

**Unlabeled Data Classification**  
**by**  
**Support Vector Machines**

**Glenn Fung & Olvi L. Mangasarian**  
**University of Wisconsin – Madison**

[www.cs.wisc.edu/~olvi](http://www.cs.wisc.edu/~olvi)

[www.cs.wisc.edu/~gfung](http://www.cs.wisc.edu/~gfung)

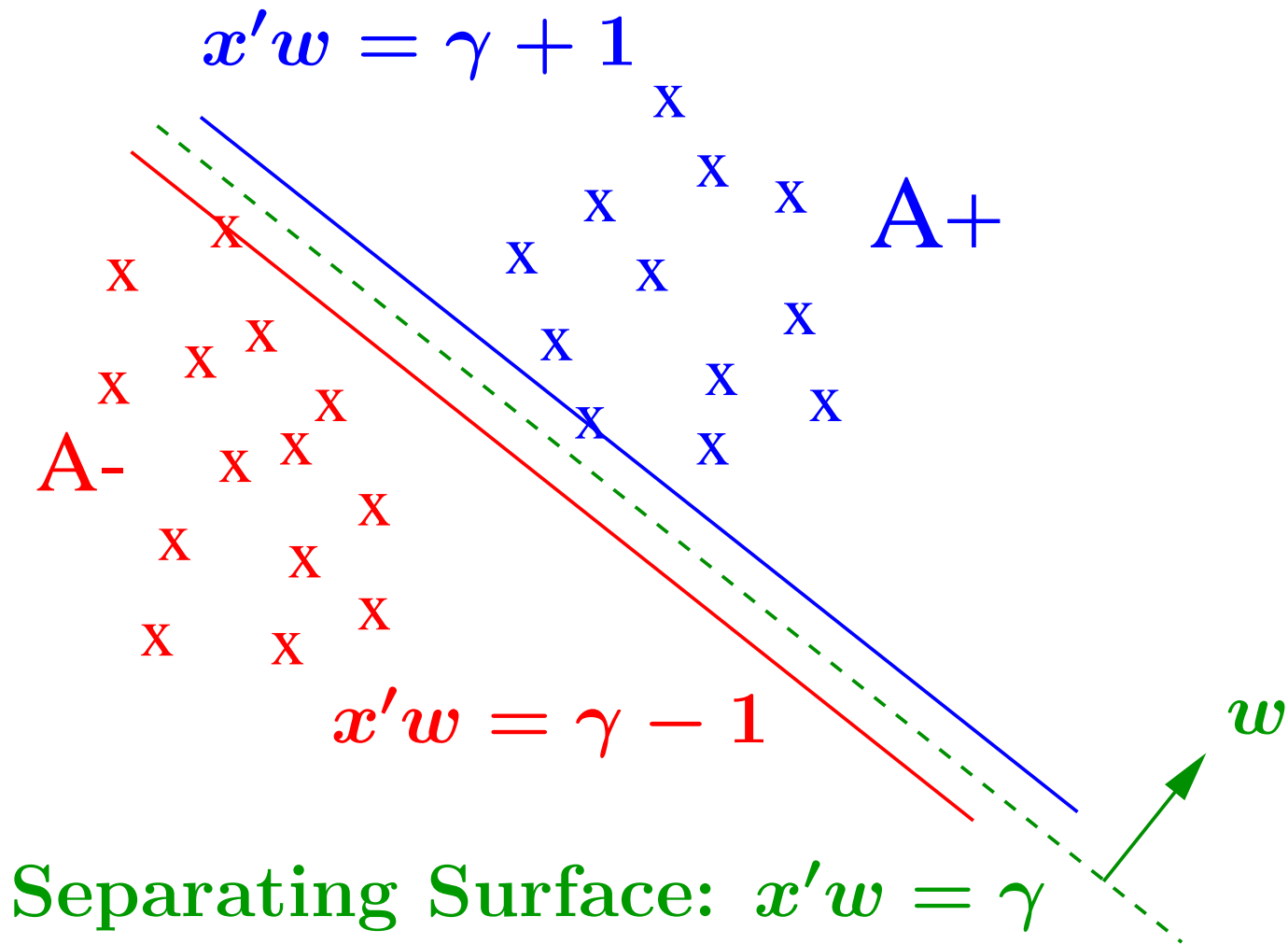
## The General Problem

- Given:
  - Points labeled **Benign** & **Malignant**
  - **Unlabeled** points
- How should we label the **unlabeled** points so as to get a “best” overall separation between **Benign** & **Malignant** points?
- How should we proceed if all the points are **unlabeled**?

