Example:

Suppose we want to build a classifier that recognizes WebPages of graduate students.

How can we find training data?

We can browse the web and collect a sample of WebPages of graduate students of various universities.

Now we have a collection of ***positive examples***.

How about ***negative examples*** ?

The negative examples are… *the rest of the web that is not "a graduate student webpage"*.

So: the negatives examples come from an unknown number of different "negatives" classes.

Thus: It is hopeless, and wrong, to trying to characterize the distribution of the negatives; they can belong to *any* class. ("**Each negative examples is negative in its own way**.")

We just *cannot* formulate this problem as a two class classification problem.

It can be seen as a *(1+x)-class* learning problem:

There are an unknown number (*x*) of classes, but the user is interested in one class, i.e. the user is biased toward one class.

Similarly: in *content-based image retrieval*, and *document retrieval* in general.

How do we approach this problem then?

It is reasonable to assume that positive examples cluster in a certain way. ("**All positive example are alike**.")

Thus: We can attempt to capture the distribution of the positive examples.

One-class SVMs offer a solution to the (1+x)-class problem.

One-Class SVM for Learning in Image Retrieval

Y. Chen, X. S. Zhou, T. Huang
University of Illinois at Urbana-Champaign

IEEE International Conference on Image Processing
2001

Undesirable result reached by a two-class SVM

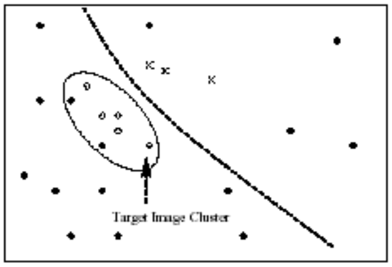


Figure 1 Decision boundary of a two-class SVM: The circles are the positive images, the crosses are the negative ones, and the black dots are the unlabeled images. The decision boundary (the dashed line) will classify many non-target images as positive.

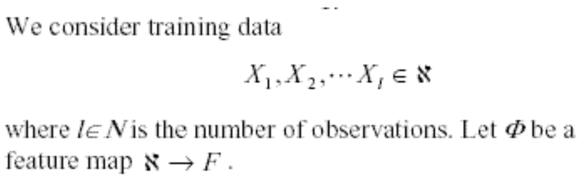
The Proposed Approach

Try to fit a tight hyper-sphere (in a transformed space) to include *most* positive training examples.

Such hyper-sphere tries to capture the support within which the positive examples are clustered, in an effort to separate them from “the rest of the world”.

The hyper-sphere will include *most*, but *not all*, training data to avoid *overfitting*.

One class SVM



We consider training data

$$X_1, X_2, \cdots X_l \in \aleph$$

where $l \in N$ is the number of observations. Let Φ be a feature map $\aleph \to F$.

One class SVM

radius

$$\min_{R\in\Re,\zeta\in\Re^{l},c\in F} R^2 + \frac{1}{\nu l}\sum_{i} \zeta_i$$

$$s.t.\quad \|\Phi(X_i) - c\|^2 \le R^2 + \zeta_i, \quad \zeta_i \ge 0 \ for\ i \in [l]$$

center

One class SVM

We can solve this optimization with Lagrangian multipliers:

$$L(R,\zeta,c,\alpha,\beta) = R^2 + \sum_{i=1}^{l} \alpha_i \left[\|\Phi(X_i) - c\|^2 - R^2 - \zeta_i \right]$$

