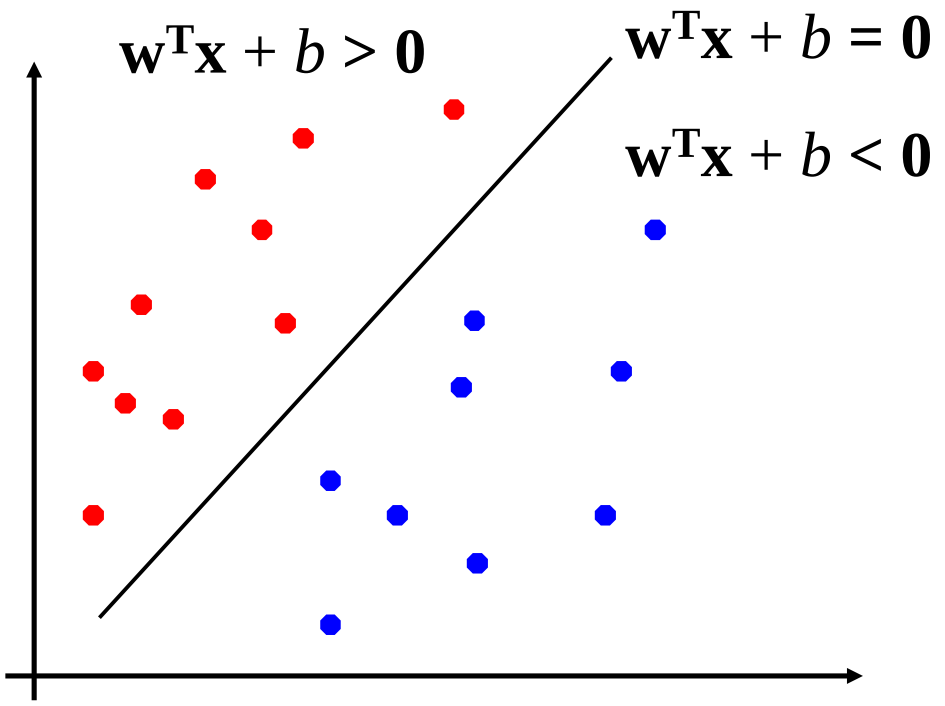
Support Vector Machines

Linear Classifier

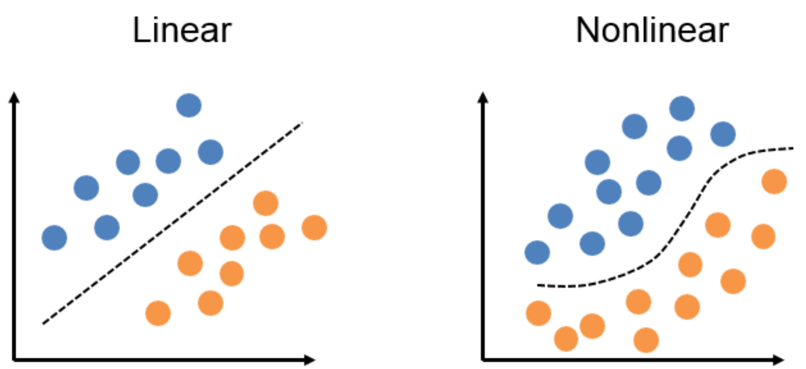
- Linear Classifier is a classifier that makes decision based on the linear combination of the attributes (e.g.: Perceptron, Naïve Bayes, Logistic Regression)
- SVM is one of the state-of-the-art classifiers and also a Linear Classifier (Yes!)

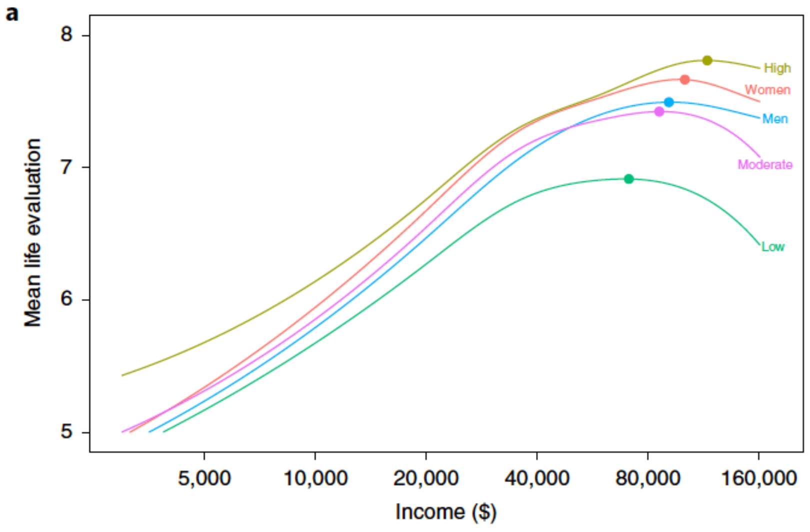


$$f(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^T\mathbf{x} + b)$$

Linear Classifier

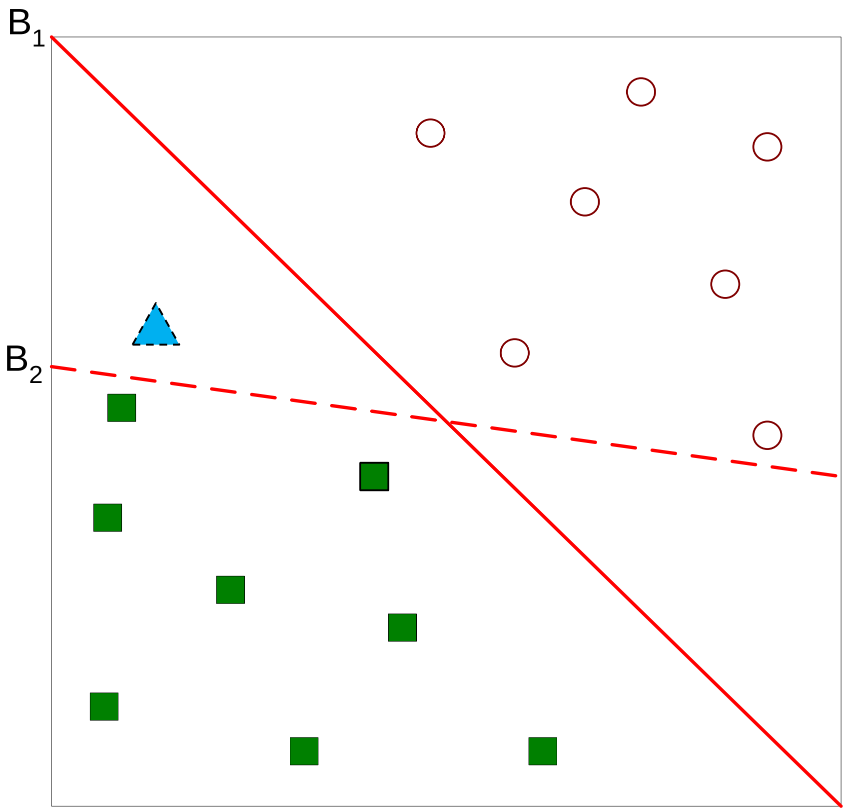
- Linear classifier can be applied to linear problem only, which makes linear classifier ‘toy method’
- Vast majority of real world problem is non-linear
- Most machine learning methods are non-linear classifiers
- Many people are mistaken by the non-linearity of real world





