Linear discrimination

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Books

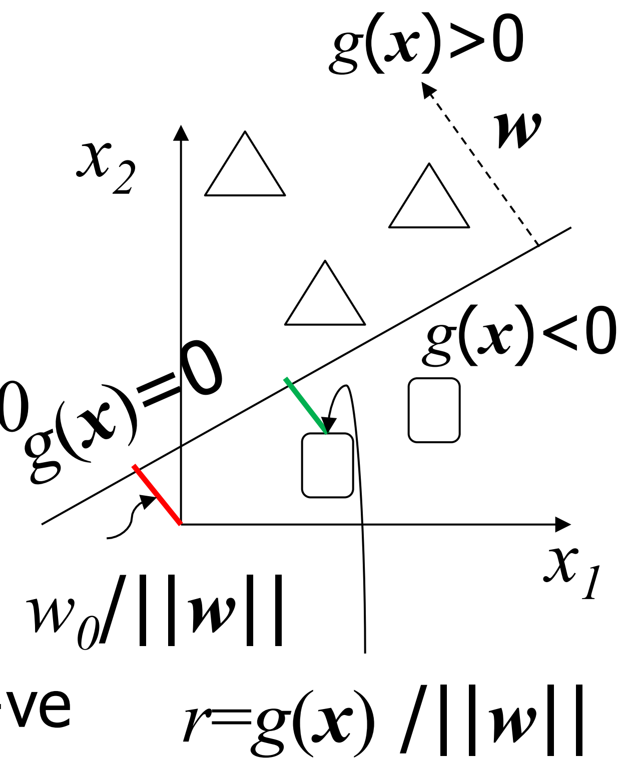
- Chapter 10 and 13 of “Introduction to Machine Learning” by Ethem Alpaydin.
- Chapter 5 of “Pattern Classification” by R.O. Duda, P. E. Hart and D. G. Stork

Discriminant functions

- Choose C_i if $g_i(\boldsymbol{x})=\max_j g_j(\boldsymbol{x})$
- Linear function
  - $g_i(\boldsymbol{x}|\boldsymbol{w_i},w_{i0})=\boldsymbol{w_i}^T\boldsymbol{x}+w_{i0}=\Sigma^d_{j=1}w_{ij}x_j+w_{i0}$
  - Simple model, linear in form
  - $O(d)$ storage and time of computing $g(.)$.
- Quadratic function
  - $g_i(\boldsymbol{x}|\boldsymbol{W_i},\boldsymbol{w_i},w_{i0})=\boldsymbol{x}^T\boldsymbol{W_i}\boldsymbol{x}+\boldsymbol{w_i}^T\boldsymbol{x}+w_{i0}$
  - $O(d^2)$ storage and time of computing $g(.)$.

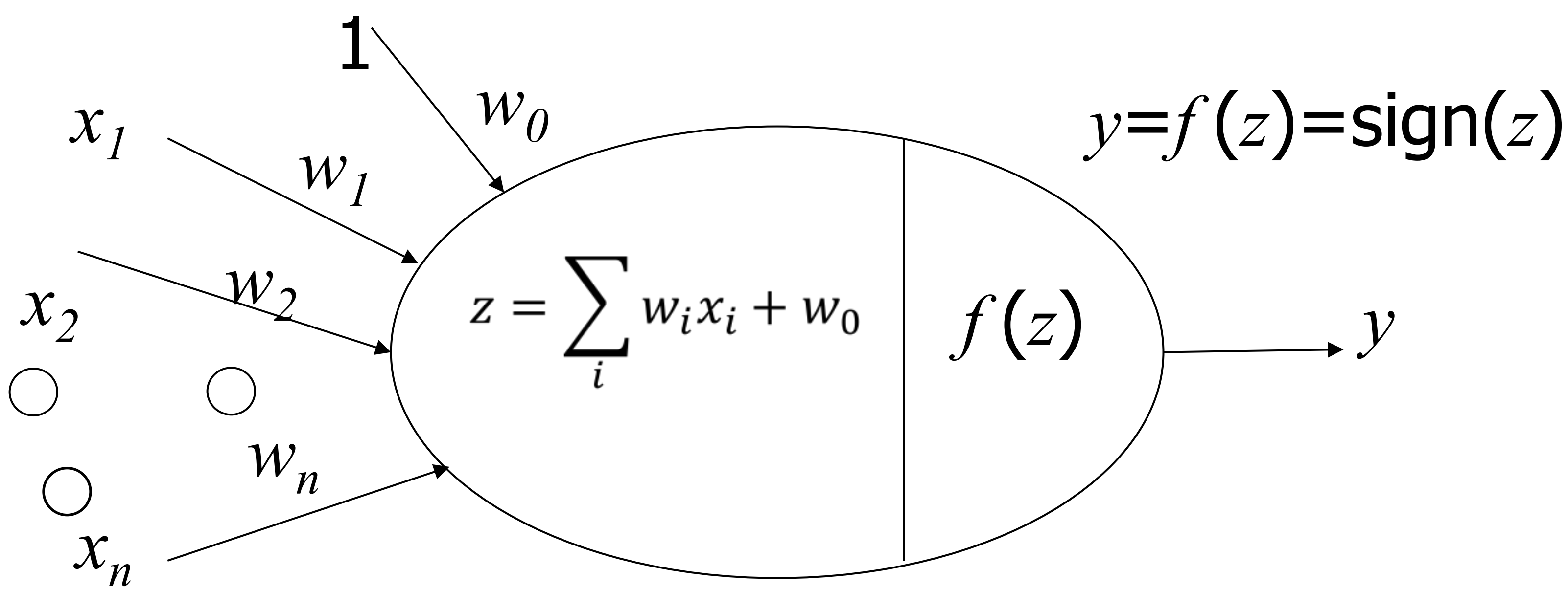
Two classes

- One discriminant function sufficient
  - $g(\boldsymbol{x})=g_1(\boldsymbol{x}|\boldsymbol{w_1},w_{10})\text{-}g_2(\boldsymbol{x}|\boldsymbol{w_2},w_{20})$
  - $\quad =\boldsymbol{w_1}^T\boldsymbol{x}+w_{10}-\boldsymbol{w_2}^T\boldsymbol{x}-w_{20}$
  - $\quad =(\boldsymbol{w_1}-\boldsymbol{w_2})^T\boldsymbol{x}-(w_{10}-w_{20})$
  - $\quad = \boldsymbol{w}^T\boldsymbol{x}+w_0$
- If $g(\boldsymbol{x}) > 0$ assign C_1 else C_2.
- Hyper-plane dividing classes: $g(\boldsymbol{x})=0$
- Extend to more than 2 classes
  - Pairwise separation.
  - Only samples of the class lies in the +ve half.



