MACHINE LEARNING IN BIOINFORMATICS

Part 5: Support Vector Machines

František Mráz
KSVI MFF UK

(Sources: slides by Raymond J. Mooney, Chris Manning and Pandu Nayak and tutorial by Anoop M. Namoodiri)

Perceptron

- Given a training set of pairs $\{(x_i, y_i)\}_{i=1,\dots,N}$ where $x_i \in \mathbb{R}^d$, $y_i \in \{-1,1\}$, for $i = 1, \dots, N$ and some integer $d > 0$, we would like to find vector w and a constant b such that
  - $w^T x + b \geq 0$ if $y_i = 1$
  - $w^T x + b < 0$ if $y_i = -1$

> - The elements of w are called weights
> - b is called threshold (or bias)

- **Perceptron learning algorithm:**
  1. Initialize the weight vector $w(0)$ and $b(0)$ with small random values; $t := 0$
  2. For $i = 1, \dots, N$

     Compute the actual output of the perceptron $o = \text{sgn}(w(t)^T x_i + b(0))$

     If $o = y_i$ then $w(t+1) = w(t)$; $b(t+1) = b(t)$;

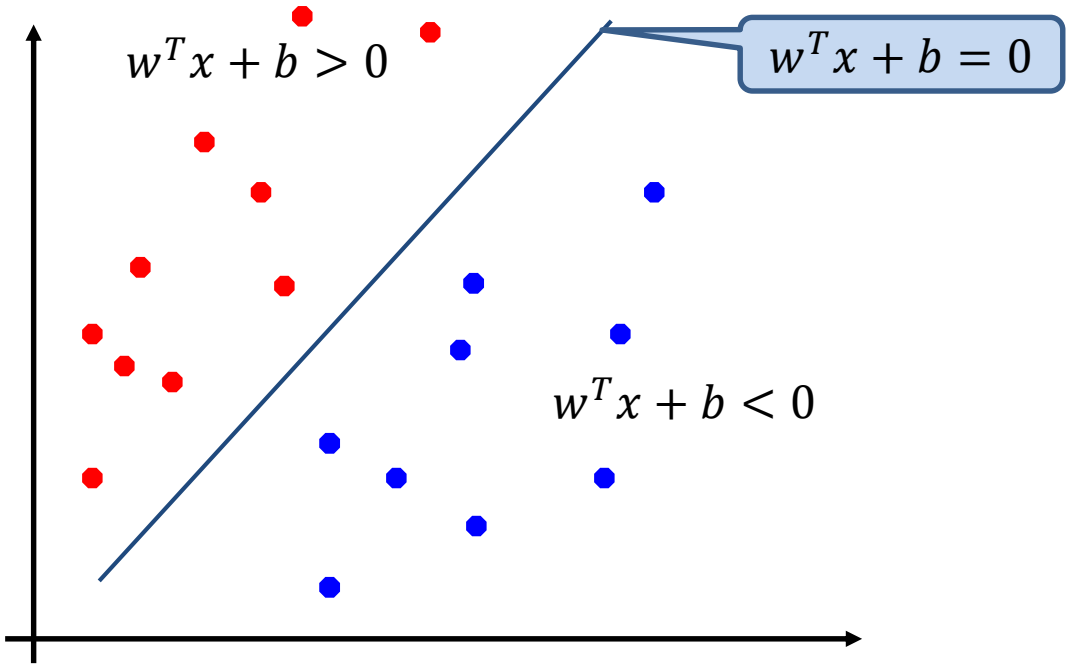
     else if $o = -1$ and $y_i = +1$ then $w(t+1) = w(t) + x_i$; $b(t+1) = b(t) + 1$

     else if $o = +1$ and $y_i = -1$ then $w(t+1) = w(t) - x_i$; $b(t+1) = b(t) - 1$

     $t := t + 1$
  3. Repeat step 2. until some stop condition

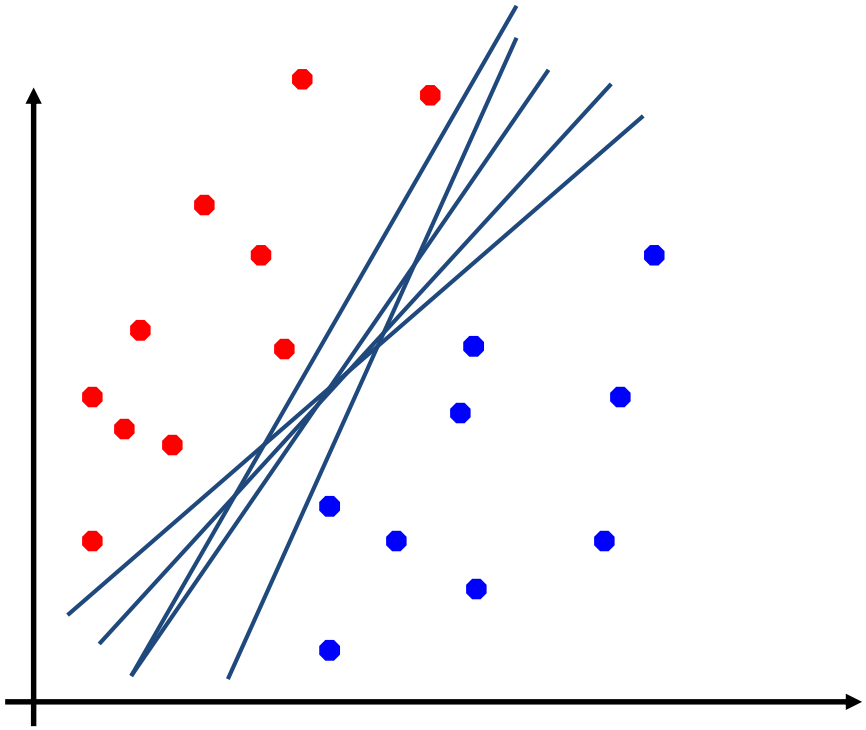
Perceptron

- If the training set is linearly separable, the perceptron learning algorithm always finds a separating hyperplane such that all training samples will be correctly classified



