c^k = \frac{c^{k-1}}{\eta}$
  end if
end for

Table 1. Three data sets for experiments

Data set	#Positive	#Negtive	Dimension
IG02: *Car* & *Bike*	420	365	90×120
CIFAR-10: *Airplane* & *Bird*	1005	1032	32×32
INRIA person	902	1212	160×96

The learning alrorithm based on ADMM for our TV-SVM is summarized in Algorithm 1. Since objective function of TV-SVM is convex, the convergence of Algorithm 1 is guaranteed and the global optimal solution is promised based on the theoretical results in [9] [10].

4. EXPERIMENTAL RESULTS

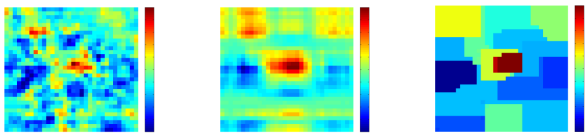
To verify the superiority of TV-SVM, we perform experiments on three image classification data sets: INRIA anno-

Table 2. Training time (unit: second) and classification accuracy (unit: %) on three data sets

Method	INRIA person		CIFAR-10: *Airplane* & *Bird*		IG02: *Car* & *Bike*	
	Time	Accuracy	Time	Accuracy	Time	Accurcy
L-SVM	-	80.12(1.71)	-	67.60(1.69)	-	72.91(2.32)
SMM	**42.21**(7.23)	81.98(1.15)	15.40(10.53)	**68.62**(1.58)	**11.62**(4.53)	73.68(3.61)
TV-SVM	42.87(7.29)	**82.99**(0.43)	**14.79**(10.88)	68.23(1.90)	11.80(4.37)	**73.86**(2.62)

Time and accuracy are represented by the form of "average(standard deviation)", and bold numbers represent the best average performance.

tations for Graz-02 (IG02)[2], CIFAR-10 [13], and INRIA person [3]. We compare the performance of TV-SVM with those of the standard linear SVM (L-SVM) [1] and SMM [4]. Experiments are preformed using Matlab R2014b on a PC with Inter i7-4790 CPU 3.60GHz (8 cores), 16GB RAM in ubuntu 14.04 system. IG02 is a re-edition of the popular natural-scene object category data set made by Graz University of Technology. We select two subsets of IG02: *Car* and *Bike*. We resize the original images into 90×120, and then set *Car* as positive samples and *Bike* as negative samples. CIFAR-10 data set is labeled subsets with 80 million tiny images collected by Krizhevsky et al. We use first 10,000 training samples of CIFAR-10 data set, and then set *Airplane* as positive samples and *Bird* as negative samples because this pair of contents is a little more difficult to classify. INRIA person data set has been collected to detect whether people exist in an image. We resize the images into 160×96. Note that we only use the normalized gray images of three date sets as input features without any other preprocessing. Table 1 summarizes the information of three data sets including the numbers and dimensions of input samples. In each data set, we randomly select 70% samples for training and the rest for testing. We select the hyper parameters, i.e. (C, τ), by cross validation. For each (C, τ) pair, we run both TV-SVM and SMM 10 times to calculate the mean and standard deviations of the classification accuracy and training time. The results are summarized in Table 2. It can be observed that TV-SVM is competitive to SMM, i.e. the state-of-the-art method, in both classification accuracy and computational complexity. Specifically, the classification accuracy of TV-SVM is higher than that of SMM on IG02 and INRIA person data sets. In CIFAR-10, TV-SVM is inferior to SMM in classification accuracy, but still outperforms L-SVM. The computation complexity of TV-SVM and SMM are nearly the same. Normalized regression matrices of L-SVM, SMM, and TV-SVM, i.e. $\mathbf{w}$, learned from CIFAR-10 data set are shown in Fig. 1. It can be observed that $\mathbf{w}$ of SMM and TV-SVM is more regular than $\mathbf{w}$ of L-SVM. Specifically, the regression matrix of SMM in Fig. 1(b) has strong row corrections while the regression matrix of TV-SMM in Fig. 1(c) is very smooth and its boundaries between blocks are obvious. These phenomenons are consistent with theoretical analysis. This is because SMM is based on the low rank assumption and TV-SVM is mainly based on the local smoothness and gradient sparse assumption, while L-SVM has no extra constraint to $\mathbf{w}$ except L_2 norm. Although we evaluate TV-SVM on 2D data sets, it is easy to extend TV-SVM to higher dimensional data because the optimized taut-string method [11] can handle higher dimensional TV problems. However, SMM is not easy to be generalized to higher dimensional input data because SMM involves singular value decomposition of the regression matrix.



(a) (b) (c)

Fig. 1. Normalized regression matrices learned from CIFAR-10 data set. (a) L-SVM. (b) SMM. (c) TV-SVM.

5. CONCLUSIONS

In this paper, we have proposed TV-SVM for data classification. To leverage the local smoothness and gradient sparse properties of input 2D data, TV-SVM introduces TV regularization of the regression matrix into the standard L-SVM. We have derived an iteration algorithm based on the fast ADMM to solve TV-SVM. We have verified the superiority of TV-SVM on three image classification data sets. Experimental results demonstrate that TV-SVM consistently performs better than L-SVM and is competitive to SMM, i.e. the state-of-the-art method, both in classification accuracy and computational complexity. Moreover, TV-SVM can be straightforwardly generalized to higher dimensional input data.

6. REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," 1990.

[3] L. Wolf, H. Jhuang, and T. Hazan, "Modeling appearances with low-rank svm," in *Proceedings of IEEE Con-*

[2] http://lear.inrialpes.fr/people/marszalek/data/ig02/

[3] http://pascal.inrialpes.fr/data/human/

ference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1–6.

[4] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Bilinear classifiers for visual recognition," in *Advances in Neural Information Processing Systems*, 2009, pp. 1482–1490.

[5] L. Luo, Y. Xie, Z. Zhang, and W.-J. Li, "Support matrix machines," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 938–947.

[6] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," *Lecture Notes in Computer Science*, vol. 3559, pp. 545–560, 2005.

[7] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "Tilt: Transform invariant low-rank textures," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 1–24, 2012.

[8] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms ," *Physica D: Nonlinear Phenomena*, vol. 60, no. 14, pp. 259–268, 1992.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[10] T. Goldstein, B. Donoghue, and S. Setzer, "Fast alternating direction optimization methods," *Siam Journal on Imaging Sciences*, vol. 7, no. 3, 2014.

[11] A. Barbero and S. Sra, "Modular proximal optimization for multidimensional total-variation regularization," *Mathematics*, vol. 4, pp. 58, 2014.

[12] L. Condat, "A direct algorithm for 1d total variation denoising," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1054–1057, 2013.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.