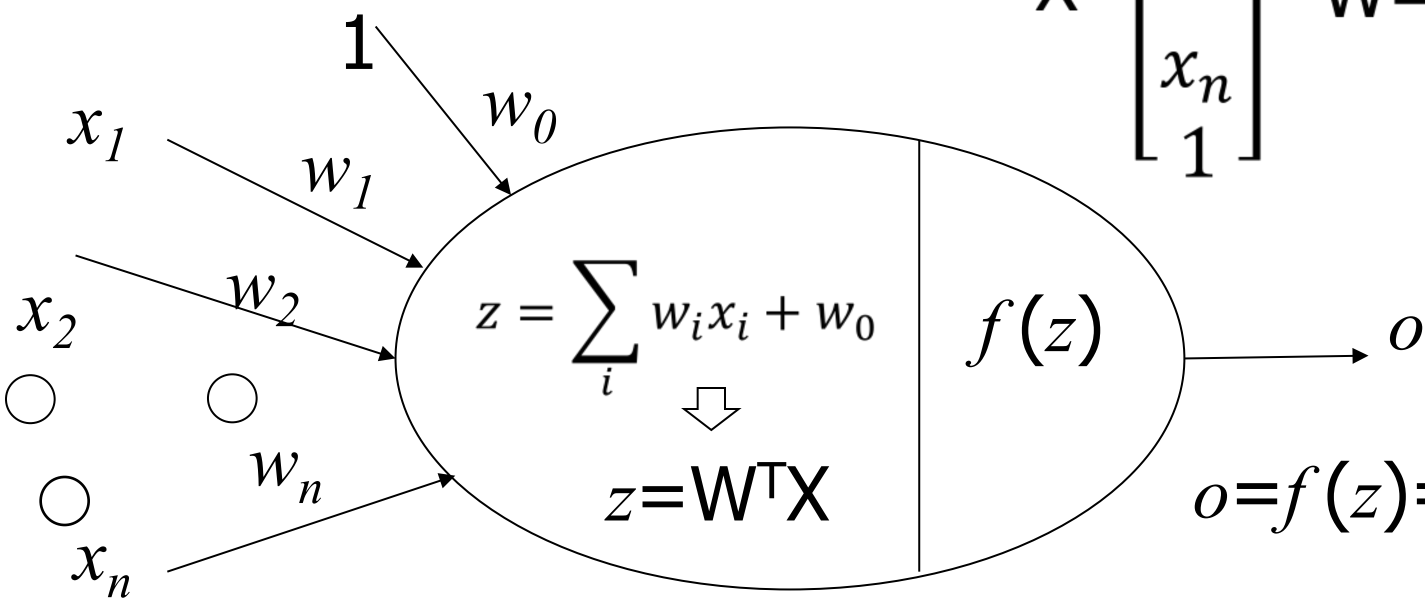
Perceptron classifier

- A linear classifier with a different perspective.



Augmented input and linear form

$$X=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \quad W=\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_0 \end{bmatrix}$$

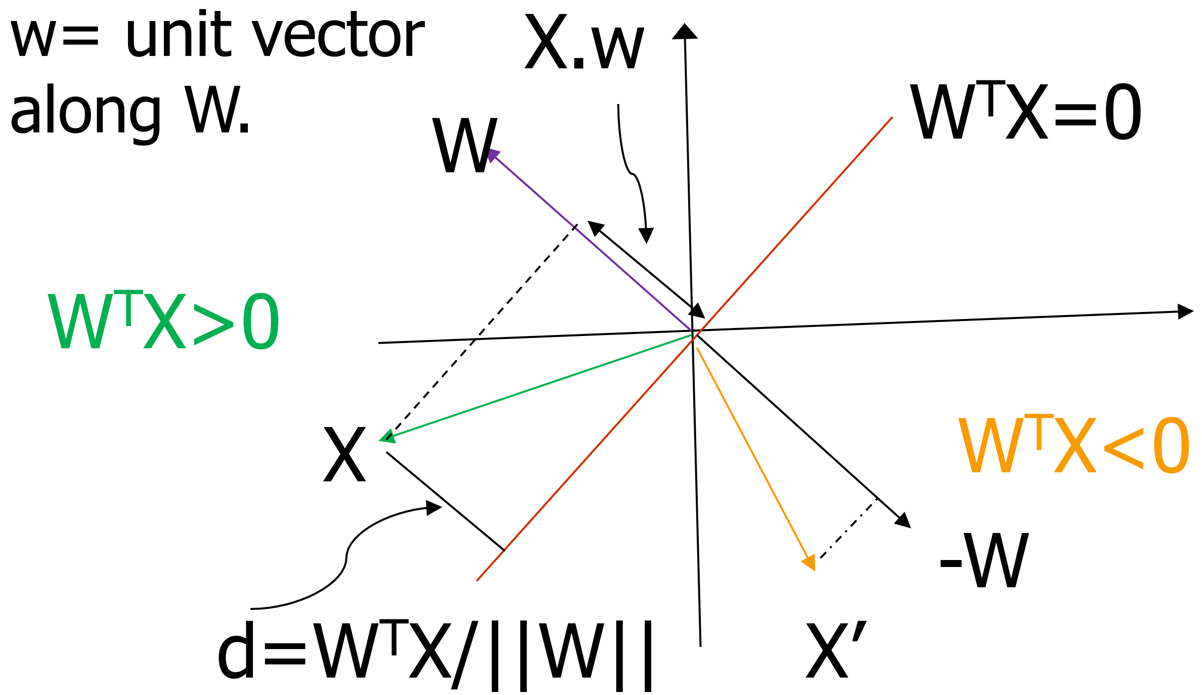


$o=f(z)=\text{sign}(z)$

Given $\{y_i,X_i\}$, compute optimum W minimizing classification error.

Interpretation of W^TX

Consider the hyperplane $W^TX=0$ separating two samples X and X' of classes 1 and 2.