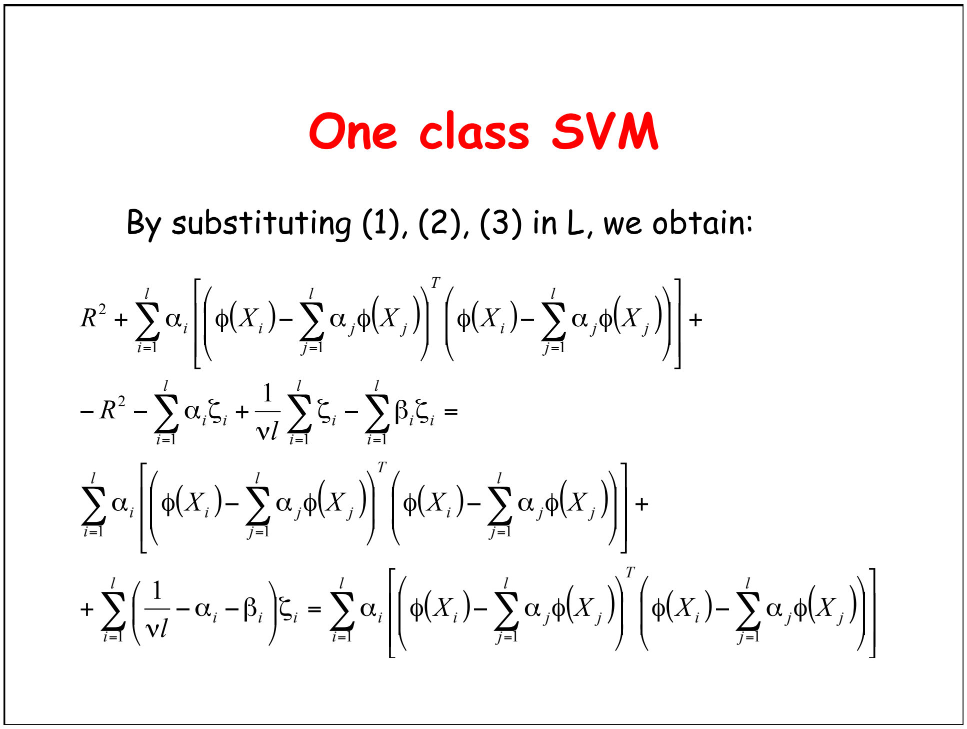
$$+ \frac{1}{\nu l}\sum_{i=1}^{l} \zeta_i - \sum_{i=1}^{l} \beta_i \zeta_i$$

$$\frac{\partial L}{\partial R} = 2R(1 - \sum \alpha_i) = 0 \ \Rightarrow\ \sum \alpha_i = 1 \qquad (1)$$

$$\frac{\partial L}{\partial \zeta_i} = \frac{1}{\nu l} - \alpha_i - \beta_i = 0 \ \Rightarrow\ 0 \le \alpha_i \le \frac{1}{\nu l} \qquad (2)$$

$$\frac{\partial L}{\partial c} = -\sum 2\alpha_i (\Phi(X_i) - c) = 0 \qquad (3)$$

$$\Rightarrow \boxed{c = \sum \alpha_i \Phi(X_i)}$$



One class SVM

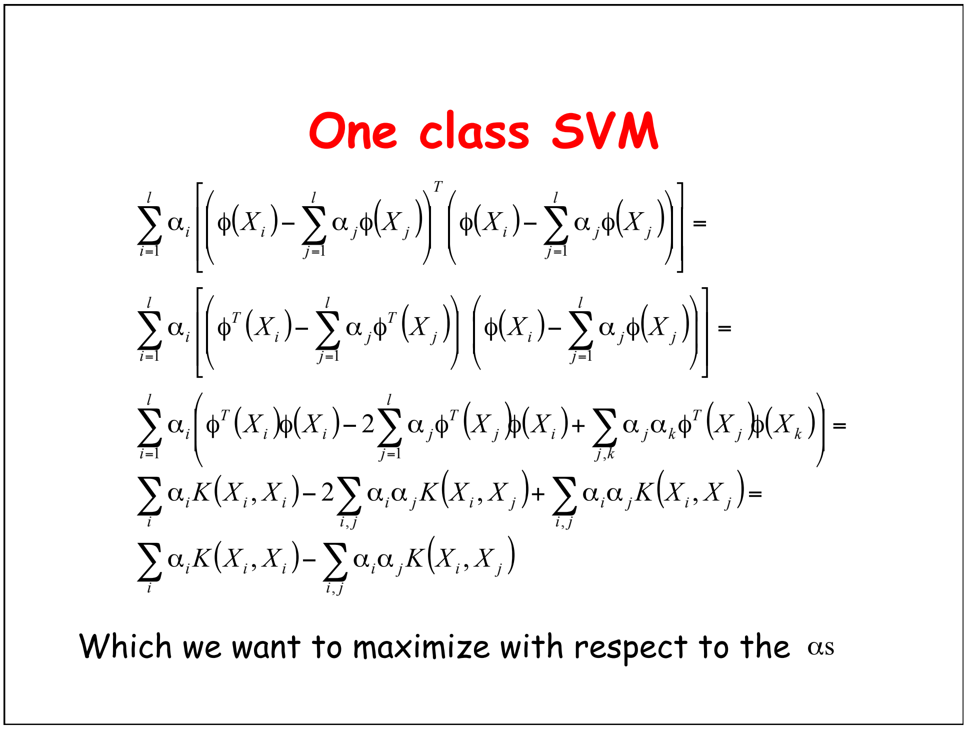
By substituting (1), (2), (3) in L, we obtain:

$$R^2 + \sum_{i=1}^{l} \alpha_i \left[\left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right)^T \left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right) \right] +$$

$$- R^2 - \sum_{i=1}^{l} \alpha_i \zeta_i + \frac{1}{\nu l} \sum_{i=1}^{l} \zeta_i - \sum_{i=1}^{l} \beta_i \zeta_i =$$

$$\sum_{i=1}^{l} \alpha_i \left[\left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right)^T \left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right) \right] +$$

$$+ \sum_{i=1}^{l} \left(\frac{1}{\nu l} - \alpha_i - \beta_i \right) \zeta_i = \sum_{i=1}^{l} \alpha_i \left[\left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right)^T \left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right) \right]$$



One class SVM

$$\sum_{i=1}^{l} \alpha_i \left[\left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right)^T \left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right) \right] =$$

$$\sum_{i=1}^{l} \alpha_i \left[\left(\phi^T(X_i) - \sum_{j=1}^{l} \alpha_j \phi^T(X_j) \right) \left(\phi(X_i) - \sum_{j=1}^{l} \alpha_j \phi(X_j) \right) \right] =$$

$$\sum_{i=1}^{l} \alpha_i \left(\phi^T(X_i) \phi(X_i) - 2 \sum_{j=1}^{l} \alpha_j \phi^T(X_j) \phi(X_i) + \sum_{j,k} \alpha_j \alpha_k \phi^T(X_j) \phi(X_k) \right) =$$

$$\sum_{i} \alpha_i K(X_i, X_i) - 2 \sum_{i,j} \alpha_i \alpha_j K(X_i, X_j) + \sum_{i,j} \alpha_i \alpha_j K(X_i, X_j) =$$

$$\sum_{i} \alpha_i K(X_i, X_i) - \sum_{i,j} \alpha_i \alpha_j K(X_i, X_j)$$

Which we want to maximize with respect to the αs

One class SVM

Thus: the dual objective function can be written using a kernel function:

$$\max_{\alpha} \sum_{i,j} \alpha_i \alpha_j k(X_i, X_j) - \sum_i \alpha_i k(X_i, X_i)$$

$$s.t. \quad 0 \le \alpha_i \le \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1$$

The solution of this optimization problem gives the optimal αs values

One class SVM

Note :

$$\|\phi(X) - c\|^2 = \left(\phi(X) - \sum_{j=1}^{l} \alpha_j \phi(X_j)\right)^T \left(\phi(X) - \sum_{j=1}^{l} \alpha_j \phi(X_j)\right) =$$

$$\left(\phi^T(X) - \sum_{j=1}^{l} \alpha_j \phi^T(X_j)\right)\left(\phi(X) - \sum_{j=1}^{l} \alpha_j \phi(X_j)\right) =$$

$$\phi^T(X)\phi(X) - 2\sum_{j=1}^{l} \alpha_j \phi^T(X_j)\phi(X) + \sum_{j,k} \alpha_j \alpha_k \phi^T(X_j)\phi(X_k) =$$

$$K(X,X) - 2\sum_{j} \alpha_j K(X_j, X) + \sum_{j,k} \alpha_j \alpha_k K(X_j, X_k)$$