## Outline

- SVM: The linear Support Vector Machine for labeled data
- S<sup>3</sup>VM: Semi-Supervised SVM [Bennett & Demiriz]
  - An integer programming approach to partially labeled data
- VS<sup>3</sup>VM: Concave S<sup>3</sup>VM for partially labeled data
- k-median Clustering
- CVS<sup>3</sup>VM: Clustering for VS<sup>3</sup>VM for unlabeled data
- Numerical Tests:
  - Higher test set accuracy on 9 public datasets (CVS<sup>3</sup>VM) compared to randomly selected & labeled training set
- Conclusions

Geometry of the Classification Problem  
The Fundamental 2-Category Linearly Separable Case



# Algebra of the Classification Problem

## The Fundamental 2-Category Linearly Separable Case

- Given  $m$  points in the  $n$  dimensional real space  $R^n$
- Represented by an  $m \times n$  matrix  $A$
- Membership of each point  $A_i$  in the classes 1 or -1 is specified by:
  - An  $m \times m$  diagonal matrix  $D$  with  $\pm 1$  along its diagonal
- Separate by two bounding planes:  $x'w = \gamma \pm 1$  such that:

$$A_i w \geq \gamma + 1, \text{ for } D_{ii} = +1,$$

$$A_i w \leq \gamma - 1, \text{ for } D_{ii} = -1.$$

- More succinctly:

$$D(Aw - e\gamma) \geq e,$$

where  $e$  is a vector of ones.

## Robust Linear Programming (RLP)

Preliminary Approach to the (Linear) Support Vector Machine:

Solve the following mathematical program:

$$\begin{array}{ll} \min_{w, \gamma, y} & e'y \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e \\ & y \geq 0, \end{array} \quad (RLP)$$

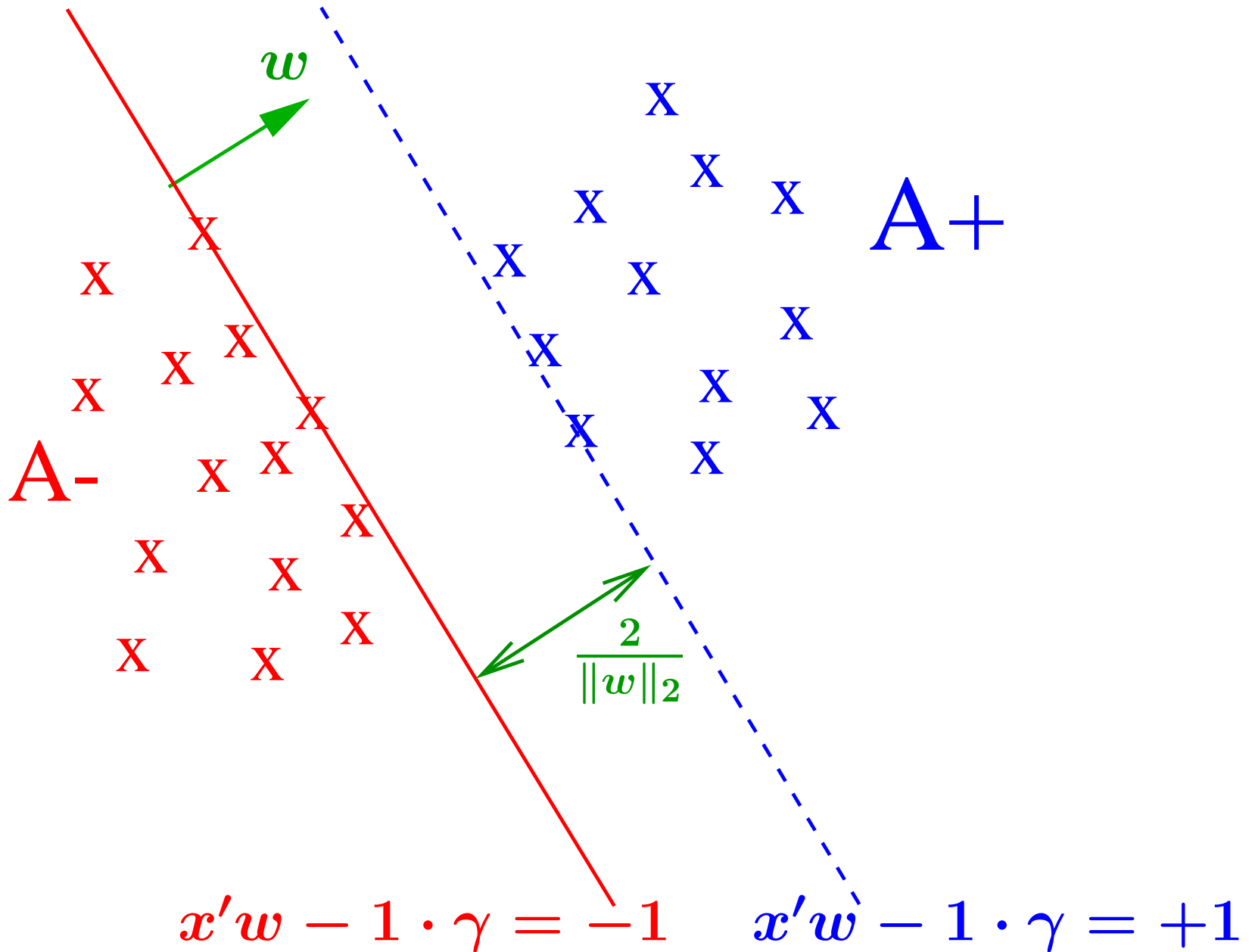
where:

$y :=$  nonnegative error (slack) vector

Note:  $y = 0$  iff convex hulls of  $A_+$  and  $A_-$  do not intersect.

# The (Linear) Support Vector Machine

## Maximize Margin between Separating Planes



## The (Linear) Support Vector Machine Formulation Linear Programming Formulation

Solve the following mathematical program for some  $\mu > 0$ :

$$\begin{aligned} & \min_{w, \gamma, y, z} e' y + \mu \|w\|_1 \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \\ & y \geq 0. \end{aligned}$$

Which is equivalent to the LP:

$$\begin{aligned} & \min_{w, \gamma, y, z} e' y + \mu e' z \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \\ & -z \leq w \leq z \\ & y \geq 0. \end{aligned} \tag{LP}$$

Other norms and functions lead to other formulations:

$SVM_{\|\cdot\|_2}$ ,  $SVM_{\|\cdot\|_2^2}$  &  $FSV$ .

## Semi-Supervised Support Vector Machines (S<sup>3</sup>VM)

We consider here data consisting of  $m$  labeled points and  $p$  unlabeled points all in  $R^n$ .

- $m$  labeled points represented by  $A \in R^{m \times n}$
- $p$  unlabeled points represented by  $B \in R^{p \times n}$ .

Bennett and Demiriz formulate the semi-supervised support vector machine for this problem as follows:

$$\begin{aligned} & \min_{w, \gamma, y, z, r, s} e' y + \mu e' z + \nu e' \min \{r, s\} \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \\ & -z \leq w \leq z \\ & Bw - e\gamma + r \geq e \\ & -Bw + e\gamma + s \geq e \\ & y \geq 0, r \geq 0, s \geq 0. \end{aligned}$$

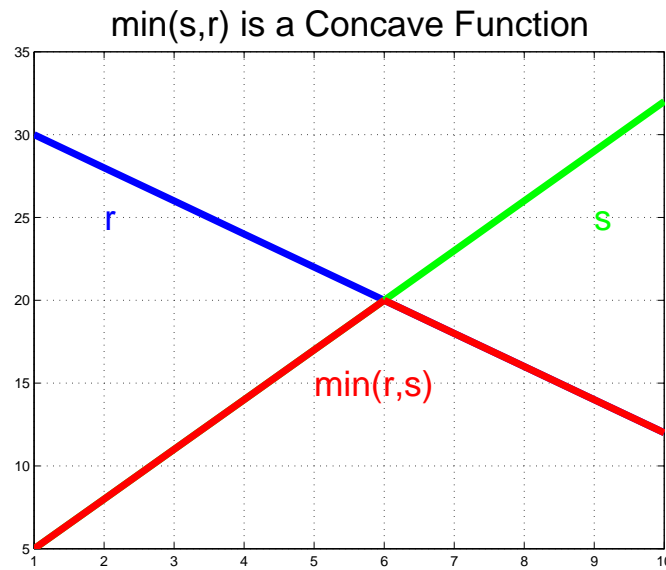
## S<sup>3</sup>VM: A Mixed Integer Program Approach (MIP)

Bennett and Demiriz (1998), formulate this problem as mixed integer program (MIP) by assigning a binary decision variable to each row of the unlabeled matrix.

- Only relatively **small** unlabeled data can be handled.
- Solver sometimes fails due to **excessive branching**.
- Reaching an optimal solution may take **long time**.

## VS<sup>3</sup>VM: A Concave S<sup>3</sup>VM

The term  $\min\{r, s\}$  in the objective function of the S<sup>3</sup>VM formulation is concave because it is the minimum of two linear functions.



### Sketch of Algorithm

- Linearize  $\min\{r, s\}$  around the current iterate  $(r^i, s^i)$ 
  - Take a supporting plane (generalization of a tangent plane for a nondifferentiable concave function) as an approximation to the function at  $(r^i, s^i)$
- Solve the resulting linear program

## VS<sup>3</sup>VM: Successive Linear Approximation for S<sup>3</sup>VM

Choose positive values for the parameters  $\mu, \nu$ . Start with a random  $(r^0, s^0) \geq 0$ .

Having  $(r^i, s^i)$  determine  $(w^{i+1}, \gamma^{i+1}, y^{i+1}, z^{i+1}, r^{i+1}, s^{i+1})$  by solving the linear program:

$$\begin{array}{ll}
 \min_{w, \gamma, y, z, r, s} & e'y + \mu e'z + \nu \partial(e' \min \{r^i, s^i\}) \begin{bmatrix} r - r^i \\ s - s^i \end{bmatrix} \\
 \text{s.t.} & D(Aw - e\gamma) + y \geq e \\
 & -z \leq w \leq z \\
 & Bw - e\gamma + r \geq e \\
 & -Bw + e\gamma + s \geq e \\
 & y \geq 0, r \geq 0, s \geq 0,
 \end{array}$$

where  $\partial(e' \min \{r^i, s^i\})$  denotes the supergradient of  $e' \min \{r, s\}$  at  $\{r^i, s^i\}$ .

## Stopping Criterion

**Stop** when the following necessary optimality condition holds:

$$e'(y^{i+1} - y^i) + \mu e'(z^{i+1} - z^i) + \nu \partial(e' \min \{r^i, s^i\}) \begin{bmatrix} r^{i+1} - r^i \\ s^{i+1} - s^i \end{bmatrix} = 0.$$

The Algorithm terminates after a **finite** number of linear programs at a point satisfying the **necessary optimality condition** for the original problem.

For a concave function  $f : R^n \rightarrow R$  the supergradient  $\partial(f(x))$  of  $f$  at  $x$  is a vector in  $R^n$  satisfying:

$$f(y) - f(x) \leq \partial f(x)(y - x),$$

for all  $y \in R^n$ . The supergradient reduces to the ordinary gradient  $\nabla f(x)$ , when  $f$  is differentiable at  $x$ .

## Supergradient of $e' \min \{r, s\}$

In our case  $e' \min \{r, s\} : R^{2p} \longrightarrow R$  is a non-differentiable concave function and its supergradient is given by:

$$\partial(e' \min \{r, s\}) = \sum_{j=1}^p \left\{ \begin{array}{ll} \begin{pmatrix} I_j \\ 0_p \end{pmatrix} & \text{if } r_j < s_j \\ (1 - \lambda) \begin{pmatrix} I_j \\ 0_p \end{pmatrix} + \lambda \begin{pmatrix} 0_p \\ I_j \end{pmatrix} & \text{if } r_j = s_j \\ \begin{pmatrix} 0_p \\ I_j \end{pmatrix} & \text{if } r_j > s_j \end{array} \right.$$

Here  $0_p \in R^p$  is a vector of zeros,  $I_j \in R^p$  is a vector with 1 in position  $j$  and 0 elsewhere, and  $\lambda \in [0, 1]$ . In all our computations we set  $\lambda = 0.5$ .