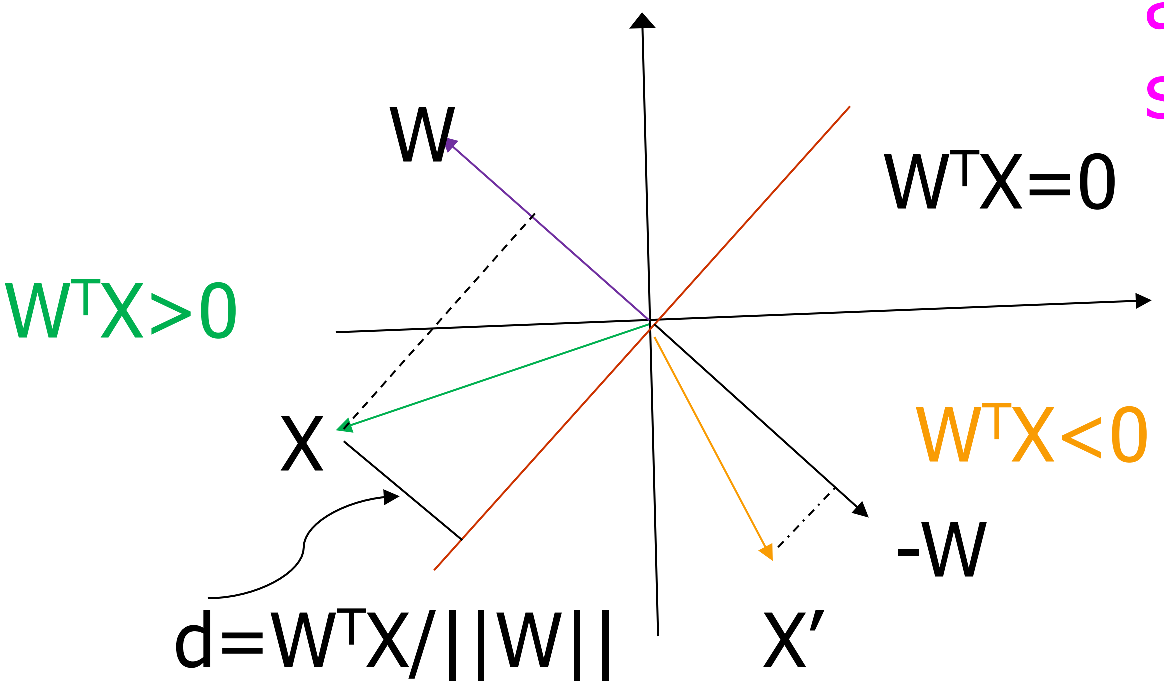


Distance of X from the hyperplane.

Linearly separable classes

To find a hyperplane separating data points of two classes.

If a solution exists, the classes are called linearly separable.



Distance of X from the hyperplane.

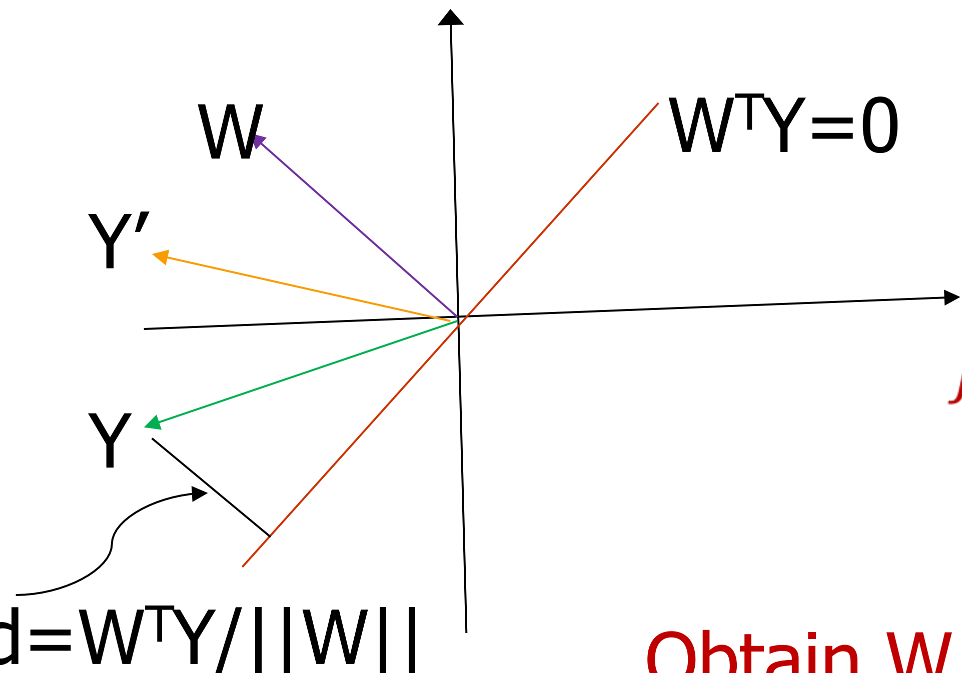
An error function

Data Normalization:

$Y = X$, if X in class 1 (o=1).

$= -X$, if X in class 2 (o=-1)

For correct classification, $W^T Y > 0$, for all Y.



Error function (Perceptron Criterion):

$$J(W) = \sum_{Y \, misclassified} -W^T Y$$

Always +ve

Obtain W which minimizes J(W).

Gradient descent method for iterative optimization

- To obtain W which minimizes J(W).
- Start with an initial vector $W^{(0)}$.
- Compute the gradient vector $\nabla J(W^{(0)})$
- Move closer to minimum by updating W.

$$W^{(i)} = W^{(i-1)} - \eta(i)\nabla J(W)$$

Positive scale factor (learning rate)

Iterative gradient descent Optimization

Data Normalization:

Y=X, if X in class 1 (o=1).

=-X, if X in class 2 (o=-1)

$$J_p(W) = \sum_{Y\ misclassified.} -W^T Y$$

Iterative Optimization using gradient descent

1. Start with $W^{(0)}$.
2. Update W

$$W^{(i)} = W^{(i-1)} + \eta(i) \sum_{Y\ misclassified} Y$$

May be taken as a constant.

$$W^{(i)} = W^{(i-1)} - \eta(i) \nabla J_p(W)$$

3. Continue step 2 till converges.

Other forms of the error function

- There could be other forms of the criterion function.
  - $J_p(W)$: not continuous
  - $J_q(W)$: continuous.
    - Very smooth in boundary.
    - May get stuck there.
    - Value dominated by long Y's.

$$J_q(W) = \sum_{Y\ misclassified.} (W^T Y)^2$$

Another error function (Relaxation criterion)

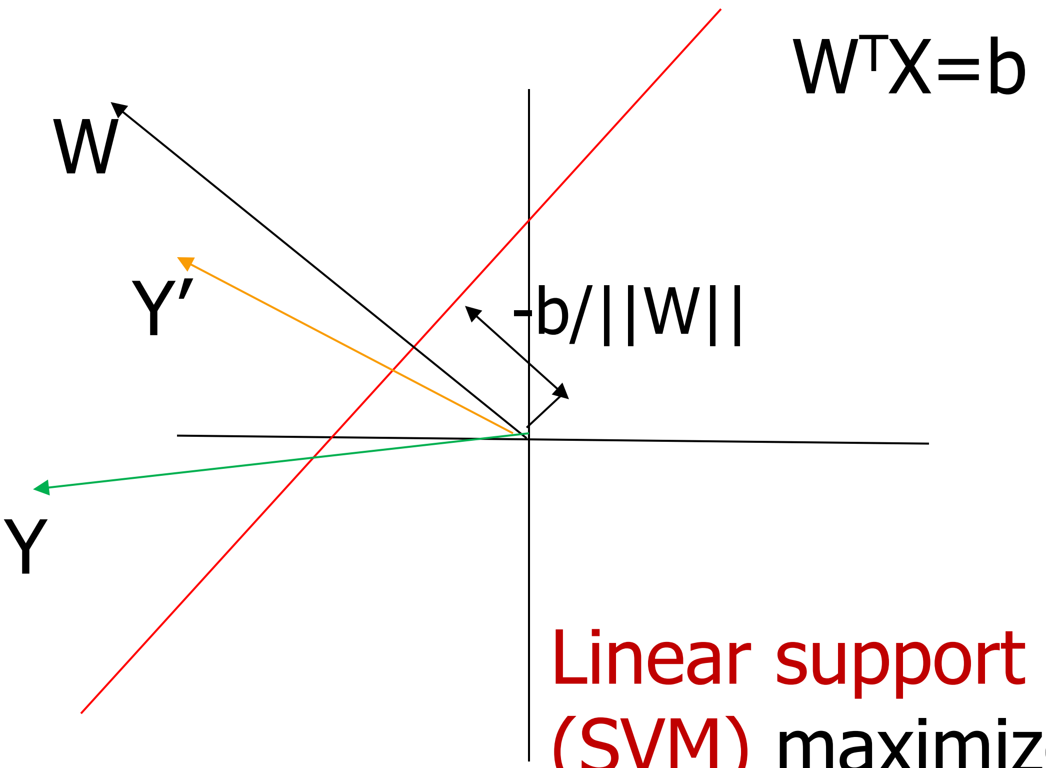
$$J_r(W) = \frac{1}{2} \sum_{\substack{Y \text{ misclassified} \\ W^T Y \le b}} \frac{(W^T Y - b)^2}{\|Y\|^2}$$

