The Efficient SVM Kernel Method for Image Compression and Image Recognition

A.Athiraja[1], A.Baranidharan[2], D.Nagendra Kumar[3], S.Vignesh Manickam[4]
M.E., Department of Computer Science, Vickram College of Engineering
Enathi, Sivagangai(dst), India[1,2,3,4]

Abstract

SVM have met the significant success in numerous real-world applications. The SVM is widely used classifier. Obtaining the best result with SVM requires an understanding of their working and various ways a user an influence their accuracy, so we provide the concept about the SVM algorithm. We introduce a algorithm for performing image compression based on the SVM algorithm. This algorithm is used to form the clustering. We also to use the SVM algorithm for image recognition application.

Keywords

SVM algorithm or classifier, image compression, clustering, image recognition

1. Introduction

SVMs (Support Vector Machines) are a useful technique for data classification. Al-though SVM is considered easier to use than Neural Networks, users not familiar with it often get unsatisfactory results at first. Here we outline a \cookbook" approach which usually gives reasonable results.

Note that this guide is not for SVM researchers nor do we guarantee you will achieve the highest accuracy. Also, we do not intend to solve challenging or difficult problems. Our purpose is to give SVM novices a recipe for rapidly obtaining acceptable results.

Although users do not need to understand the underlying theory behind SVM, we brie y introduce the basics necessary for explaining our procedure. A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one “target value" (i.e. the class labels) and several “attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Given a training set of instance-label pairs (xi; yi); i = 1; : : : ; l where xi Rn and y f{1; -1 }the support vector machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) require the solution of the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1-\xi_i,$$

$$\xi_i \geq 0.$$

Here training vectors xi are mapped into a higher (maybe infinite) dimensional space by the function. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term. Furthermore, K(xi; xj) (xi)T (xj) is called the kernel function. Though new kernels are being proposed by researchers, beginners may find in SVM books the following four basic kernels:

linear: K(xi; xj) = xTi xj.
polynomial: K(xi; xj) = (xiT xj + r)d, > 0.
radial basis function (RBF): K(xi; xj) = exp(kxi xjk2), > 0.
sigmoid: K(xi; xj) = tanh(xiT xj + r).
Here , r, and d are kernel parameters.

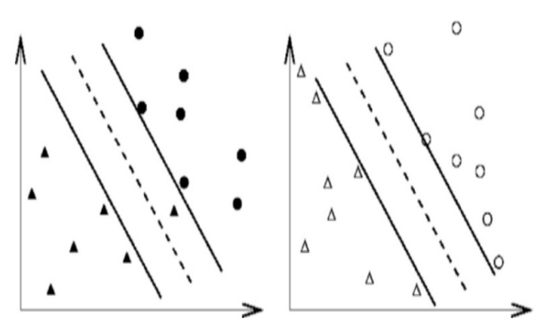
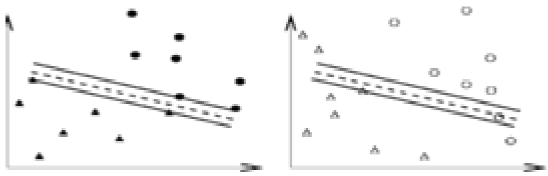


Fig 1. a) Training data and an over fitting classifier (b) Applying an over fitting classifier on testing data



(c) Training data and a better classifier
(d) Applying a better classifier on testing data

2. Image Cluster Compression

Extending image compression to multiple images has not attracted much research so far. The only exceptions are the areas of hyper spectral compression and, of course, video compression, which both handles the special case of compressing highly correlated images of exactly the same size.
compression, we Concerning generalized image group recently researched an algorithm which works by building a special eigenimage library for extracting principal component based similarities between images.
While the algorithm presented in is quite fast, and manages to merge low-scale redundancy from multiple images, it fails to detect more global scale redundancies (in particular, similar image parts which are both translated and scaled), and also has the problem of becoming "saturated" quite fast (i.e., the more images in a group, the worse the additional compression rate of the individual images), which limits the size of possible image groups.