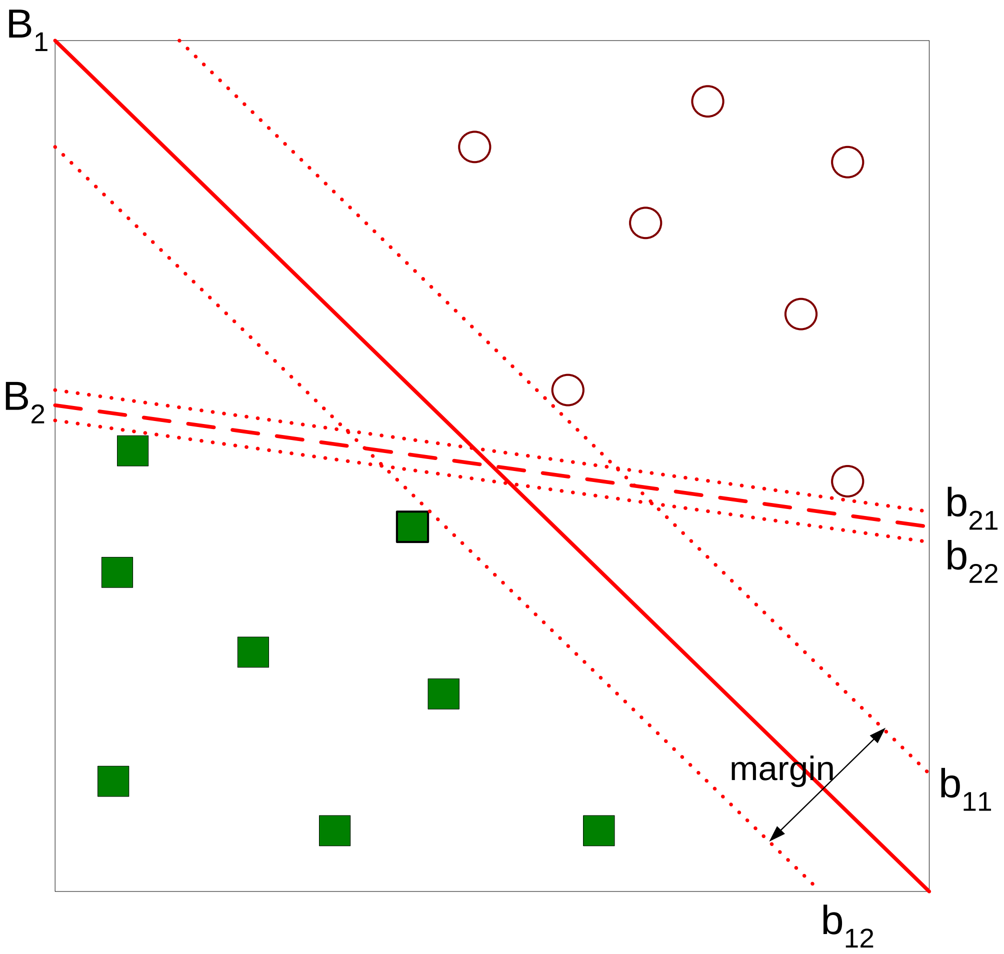
https://www.richardcarrier.info/archives/13954

Support Vector Machines



- SVM is a linear classifier
- Look for a linear line(hyperplane) which separates circles and squares
- Which one is better? B1 or B2?
- How do you define "better" ?

Support Vector Machines



- Find hyperplane maximizes the margin => B1 is better than B2
- Why do we maximize ?

Vapnik Chervonenkis(VC) Dimension

- Typically, a classifier with many parameters is very flexible, but there are also exceptions
- Vapnik argues that the fundamental problem is not the number of parameters to be estimated. Rather, the problem is about the flexibility of a classifier
- A sine wave has infinite VC dimension and only 2 parameters! By choosing the phase and period carefully we can shatter any random collection of one-dimensional datapoints (except for nasty special cases).

$$f(x) = a \sin(b\,x)$$

- Vapnik argues that the flexibility of a classifier should not be characterized by the number of parameters, but by the flexibility (capacity) of a classifier

  \- This is formalized by the "VC-dimension" of a classifier

VC Dimension

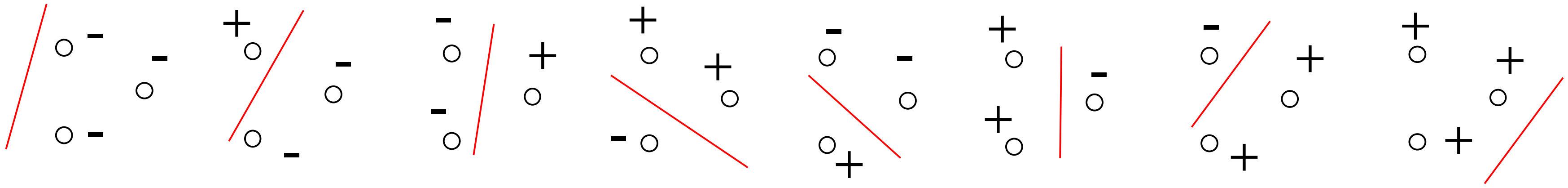
- Classifier f can *shatter* a set of points $x_1, x_2 .. x_r$ if and only if…

  For every possible labeling of the form (x_1,y_1) , (x_2,y_2) ,… (x_r ,y_r),

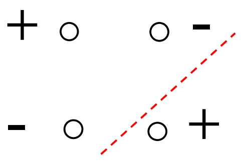
  f can correctly classify these.