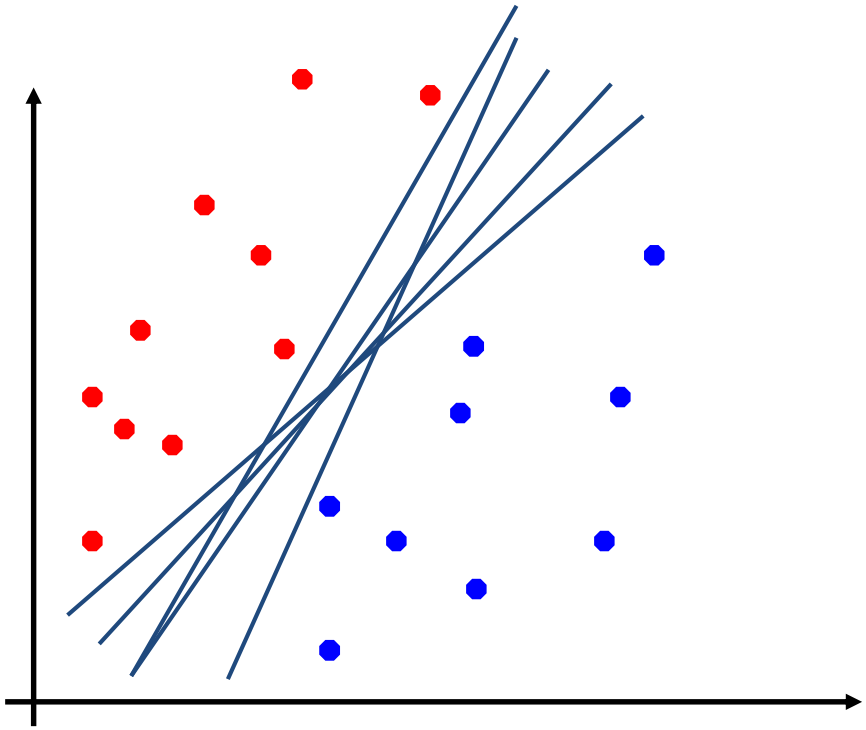
Perceptron

- The perceptron can learn any of the following separating hyperplanes



Perceptron

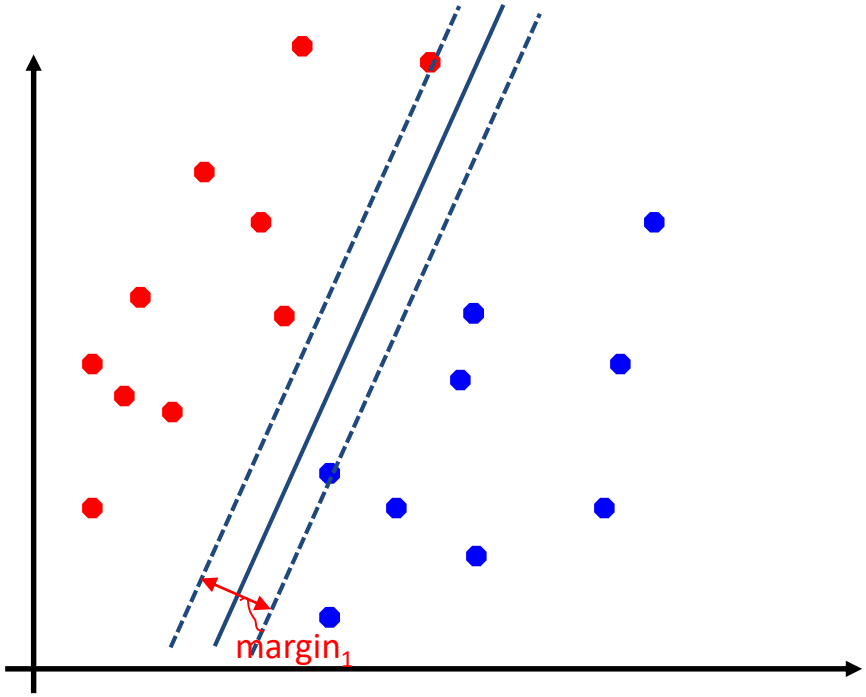
- The perceptron can learn any of the following separating hyperplanes



Are all solutions equally good?

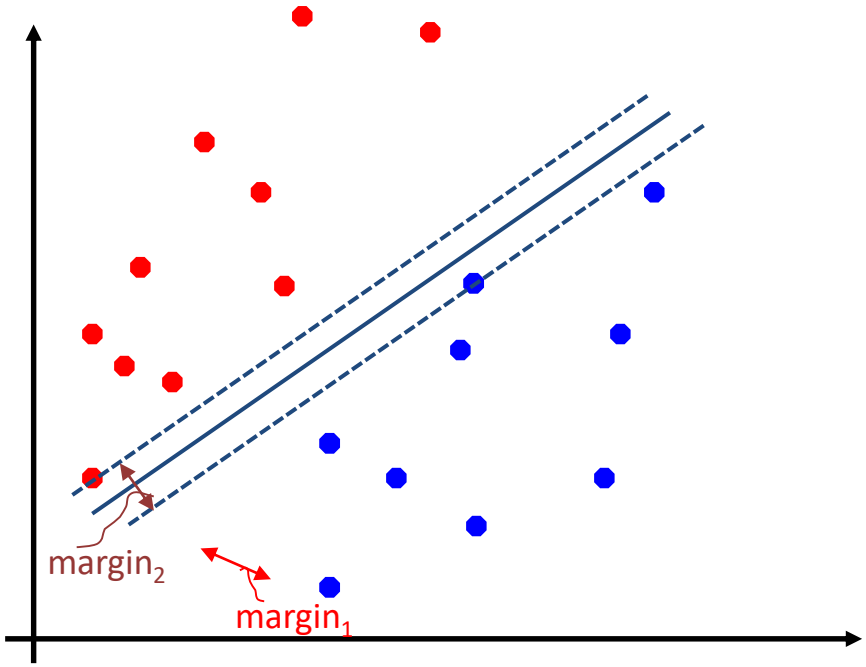
Classification Margin

- Margin is the width of the (symmetric) band around decision boundary without any training samples

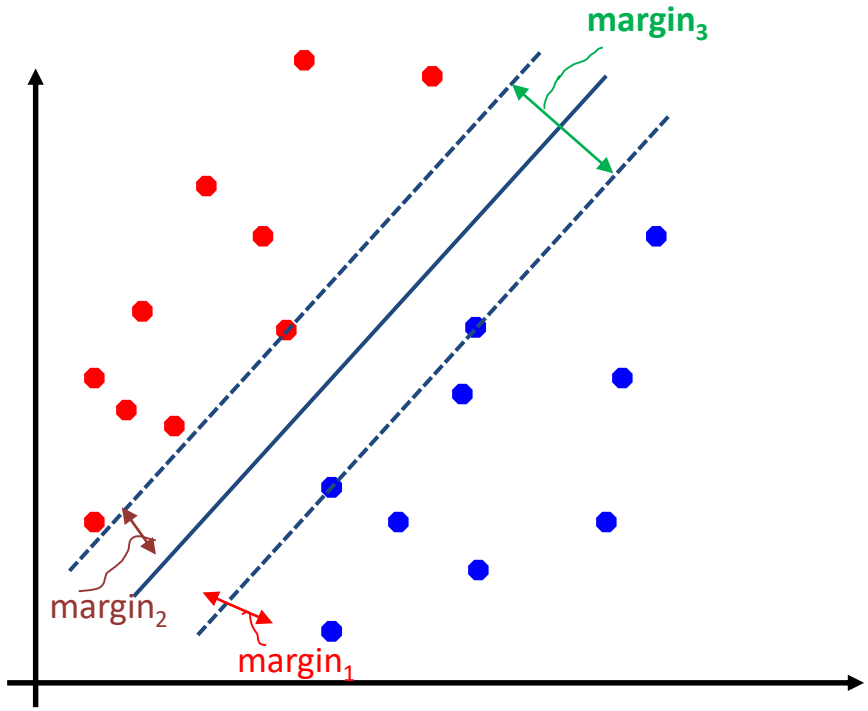


Classification Margin

- Margin is the width of the (symmetric) band around decision boundary without any training samples