Stronger linear separability

Gradient computation:

$$\boxed{\nabla J_r(W) = \sum_{W^T Y \le b} \frac{Y(W^T Y - b)}{\|Y\|^2}}$$

More stringent criteria of linear separability



Linear support vector machines (SVM) maximize this margin of separation between two linearly separable data points of classes.

The algorithm (Batch relaxation with margin)

- Initialize W to $W^{(0)}$.
- Iterate till convergence
- Compute the set M of misclassified samples (with margin b), so that
  - $M=\{Y|W^TY<=b\}$
- Compute gradient.

$$\nabla J_r(W) = \sum_{W^TY \le b} \frac{Y(W^TY-b)}{\|Y\|^2}$$

- Update W.

$$W^{(i)} = W^{(i-1)} - \eta(i)\nabla J_r(W^{(i-1)})$$

Single sample relaxation with margin

- Initialize W to $W^{(0)}$.
- Perform the update of W by considering samples one by one in every iteration.
- Consider an i th sample Y_i at k th iteration.
- If (W^TY_i <=b)
  - Update W.

$$W^{(k)} = W^{(k-1)} + \eta(k)\frac{b - W^TY_i}{||Y_i||^2}Y_i$$

- Stop when very little change in updates at the end of an iteration.

Support Vector Machine (SVM)

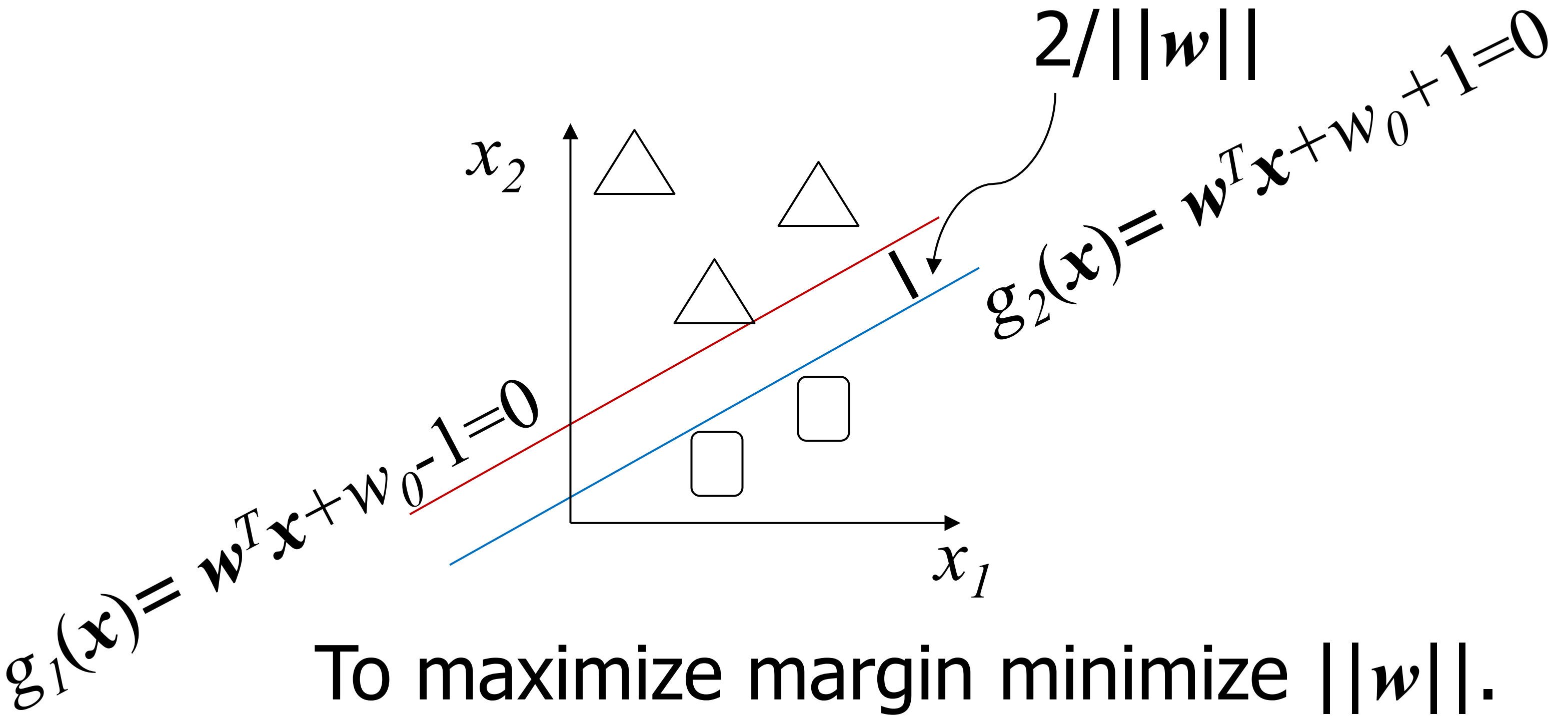
- A linear discriminant classifier.
- Uses Vapnik's principle:
  - to never solve a more complex problem as a first step before the actual problem.
    - Classification: Sufficient to compute class boundaries (where $P(C_1|\boldsymbol{x})=P(C_2|\boldsymbol{x})$) without computing class distributions $P(C_i|\boldsymbol{x})$, etc.
    - Outlier detection: Compute boundaries separating those x having low P(x).
- After training the weight vector can be written in terms of training samples lying in class boundaries.
  - Support vectors.

