ISTA 421 + INFO 521
Introduction to Machine Learning

Lecture 22:
Support Vector Machines - I

Clay Morrison
claytonm@email.arizona.edu
Harvill 437A
Phone 621-6609

13 November 2017

1

Support Vector Machines

(SVMs)

2

Support Vector Machines (SVMs)

- Considered one of the best "off-the-shelf" classifiers for many problems – state of the art.
- **BUT**, "No free lunch": not guaranteed the best
  - Wolpert & Macready 1997
  - "…any two optimization algorithms are equivalent when their performance is averaged across all possible problems." (from 2005)
- SVMs are particularly useful in applications where the number of attributes is **much larger** than the number of training objects
  - Number of parameters is based on the number of training objects, not the number of attributes!

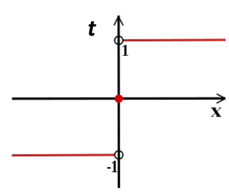
Support Vector Machines (SVMs)

- Standard SVM uses linear decision boundary given by: $\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\mathsf{new}} + b$
- SVM ***decision function*** for test point:

$$t_{\mathsf{new}} = \mathrm{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\mathsf{new}} + b)$$

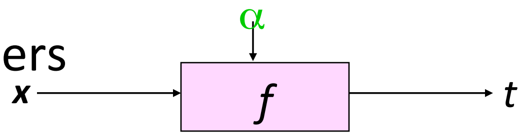
labels are $\{1, -1\}$ rather than $\{0, 1\}$

$$\mathrm{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$



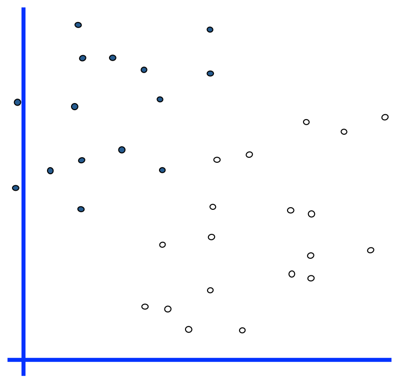
- **Goal**: find **w** and *b* based on training data
- **Criteria**: Maximize the ***margin***

Linear Classifiers



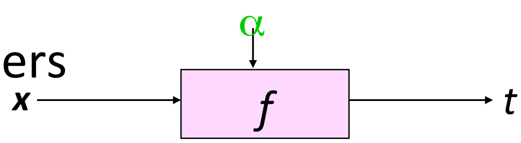
$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$

- denotes +1
- denotes -1



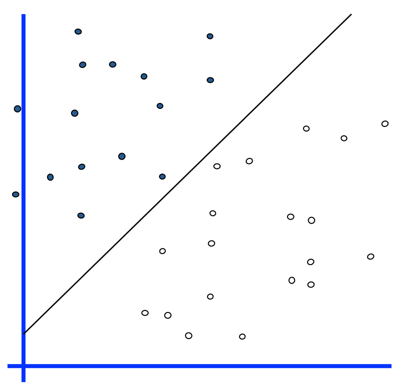
How would you classify this data?

Linear Classifiers



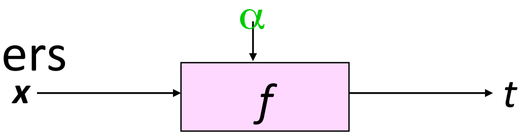
$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$

- denotes +1
- denotes -1



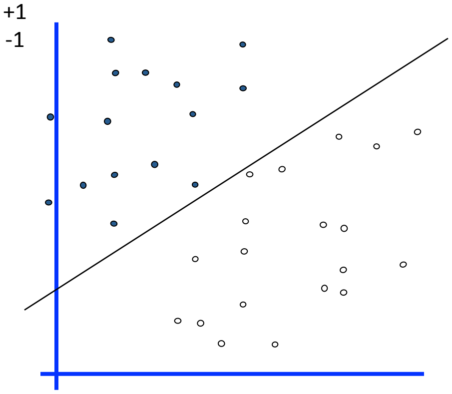
How would you classify this data?

Linear Classifiers



$$t_{new} = \text{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$$

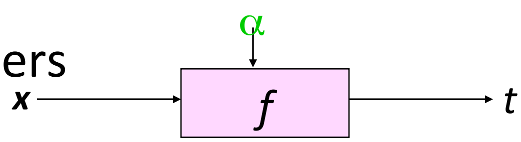
- denotes +1
- denotes -1



How would you classify this data?

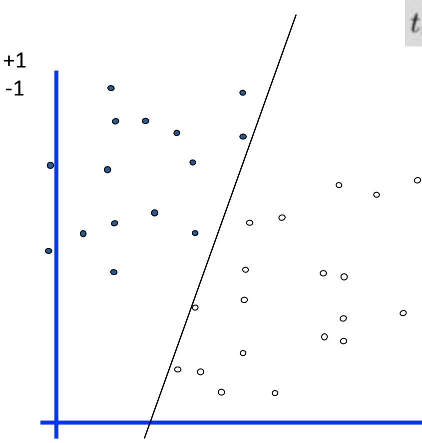
SISTA 7

Linear Classifiers



$$t_{new} = \text{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$$

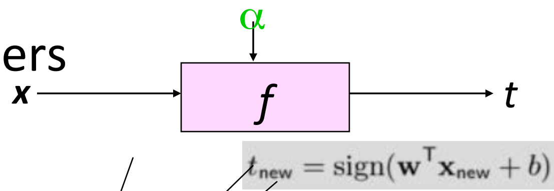
- denotes +1
- denotes -1

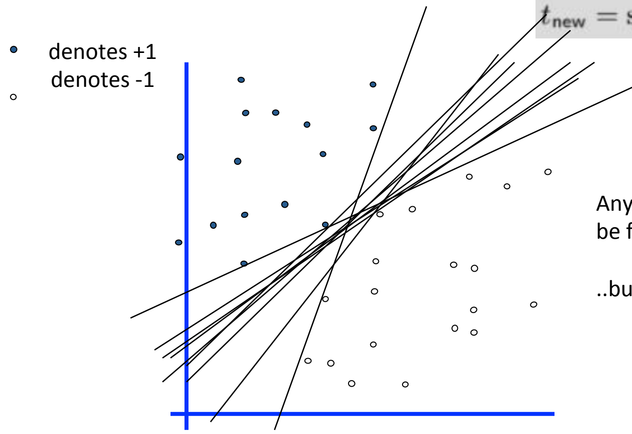


How would you classify this data?

SISTA 8

Linear Classifiers

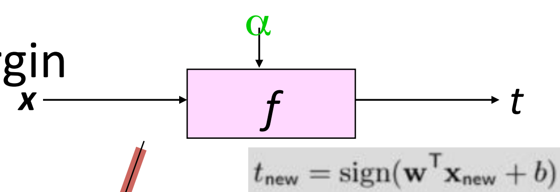


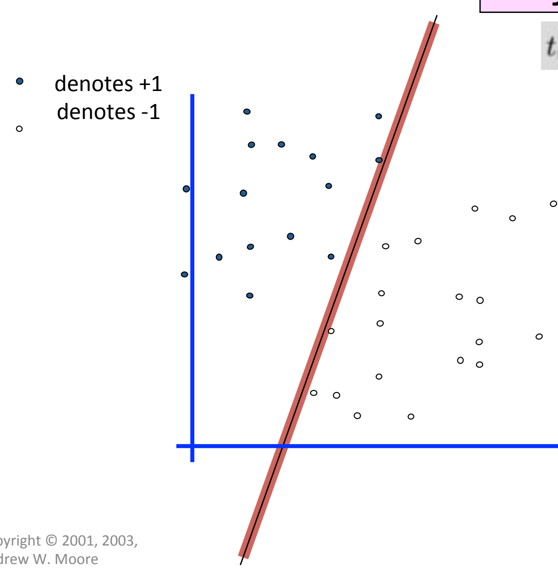


Any of these would be fine..

..but which is best?

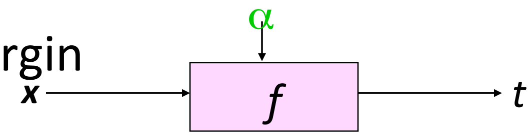
Classifier Margin



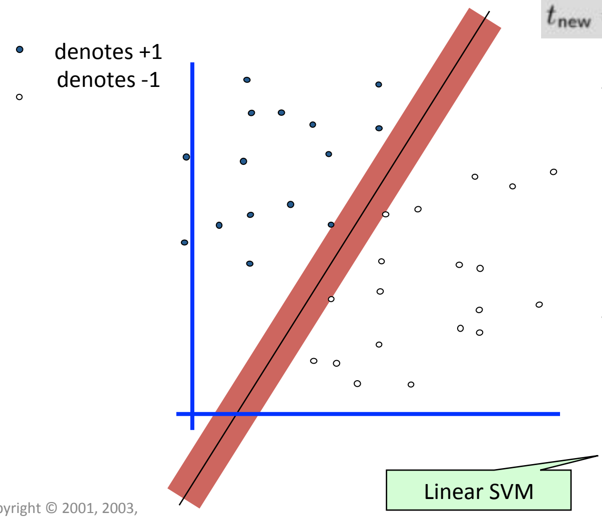


Define the margin of a linear classifier as the width that the boundary (separating the classes) could be increased by before hitting a datapoint.

Maximum Margin

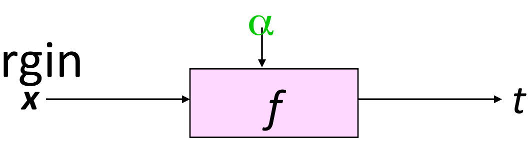


$$t_{new} = \mathrm{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$$

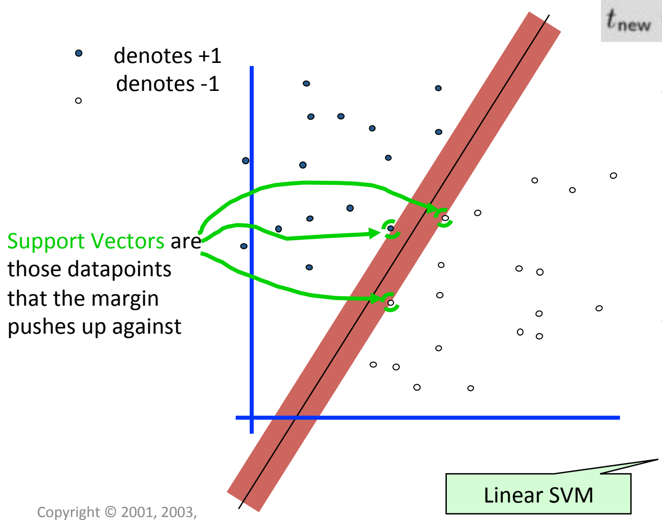


The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

Maximum Margin



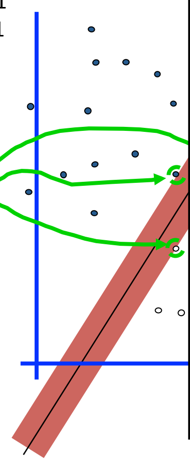
$$t_{new} = \mathrm{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$$



The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

Why Maximum Margin?

denotes +1
denotes -1



1. Intuitively this feels safest.
2. If we've made a small error in the location of the boundary (it's been jolted in its perpendicular direction) this gives us least chance of causing a misclassification.
3. LOOCV is easy since the model is immune to removal of any non-support-vector datapoints.
4. There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
5. Empirically it works very very well.

SISTA 13

Recall: Relation of a^Tb to Geometry

- $\mathbf{a}^T\mathbf{b}$ is special (also $\mathbf{a}\cdot\mathbf{b}$), called the *dot product*

$$\mathbf{a}\cdot\mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

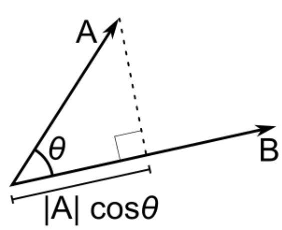
(This is actually a species of a more general operation called an *inner product*)

- Plays a role in defining
  - Euclidean vector length (norm)

$$\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}\cdot\boldsymbol{x}}.$$

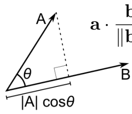
  - Angles

$$\mathbf{a}\cdot\mathbf{a} = \|\mathbf{a}\|^2$$
$$\mathbf{a}\cdot\mathbf{b} = \|\mathbf{a}\|\ \|\mathbf{b}\|\cos\theta$$
$$\theta = \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\ \|\mathbf{b}\|}\right).$$



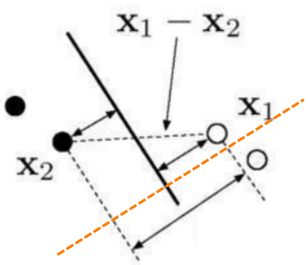
The dot (inner) project is therefore a measure of the length of vector **a** when we *project* it onto **b** (and normalize by the length (norm) of **b**):

$$\mathbf{a}\cdot\frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{1}{\|\mathbf{b}\|}\|\mathbf{a}\|\|\mathbf{b}\|\cos\theta = \|\mathbf{a}\|\cos\theta$$



$\mathbf{a} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\| \cos\theta \qquad \|\boldsymbol{x}\| := \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}.$

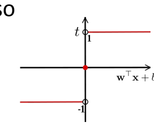
Defining the Margin



w is in the direction **perpendicular** to the boundary.
Normalize it to get the "unit vector": $\frac{\mathbf{w}}{\|\mathbf{w}\|}$

Our "*decision function*", which we will also refer to as *D*(*x*):

$$t_{\text{new}} = \text{sign}\ (\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$$



Another complication:

We note that the argument in $D(x)$ is invariant under a rescaling: $\mathbf{w} \to \lambda\mathbf{w}$, $b \to \lambda b$.

We will implicitly fix a scale with:

$$\mathbf{w} \cdot \mathbf{x_1} + b = 1$$
$$\mathbf{w} \cdot \mathbf{x_2} + b = -1$$

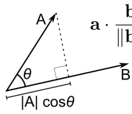
for the support vectors (*canonical hyperplanes*).

Combine both **constraints** by subtracting one from the other, to get the following:

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

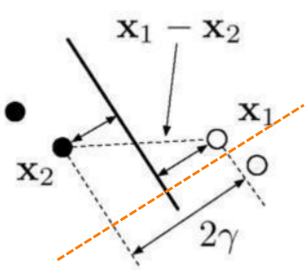
The margin (the length of the distance between the support vector canonical hyperplanes) will be given by the ***projection*** of the vector ($x_1 - x_2$) onto the normal vector to the hyperplane!
The projection is accomplished by taking the inner product of these two quantities

SISTA 15



$\mathbf{a} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\| \cos\theta \qquad \|\boldsymbol{x}\| := \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}.$

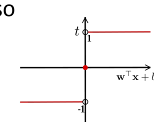
Defining the Margin



w is in the direction **perpendicular** to the boundary.
Normalize it to get the "unit vector": $\frac{\mathbf{w}}{\|\mathbf{w}\|}$

Our "*decision function*", which we will also refer to as *D*(*x*):

$$t_{\text{new}} = \text{sign}\ (\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$$



Another complication:

We note that the argument in $D(x)$ is invariant under a rescaling: $\mathbf{w} \to \lambda\mathbf{w}$, $b \to \lambda b$.

We will implicitly fix a scale with:

$$\mathbf{w} \cdot \mathbf{x_1} + b = 1$$
$$\mathbf{w} \cdot \mathbf{x_2} + b = -1$$

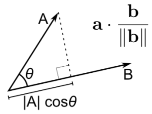
for the support vectors (*canonical hyperplanes*).

Combine both **constraints** by subtracting one from the other, to get the following:

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

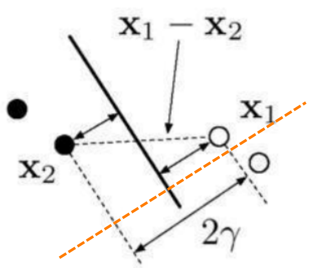
The margin (the length of the distance between the support vector canonical hyperplanes) will be given by the ***projection*** of the vector ($x_1 - x_2$) onto the normal vector to the hyperplane!
The projection is accomplished by taking the inner product of these two quantities

SISTA 16



$\mathbf{a} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\| \cos\theta \quad \|\boldsymbol{x}\| := \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}.$

Defining the Margin



$$
\begin{aligned}
2\gamma &= \frac{1}{||\mathbf{w}||}\mathbf{w}^{\mathsf{T}}(\mathbf{x}_1 - \mathbf{x}_2) \\
&= \frac{1}{||\mathbf{w}||}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_1 - \mathbf{w}^{\mathsf{T}}\mathbf{x}_2) \\
&= \frac{1}{||\mathbf{w}||}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_1 + b - \mathbf{w}^{\mathsf{T}}\mathbf{x}_2 - b) \\
&= \frac{1}{||\mathbf{w}||}(1 + 1) \\
\gamma &= \frac{1}{||\mathbf{w}||}.
\end{aligned}
$$

w is in the direction **perpendicular** to the boundary.
Normalize it to get the "unit vector": $\frac{\mathbf{w}}{||\mathbf{w}||}$

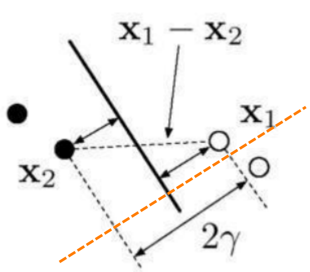
Constraints:
$\mathbf{w} \cdot \mathbf{x_1} + b = 1$
$\mathbf{w} \cdot \mathbf{x_2} + b = -1$
$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$

The margin (the length of the distance between the support vector canonical hyperplanes) will be given by the ***projection*** of the vector ($x_1 - x_2$) onto the normal vector to the hyperplane!
The projection is accomplished by taking the inner product of these two quantities

SISTA 17

Maximizing the Margin

$$\gamma = \frac{1}{||\mathbf{w}||}$$



Recall, we set a scale for the closest points to the margin:

$\mathbf{w} \cdot \mathbf{x_1} + b = 1$ For the s.v. class 1
$\mathbf{w} \cdot \mathbf{x_2} + b = -1$ For the s.v. class 2

Therefore, **w** must be chosen such that:

$$
\begin{aligned}
&\mathbf{w} \cdot \mathbf{x_n} + b \geq 1 \quad \text{for all } \mathbf{x_n} \text{ in class 1} \\
&\mathbf{w} \cdot \mathbf{x_n} + b \leq -1 \quad \text{for all } \mathbf{x_n} \text{ in class 2}
\end{aligned}
$$

Combining both constraints is easy:

$t_n(\mathbf{w}^{\top}\mathbf{x}_n + b) \geq 1$ For all $\mathbf{x}_n$

There are a total of N constraints

Easier to *minimize* $\frac{1}{2}||\mathbf{w}||^2$

SISTA 18

Constrained Optimization with Lagrange (KKT) Multipliers

- Find values of a set of parameters that maximize (or minimize) an objective function, but also satisfy some constraints.
- Create new objective function that includes the original plus an additional term for each constraint.

For example, minimize $f(x)$ subject to the constraint $g(w) \leq a$

$$\underset{w}{\operatorname{argmin}} \quad f(w)$$
$$\text{subject to} \quad g(w) \leq a$$

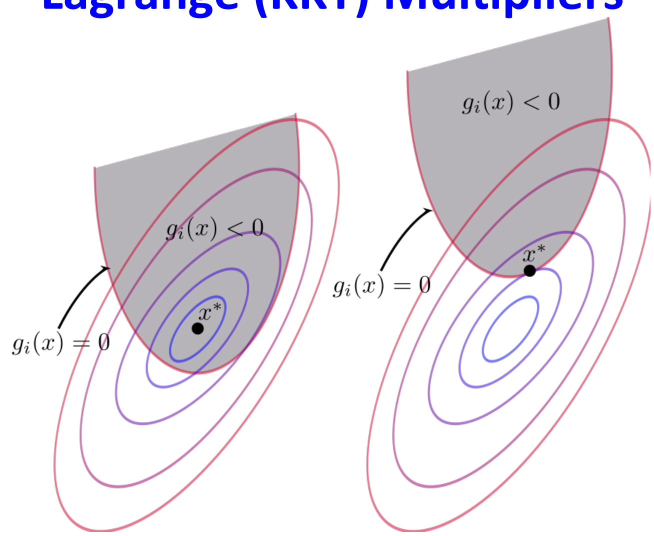
Add Lagrange term of the form $\lambda(a - g(w))$ and optimize for w and λ

$$\underset{w,\lambda}{\operatorname{argmin}} \quad f(w) - \lambda(a - g(w))$$
$$\text{subject to} \quad \lambda > 0$$

(Strictly speaking, Lagrange multipliers are just for equality constraints, whereas constrained optimization problems with inequalities involve KKT (Karush-Kuhn-Tucker) conditions; so you may see the lambda above referred to as a “KKT” multiplier…)

19

Constrained Optimization with Lagrange (KKT) Multipliers



(Shamelessly cribbed from Wikipedia…)

20

Maximizing the Margin $\gamma = \frac{1}{\|\mathbf{w}\|}$

$\frac{1}{2}\|\mathbf{w}\|^2 \qquad t_n(\mathbf{w}^\top \mathbf{x}_n + b) \geq 1$

Maximizing the margin, γ, becomes a ***constrained optimization*** problem, namely, to minimize the following:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to} \quad t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1, \ \text{for all } n$$

We can incorporate the inequalities into the minimization by introducing ***Lagrange multipliers***, resulting in the following:

note: $||\mathbf{w}||^2 = \mathbf{w}^\top \mathbf{w}$

$$\underset{\mathbf{w},\alpha}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} - \sum_{n=1}^{N} \alpha_n(t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1)$$
$$\text{subject to} \quad \alpha_n \geq 0, \ \text{for all } n,$$

Recall how we maximize/minimize!

$$\frac{\partial}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n t_n \mathbf{x}_n$$
$$\frac{\partial}{\partial b} = -\sum_{n=1}^{N} \alpha_n t_n.$$

Set to 0!

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n t_n \mathbf{x}_n$$
$$\sum_{n=1}^{N} \alpha_n t_n = 0$$

These two identities must be satisfied at the optimum

Maximizing the Margin $\gamma = \frac{1}{\|\mathbf{w}\|}$

$\frac{1}{2}\|\mathbf{w}\|^2 \qquad t_n(\mathbf{w}^\top \mathbf{x}_n + b) \geq 1$

$$\underset{\mathbf{w},\alpha}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} - \sum_{n=1}^{N} \alpha_n(t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1)$$
$$\text{subject to} \quad \alpha_n \geq 0, \ \text{for all } n,$$

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n t_n \mathbf{x}_n$$
$$\sum_{n=1}^{N} \alpha_n t_n = 0$$