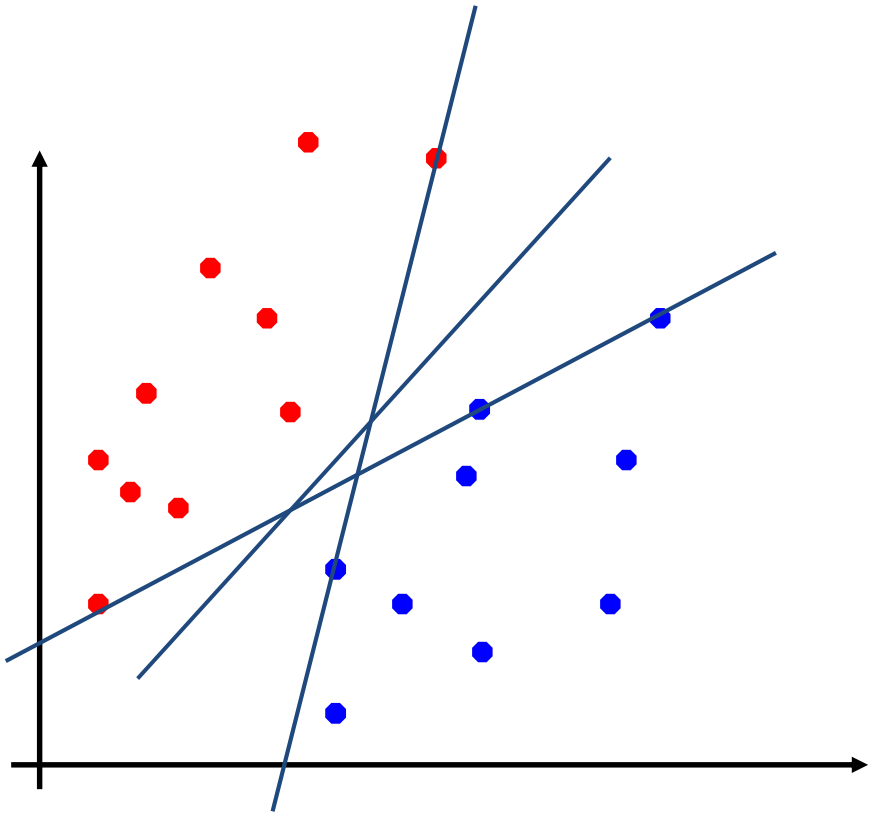


Classification Margin



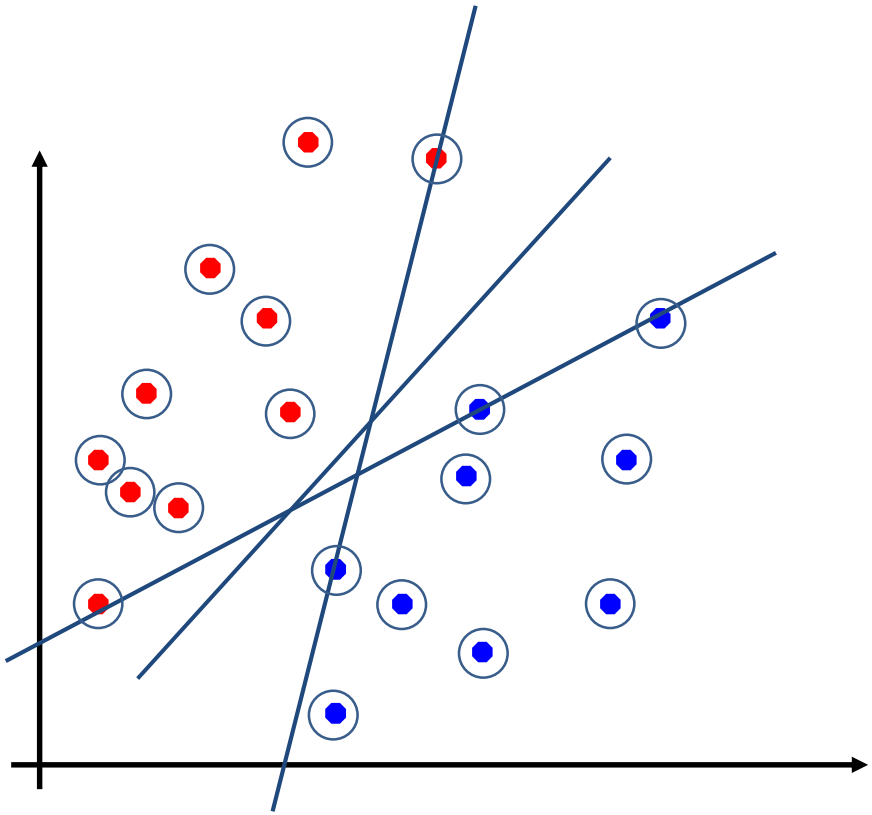
- Margin is the width of the band around decision boundary without any training samples
- **Maximize the margin!**
- *Is larger margin better? Why?*
  - *Intuition, but we will answer this later.*

Classification Margin: Bubbles around Samples



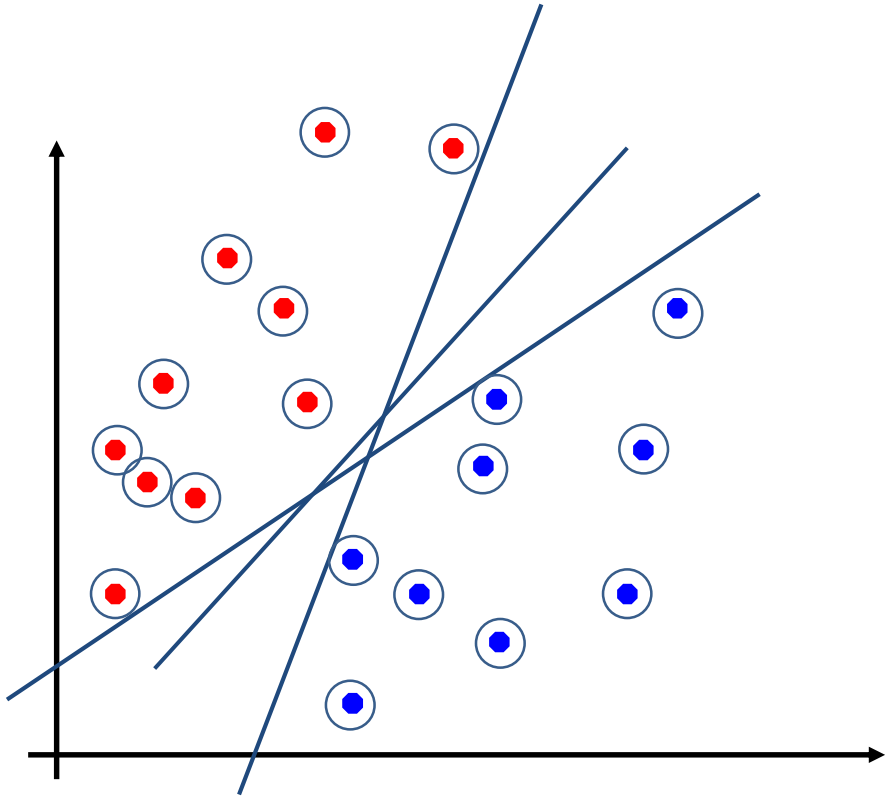
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces

Classification Margin: Bubbles around Samples



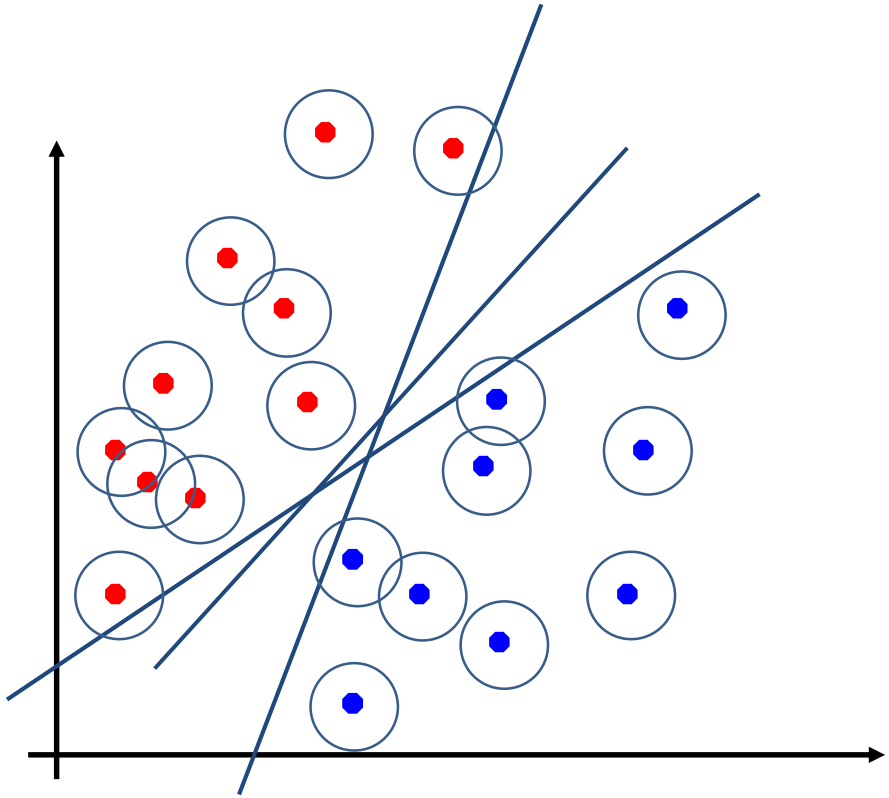
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces

Classification Margin: Bubbles around Samples



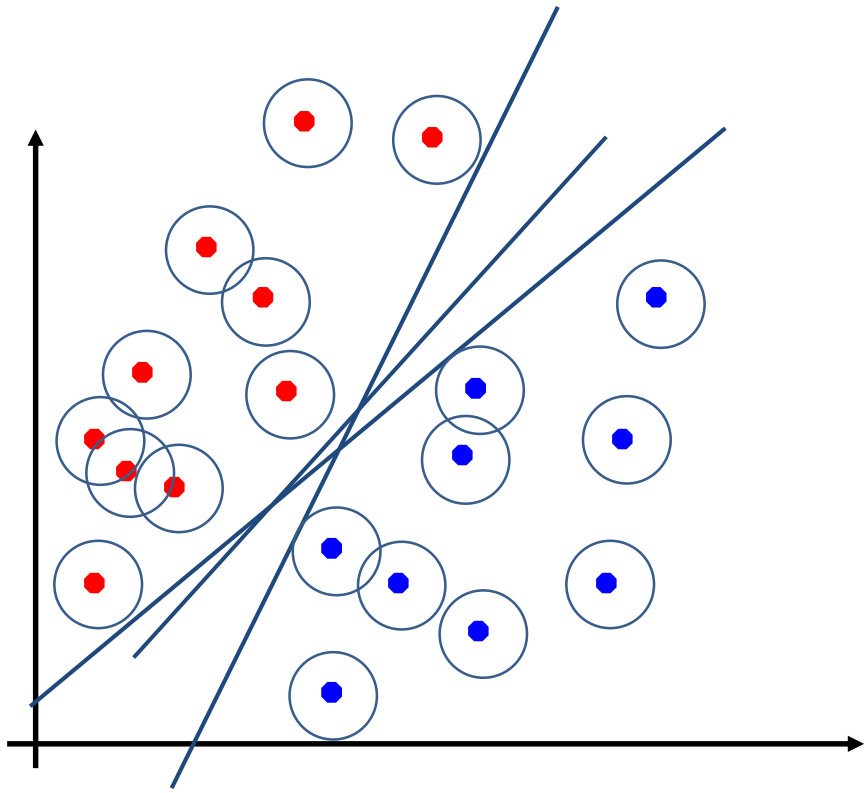
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces

Classification Margin: Bubbles around Samples



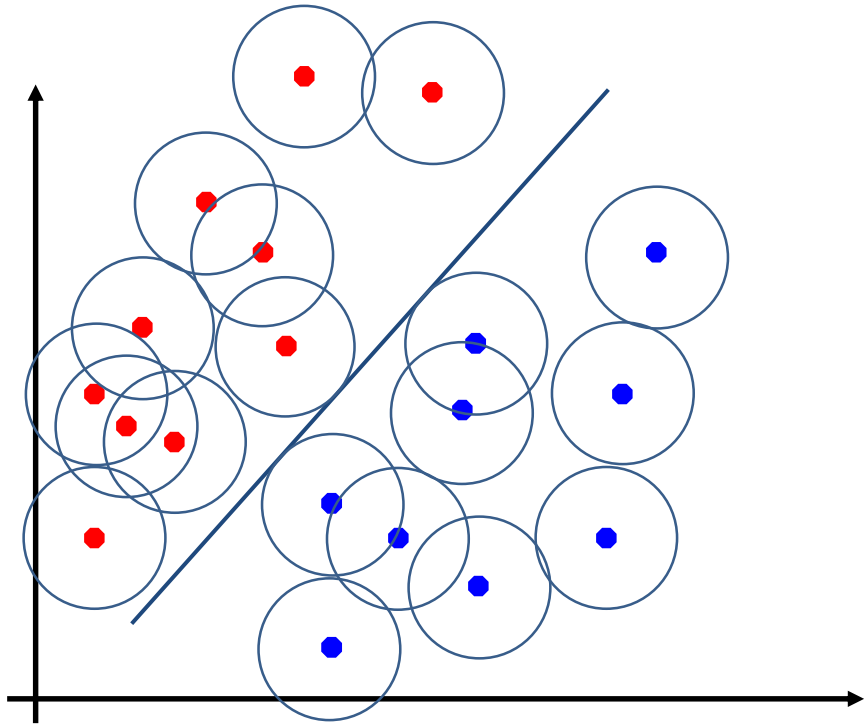
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces

Classification Margin: Bubbles around Samples



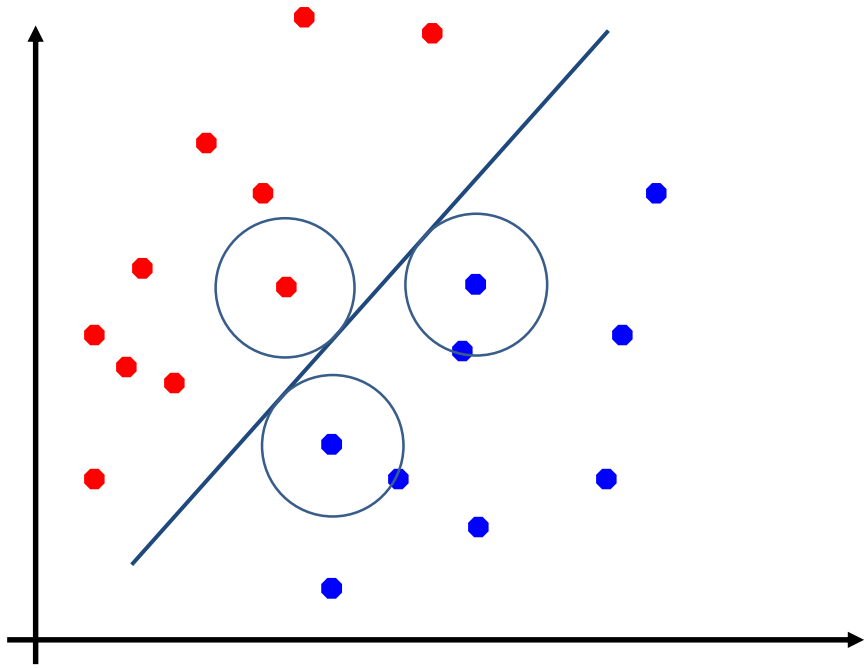
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces

Classification Margin: Bubbles around Samples



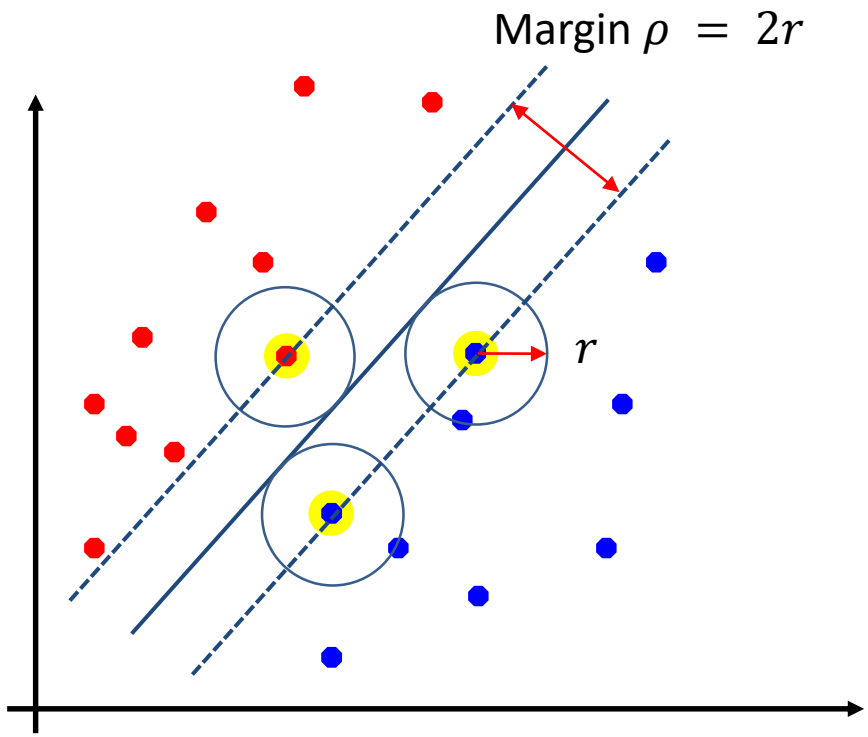
- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region (for the separating hyperplane) reduces – until single feasible separating hyperplane remains

Classification Margin: Bubbles around Samples



- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region reduces – until single feasible separating hyperplane remains
- *Only few circles touch the decision boundary – few samples called* ***support vectors*** *control the decision boundary*

Classification Margin: Band vs. Bubbles



- Margin is the width of the band around decision boundary without any training samples
- As margin increases, the feasible region reduces – until single feasible separating hyperplane remains
- *Only few circles touch the decision boundary – few samples called* ***support vectors*** *control the decision boundary*

Justification for Maximizing the Margin

- Work by Vapnik and Červonenkis in 1970's:
  1. V. N. Vapnik, A. Ya. Červonenkis: On Uniform Convergence of the Frequencies of Events to their Probabilities. Teoria Verojatnosti i Primenenia, 1971, 16(2), pp. 264-279
  2. V. N. Vapnik: Estimation of Dependencies Based on Empirical Data. Nauka, Moscow, 1979 [in Russian]
  3. V. N. Vapnik: The Nature of Statistical Learning Theory. Springer Verlag, New York, 1995

- Bound on expected loss [3]:

$$R(\alpha) \leq R_{train}(\alpha) + \sqrt{\frac{f(h)}{N}}$$

VC dimension

- Let us have a dataset containing n points. These n points can
- be labeled in 2^n ways as positive and negative. Therefore, 2^n different learning problems can be defined by n data points.
- If for any of these problems, we can find a hypothesis $\gamma \in \mathcal{H}$ that separates the positive examples from the negative, then we say $\mathcal{H}$ *shatters* n points. That is, any learning problem definable by n examples can be learned with no error by a hypothesis drawn from $\mathcal{H}$.
- The maximum number of points that can be shattered by $\mathcal{H}$ is called the *Vapnik-Chervonenkis* (VC) *dimension* of $\mathcal{H}$, is denoted as h
- Ex.: for $\mathcal{H}$ = set of all rectangles in 2D, $h = 4$
- Ex.: for $\mathcal{H}$ = set of all hyperplanes in d dimensional space, $h = d + 1$

Justification for Maximizing the Margin

- Bound on expected loss [3]:

$$R(\alpha) \leq R_{train}(\alpha) + \sqrt{\frac{f(h)}{N}}$$

  - $R(\alpha)$ is the risk (loss) when we select decision boundary α
  - $R_{train}(\alpha)$ is the risk (loss) from training when we select decision boundary α
  - h is a VC-dimension, $f(h) = h + h\log(2N) - h\log(h) - c$
  - $\sqrt{\frac{f(h)}{N}}$ is a monotonically increasing function

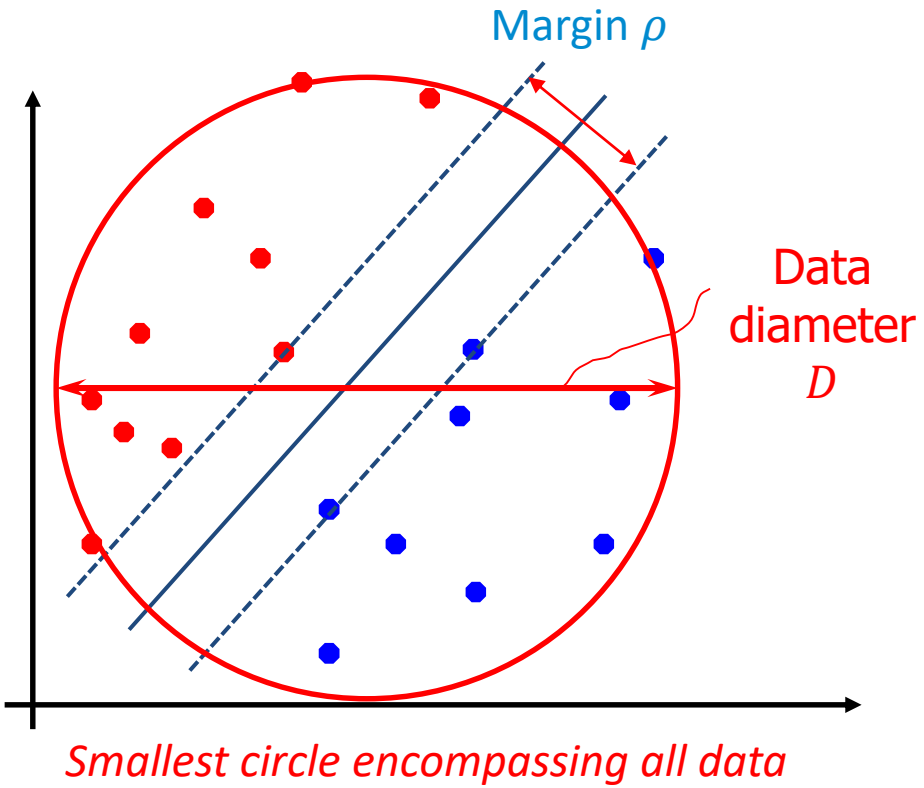
  - "Future error on a (new) test sample is limited by the error on the training samples plus a monotonically increasing function of h."

Why Maximize the Margin?

- To reduce test error, keep training error low (say 0) and minimize the VC-dimension h
- Relative margin $\frac{\rho}{D}$
- VC-dimension is limited

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

- Regardless of dimensionality d, we can minimize VC-dimension by maximizing the margin ρ – we can make h independent of the dimensionality d.
- Maximizing margin improves generalization.