## Initial Point $(r^0, s^0)$

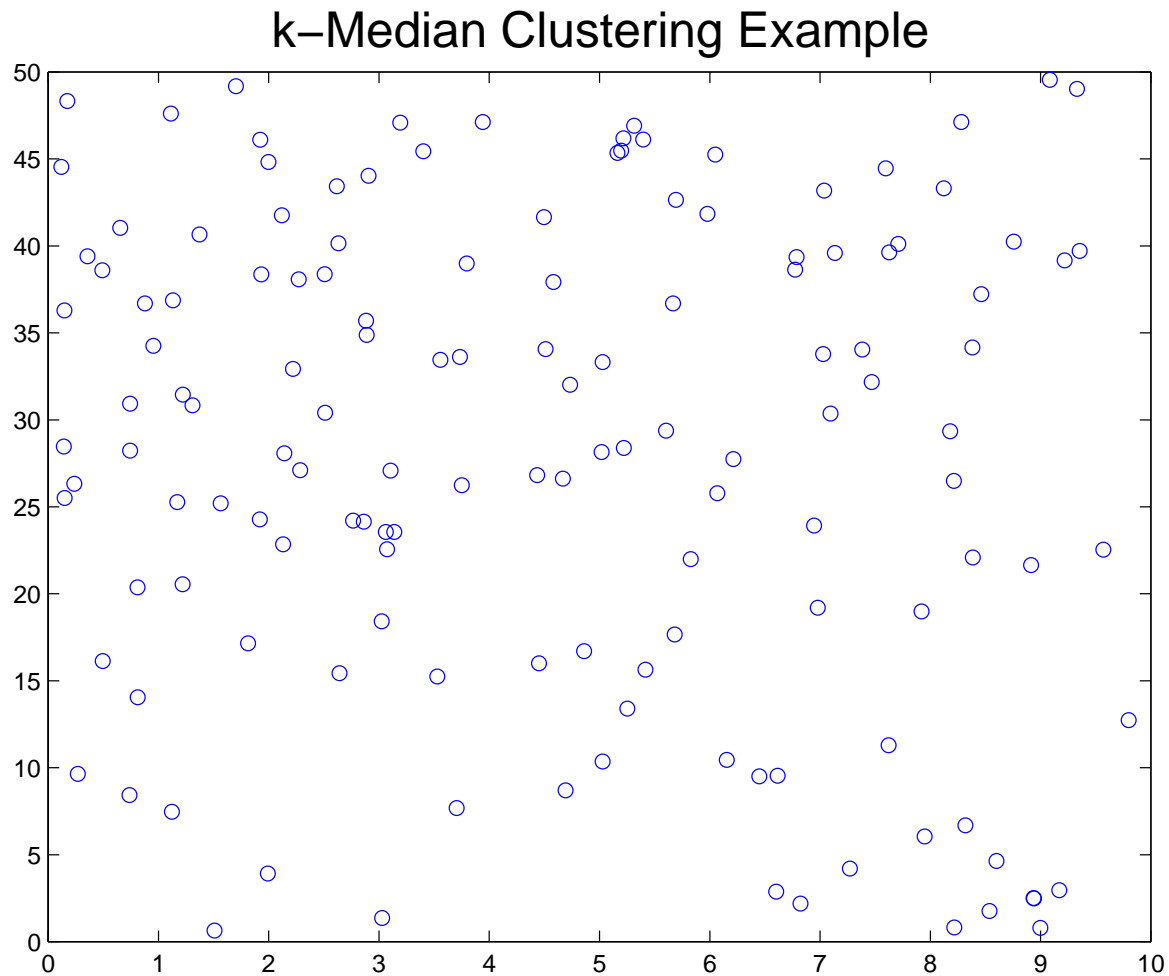
Our numerical experiments showed that instead of a random starting point  $(r^0, s^0) \geq 0$ , a much better starting point for the Algorithm can be obtained by solving the following linear program:

$$\begin{aligned} & \min_{w, \gamma, y, z, r, s} e'y + \mu e'z + \frac{\nu}{2}(e'(r + s)) \\ \text{s.t.} \quad & D(Aw - e\gamma) + y \geq e \\ & -z \leq w \leq z \\ & Bw - e\gamma + r \geq e \\ & -Bw + e\gamma + s \geq e \\ & y \geq 0, r \geq 0, s \geq 0, \end{aligned}$$

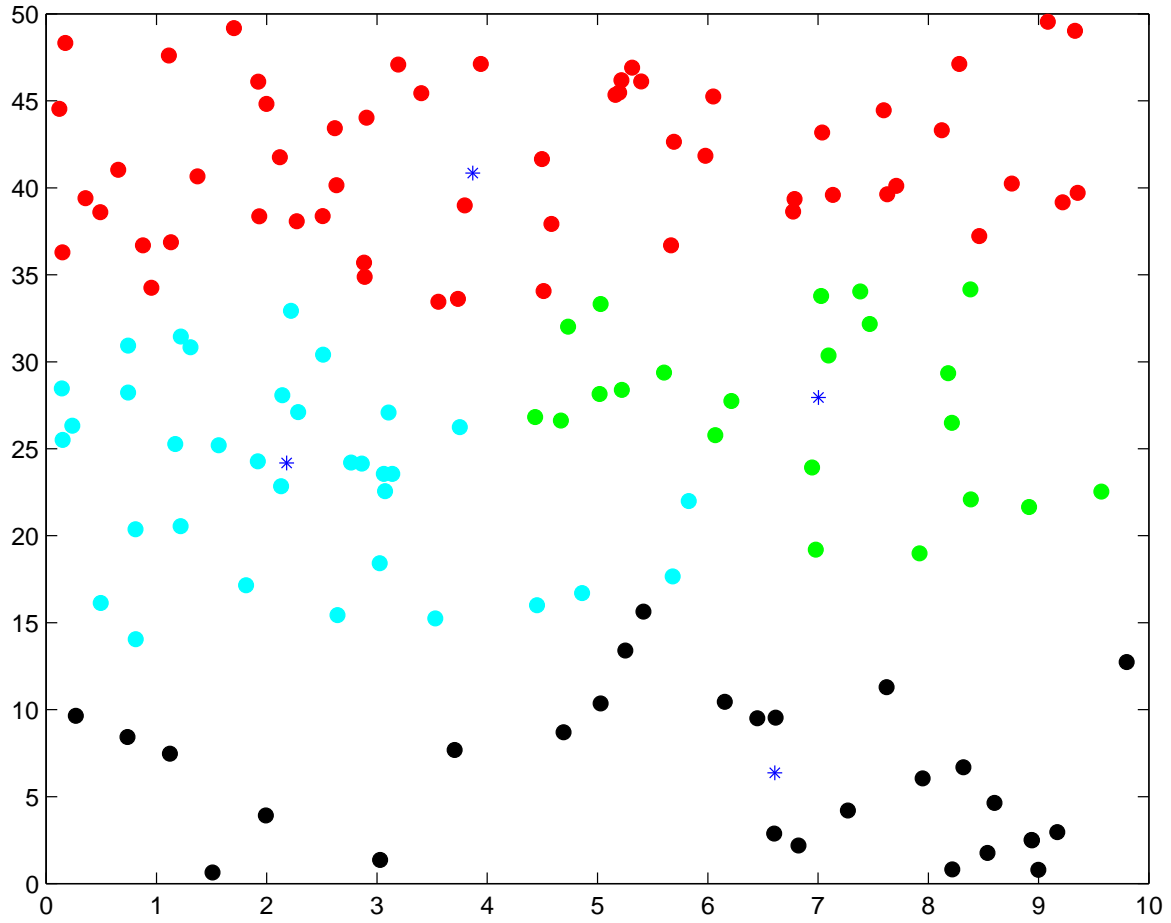
This corresponds to a regular iteration of the Algorithm with a supergradient of  $e' \min\{r, s\}$  evaluated at  $r = s$  with  $\lambda = 0.5$ .

## k-Median Clustering

Consider a set of  $m$  data points in  $R^n$  represented by the matrix  $A \in R^{m \times n}$ . We wish to find  $k$  clusters of this data such that the the sum of 1-norm distances from each point to its closest cluster center  $C_l, l = 1, \dots, k$ , is minimized.



### k-Median Clustering Example



## k-Median Algorithm

Given  $C_1^j, \dots, C_k^j$  at iteration  $j$ , compute  $C_1^{j+1}, \dots, C_k^{j+1}$  by the following two steps:

- (a) **Cluster Assignment:** For each  $A'_i$ ,  $i = 1, \dots, m$ , determine  $\ell(i)$  such that  $C_{\ell(i)}^j$  is closest to  $A'_i$  in the one norm.
- (b) **Cluster Center Update:** For  $\ell = 1, \dots, k$  choose  $C_\ell^{j+1}$  as a median of all  $A'_i$  assigned to  $C_\ell^j$ .

**Stop** when  $C_\ell^{j+1} = C_\ell^j$ .

## Unlabeled Data Clustering for VS<sup>3</sup>VM

Suppose now:

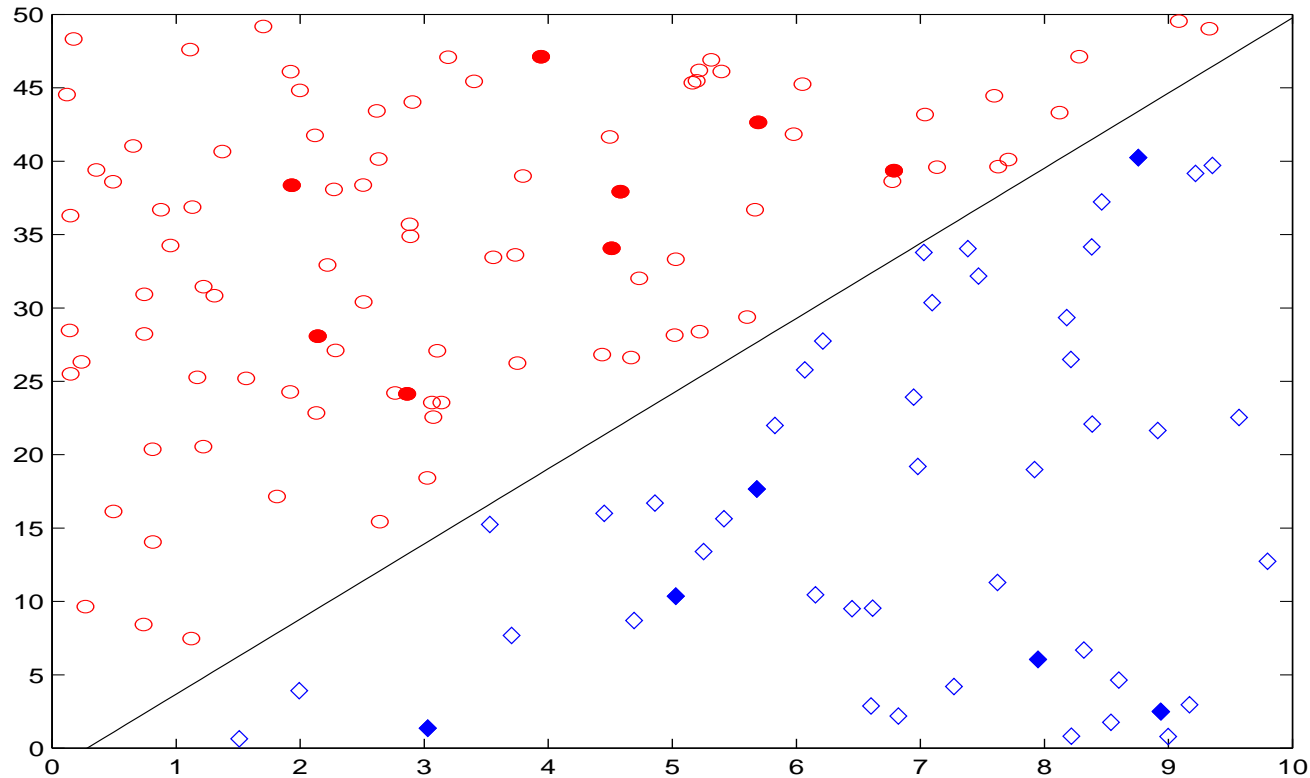
- All our data is unlabeled
- Relatively few points can be labeled through expensive or time consuming services of an expert or an oracle

Our approach here will consist of the following:

- Use k-Median clustering to find k cluster centers
- Select a small subset surrounding each cluster center to be labeled by oracle or expert.
- Give the resulting labeled-unlabeled dataset to VS<sup>3</sup>VM Algorithm.

We call this approach: CVS<sup>3</sup>VM: Clustered VS<sup>3</sup>VM

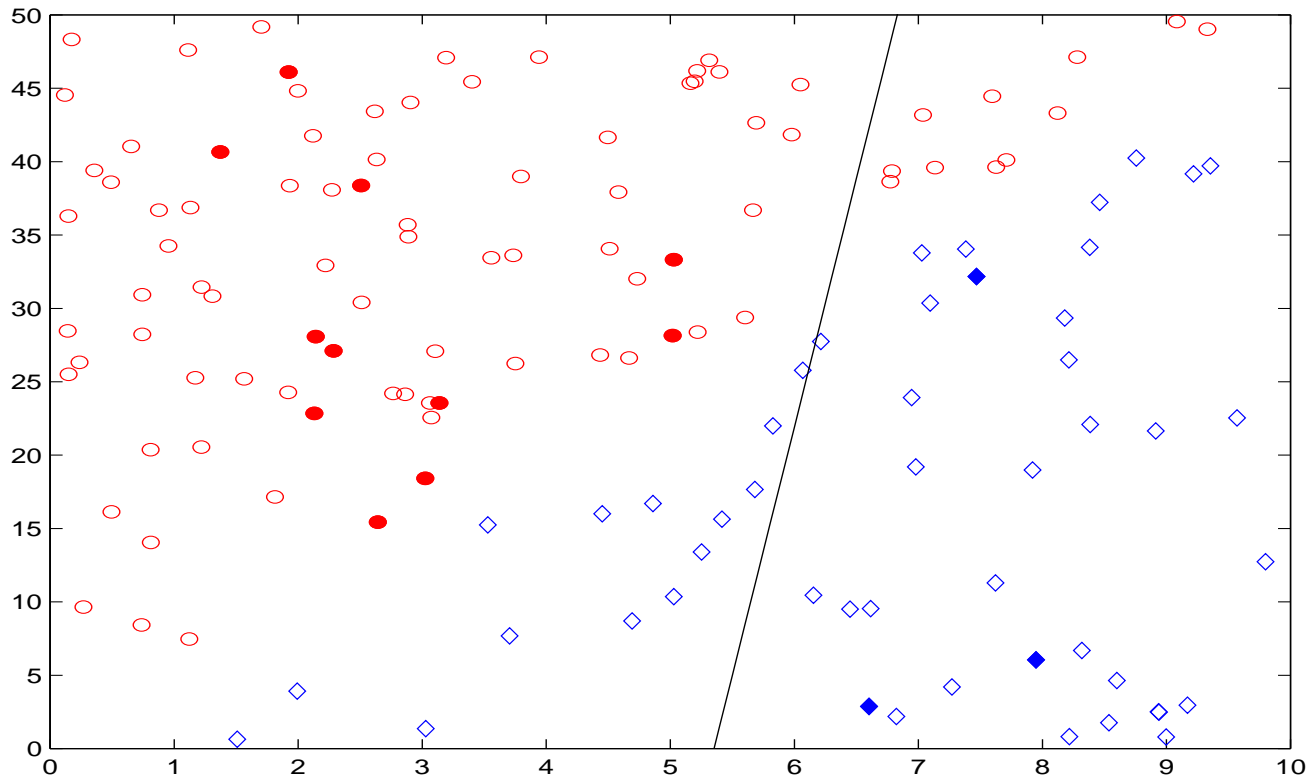
## CVS<sup>3</sup>VM Graphic Example



### CVS<sup>3</sup>VM for Unlabeled Data:

- Solid shapes, 10% of dataset, whose labels, diamonds and circles, are unknown to the  $k$ -median algorithm, which selected them as two clusters
- Expert determines shape of solid points
- VS<sup>3</sup>VM uses 10% as labeled data and 90% as unlabeled
- Resulting separating plane correctly classifies all points in both classes.

## CV<sup>3</sup>VM Graphic Example



### VS<sup>3</sup>VM for Unlabeled Data:

- Same example with the solid 10% shapes randomly selected to be used as a labeled training set in the robust linear programming (RLP) algorithm
- Resulting separating plane incorrectly classifies 18% of the data.

## Numerical Tests & Comparisons

			% Test Correctness		
Dataset	Dimension	Points	CVS <sup>3</sup> VM	10% RLP	Full RLP
Galaxy Bright	14	2462	98.0	97.7	98.3
Cancer Diagnosis	9	683	95.7	93.4	96.7
Cancer Prognosis	30	569	94.6	90.9	95.4
Heart	13	297	78.3	70.4	82.8
Housing	13	506	85.8	79.4	84.8
Ionosphere	34	351	83.9	78.0	88.2
Musk	166	476	69.8	66.0	82.5
Pima	8	769	74.2	73.4	77.4
Sonar	60	208	77.1	64.0	77.4

- **CVS<sup>3</sup>VM:** 10% of data picked by clustering to be labeled by an expert, remaining 90% remains unlabeled.
- **10% RLP:** Ten-fold testing set correctness of plain linear programming approach RLP with a randomly chosen 10% labeled training set.
- **Full RLP:** Ten-fold testing set correctness of plain linear programming approach RLP with full labeled training set.

## Conclusions

- $VS^3VM$ : A new formulation of semi-supervised support vector machines as a concave minimization problem.
- $VS^3VM$  can handle much larger unlabeled datasets than a MIP-based  $S^3VM$
- For totally unlabeled data,  $CVS^3VM$  combines k-median clustering and  $VS^3VM$  to obtain better results than training on a randomly selected set that is labeled by an expert.
- Future directions:
  - $VS^3VM$  using kernels for nonlinear separation.
  - Multicategory unlabeled data classification.
  - $VS^3VM$  for incremental data mining where a small portion of the dataset is labeled dynamically as new data becomes available.