One class SVM

To use the one-class SVM to rank images:

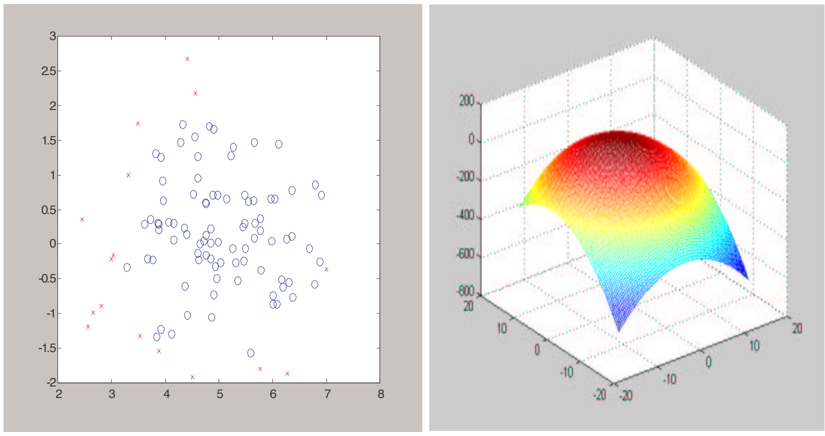
$$f(X) = R^2 - \|\phi(X) - c\|^2 =$$

$$R^2 - K(X,X) + 2\sum_j \alpha_j K(X_j, X) - \sum_{j,k} \alpha_j \alpha_k K(X_j, X_k)$$

The closer the image is to the center of the hyper-sphere, the higher is the score, and more likely the image is to be a target image.

Two nice toy examples

Linear One class SVM

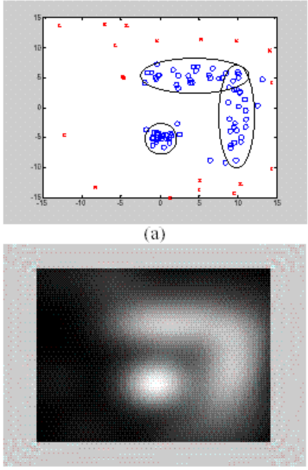


(a) (b)

In practice we cannot assume that real data are clustered in spherical shapes as in the previous example. Real data (e.g. images) can have multi-mode distributions.

The use of a kernel allows to handle the more general case. We look for spherical shapes in the transformed space.

Non-linear One class SVM



(a)

(b)

Real data

- Fully labeled image database;
- 5 classes with 100 images each;
- Classes: airplanes, cars, horses, eagles, stained glasses;
- Each image is a vector of 37 dimensions: statistical moments, edge-based structure features, etc;

Experiment with real data

- For each class, 10 images are randomly chosen as training examples;
- The learned decision function is used to rank all the 500 images in the database;
- The hit rates in the top ranked 20 and 100 images are used as performance measures;
- For each class, the experiment is repeated 100 times, and average error rates are reported.

Experimental Results

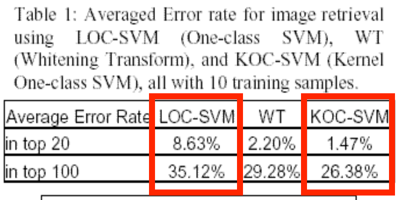
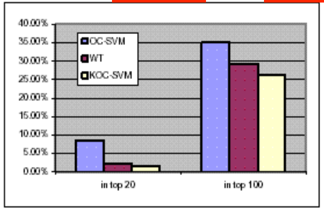


Table 1: Averaged Error rate for image retrieval using LOC-SVM (One-class SVM), WT (Whitening Transform), and KOC-SVM (Kernel One-class SVM), all with 10 training samples.

Average Error Rate	LOC-SVM	WT	KOC-SVM
in top 20	8.63%	2.20%	1.47%
in top 100	35.12%	29.28%	26.38%



Conclusions

- Effective training was performed with a small number of examples;
- A Gaussian kernel was used: how does it compare with using different kernels?
- The method requires the tuning of two parameters: the spread of the Gaussian kernel, and the regularization term for errors.