Formalizing the Margin

- Let a training set of pairs $\{(x_i, y_i)\}_{i=1,\dots,N}$ where $x_i \in \mathbb{R}^d$, $y_i \in \{-1,1\}$, for $i = 1, \dots, N$ and some integer $d > 0$, be separated by a hyperplane, then for each training sample (x_i, y_i)

$$\left.\begin{array}{ll} w^T x + b \geq 0 & \text{if } y_i = +1 \\ w^T x + b \leq 0 & \text{if } y_i = -1 \end{array}\right\} \Leftrightarrow y_i(w^T x_i + b) \geq 0$$

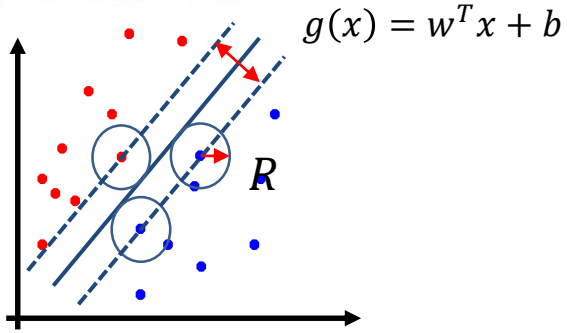
- For every support vector x_s the its distance from the separating hyperplane $w^T x + b = 0$ is

$$\frac{\rho}{2} = \frac{|w^T x_s + b|}{\|w\|} = \frac{y_s(w^T x_s + b)}{\|w\|}$$

- Then for all vectors x we have:

$$y_i(w^T x_i + b) \geq \frac{\rho}{2}\|w\|$$

Formalizing the Margin



- Then for all vectors x we have:

$$y_i(w^T x_i + b) \geq \frac{\rho}{2}\|w\|$$

- We want to maximize ρ, hence we can
  1) Either keep $\|w\| = 1$, and maximize $y(w^T + b)$, or
  2) let $y(w^T + b) \geq 1$, and minimize $\frac{1}{2}\|w\|$
- We use approach 2) and formulate the problem as

Minimize: $\frac{1}{2} w^T w$
Subject to: $y_i(w^T x_i + b) - 1 \geq 0, \ \forall i$

The Optimization Problem

Minimize: $\frac{1}{2} w^T w$
Subject to: $y_i(w^T x_i + b) - 1 \geq 0, \ \forall i$

- Quadratic objective function with linear inequalities as constraints $\Rightarrow$ use a QP solver
- Integrating the constraints into the Lagrangian form, we get:

Minimize: $J(w, b, \boldsymbol{\alpha}) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^{N} \alpha_i$
Subject to: $\alpha_i \geq 0 \ \forall i$

- Minimize J with respect to w and b, and maximize with respect to $\boldsymbol{\alpha}$
  - search for a saddle point

Lagrange Theory
Karush-Kuhn-Tucker Condition

- Min. $J(w)$
- S.t. $h_j(w) = 0$ m equality constraints
  $g_i(w) \leq 0$ N inequality constraints

- Lagrangian: $L(w, \alpha, \beta) = f(x) + \sum_{i=1}^{N} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w)$
- If all $J(w), h_j(w)$ and $g_i(w)$ are convex functions, a solution exists where for the optimal value J^* of $J(w)$ it holds:

Primal problem

$$J^* = \min_{w} \max_{\boldsymbol{\alpha} \geq 0, \beta} L(w, \boldsymbol{\alpha}, \beta) = \max_{\boldsymbol{\alpha} \geq 0, \beta} \min_{w} L(w, \boldsymbol{\alpha}, \beta) = D^*$$

Dual problem

1. Gradient of the Lagrangian: $\nabla L = 0$
2. Constraints: $h_j(w^*) = 0$ & $g_i(w^*) \leq 0$
3. Complementary Slackness: $\alpha_i^* g_i(w^*) = 0$
4. Sign condition on the inequality multipliers: $\alpha_i^* \geq 0$

w^*, α^* optimal values

Lagrange Theory
Karush-Kuhn-Tucker Condition

$$J^* = \min_{w} \max_{\boldsymbol{\alpha} \geq 0, \beta} L(w, \boldsymbol{\alpha}, \beta) = \max_{\boldsymbol{\alpha} \geq 0, \beta} \min_{w} L(w, \boldsymbol{\alpha}, \beta) = D^*$$

1. Gradient of the Lagrangian: $\nabla L = 0$
2. Constraints: $h_j(w^*) = 0$ & $g_i(w^*) \leq 0$
3. Complementary Slackness: $\alpha_i^* g_i(w^*) = 0$
4. Sign condition on the inequality multipliers: $\alpha_i^* \geq 0$

- For convex problems KKT conditions are necessary and sufficient condition for primal and dual solution

Converting to the Dual Form

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

$$1: \quad \frac{\partial J(w, b, \boldsymbol{\alpha})}{\partial w} = 0 \quad \text{and} \quad 2: \frac{\partial J(w, b, \boldsymbol{\alpha})}{\partial b} = 0$$

Karush-Kuhn-Tucker condition

- At optimum

$$1: \quad w^* = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad 2: \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad 3: \quad \alpha_i [y_i (w^{*T} x_i + b^*) - 1] = 0$$

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

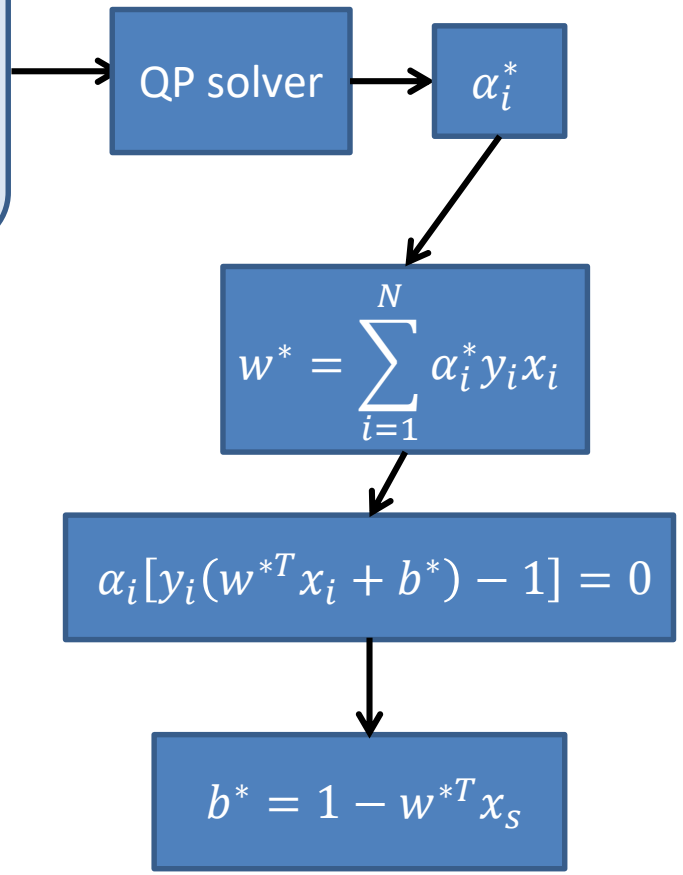
- Obj. $D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i + \frac{1}{2} w^T w - w^T \sum_{i=1}^{N} \alpha_i y_i x_i - b \sum_{i=1}^{N} \alpha_i y_i$