- There are 2^r such labelings to consider, each with a different combination of +1's and –1's for the y's

- Linear line (linear classifier) *shatters* 3 points. No matter how those points are labeled, we can classify them perfectly



- Linear line can't shatter 4 points



VC Dimension

- Given classifier ***f***, the *VC-dimension* *h* is the maximum number of points that can be shattered by ***f***
- VC (Vapnik Chervonenkis) dimension represents the expressive power (flexibility) of classifier f
- The VC-dimension of a linear classifier in a 2D space is 3 because, if we have 3 points in the training set, perfect classification is always possible irrespective of the labeling, whereas for 4 points, perfect classification can be impossible
- The VC-dimension of a linear classifier is (number of dimension + 1)
- The VC-dimension of the nearest neighbor classifier is infinity, because no matter how many points you have, you get perfect classification on training data

  - The higher the VC-dimension, the more flexible a classifier is
  - VC-dimension, however, is a theoretical concept; the VC dimension of most classifiers, in practice, is difficult to be computed exactly
  - Qualitatively, if we think a classifier is flexible, it probably has a high VC-dimension

Theoretical Justification for Maximum Margins

- Vapnik has proved the following:

  The class of optimal linear separators has VC dimension h bounded from above as

  $$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

  where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.

- Intuitively, this implies that regardless of dimensionality m_0 we can minimize the VC dimension by maximizing the margin *ρ.*

- Thus, complexity of the classifier is kept small regardless of dimensionality.

Structural Risk Minimization (SRM)

- SRM: A fancy term, but it simply means: we should find a classifier that minimizes training error (empirical risk) and a term that is a function of the flexibility of the classifier (model complexity)

- VC dimension represents the expressive power (flexibility) of classifier

- Now the VC dimension has utility in statistical learning theory, because it can predict a probabilistic upper bound on the test error of a classification model.

The Probabilistic Guarantee

- Vapnik proved that the probability of the test error distancing from an upper bound is given by

$$E_{test} \leq E_{train} + \left(\frac{h + h\log(2N/h) - \log(p/4)}{N} \right)^{\frac{1}{2}}$$

where N = size of training set

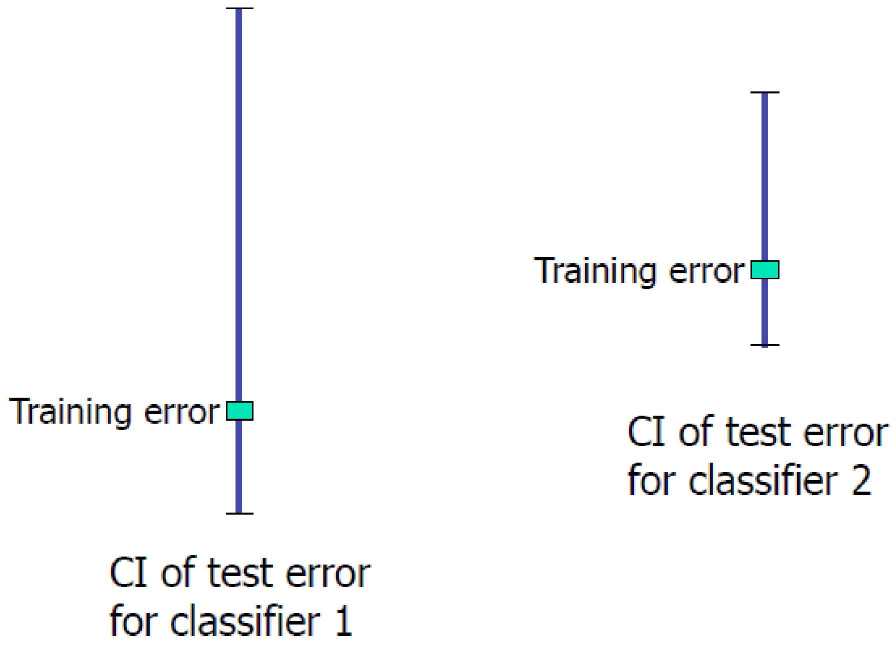
h = VC dimension of the model class

p = upper bound on probability that this bound fails

- So if we train models with different complexity, we should pick the one that minimizes this bound
- Actually, this is only sensible if we think the bound is fairly tight, which it isn't
- The theory provides insight, but in practice we still need some witchcraft

Structural Risk Minimization (SRM)

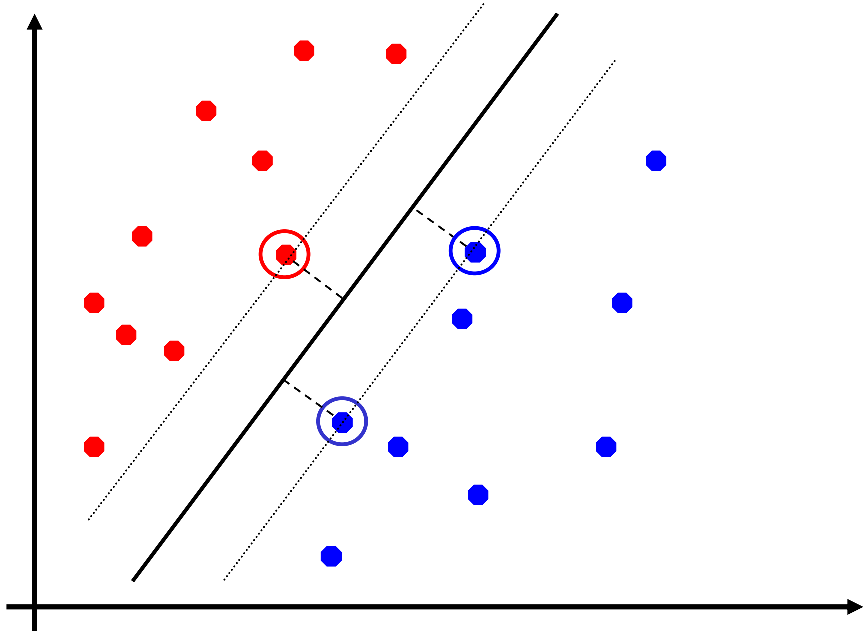
- Roughly speaking, simpler models have smaller VC dimension.
- Pruning in Decision Tree is actually the process of reducing VC dimension of decision tree.



- SRM prefers classifier 2 although it has a higher training error, because the upper limit of confidence interval(CI) is smaller

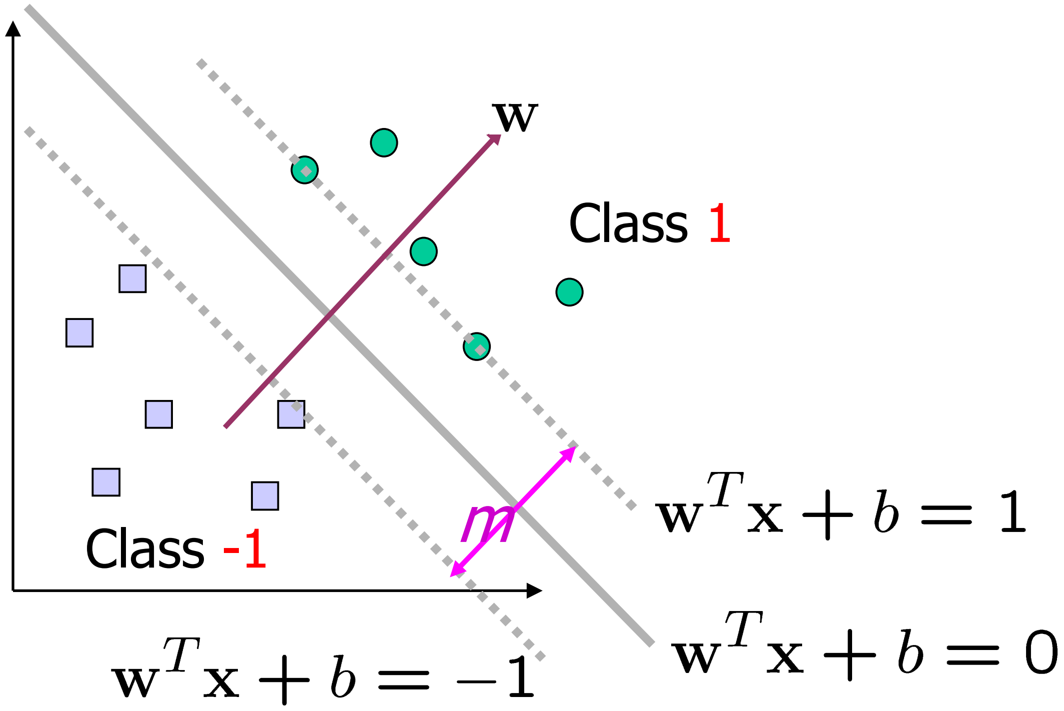
Maximum Margin Classification

- Maximizing the margin is good according to intuition and computational learning theory.
- Why maximize margin? Refer to VC dimension.
- The circled data below are called support vector
- Implies that only support vectors matter; other data are ignorable.



Maximum Margin Classification

- Suppose $\mathbf{w}^T\mathbf{x} + b = 0$ is the maximum margin linear classifier(SVM)
- The following also must be satisfied
  - data below the line $\mathbf{w}^T\mathbf{x} + b = -1$ should be classified as -1
  - data above the line $\mathbf{w}^T\mathbf{x} + b = 1$ should be classified as 1



$$m = \frac{2}{||\mathbf{w}||}$$

$\|w\|$ means $\|w\|_2$

$$\|w\|_p = \sqrt[q]{\sum_i |w_i|^q}$$

Maximum Margin Classification

- Let $\{x_1, ..., x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of x_i
- The condition of 'data below the line $\mathbf{w}^T\mathbf{x} + b = -1$ should be classified as -1' can be represented as (y_i: class value)

$$wx_i + b \leq -1 \qquad \text{if } y_i = -1$$

- The condition of 'data above the line $\mathbf{w}^T\mathbf{x} + b = 1$ should be classified as 1' can be represented as

$$wx_i + b \geq 1 \qquad \text{if } y_i = 1$$

- By combining these two

$$y_i(wx_i + b) \geq 1 \quad \text{for all } i$$

Linear SVM

- Our goal is to 1) maximize the margin and 2) satisfy two conditions
- Goal: 1) Maximize the margin $m = \frac{2}{||\mathbf{w}||}$ is same as minimize $\frac{1}{2} w^t w$

  2) Correctly classify all training data

$$\left.\begin{array}{ll} wx_i + b \geq 1 & \text{if } y_i = 1 \\ wx_i + b \leq -1 & \text{if } y_i = -1 \end{array}\right\}$$

Combine

$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$

- Minimize $\frac{1}{2} w^t w = \frac{1}{2}||\mathbf{w}||^2$

  Subject to $y_i(wx_i + b) \geq 1 \quad \forall i$

Linear SVM

- The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

- This problem is called primal form of SVM
- How to solve this optimization problem ?
- It can be solved by the Constrained Optimization Method (aka Lagrangian multipler method) because it is quadratic (convex), the surface is a paraboloid, with just a single global minimum.

Constrained Optimization Method

- Minimize $f(x_1, x_2, \ldots, x_d)$ where $g_i(X) = 0, \quad i = 1,2,\ldots p$

  $h_i(X) \le 0, \quad i = 1,2,\ldots q$

- If f & $h_i(X)$ are convex and $g_i(X)$ are affine($g_i(X) = a^T X + b$), we can find optimal answer by using the following techniques.

- Define the Lagrangian multipliers:

$$J = f(X) + \sum_{i=1}^{p} \lambda_i g_i(X) + \sum_{i=1}^{q} v_i h_i(X)$$

- Karush-Kuhn-Tucker(KKT) conditions

$$\frac{dJ}{dx_i} = 0, \qquad \forall i = 1, \ldots, d$$

$$v_i \ge 0, \quad \forall i = 1,\ldots q$$

$$v_i h_i(x) = 0, \quad \forall i = 1,\ldots q$$

$$h_i(x) \le 0, \quad \forall i = 1,\ldots q \qquad g_i(X) = 0, \quad i = 1,2,\ldots p$$

Lagrangian Function

- Now, we construct the Lagrangian function:

$$J = \frac{1}{2} \| W \|^2 - \sum_{i=1}^{N} \alpha_i (y_i (w^T x_i + b) - 1) \qquad \alpha_i \geq 0 \quad \forall i$$

where the nonnegative variables α are called Lagrange multipliers.

- Optimization theory says that an optimal solution must satisfy certain conditions, called **Kuhn-Tucker conditions**, which are necessary (but not sufficient)
- For non-convex problems, the KKT conditions are generally necessary but not sufficient.
- If the problem is convex, or the constraints satisfy a regularity condition known as the constraint qualification, the solution is the optimal.