Vapnik, V. 1995. The Nature of Statistical Learning Theory. New York: Springer

Two class problem

- X={$\boldsymbol{x}^t, r^t$}, t=1,2,..N
  - $\boldsymbol{x}^t$ in R^d, r^t in {+1, -1}.
- To compute $\boldsymbol{w}$ and w_0 such that for all t
  - $\boldsymbol{w}^T\boldsymbol{x}^t+w_0$ > +1 if r^t=+1
  - $\boldsymbol{w}^T\boldsymbol{x}^t+w_0$ < -1 if r^t=-1
- Rewritten as:
  - $r^t(\boldsymbol{w}^T\boldsymbol{x}^t+w_0) \geq$ +1 for all t
    - Note: it is harder than $r^t(\boldsymbol{w}^T\boldsymbol{x}^t+w_0) \geq$ 0
    - A margin left between zones of two classes.

Margin between classes



To maximize margin minimize $||\boldsymbol{w}||$.

To minimize $||\boldsymbol{w}||^2/2$ subject to $r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) \geq +1$ for all t

Optimization problem

- Constrained optimization problem
  - To minimize $||\boldsymbol{w}||^2/2$
    - subject to $r^t(\boldsymbol{w}^T\boldsymbol{x}^t+w_0) \geq +1$ for all t
- Unconstrained problem:

Lagrange multipliers

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{t=1}^{N} \alpha^t [r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) - 1]$$

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) + \sum_{t=1}^{N} \alpha^t$$

- To be minimized w.r.t $\boldsymbol{w}$ and w_0 and maximized w.r.t. Lagrange multipliers.

Convex quadratic optimization problem

- Unconstrained problem:

Lagrange multipliers

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t (\boldsymbol{w}^T \boldsymbol{x}^t + w_0) + \sum_{t=1}^{N} \alpha^t$$

  - Convex objective function and linear constraints.

- Dual problem

- To be maximized w.r.t. Lagrange multipliers (>0) subject to that gradients w.r.t $\boldsymbol{w}$ and w_0 should be 0.

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_t \alpha^t r^t \boldsymbol{x}^t \qquad \frac{\partial L_p}{\partial w_0} = 0 \implies \sum_t \alpha^t r^t = 0$$

Dual optimization problem

- Primary problem:

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t (\boldsymbol{w}^T \boldsymbol{x}^t + w_0) + \sum_{t=1}^{N} \alpha^t$$

- Dual problem derived by applying following conditions in the primary objective function.

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \sum_t \alpha^t r^t \boldsymbol{x}^t \qquad \frac{\partial L_p}{\partial w_0} = 0 \implies \sum_t \alpha^t r^t = 0$$

$$L_d = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{w}^T \overbrace{\sum_t \alpha^t r^t \boldsymbol{x}^t}^{\boldsymbol{w}} - w_0 \overbrace{\sum_t \alpha^t r^t}^{0} + \sum_t \alpha^t$$

$$\Downarrow$$

$$L_d = -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_t \alpha^t$$

Dual optimization problem

- Dual problem:

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_t \alpha^t r^t \boldsymbol{x}^t$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_t \alpha^t r^t = 0$$

$$\alpha^t \geq 0$$

$$L_d = -\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_t \alpha^t$$

- Expand $\boldsymbol{w}$ from the condition.

$$L_d = -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\boldsymbol{x}^t)^T \boldsymbol{x}^s + \sum_t \alpha^t$$

- To maximize L_d w.r.t. α^t ← Most of them will be 0.
- Apply quadratic optimization technique:
  - $O(N^3)$ time and $O(N^2)$ space complexity.

Solution

- Dual problem:

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_t \alpha^t r^t \boldsymbol{x}^t$$

$$L_d = -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\boldsymbol{x}^t)^T \boldsymbol{x}^s + \sum_t \alpha^t$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_t \alpha^t r^t = 0$$

$$\alpha^t \geq 0$$

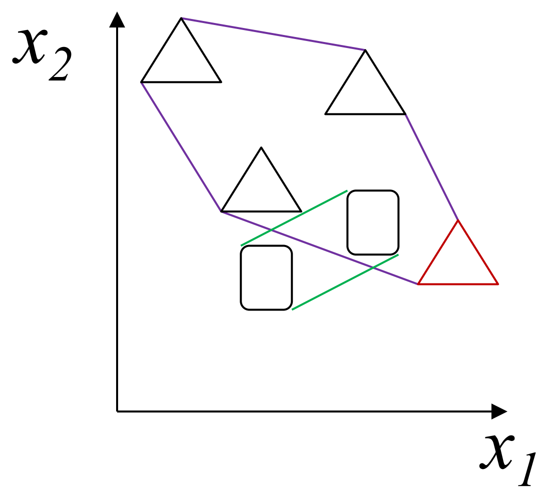
- Apply quadratic optimization technique:
  - $O(N^3)$ time and $O(N^2)$ space complexity.
- Most of α^t will be zero.
- Samples with positive (non-zero) α^t are support vectors.
  - Provide $\boldsymbol{w}$ as a linear combination of input samples (Condition 1).
  - w_0 is obtained from **any** of the support vector which lies in the boundary.
  - $r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) = +1$ ➔ $w_0 = r^t - \boldsymbol{w}^T\boldsymbol{x}^t$ (For numerical stability take **avg.**).

SVM- Testing

- Check only the sign of discriminant value.
  - Margin not enforced.
- Only support vectors decide class boundaries.
  - Other samples do not influence the classifier.