In this paper, we present a novel algorithm for image groups, which is based on SVM compression and thus manages to exploit several high-level redundancies, in particular scaled image parts.
Compression of image sequences using PIFS was done previously (in the context of video compression) . However, in these papers, both the frames/images contributing to one compression group as well as the order of those images is predetermined by the video sequence. Furthermore, images need to be of the same size, which can't be assumed for most real-world image databases. Here, they specify a multi-image PIFS algorithm which works on images of arbitrary sizes, and also allows to cluster image databases into groups so that compression of each group is optimized.
The rest of previous paper is organized as follows: they first derive the multi-image PIFS algorithm by generalizing the single-image PIFS algorithm. We also describe a way to optimize said algorithm using DFT lookup tables. Afterwards, they take on the problem of combining the "right" images into groups, by first describing efficient ways to compute a distance function between two images, and then, in the next session, comparing a number of clustering algorithms working on such a distance. The final algorithm is evaluated by compression runs over a photo database consisting of 3928 images.

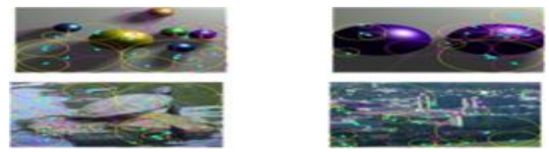


Fig. 2 Database images

they found that by using clustering algorithms (a type of algorithm usually more common in the fields of data analysis and image segmentation), they can find approximations to the image grouping problem while using significantly less computing time.

They considered a number of different clustering algorithms, which all have different advantages and disadvantages, and which will described in the following.

- MST clustering: An algorithm which calculates the spanning tree from the distance metric, and then splits the tree into clusters by cutting off edges.
- nCut clustering: A hierarchical method which treats the complete data set as one big cluster, and then starts split-ting the nodes into two halves until the desired number of clusters is reached (Splitting is done by optimizing the nCut metric).
- SAHN clustering: Another hierarchical method, which in each step, combines a node (or cluster) and another node (or cluster), depending on which two nodes/clusters have the smallest distance to each other. Distances between clusters are evaluated using the sum over all distances between all nodes of both clusters, divided by the number of such distances.
- Relational k-Means: An extension of the "classical" k-Means of multidimensional data [21], which computes centers not by the arithmetic mean, but by finding a "median" node with the lowest mean distance to all other nodes .
- Random clustering: Distributes nodes

between clusters arbitrarily. This algorithm was included for comparison purposes.

They did a comparison run of the aforementioned clustering algorithms on a small image database (128 images) using both

$$\text{distance} = \frac{2}{\|\mathbf{w}\|}$$

$$\textit{find } w, b, \textit{solve}$$

$$\max \frac{2}{\|\mathbf{w}\|}$$

$$s.t.\ w^T x + b \geq 1 \ if \ \ y_i = 1$$

$$w^T x + b \leq 1 \ if \ \ y_i = -1$$

the Gabor filter metric as well as the full NCD metric, in order to evaluate how much difference a more precise distance metric makes.

We have introduce the svm algorithm based image clustering using image compression first to find the *Maximal margin = distance between*

$$x_1, x_2, \cdots, x_l \in R^n$$

Example ↓

$$\phi(x_1), \phi(x_2), \cdots \phi(x_l) \in R^m$$

Find a linear separating hyperplane

$$\max \frac{2}{\|\mathbf{w}\|}$$

$$s.t.\ w^T \phi(x_i) + b \geq 1 \ if \ \ y_i = 1$$

$$w^T \phi(x_i) + b \leq 1 \ if \ \ y_i = -1$$

$$\max \frac{2}{\|\mathbf{w}\|} \equiv \min \frac{\|\mathbf{w}\|}{2} = \min_{w,b} \frac{w^T w}{2}$$

$$subject.to.\ y_i(w^T \phi(x_i) + b) \geq 1 \ i = 1, \cdots l$$

Example: $x \in R^3, \phi(x) \in R^{10}$

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

non-linear curves: linear hyperplane in high dimension space (feature space)

Fig.3 SVC formulations (the soft margin hyper plane)

$$\min_{w,b,\xi} \ \frac{1}{2} w^T w + C(\sum_{i=1}^{l} \xi_i)$$

$$y_i((w^T \phi(x_i)) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, i = 1, \cdots, l$$

Expect: if separable,

$$\xi_i = 0$$

$$C \sum_{i=1}^{l} \xi_i \ : \ penalty \ \ term$$

$$C : a \ \text{constant}$$

$$h_1(x) = 0, \cdots, h_m(x) = 0$$

find necessary condition

if x is an opt. and x satisfied regularity cond.

$$\Rightarrow \nabla f(x) = \lambda_1 \nabla g_1(x) + \cdots + \lambda_m \nabla g_m(x) +$$

$$\mu_1 \nabla h_1(x) + \cdots + \mu_n \nabla h_n(x)$$

Ho

$$\lambda_i g_i(x) = 0$$

$$\lambda_i \geq 0, \ g_i(x) \geq 0$$

$$h_j(x) = 0$$

w to solve an opt. problem with constraints? Using Lagrangian multipliers Given an optimization problem

$$\min \ f(w)$$

$$\text{subject to } g_i(w) \leq 0, i = 1, \cdots l$$

$$h_i(w) = 0, i = 1, \cdots, m$$

we define the generalised Lagrangian function as

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{l} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

$$= f(w) + \alpha^T g(w) + \beta^T h(w)$$

Consider the following primal problem

$$\text{minimise}_{\xi, w, b} \quad w^T \cdot w + C \sum_{i=1}^{l} \xi_i$$

$$\textit{subject to} \quad y_i(w^T \cdot \phi(x_i) + b) \geq 1 - \xi_i \ , i = 1, \cdots, l$$

$$\xi_i \geq 0 \ , i = 1, \cdots, l$$

(P) # variables: w→ dimension of ϕ(x) (very big number) , b→1, ξ→ l ,(D) # variables:

lDerive its dual. The primal Lagrangian for the problem is :

$$L(w,b,\xi,\alpha,\mu)=\frac{1}{2}(w^T\cdot w)+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i[y_i((w^T\cdot x_i)+b)-1+\xi_i]-\sum_{i=1}^{l}\mu_i\xi_i$$

The corresponding dual is found by differentiating with respect to w, ξ, and b.

$$\frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial w}=w-\sum_{i=1}^{l}\alpha_i y_i x_i=0 \quad \Rightarrow w=\sum_{i=1}^{l}\alpha_i y_i x_i$$

$$\frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial \xi_i}=C-\alpha_i-\mu_i=0$$

$$\frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial b}=\sum_{i=1}^{l}\alpha_i y_i=0$$

Re constituting the relations obtained into the primal to obtain the following adaptation of the dual objective function:

$$L(w,b,\xi,\alpha,\mu)=-\frac{1}{2}\sum_{i,j=1}^{l}y_i y_j\alpha_i\alpha_j(x_i^T\cdot x_j)+\sum_{i=1}^{l}\alpha_i$$

Let $Q_{ij}=y_i y_j x_i^T\cdot x_j$ then

$$L(w,b,\xi,\alpha,\mu)=-\frac{1}{2}\alpha^T Q\alpha+e^T\cdot\alpha$$ Hence,

maximizing the above objective over is equivalent to maximizing

$$\max \quad \Phi(\alpha)=-\frac{1}{2}\alpha^T Q\alpha+e^T\cdot\alpha$$

$$\textit{subject to} \quad y^T\alpha=0$$

$$0\le\alpha\le C$$

$$(\because C-\alpha_i-\mu_i=0 \text{ and } \mu_i\ge 0\Rightarrow\alpha_i\le C)$$

the constraint $C-\alpha_i-\mu_i=0$, together with $\mu_i\ge 0$, enforces $\alpha_i\le C$

Primal and dual problem have the same KKT conditions Primal: # variables very large (shortcoming) Dual: # of variable = l

$$Q_{ij}=y_i y_j\phi(x_i)^T\cdot\phi(x_j)$$

High dim. Inner product Reduce its computational timeFor special ϕ question can be efficiently calculated

Kernel function

$$K(x,y)\equiv\phi(x)^T\cdot\phi(y)$$

The solution of a SVM has the form :

$$f(x)=sign\left(\sum_{i=1}^{l}y_i\alpha_i^* K(x,x_i)+b^*\right)$$

Commonly used kernels :

Linear : $K(x, y)=(x\cdot y)$

Gaussian RBF :

$$K(x,y)=\exp(-\gamma\|x-y\|^2)$$

Polynomial of degree d :

$$K(x,y)=(1+x\cdot y)^d$$

The linear classifier relies on dot product between vectors $K(x_i,x_j)=x_i^T x_j$

If every data point is mapped into high-dimensional space via some transformation Φ: $x\rightarrow\varphi(x)$, the dot product becomes: $K(x_i,x_j)=\varphi(x_i)^T\varphi(x_j)$ A kernel function is some function that corresponds to an inner product in some expanded feature space.

Example: 2-dimensional vectors $x=[x_1 \quad x_2]$; let $K(x_i,x_j)=(1+x_i^T x_j)^2$, Need to show that $K(x_i,x_j)=\varphi(x_i)^T\varphi(x_j)$:

$K(x_i,x_j)=(1+x_i^T x_j)^2=1+x_{i1}^2 x_{j1}^2+2\,x_{i1}x_{j1}\,x_{i2}x_{j2}+x_{i2}^2 x_{j2}^2+2x_{i1}x_{j1}+2x_{i2}x_{j2}=[1\ \ x_{i1}^2\ \ \sqrt{2}\,x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T[1\ \ x_{j1}^2\ \ \sqrt{2}\,x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]=\varphi(x_i)^T\varphi(x_j)$, where $\varphi(x)=[1\ \ x_1^2\ \ \sqrt{2}\,x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]$

For some functions $K(x_i,x_j)$ checking that $K(x_i,x_j)=\varphi(x_i)^T\varphi(x_j)$ can be cumbersome. Mercer's theorem: Every semi-positive definite symmetric function is a kernel .Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_N)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_N)$
…	…	…	…	…
$K(\mathbf{x}_N,\mathbf{x}_1)$	$K(\mathbf{x}_N,\mathbf{x}_2)$	$K(\mathbf{x}_N,\mathbf{x}_3)$	…	$K(\mathbf{x}_N,\mathbf{x}_N)$

Fig. 4 Examples of Kernel Functions

Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$. Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^p$

Gaussian (radial-basis function network): $K(\mathbf{x_i},\mathbf{x_j}) = \exp(-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2})$

Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0 \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j} + \beta_1)$

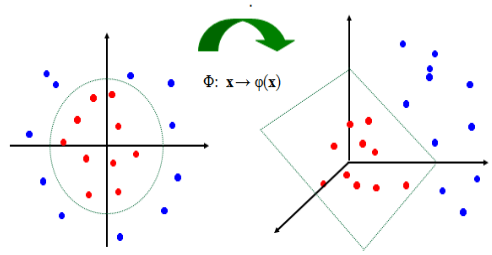


Fig.5 Example

3. Image recognition

Face recognition is a rapidly growing field day for is many uses in the fields of biometric authentication, security, and many other areas. There are many problems that exist due to the many factors that can affect the photos. When processing images one must take into account the variations in light, image quality, the persons pose and facial expressions along with others. In order to successfully be able to identify individuals correctly there must be some way to account for all these variations and be able to come up with a valid answer.

Fig. 6 Differences in Lighting and Facial Expression

4. Approach

In order to come up with a method that will help increase the chances of correct matches I propose to apply methods we have learned this year to "preprocess" the images before they are sent into the database to be matched. This should help to remove some of the major differences that can show up in the images. In order to verify the results of this processing I am going to implement the eigenface approach proposed by Turk and Pentland which can be found in there paper here.

5. Process

First we implemented the calculations for the eigenfaces which I will give a brief overview of taken from Turk and Pentland.

1) Acquire an initial set of face images (the training set) Γ1, Γ2,....ΓM

2) Calculate the eigenfaces from the training set, keeping the M best images and there corresponding eigenvalues to make up the face space. In order to calculate the eigenfaces I followed the method I have outlined below Recognition Process.

Once the eigenfaces are know you can take an input image and in the same way calculate it's eigenfaces from the known data and use this to classify it to a known face value. I chose to use the Euclidean distance as done by Turk and Pentland to calculate the known face.

Characteristics of regions face

An area of skin contains at least two non-skin regions (eg eyes) (Fig 7). The ratio of area surface must be between certain values (Fig 7). There is a distance d belongs to interval [d0, d1] between two sets of non-skin regions in this region. (The eyes should be distributed between the left and right side of the face for example) (Fig 7). In addition, non-skin regions should exist in the upper face. The distance between the ordinates of the centres of gravity of non-skin regions of the current region must belong to a certain interval (Fig 7).

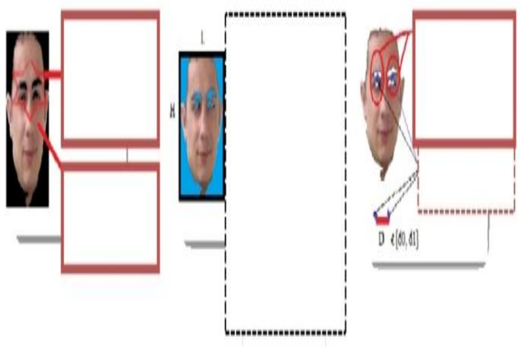


Fig. 7 Characteristics of regions face

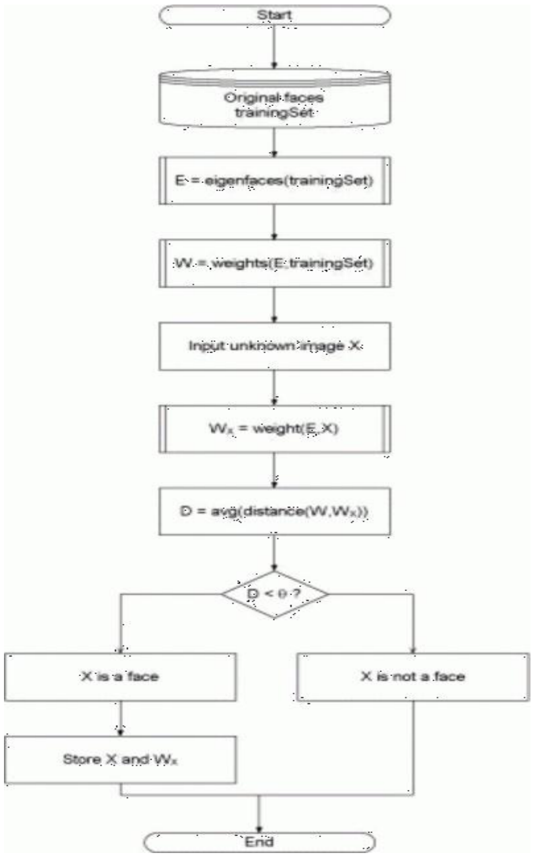


Fig. 8 Flow Chart

6. Conclusion

In this paper, we have addressed the problem of image clustering based on PFS and also explain the image clustering based on the svm algorithm. It is used in the image compression application. We have introduced concept of svm based image recognition, it is used in the face detection application. Future work will focus on svm based image compression and recognition using VLSI.

References

[1] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003).

[2] E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classi_ers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pages 144{152. ACM Press, 1992.

[3] Kramm, Matthias. "Image cluster compression using partitioned iterated function systems and efficient inter-image similarity features." In Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on, pp. 989-996. IEEE, 2007.

[4] M. Kramm, "Compression of image clusters using Karhunen Loeve transformations," in Electronic Imaging, Human Vision, vol. XII, no. 6492, 2007, pp. 101–106.

[5] M. Barnsley and A. Sloan, "A better way to compress images." Byte, vol. 13, no. 1, pp. 215–223, 1988.

[6] Moudani, Walid, Ahmad Shahin, Fadi Chakik, A. R. Sayed, and Félix Mora-Camino. "An Efficient Approach for Image Recognition using Data Mining." International Journal on Computer Science and Engineering 3, no. 1 (2011).