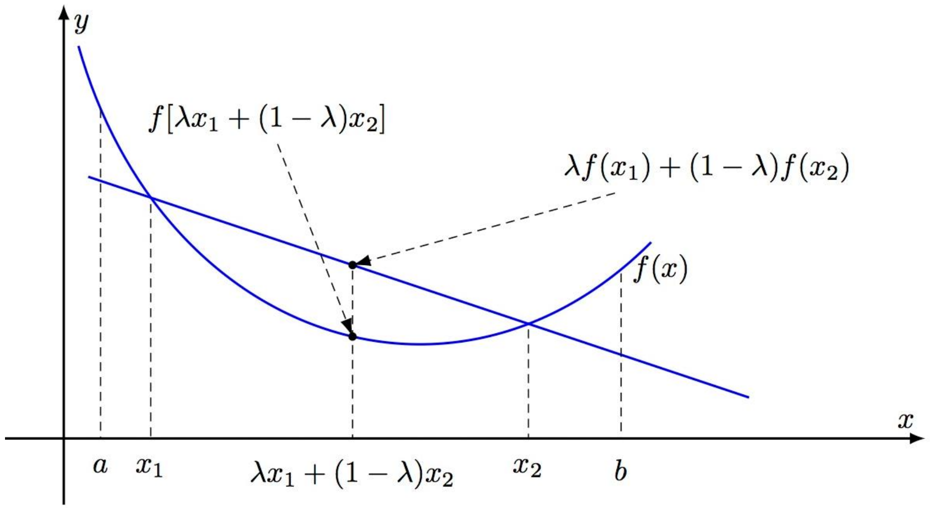
Convex function

- If f is strict *convex*, the previous solution is global optimum (if exists)

- Convex function:

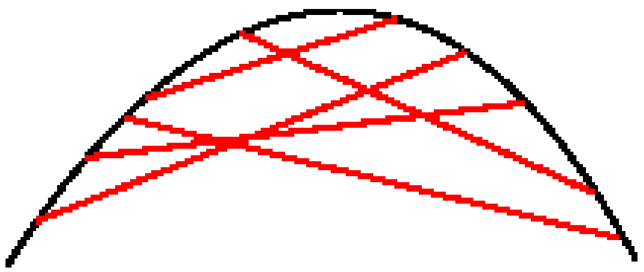
$\forall \lambda \in [0,1]$, for any x1 & x2

$$f[\lambda x_1 + (1-\lambda)x_2] \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

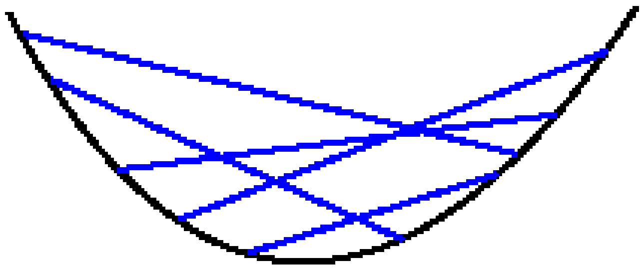


line joining $(x_1,f(x_1))$ and $(x_2,f(x_2))$ lies above the f graph

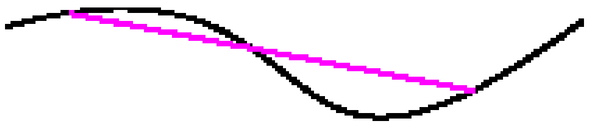
Convex function



A concave function.
No line segment lies above the graph at any point.



A convex function.
No line segment lies below the graph at any point.



A function that is neither concave nor convex.
The line segment shown lies above the graph at some points and below it at other points.

Plug In

- We now solve the primal problem using constrained optimization method (aka Lagrange multipliers)

convex

- Minimize $f(x_1, x_2, \ldots, x_d)$ where $h_i(X) \le 0, \quad i = 1, 2, \ldots q$

$$\boxed{\frac{1}{2} \|W\|^2}$$

$$\boxed{\begin{gathered} y_i(wx_i + b) \ge 1 \Rightarrow \\ -y_i(wx_i + b) + 1 \le 0 \end{gathered}}$$

- Lagrangian : $J = f(X) + \sum_{i=1}^{p} \lambda_i g_i(X) + \sum_{i=1}^{q} \nu_i h_i(X)$

$$\boxed{J = \frac{1}{2} \|W\|^2 + \sum_{i=1}^{N} \alpha_i \left(y_i \left(w^T x_i + b\right) - 1\right)}$$

α_i means ν_i

Lagrangian Function

- Now we solve the constrained optimization problem using the Lagrangian and KKT method

$$J(w,b,a) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i [y_i (w^T x_i + b) - 1]$$

$$\frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \qquad \Rightarrow W = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$KKT\ cond : \alpha_i (y_i (\mathbf{w}\mathbf{x}_i + b) - 1) = 0$$

Lagrangian Function

- The previous Lagrangian function can be expanded term by term, as follows:

$$J(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i w^T x_i - b \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i$$

- The third term on the right-hand side is zero by virtue of the optimality condition. Furthermore, we have

$$w^T w = \sum_{i=1}^{N} \alpha_i y_i w^T x_i = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Lagrangian Function

$$J(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i w^T x_i - b \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i$$

$$w^T w = \sum_{i=1}^{N} \alpha_i y_i w^T x_i = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \qquad \text{since} \quad W = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$- \sum_{i=1}^{N} \alpha_i y_i w^T x_i = -w^T \sum_{i=1}^{N} \alpha_i y_i x_i = -w^T w \qquad \text{since} \quad W = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$- b \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \text{since} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

Dual Problem

- Setting the objective function J(w,b, α)=Q(α), we may reformulate the Lagrangian equation as:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$(1) \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$(2) \alpha_i \geq 0$$

- It is called dual problem:

- Find the Lagrange multipliers α_i that maximize the objective function Q(α), subject to the constraints

Dual Problem

$$\text{max.} \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Advantages of dual problem
  - Because it contains $\mathbf{x}_i^T \mathbf{x}_j$!
  - This is a quadratic programming (QP) problem
    - A global maximum of a_i can always be found
  - **w** can be recovered by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$ (p. 23)

Advantages of Dual Problem

- Problem of getting a_i instead of w and b
- Quadratic problem in terms of a_i
- Many of the a_i are zero
  - **w** is a linear combination of a small number of data points
  - This "sparse" representation can be viewed as data
- Equality constraints, instead of inequality constraints
  - much easier to solve
- Inner product form of features: $\mathbf{x}_i^T \mathbf{x}_j$
  - basis for non-linear SVM

Characteristics of the Solution

- KTT condition indicates many of the α_i are zero
  - **w** is a linear combination of a small number of data points (support vectors)

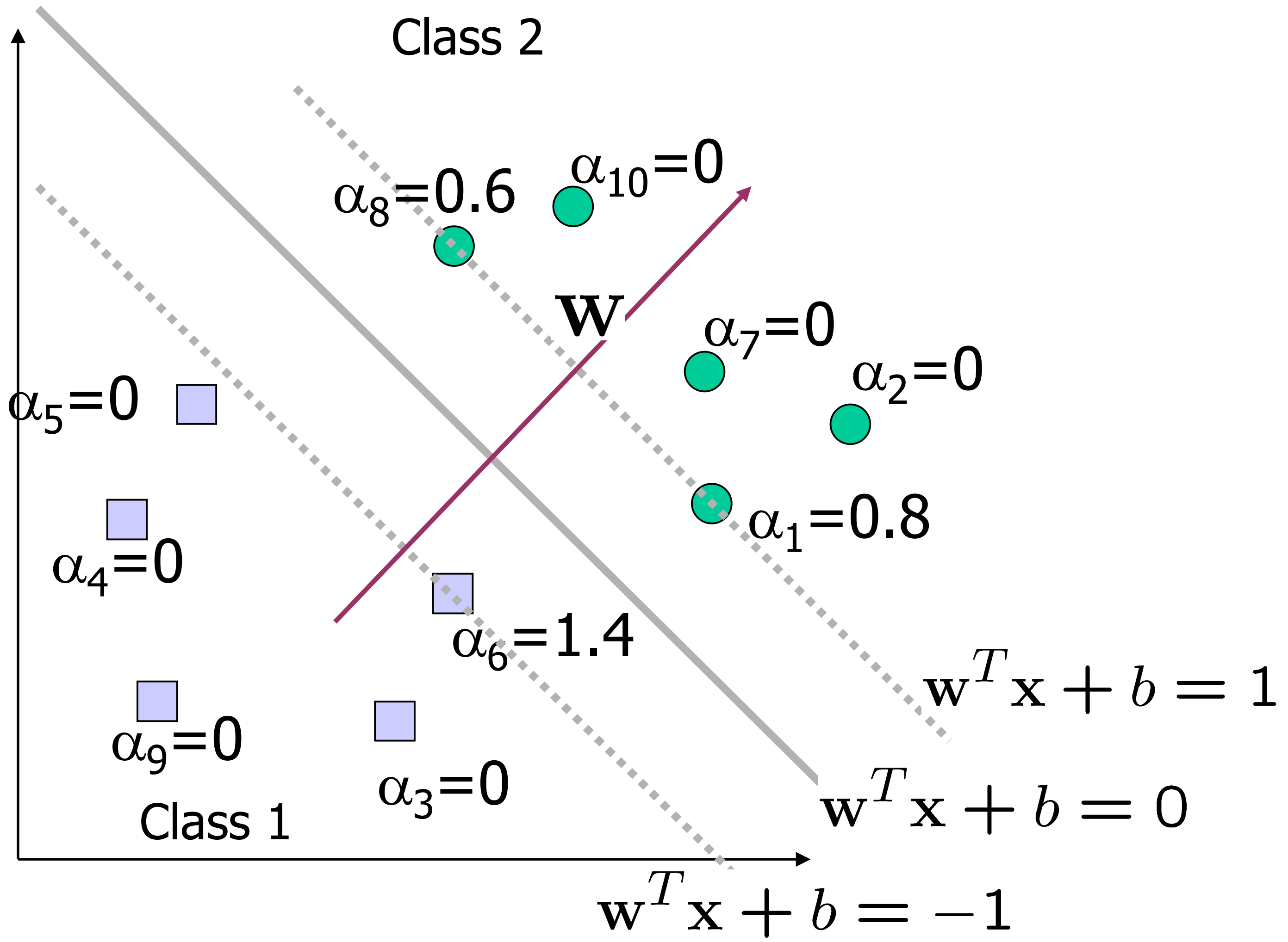
- $\mathbf{x}_i$ with non-zero α_i are called support vectors (SV)
  - The decision boundary is determined only by the SV
  - Let t_j (j=1, ..., s) be the indices of the s support vectors. We can write

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

- For testing with a new data **z**,
  - Compute $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$

    and classify **z** as class 1 if the sum is positive, and class 2 otherwise.

Support Vectors



Solving Linear SVM Problem: Summary

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi(w)} = \mathbf{1/2 w^T w}$ is minimized
and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i (\mathbf{w^T x}_i + b) \geq 1$

primal

- Need to optimize a *quadratic* function subject to *linear* constraints.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every inequality constraint in the primal (original) problem:

Find $\alpha_1 \ldots \alpha_n$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

dual

Solving Linear SVM Problem: Summary

- Given a solution α_1 ... α_n to the dual problem, solution to the primal is:

$$\boxed{\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x}_i \qquad b = y_k - \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_k \quad \text{for any } \alpha_k > 0}$$

- Each non-zero α_i indicates that corresponding $\mathbf{x}_i$ is a support vector.
- Then the classifying function is (note that we don't need **w** explicitly):

$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}} \mathbf{x} + b}$$

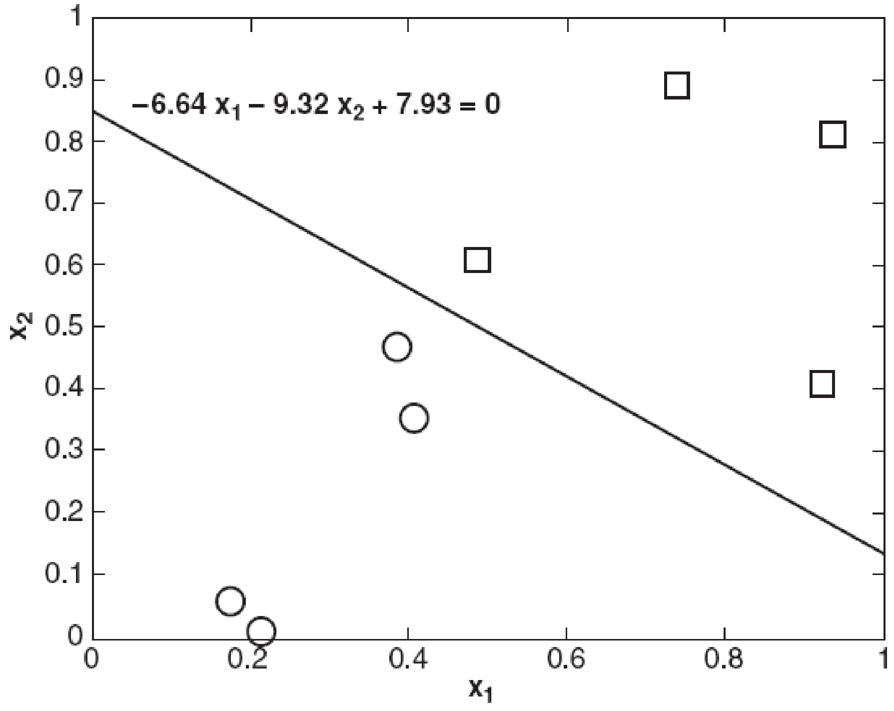
- Notice that it relies on an *inner product* between the test point **x** and the support vectors $\mathbf{x}_i$
  - we will return to this later.
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$ between all training points.

Example 1

Given

x_1	x_2	y	Lagrange Multiplier
0.3858	0.4687	1	65.5261
0.4871	0.611	−1	65.5261
0.9218	0.4103	−1	0
0.7382	0.8936	−1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	−1	0
0.2146	0.0099	1	0

SVM



Example 1

$$w_1 = \sum_i \alpha_i \, y_i x_{i1} = 65.5621 * 1 * 0.3858 + 65.5621 * (-1) * 0.4871 = -6.64$$

$$w_2 = \sum_i \alpha_i \, y_i x_{i2} = 65.5621 * 1 * 0.4687 + 65.5621 * (-1) * 0.611 = -9.32$$

$$b_1 = 1 - W_1 \cdot X_1$$
$$= 1 - (-6.64)(0.3858) - (-9.32)(0.4687) = 7.93$$

b2 value is same as b1

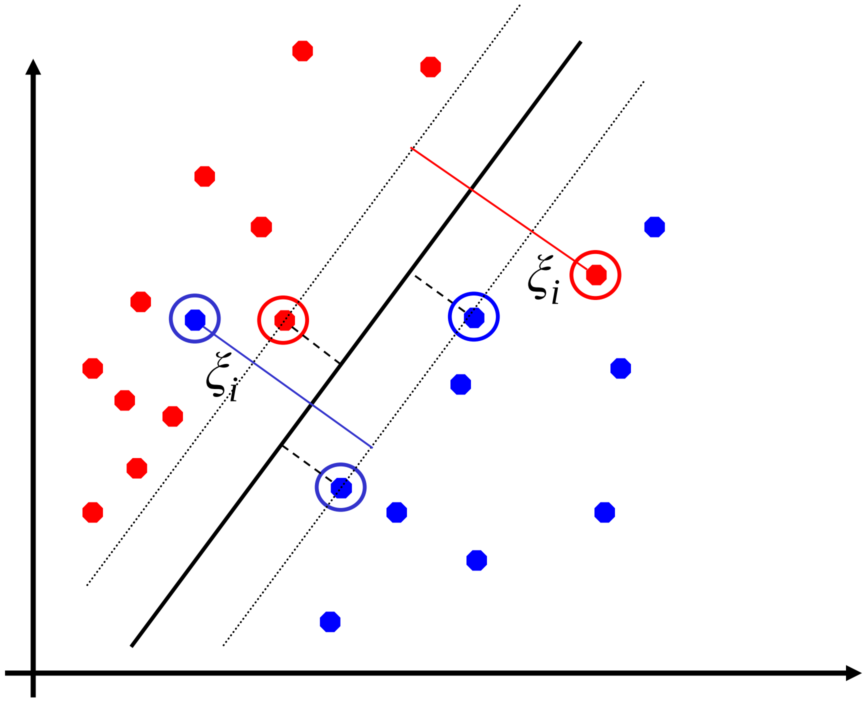
$$b = 7.93$$

SOFT MARGIN SVM

Soft Margin Classification

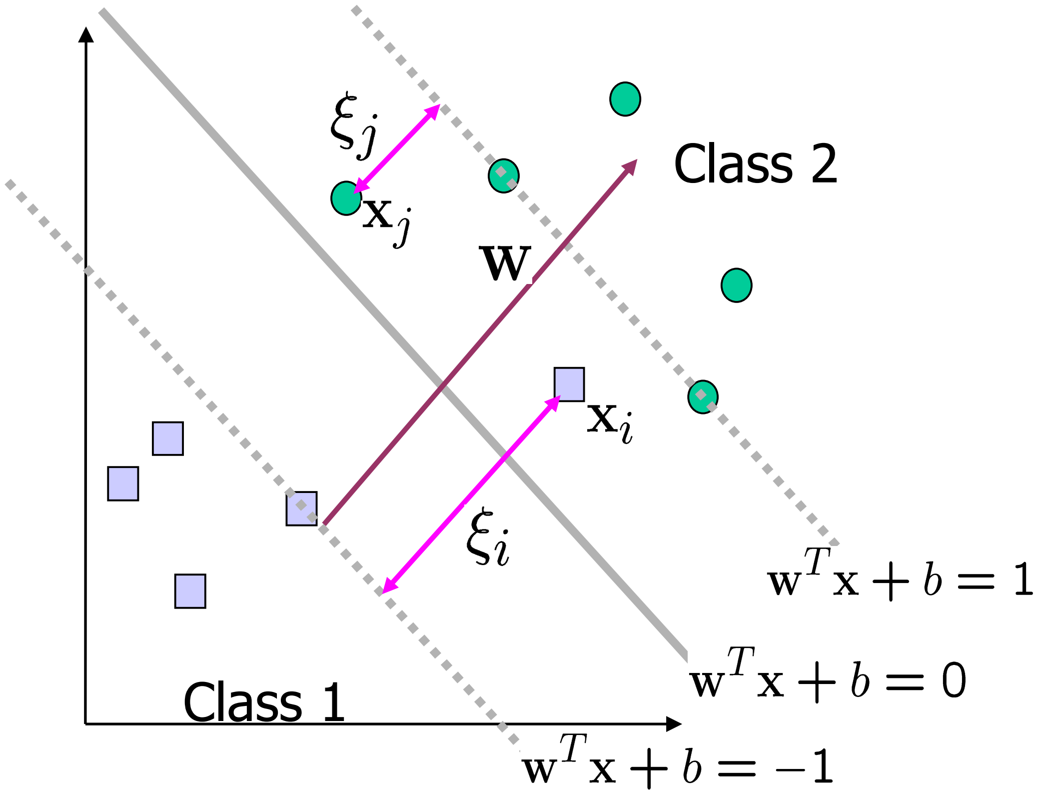
- What if the training set is not linearly separable ?
  - Add a bit of error in SVM (Soft margin SVM)

- *Slack variables* ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called *soft*.



Soft Margin Classification

- We allow "error" ξ_i in classification (ξ_i : size of error for x_i)



- Compared with ordinary SVM, the constraints are changed to

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

Soft Margin SVM

- In soft margin SVM, we want to
  1) maximize the margin and 2) minimize the total x_i

- Therefore, we minimize $\frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$

  C : tradeoff parameter between error and margin

- The optimization problem now becomes

$$\boxed{\begin{array}{l} \text{Minimize } \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{array}}$$

- Primal problem of soft margin SVM

The Optimization Problem

- Using the same Lagrangian Method in ordinary SVM,
- The dual problem of soft margin SVM is

$$\text{max.} \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on a_i now
- Once again, a QP solver can be used to find α_i

Soft Margin SVM : Summary

- The old formulation:

> Find **w** and b such that
> $\mathbf{Q}(\mathbf{w}) = \mathbf{1/2w}^{\mathrm{T}}\mathbf{w}$ is minimized
> and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $\quad y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq 1$

- Modified formulation incorporates slack variables:

> Find **w** and b such that
> $\mathbf{Q}(\mathbf{w}) = \mathbf{1/2w}^{\mathrm{T}}\mathbf{w} + C\Sigma\xi_i$ is minimized
> and for all $(\mathbf{x}_i, y_i)$, $i=1..n$: $\quad y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

- Parameter *C* can be viewed as a way to control overfitting: it “trades off” the relative importance of maximizing the margin and fitting the training data.

Soft Margin Classification – Solution

- Dual problem is identical to

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½ \Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

- Again, $\mathbf{x}_i$ with non-zero α_i will be support vectors.
- Solution to the dual problem is:

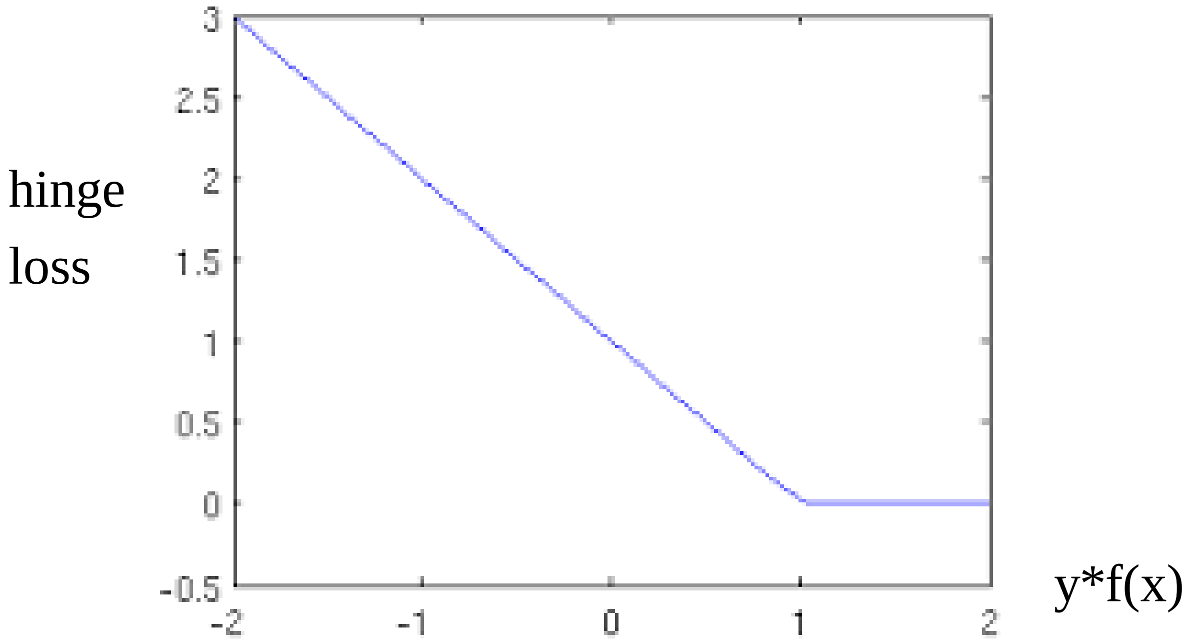
$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x}_i$
$b = y_k(1 - \xi_k) - \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_k$ for any k s.t. $\alpha_k > 0$

- Again, we don't need to compute **w** explicitly for classification:

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^{\mathbf{T}}\mathbf{x} + b$

Recap: Hinge Loss

- Hinge loss: max{0, (1-y*f(x))}
  - y: true value, f(x): predicted value
  - gives high penalty for wrong answers



Gradient Method in Soft Margin SVM

- In soft margin SVM, the constraints are changed to

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

- ξ_i are "slack variables" in optimization; ξ_i =0 if there is no error for $\mathbf{x}_i$, and ξ_i is an upper bound of the errors

- ξ_i can be written as follows

$$\xi_i = \begin{cases} 1 - y_i\left(w^T x_i + b\right), & \text{if } y_i\left(w^T x_i + b\right) < 1 \\ 0, & \text{if } y\left(w^T\ x_i + b\right) \geq 1 \end{cases}$$

Gradient Descent SVM

- ξ_i is defined as

$$\xi_i = \begin{cases} 1 - y(w^T x + b), & \text{if } y(w^T x + b) < 1 \\ 0, & \text{if } y(w^T x + b) \geq 1 \end{cases}$$

- Above formula is equivalent to the following form:

$$\xi_i = \max\{0, 1 - y_i(w^T x_i + b)\}$$

- The primal problem of soft-margin SVM is

Minimize $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i$

subject to $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

Gradient Descent SVM

- If we plug ξ_i into the objective of our SVM problem, we obtain the following loss function and regularizer:

$$\boxed{\min \quad \sum_i C_i * \underbrace{max(0,\ 1 - y_i(wx_i + b)\ )}_{\text{Hinge loss}} + \underbrace{\frac{1}{2}\|\mathrm{w}\|^2}_{\text{Ridge regularization}}}$$

- We see that SVM now is a method that minimizes Hinge error function with Ridge regularization.

- This formulation allows us to optimize the SVM parameter (w, b) by using gradient descent
- The only difference is that we have hinge-loss instead of cross-entropy loss (logistic loss).