The non-separable case: Soft margin hyperplane

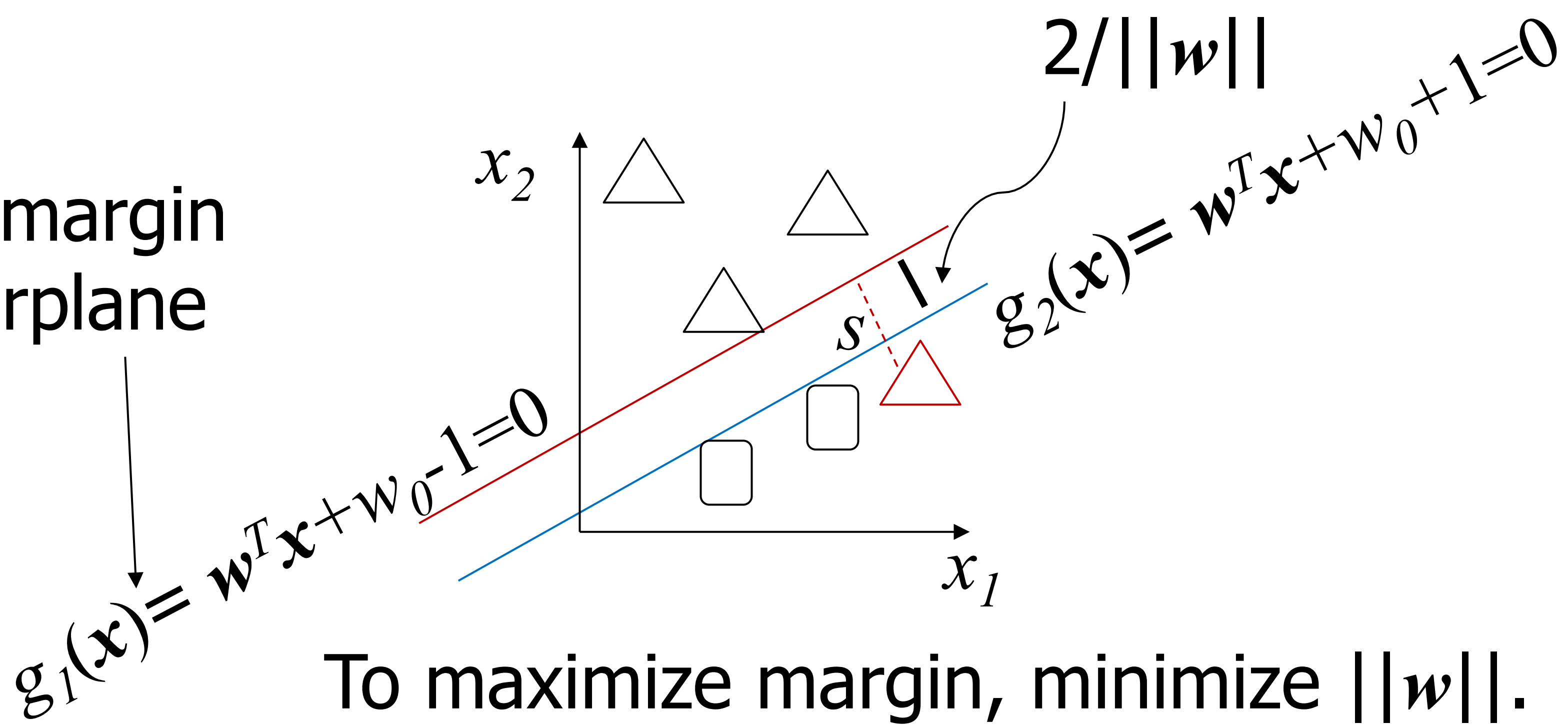
- Classes may not be linearly separable.



- Use of slack variable, $\{s^t\}$, t=1,2,..N

Slack variable to define soft margin

Soft-margin hyperplane



To maximize margin, minimize $||\boldsymbol{w}||$.

To minimize $||\boldsymbol{w}||^2/2$ subject to $r^t(\boldsymbol{w}^T\boldsymbol{x}^t+w_0) \geq +1-s^t$ for all t

Constraints with Soft margin hyperplanes

- Use of slack variable, $\{s^t\}$, t=1,2,..N to define constraints.
  - $r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) \geq 1 - s^t$ for all t
    - $0 < s^t < 1$, $\boldsymbol{x}^t$ correctly classified.
  - If $s^t \geq 1$, $\boldsymbol{x}^t$ misclassified.
  - #$[s^t > 1]$: Number of misclassified points.
  - #$[s^t > 0]$: Number of non-separable points.
  - Soft error= $\Sigma_t\, s^t$

Optimization problem

- Add penalty term for soft error to define the objective function for minimization.
  - $L_p = ||\mathbf{w}||^2/2 + C \sum_t s^t$
    - subject to $r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - s^t$ for all t
    - C is the penalty factor.

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t s^t - \sum_{t=1}^{N} \alpha^t[r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1 + s^t] - \sum_t \mu^t s^t$$

  - μ^t are the new Lagrange parameters to guarantee that $s^t > 0$.

Optimization problem

- Primary problem:

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_t s^t - \sum_{t=1}^{N} \alpha^t [r^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) - 1 + s^t] - \sum_t \mu^t s^t$$

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \;\Rightarrow \boldsymbol{w} = \sum_t \alpha^t r^t \boldsymbol{x}^t \qquad \frac{\partial L_p}{\partial w_0} = 0 \;\Rightarrow \textstyle\sum_t \alpha^t r^t = 0$$

$$\frac{\partial L_p}{\partial s^t} = C - \alpha^t - s^t = 0 \;\Rightarrow\; 0 \le \alpha^t \le C \text{ as } s^t > 0$$

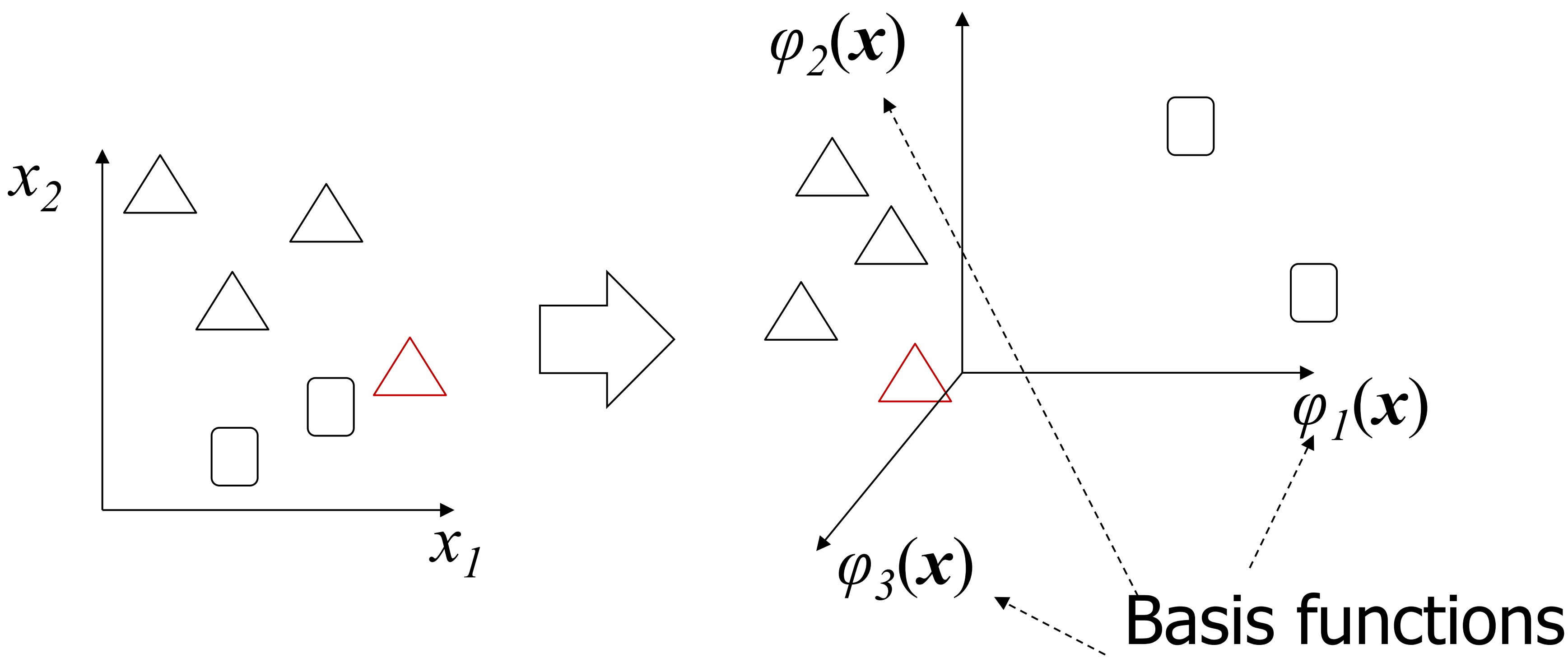
- Dual problem: $L_d = -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\boldsymbol{x}^t)^T \boldsymbol{x}^s + \sum_t \alpha^t$
  - Subject to: $\sum_t \alpha^t r^t = 0$ and $0 \le \alpha^t \le C$

The solution

- The same quadratic optimization technique to be used.
- The support vectors have $\alpha^t > 0$
- Out of them whose values are less than C are used for deriving w_0.
  - $w_0 = r^t - \boldsymbol{w}^T\boldsymbol{x}^t$
  - Take average.

Projecting to higher dimensional space

- May make them linearly separable!



Solving by projecting to a high-dimensional space.

- $\boldsymbol{z} = \varphi(\boldsymbol{x})$, *where* $z_j = \varphi_j(\boldsymbol{x})$, $j=1,2,..,k$
- $g(\boldsymbol{z}) = \boldsymbol{w}^T\boldsymbol{z}$
  - Assume z_1=1 (for taking care of the constant term w_0 as used previously).
- $g(\boldsymbol{x}) = \Sigma_j w_j\, \varphi_j(\boldsymbol{x})$
- No guarantee that the classes are linearly separable in the space of basis functions.
- Similar optimization problem:
  - $L_p = ||\boldsymbol{w}||^2/2 + C\, \Sigma_t\, s^t$
    - subject to $r^t \boldsymbol{w}^T \varphi\, (\boldsymbol{x}^t) \geq 1 - s^t$ for all t
    - C is the penalty factor.

Primal-Dual problems

- Primal problem:
- To minimize w.r.t $\boldsymbol{w}$

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_t s^t - \sum_{t=1}^{N} \alpha^t [r^t(\boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}^t) + w_0) - 1 + s^t] - \sum_t \mu^t s^t$$

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_t \alpha^t r^t \varphi(\boldsymbol{x}^t) \quad \frac{\partial L_p}{\partial s^t} = C - \alpha^t - s^t = 0 \quad \sum_t \alpha^t r^t = 0$$

- Dual problem

Kernel function

$$L_d = -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s \overbrace{(\varphi(\boldsymbol{x}^t))^T \varphi(\boldsymbol{x}^s)}^{K(\boldsymbol{x}^t, \boldsymbol{x}^s)} + \sum_t \alpha^t$$

- Subject to $\sum_t \alpha^t r^t = 0$ and $0 \le \alpha^t \le C$

Kernel machines

- Discriminant function

$$g(\boldsymbol{x}) = w^T \varphi(\boldsymbol{x}) = \sum_t \alpha^t r^t \varphi(\boldsymbol{x}^t)^T \varphi(\boldsymbol{x})$$

$$\Downarrow$$

$$g(\boldsymbol{x}) = \sum_t \alpha^t r^t K(\boldsymbol{x}^t, \boldsymbol{x})$$

A real symmetric nxn matrix M is +ve semidefinite iff $z^T M z \geq 0$ for any non-zero z in R^n.

- No need to compute with basis functions and also performing dot products with z.
- Gram matrix: The matrix of kernel values **K**, where $K_{t,s} = K(x^t, x^s)$
  - Should be symmetric and +ve semidefinite.
  - To be provided.

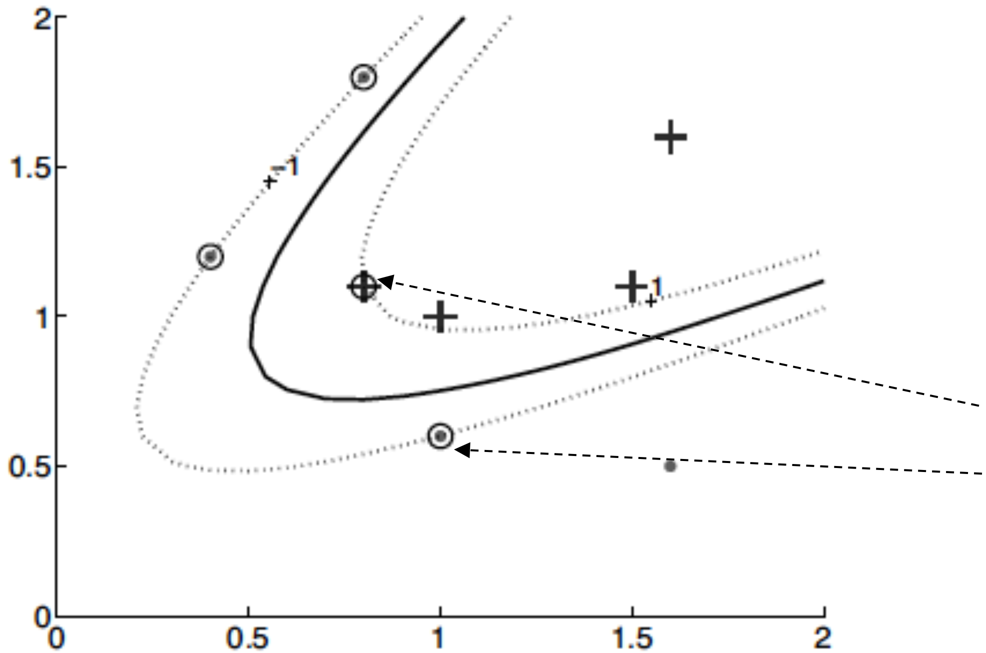
Vectorial kernel functions

- Polynomials of degree q
  - $K(\boldsymbol{x}^t,\mathrm{x})=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}^t+1)^q$
- Radial basis functions
  - $K(\boldsymbol{x}^t,\mathrm{x})=\exp[-\|\boldsymbol{x}-\boldsymbol{x}^t\|^2/(2s^2)]$
- Mahalanobis kernel function
  - $K(\boldsymbol{x}^t,\mathrm{x})=\exp[-(\boldsymbol{x}-\boldsymbol{x}^t)^{\mathrm{T}}S^{-1}(\boldsymbol{x}-\boldsymbol{x}^t)/2]$
- Distance function based
  - $K(\boldsymbol{x}^t,\mathrm{x})=\exp[-\mathrm{D}(\mathbf{x},\mathbf{x}^{\mathrm{t}})\,/(2s^2)]$
- Sigmoidal function: $K(\boldsymbol{x}^t,\mathrm{x})=\tanh(2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}^t+1)$

A typical example:

- Decision Boundaries of Quadratic Kernel in 2D:
  - $K(x,y)=(x^T y+1)^2$

$$\varphi(\boldsymbol{x})=\begin{bmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$$



Circled instances are support vectors.

No need to compute $\varphi(\boldsymbol{x})$.

Courtesy: "Introduction to Machine" Learning by Ethem Alpaydin (Chapter 13, Fig. 13.4)

Defining kernels

- May be defined between a pair of objects flexibly.
  - without using any closed functional form.
- A few examples
  - Number of shared words of two documents.
  - Edit distance between two strings.
  - Number of shared paths between two graphs.
  - Empirical definition of a kernel matrix on training samples.
- The same principle applicable for designing SVM for classifying such objects.

SVM: Summary

- SVM provides maximum margin based linear discrimination for two linearly separable classes.
- Generalizes to non-separable classes by using slack variables.
- Use of basis functions to map non-separable classes to separable in a higher dimensional space.
  - Computation becomes simple and efficient with kernel functions.

Parametric discrimination revisited

- Class densities $P(\boldsymbol{x}|C_i)$ Gaussian sharing a common cov. Matrix: Σ, and $\boldsymbol{\mu}_i$: $E(\boldsymbol{x}|C_i)$.
  - The discriminant function is linear
  - $g_i(\boldsymbol{x})=\boldsymbol{w}_{\boldsymbol{i}}^T\boldsymbol{x}+w_{i0}$
    - $\boldsymbol{w}_{\boldsymbol{i}}= \Sigma^{-1}\,\boldsymbol{\mu}_i$
    - $w_{i0}=-(\boldsymbol{\mu}_i^{\mathrm{T}}\Sigma^{-1}\,\boldsymbol{\mu}_i)/2 + \log P(C_i)$
- Two classes: Let $P(C_1|\boldsymbol{x})=y$, hence $P(C_2|\boldsymbol{x})=1-y$
  - Choose C_1 if y>0.5 $\Leftrightarrow y/(1-y)$>1 $\Leftrightarrow$ log(y/(1-y))>0
  - Else Choose C_2.

logit(y) or log odds of y.

Parametric discrimination revisited

- Two classes: Let $P(C_1|\boldsymbol{x})=y$, hence $P(C_2|\boldsymbol{x})=1-y$
  - Choose C_1 if logit(y)(=log(y/1-y))>0, else Choose C_2.
- For two normal classes sharing a common covariance matrix, the log odds linear.
- $\text{logit}(P(C_1|\boldsymbol{x}))=\boldsymbol{w}^T\boldsymbol{x}+w_0$
  - $\boldsymbol{w}=\Sigma^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)$
  - $w_0=-((\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2))/2+\log P(C_1)/P(C_2)$
- The inverse of logit is the logistic function, also called *sigmoid function*.
- $P(C_1|\boldsymbol{x})=\text{logit}^{-1}(\boldsymbol{w}^T\boldsymbol{x}+w_0)=1/(1+\exp(-(\boldsymbol{w}^T\boldsymbol{x}+w_0)))$

Parametric two class classification using discriminant

- Estimate parameters, Σ, $\boldsymbol{\mu}_1$, and $\boldsymbol{\mu}_2$.
- Compute coefficients of $g(\boldsymbol{x})$: $\boldsymbol{w}$, and w_0.
- During testing:
  - Calculate $g(\boldsymbol{x})$.
  - Assign C_1 if $g(\boldsymbol{x})>0$, else C_2.
- OR
  - Calculate $y=1/(1+\exp(-(\boldsymbol{w}^T\boldsymbol{x}+w_0))$
  - Assign C_1 if $y>0.5$ else C_2.

Logistic discrimination of two classes

- Ratio of class densities modeled: $P(\boldsymbol{x}|C_1)/P(\boldsymbol{x}|C_2)$
- Assume log likelihood ratio is linear
  - true for normal density functions.
- $\log\big(P(\boldsymbol{x}|C_1)/P(\boldsymbol{x}|C_2)\big)= \boldsymbol{w}^T\boldsymbol{x}+w_0{}'$

$\Rightarrow$ $\text{logit}(P(C_1|\boldsymbol{x}))=\log(P(C_1|\boldsymbol{x})/P(C_2|\boldsymbol{x}))$

$=\log\big(P(\boldsymbol{x}|C_1)/P(\boldsymbol{x}|C_2)\big) + \log\big(P(C_1)/P(C_2)\big)$

$= \boldsymbol{w}^T\boldsymbol{x}+w_0$ (when $w_0=w_0{}'+\log(P(C_1)/P(C_2))$)

- $y=P(C_1|\boldsymbol{x}))=1/(1+\exp(-(\boldsymbol{w}^T\boldsymbol{x}+w_0))$

Learning weights of logit functions.

- Data: $X=\{\boldsymbol{x}^t, r^t\}, t=1,2,..N$
  - r^t=1 for C_1, and 0 for C_2.
- Let $y=P(r^t=1|\boldsymbol{x}) \sim \mathrm{Bernoulli}(y)$.
  - Directly modeling likelihood of class assignment
    - instead of likelihood of data given classes as in the parametric approach.

$$l(\boldsymbol{w}, w_0|X) = \prod_{t=1}^{N} (y^t)^{r^t}(1-y^t)^{(1-r^t)}$$

- To minimize E=-log(l) (Maximization of log likelihood)

$$E = -(\sum_t (r^t \log y^t + (1-r^t)\log(1-y^t))$$

- Use gradient descent technique to iterate on weights.

y= 1/(1+exp(-a))

Gradient descent technique

- y=sigmoid(a) ➔ dy/da=y.(1-y)

$$\frac{\partial E}{\partial w_j} = -\sum_t \left(\frac{r^t}{y^t} - \frac{1-r^t}{1-y^t}\right) y^t(1-y^t)x_j^t$$
$$= \ -\textstyle\sum_t (r^t - y^t)\, x_j^t$$
$$\frac{\partial E}{\partial w_0} = -\sum_t (r^t - y^t)$$

- Update of weights at i th iteration.

$$w_j^{(i)} = w_j^{(i-1)} - \eta \frac{\partial E}{\partial w_j}$$

Algorithm (Learning weights)

1. Assume initial $\boldsymbol{w}$, and w_0.
2. Compute y=sigmoid($\boldsymbol{w}^T\boldsymbol{x}+w_0$)
3. Compute gradients.
4. Update $\boldsymbol{w}$, and w_0.
5. Continue steps 2 to 4 till convergence.

Extended to multi-class problem by modeling $P(C_i|\boldsymbol{x})$ by softmax(.) function, and multinomial distr.

$$y_i = P(C_i|\boldsymbol{x}) = \frac{exp(\boldsymbol{w}_i^T\boldsymbol{x} + w_{i0})}{\sum_j exp(\boldsymbol{w}_j^t\boldsymbol{x} + w_{j0})}$$

$$l(\{\boldsymbol{w}_i, w_{i0}\}|X) = \prod_t \prod_i (y_i^t)^{r_i^t}$$

Summary

- Discriminant functions could be explained in the context of Bayesian inference.
- Could be explained by geometry.
- Weights of the function to be learned by minimizing an objective function (error due to miss-classification).
  - Gradient descent method.
  - Stochastic gradient descent.
- Linear SVM: optimally separable hyperplane.
- Logistic discrimination: regresses posterior directly from labelled data.

Thank you!