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

Solving the Dual Form

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

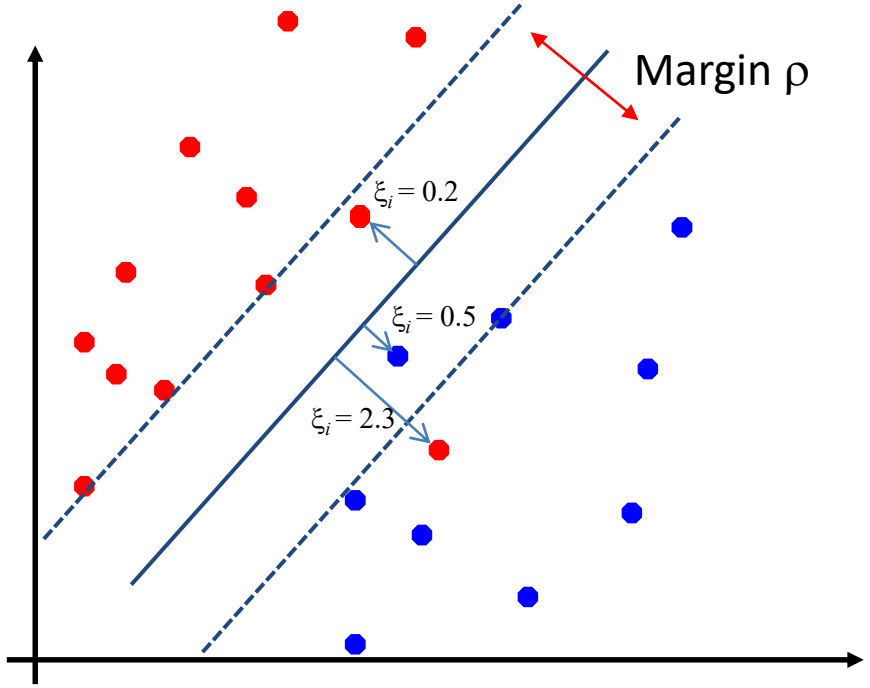
Subject to: $\alpha_i \geq 0 \ \forall i$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$



- The only unknowns (variables) are α_i's.
- The constraints are also on α_i's only.
- Data vectors appear only as dot products.
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers.

Noise in Data
Non-Separable Classes

Minimize: $\frac{1}{2}w^T w$
Subject to: $y_i(w^T x_i + b) - 1 \geq 0, \forall i$



- Introduce slack variables $\xi_i \geq 0$

Minimize: $\frac{1}{2}w^T w$
Subject to: $y_i(w^T x_i + b) \geq 1 - \xi_i, \forall i$

- Also minimize training error $\sum_{i=1}^{N} \xi_i$

Minimize: $\frac{1}{2}w^T w + C\sum_{i=1}^{N} \xi_i$ $\quad C > 0$ const

Subject to: $y_i(w^T x_i + b) \geq 1 - \xi_i;\ \xi_i \geq 0,\ \forall i$

Dual Form with Slack Variables

- Form the Lagrangian and convert it to dual problem

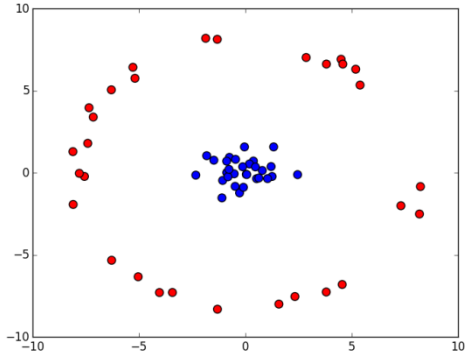
Maximize $D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$

Subject to: $0 \leq \alpha_i \leq C, \ \forall i$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

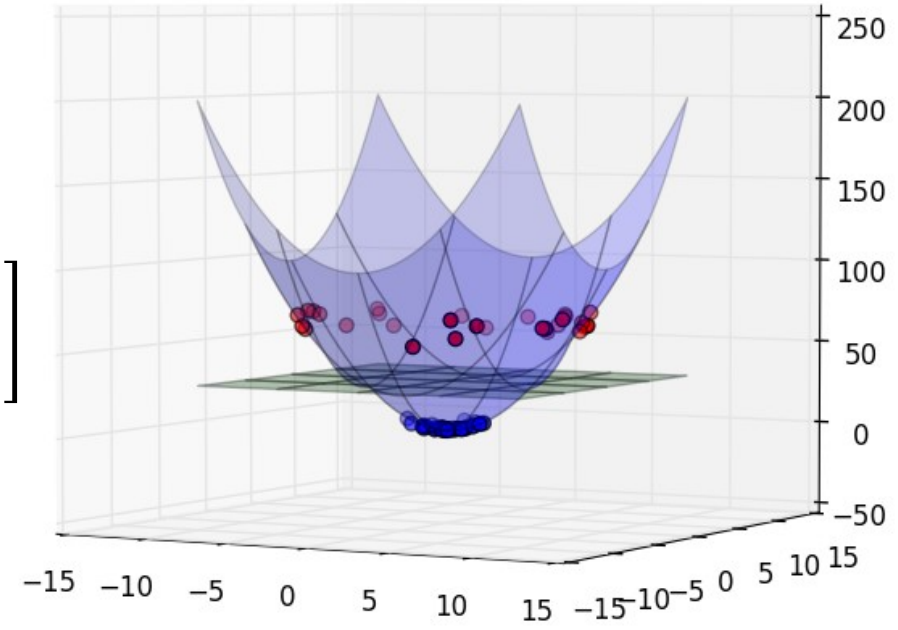
- Note that neither the slack variables, nor their Lagrange multipliers appear in the dual problem.
- The only change is the additional constraint on α_i
- The parameter C controls the relative strength between training error and the VC dimension

Non-Separable Classes
Non-Linear Boundaries

- Add a new feature by a nonlinear mapping Φ into a higher-dimensional space



$$\Phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$



SVM with the Mapping Φ

- Data vectors occur only in dot products in SVM-learning and testing
  - Training

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

Subject to $0 \leq \alpha_i \leq C, \forall i,$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

  - Testing

$$Label(x_{test}) = sign(w^* \cdot \Phi(x_{test}) + b_o)$$

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i \Phi(x_i)$$

$$Label(x_{test}) = sign\left(\sum_{i=1}^{N} (\alpha_i^* y_i \Phi(x_i) \cdot \Phi(x_{test})) + b^* \right)$$

$\boldsymbol{a} \cdot \boldsymbol{b}$ denotes dot product $\boldsymbol{a}^T \boldsymbol{b}$

SVM with the Mapping Φ

- Data vectors occur only in dot products in SVM-learning and testing
- If there exists a function $K(x, y)$ such that $K(x, y) = \Phi(x) \cdot \Phi(y)$ which is easier to compute than two mappings into a higher-dimensional space and dot-product, we can be more effective

$$D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$Label(x_{test}) = sign\left(\sum_{i=1}^{N} (\alpha_i y_i K(x_i, x_{test})) + b^* \right)$$

K is called **kernel** function

Sample Kernels: A Simple Quadratic Kernel

- Let $\Phi(x) = \Phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_2x_1 \\ x_2x_2 \end{bmatrix}$

Instead of mapping with Φ and computing dot product, we can compute $K(x, y) = (x \cdot y)^2$