Plug constraint for **w** back into the objective function, to get:

$$\frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} - \sum_{n=1}^{N} \alpha_n(t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1)$$
$$= \frac{1}{2}\left(\sum_{m=1}^{N} \alpha_m t_m \mathbf{x}_m^\mathsf{T}\right)\left(\sum_{n=1}^{N} \alpha_n t_n \mathbf{x}_n\right) - \sum_{n=1}^{N} \alpha_n \left(t_n \left(\sum_{m=1}^{N} \alpha_m t_m \mathbf{x}_m^\mathsf{T}\mathbf{x}_n + b\right) - 1\right)$$
$$= \frac{1}{2}\sum_{n,m=1}^{N} \alpha_m \alpha_n t_m t_n \mathbf{x}_m^\mathsf{T}\mathbf{x}_n - \sum_{n,m=1}^{N} \alpha_m \alpha_n t_m t_n \mathbf{x}_m^\mathsf{T}\mathbf{x}_n - \sum_{n=1}^{N} \alpha_n t_n b + \sum_{n=1}^{N} \alpha_n$$
$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n,m=1}^{N} \alpha_m \alpha_n t_m t_n \mathbf{x}_m^\mathsf{T}\mathbf{x}_n$$

This goes away by this constraint

This is the **dual optimization** problem (we have eliminated **w**!)
It is a ***quadratic optimization*** problem due to the $\alpha_m \alpha_n$ term (matlab: quadprog)

22

Making Predictions

Our final constraint problem:

$$\sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n,m=1}^{N}\alpha_m\alpha_n t_m t_n \mathbf{x}_m^{\mathsf{T}}\mathbf{x}_n$$

Subject to: $\alpha_n \geq 0, \quad \sum_{n=1}^{N}\alpha_n t_n = 0.$

Give it to a quadratic programming solver!

To predict, we need our decision function D(x)

$$t_{\text{new}} = \text{sign}\ (\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$$

But we just optimized for α's!

Recall: $\mathbf{w} = \sum_{n=1}^{N}\alpha_n t_n \mathbf{x}_n$

So rewrite the decision function as:

$$t_{\text{new}} = \text{sign}\left(\sum_{n=1}^{N}\alpha_n t_n \mathbf{x}_n^{\mathsf{T}}\mathbf{x}_{\text{new}} + b\right)$$

To find b, we will use the fact that for the closest points, $t_n\ (\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b) = 1$

$$b = t_n - \sum_{m=1}^{N}\alpha_m t_m \mathbf{x}_m^{\mathsf{T}}\mathbf{x}_n$$

(note that $t_n = 1/t_n$ in this case because $t_n = \{ 1, -1 \}$)

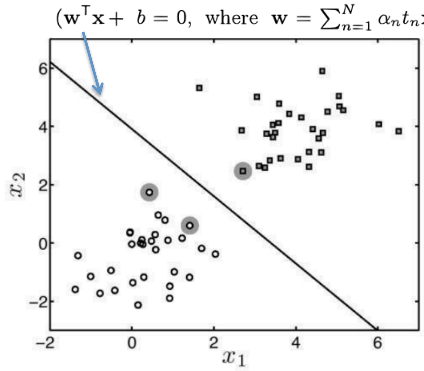
SISTA 23

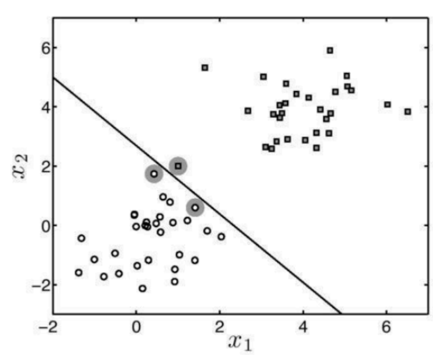
Hard Margin SMV

$$t_{new} = \text{sign}\left(\sum_{n=1}^{N}\alpha_n t_n \mathbf{x}_n^{\top}\mathbf{x}_{new} + t_n - \sum_{m=1}^{N}\alpha_m t_m \mathbf{x}_m^{\top}\mathbf{x}_n\right)$$

After optimizing for all α_n's, the only α's that are **non-zero** are the **support vectors**!

$(\mathbf{w}^{\mathsf{T}}\mathbf{x} + \ b = 0, \ \text{where} \ \ \mathbf{w} = \sum_{n=1}^{N}\alpha_n t_n \mathbf{x}_n)$





SISTA 24

Soft Margin SVM

- To allow points to lie on the wrong side of the boundary, need to “slacken” the constraints

$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{where} \quad \xi_n \geq 0$$

- If $0 \leq \xi_n \leq 1$
  - Then the point lies on the correct side of the boundary, but within the boundary margin
- If $\xi_n > 1$
  - Then the point lies on the “wrong” side of the boundary

25

Soft Margin SVM

- To allow points to lie on the wrong side of the boundary, need to “slacken” the constraints

$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{where} \quad \xi_n \geq 0$$

- The optimization task becomes:

Recall, the original constrained optimization was:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{w}||^2$$

subject to $t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1$, for all n

$$\underset{\mathbf{w}}{\operatorname{argmin}} \quad \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

subject to $\xi_n \geq 0$ and $t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 - \xi_n$ for all n

- *C* controls to what extent we are willing to allow points to sit within the margin itself or on the wrong side of the decision boundary

26

Soft Margin SVM

- To allow points to lie on the wrong side of the boundary, need to "slacken" the constraints

$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{where} \quad \xi_n \geq 0$$

- It turns out that incorporating the new constraint *C* does not change the overall optimization much!: Recall: $\sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n,m=1}^{N} \alpha_m\alpha_n t_m t_n \mathbf{x}_m^\mathsf{T}\mathbf{x}_n \quad \alpha_n \geq 0, \ \sum_{n=1}^{N}\alpha_n t_n = 0$

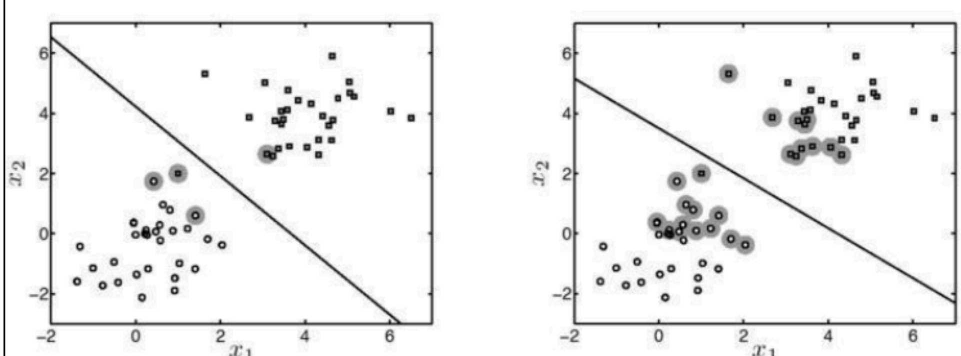
$$\underset{\mathbf{w}}{\text{argmax}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n,m=1}^{N} \alpha_n\alpha_m t_n t_m \mathbf{x}_n^\mathsf{T}\mathbf{x}_m$$

$$\text{subject to} \ \sum_{n=1}^{N}\alpha_n t_n = 0 \ \text{ and } \ 0 \leq \alpha_n \leq C, \ \text{for all } n.$$

Just adds an upper bound on the influence any training point (support vector) can have

27

Decision Boundary & Support Vectors for different *C*



(a) $C = 1$ (b) $C = 0.01$

Larger C is (i.e., the larger the upper bound), the more the optimization can *penalize* for crossing the boundary. Larger C generally leads to fewer support vectors determining the boundary.

$$\underset{\mathbf{w}}{\text{argmax}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n,m=1}^{N} \alpha_n\alpha_m t_n t_m \mathbf{x}_n^\mathsf{T}\mathbf{x}_m$$

$$\text{subject to} \ \sum_{n=1}^{N}\alpha_n t_n = 0 \ \text{ and } \ 0 \leq \alpha_n \leq C, \ \text{for all } n.$$