- Let

Here y is a vector of the same size as the vector x

$$K(x, y) = \Phi(x) \cdot \Phi(y) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ y_1y_2 \\ y_2y_1 \\ y_2^2 \end{bmatrix} = x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2$$

$$= (x_1y_1 + x_2y_2)^2 = (x \cdot y)^2$$

(3 mult.+1 add) instead of (10 mult.+3 add)

Sample Kernels: A Cubic Kernel

- Original 3-dimensional space mapped into 10 dimension
  Let $K(x,y) = (x \cdot y)^3 = (x_1y_1 + x_2y_2 + x_3y_3)^3$

- Actually we have only $K(x,y) \approx \Phi(x) \cdot \Phi(y)$

(5 mult.+2 add)
instead of
(38 mult.+9 add)

$$\Phi(x) = \Phi\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \\ x_1^2x_2 \\ x_1x_2^2 \\ x_1^2x_3 \\ x_1x_3^2 \\ x_2^2x_3 \\ x_2x_3^2 \\ x_1x_2x_3 \end{bmatrix}$$

A Generic Polynomial Kernel

- Adding two kernels results in a new kernel

$$K(x,y) = K_1(x,y) + K_2(x,y): \quad \Phi(x) = \begin{bmatrix} \Phi_1(x) \\ \Phi_2(x) \end{bmatrix}$$

$$K(x,y) = \Phi(x) \cdot \Phi(y) = \Phi_1(x) \cdot \Phi_1(y) + \Phi_2(x) \cdot \Phi_2(y)$$

$$K_p(x,y) = (1 + x \cdot y)^p = 1 + x \cdot y + (x \cdot y)^2 + \cdots + (x \cdot y)^p$$

Coefficients are left out

- Adding 1 and raising to the power of p maps the input vector into a space containing all original dimensions, all 2-products, 3-products, …, p-products

Which Functions are Kernels?

- A kernel function is a function which is equivalent to an inner product in some feature space
- **Example:**
  - 2-dimensional vectors
  - Let $K(x, y) = (1 + x \cdot y)^2$
  - we must show that $K(x, y) = \Phi(x) \cdot \Phi(y)$, for some function $\Phi(\cdot)$

$$K(x, y) = (1 + x \cdot y)^2 = 1 + x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 =$$
$$= \left[1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2\right]^T \left[1, y_1^2, \sqrt{2}y_1 y_2, y_2^2, \sqrt{2}y_1, \sqrt{2}y_2\right] =$$
$$= \Phi(x) \cdot \Phi(y),$$

where $\Phi(x) = \left[1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2\right]$

Which Functions are Kernels?

- Checking that $K(x, y) = \Phi(x) \cdot \Phi(y)$, for some function $\Phi(\cdot)$ can be cumbersome
- In fact, we do not even have to know Φ exactly as long as we are sure that it exists
- A necessary and sufficient condition for a function $K(x, y)$ to be a valid kernel is that for each sequence of vectors $\{x_i\}_{i=1,\ldots,N}$, the Gram matrix

$$\begin{pmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_N, x_1) & K(x_n, x_2) & \cdots & K(x_N, x_N) \end{pmatrix}$$

should be positive semidefinite, i.e. $x^T K x \geq 0$, for all vectors x

Further Examples of Kernels

- Linear: $K(x, y) = x^T y$
  - Mapping $\Phi(x)$ is x itself, i.e. the same dimension
- Polynomial of degree p: $K_p(x, y) = (1 + x^T y)^p$
  - Mapping $\Phi(x)$ into $\binom{d+p}{p}$ dimensions
- Gaussian (radial basis function): $K(x, y) = e^{-\frac{\|x-y\|^2}{2e^2}}$
  - Mapping $\Phi(x)$ is infinite dimensional – every point is mapped to a function (a Gaussian); the separator is a combination of the functions of the support vectors
- Higher order space still has intrinsic dimensionality the same as x itself, but linear separator in it corresponds to a non-linear separator in the original space

Further Methods for Constructing Kernels

- Let $K_1(x, y)$ and $K_2(x, y)$ be two valid kernel functions, then the following functions are kernels too ($c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with non-negative coefficients, $\varphi(x)$ is a function from $\mathcal{X}$ to $\mathbf{R}^M$, $k_3(\cdot,\cdot)$ is a valid kernel in $\mathbf{R}^M$, $\boldsymbol{A}$ is a symmetric positive semidefinite matrix, x_a and x_b are variables (not necessarily disjoint) with $x = (x_a, x_b)$, and k_a and k_b are valid kernel functions over their respective spaces):
  - $K(x, y) = c\, K_1(x, y)$
  - $K(x, y) = f(x) K_1(x, y) f(y)$
  - $K(x, y) = q(K_1(x, y))$
  - $K(x, y) = \exp(K_1(x, y))$
  - $K(x, y) = K_1(x, y) + K_2(x, y)$
  - $K(x, y) = K_1(x, y) K_2(x, y)$
  - $K(x, y) = K_3(\varphi(x), \varphi(y))$
  - $K(x, y) = x^T \boldsymbol{A} y$
  - $K(x, y) = k_a(x_a, y_a) + k_b(x_b, y_b)$
  - $K(x, y) = k_a(x_a, y_a) k_b(x_b, y_b)$

Application of SVM

- Multi-class classification
  - most widely used method: one versus all
  - Also exists a direct multi-classification using SVM
- For SVM optimization, every local solution is global due to the property of the convex objective function.
- The solution is guaranteed to be unique.
- SVM training always finds a global solution is in contrast to the case of neural networks, where many local minima usually exist.

Method of Solution

- The support vector optimization problem can be solved
  1. Analytically:
     - only when the number of training data is very small, or for the separable case when it is known beforehand which of the training data become support vectors.
     - For the general analytic case, the worst case computational complexity is of order N_S^3 (inversion of Hessian), where N_S is the number of support vectors.
  2. Numerically:
     - In most real world cases, the quadratic optimization problem must be solved numerically.
     - For small problems, any general purpose optimization package that solves linearly constrained convex quadratic programs will do
     - For large problems, divide and conquer technique is usually used

Time Complexity of Testing

- $O(MN_s)$, where
  - M is the number of operations required to evaluate the kernel. For RBF kernel, M is $O(d)$.
  - N_s is the number of support vectors.

SVM Applications in Bioinformatics

- Cancer Classification using Gene Expression Data
- Protein Mutation Stability Prediction
- Protein Secondary Structure Prediction
- Protein Fold Recognition
- Protein Contact Map Prediction
- Protein Structure Classification
- ……

Current Cancer Diagnosis

- *A reliable and precise classification of tumors is essential for successful treatment of cancer.*
- *Current methods relies on the subjective interpretation of both clinical histopathological information with an eye toward placing tumors in currently accepted categories based on the tissue of origin of the tumor.*
- *However, clinical information can be misleading or incomplete.*
- *there is a wide spectrum in cancer morphology and many tumors are atypical or lack morphologic features, which results in diagnostic confusion.*

Jia Yi, 2005

DNA Microarray-Based Cancer Diagnosis

- Molecular diagnostics offer the promise of precise, objective, and systematic cancer classification.
- Recently, DNA microarray tumor gene expression profiles have been used for cancer diagnosis.
- By allowing the monitoring of expression levels for thousands of genes simultaneously, such techniques will lead to a more complete understanding of the molecular variations among tumors, hence to a finer and more reliable classification.

Tumor Classification Types

- There are three main types of statistical problems associated with tumor classification:
  - The identification of new tumor classes using gene expression profiles – unsupervised learning.
  - The classification of malignancies into known classes – supervised learning.
  - The identifications of "marker" genes that characterize the different tumor classes – variable selection.

- Cancer datasets:
  http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi