Support Vector Machines

Dionysios N. Sotiropoulos
Ph.D
(dsotirop@gmail.com)

Presentation Summary

- Introduction
- Theoretical Justifications
- Linear Support Vector Machines
  - Hard Margin Support Vector Machines
  - Soft Margin Support Vector Machines
- Non-Linear Support Vector Machines
  - Mapping Data to High Dimensional Feature Spaces
  - Kernel Trick
  - Kernels
- Conclusions

Theoretical Justifications (1 / 6)

- Training Data:
  - We want to estimate a function $f : R^N \to \{\pm 1\}$ using training data $(x_1, y_1), \ldots, (x_l, y_l) \in R^N \times \{\pm 1\}$.
- Empirical Risk:
  - measures classifier's accuracy on training data

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l} \frac{1}{2}\left| f(x_i) - y_i \right|$$

- Risk:
  - measures classifier's generalization ability:

$$R[f] = \int \frac{1}{2}\left| f(x) - y \right| dP(x, y)$$

Theoretical Justifications (2 / 6)

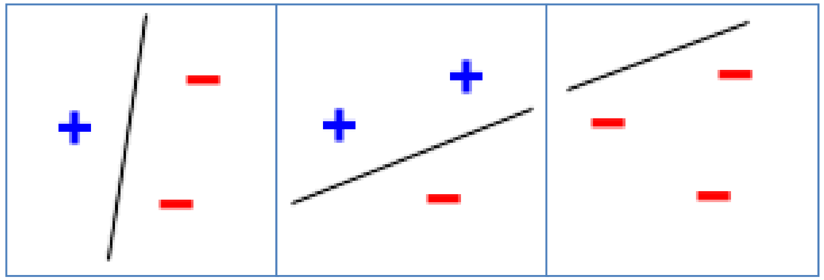
- **Structural risk minimization (SRM)** is an inductive principle.
- Commonly in machine learning, a generalized model must be selected from a finite data set, with the consequent problem of overfitting the model becoming too strongly tailored to the particularities of the training set and generalizing poorly to new data.
- The SRM principle addresses this problem by balancing the model's complexity against its success at fitting the training data.

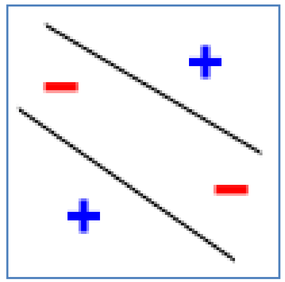
Theoretical Justifications (3 / 6)

- VC Dimension: **Vapnik – Chervonenkis dimension** is a measure of the capacity of a statistical classification algorithm defined as the cardinality of the largest set of points that the algorithm can shatter.
- Shuttering:
  - a classification model *f(*θ*)* with some parameter vector θ is said to *shatter* a set of data points $X = \{x_1, \ldots, x_l\}$ if, for all assignments of labels to those points, there exists a θ such that the model *f* makes no errors when evaluating that set of data points.

Theoretical Justifications (4 / 6)

- Examples:
  - consider a straight line as the classification model: the model used by a perceptron.
  - The line should separate positive data points from negative data points.
  - An arbitrary set of 3 points can indeed be shattered using this model (any 3 points that are not collinear can be shattered).
  - However, there exists a set of 4 points that can not be shattered. Thus, the VC dimension of this particular classifier is 3.





Theoretical Justifications (5 / 6)

- VC Theory provides bounds on the test error, which depend on both empirical risk and capacity of function class.
- The bound on the test error of a classification model (on data that is drawn i.i.d from the same distribution as the training set) is given by:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log\frac{2l}{h}+1)-\log(\frac{\eta}{4})}{l}}$$

with probability $1 - \eta$.

where *h* is the VC dimension of the classification model, and l is the size of the training set (restriction: this formula is valid when the VC dimension is small $h < l$).

Theoretical Justifications (6 / 6)

- Vapnik has proved the following:

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

  $$h \le \min\left\{\left\lceil \frac{D^2}{\gamma^2} \right\rceil, n\right\} + 1$$

  - *where γ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and n is the dimensionality.*

Introduction 1 / 2

- SVMs gained much popularity as the most important recent discovery in machine learning.
- In binary pattern classification problems
  - generalize linear classifiers in high-dimensional feature spaces through non-linear mappings defined implicitly by kernels in Hilbert space.
  - produce non-linear classifiers in the original space.

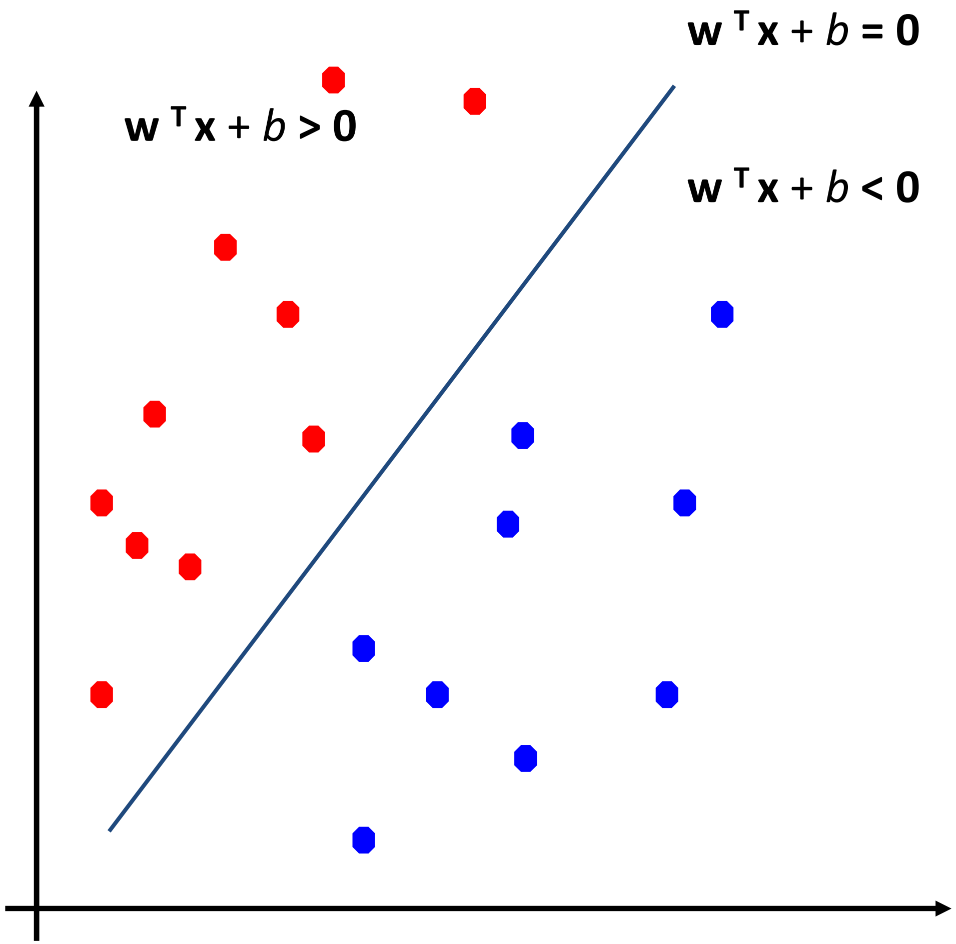
Introduction 2 / 2

- Initial linear classifiers are optimized to give maximal margin separation between classes.
- This task is performed by solving some type of mathematical programming such as quadratic programming (QP) or linear programming (LP).

Hard Margin SVM 1 /26

- Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ be a set of training patterns such that $x_i \in \Re^n$ and $y_i \in \{-1, 1\}$.
- Each training input belongs to one of two disjoints classes which are associated with the labels $y_i = +1$ and $y_i = -1$.
- If data points are linearly separable, it is possible to determine a decision function of the following form: $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \langle \mathbf{w}, \mathbf{x} \rangle + b$

Hard Margin SVM 2 / 26



$g(\mathbf{x}) = <\mathbf{w}^T, \mathbf{x}> + b$

Hard Margin SVM 3 / 26

- The decision function $g(\mathbf{x})$ defines a hyper plane in the n-dimensional vector space $\Re^n$ which has the following property:

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = \begin{cases} > 0, & \text{for } y_i = +1; \\ < 0, & \text{for } y_i = -1. \end{cases}$$

- Since training data are linearly separable, there will not be any training instances satisfying: $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$

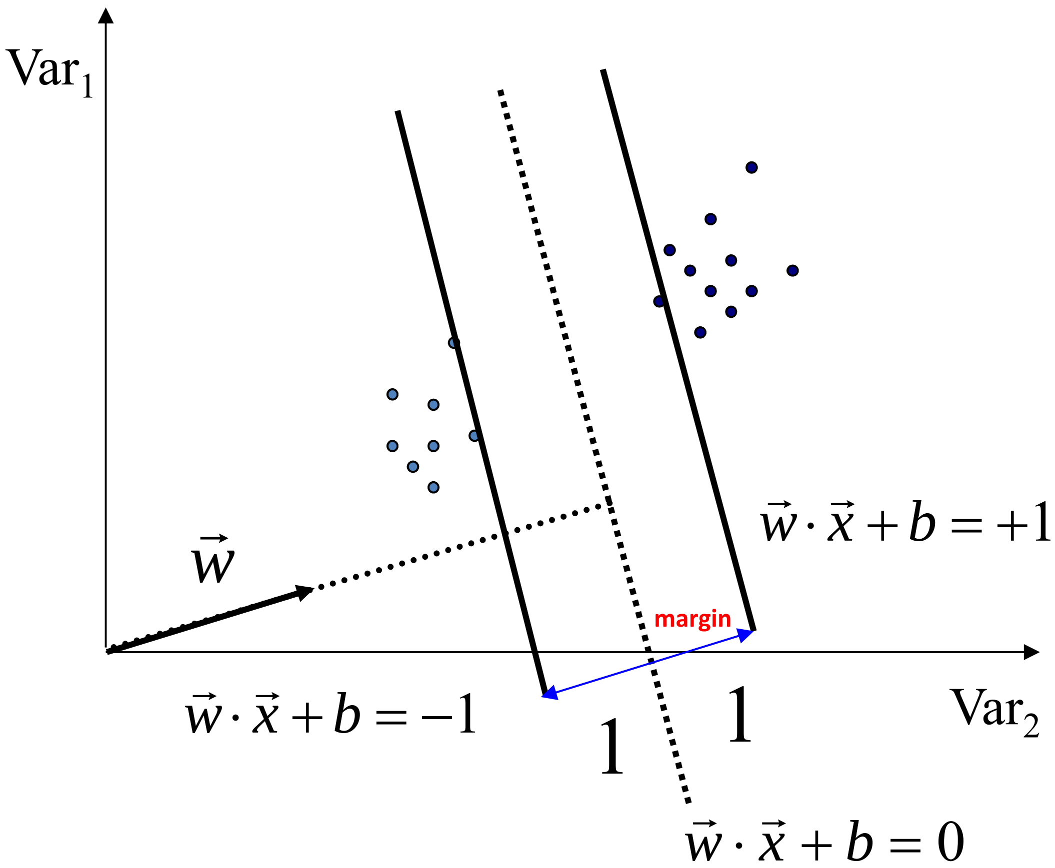
Hard Margin SVM 4 / 26

- In order to control separability we may write that:

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = \begin{cases} \geq +1, & \text{for } y_i = +1; \\ \leq -1, & \text{for } y_i = -1. \end{cases}$$

- By incorporating class labels, inequalities may be rewritten as: $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \forall i \in [l]$

Hard Margin SVM 5 / 26



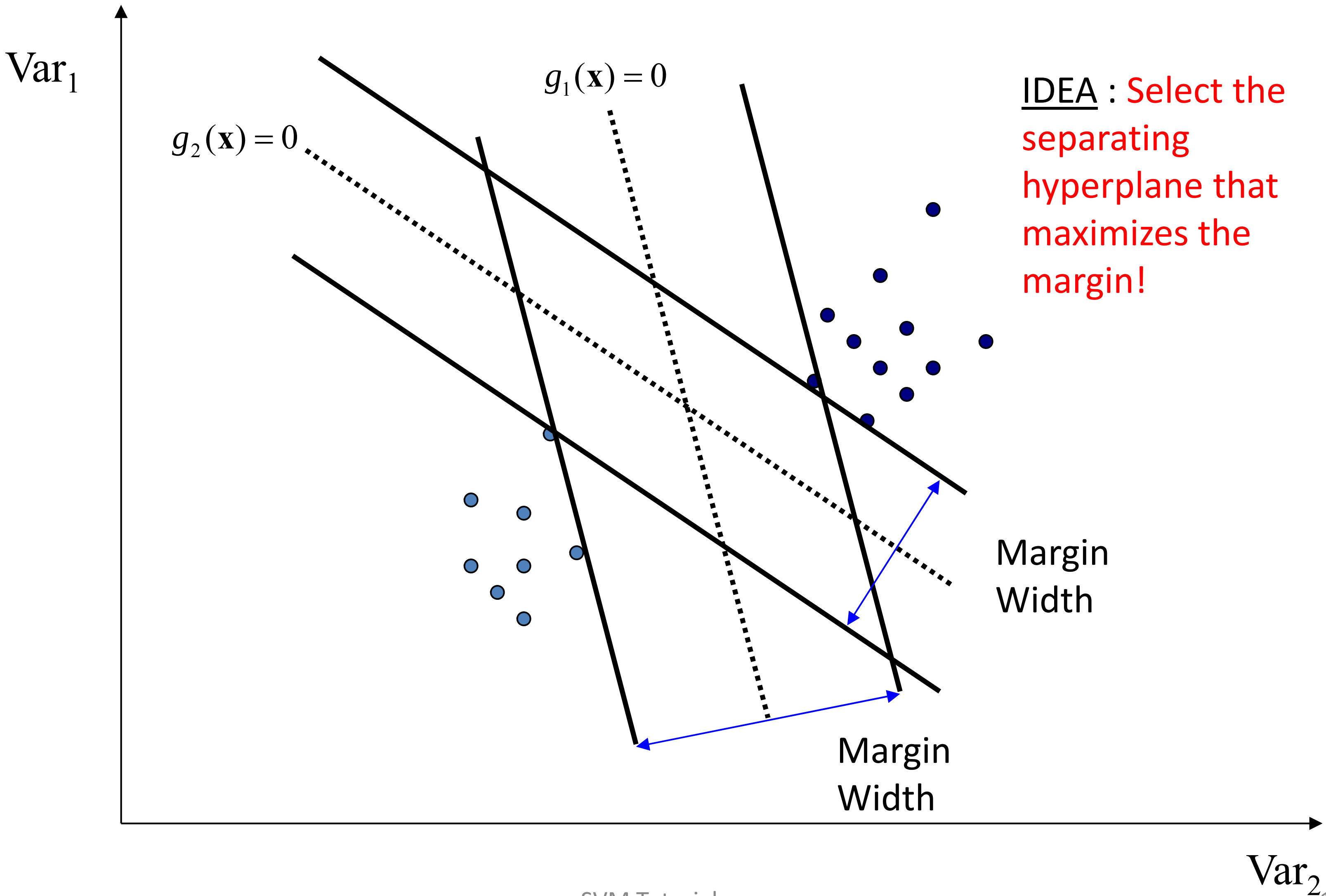
Hard Margin SVM 6 / 26

- The hyperplane $g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = c$ for $-1 < c < +1$ forms a separating hyperplane in the n-dimensional vector space $\Re^n$ that separates $\mathbf{x}_i, \forall i \in [l]$
- When $c = 0$, the separating hyperplane lies within the middle of hyperplanes $c = \pm 1$
- The distance between the separating hyperplane and the training datum nearest to the hyperplane is called the *margin*.

Hard Margin SVM 7 / 26

- Assuming that hyperplanes $g(\mathbf{x}) = +1$ and $g(\mathbf{x}) = -1$ include at least one training datum, the hyperplane $g(\mathbf{x}) = 0$ has the maximum margin for -1<c<+1.
- The region $\{x : -1 \leq g(\mathbf{x}) \leq +1\}$ is called the generalization region of the decision function.

Hard Margin SVM 8 / 26



IDEA : Select the separating hyperplane that maximizes the margin!

Hard Margin SVM 9 / 26

- Decision functions $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are separating hyperplanes.
- Such separating hyperplanes are not unique.
- Choose the one with higher generalization ability.
- Generalization ability depends exclusively on separating hyperplane location.
- Optimal Hyperplane is the one that maximizes margin.

Hard Margin SVM 10 / 26

- Assuming:
  - no outliers within the training data
  - the unknown test data will obey the same probability law as that of the training data
- Intuitively clear that generalization ability will be maximized if the optimal hyperplane is selected as the separating hyperplane

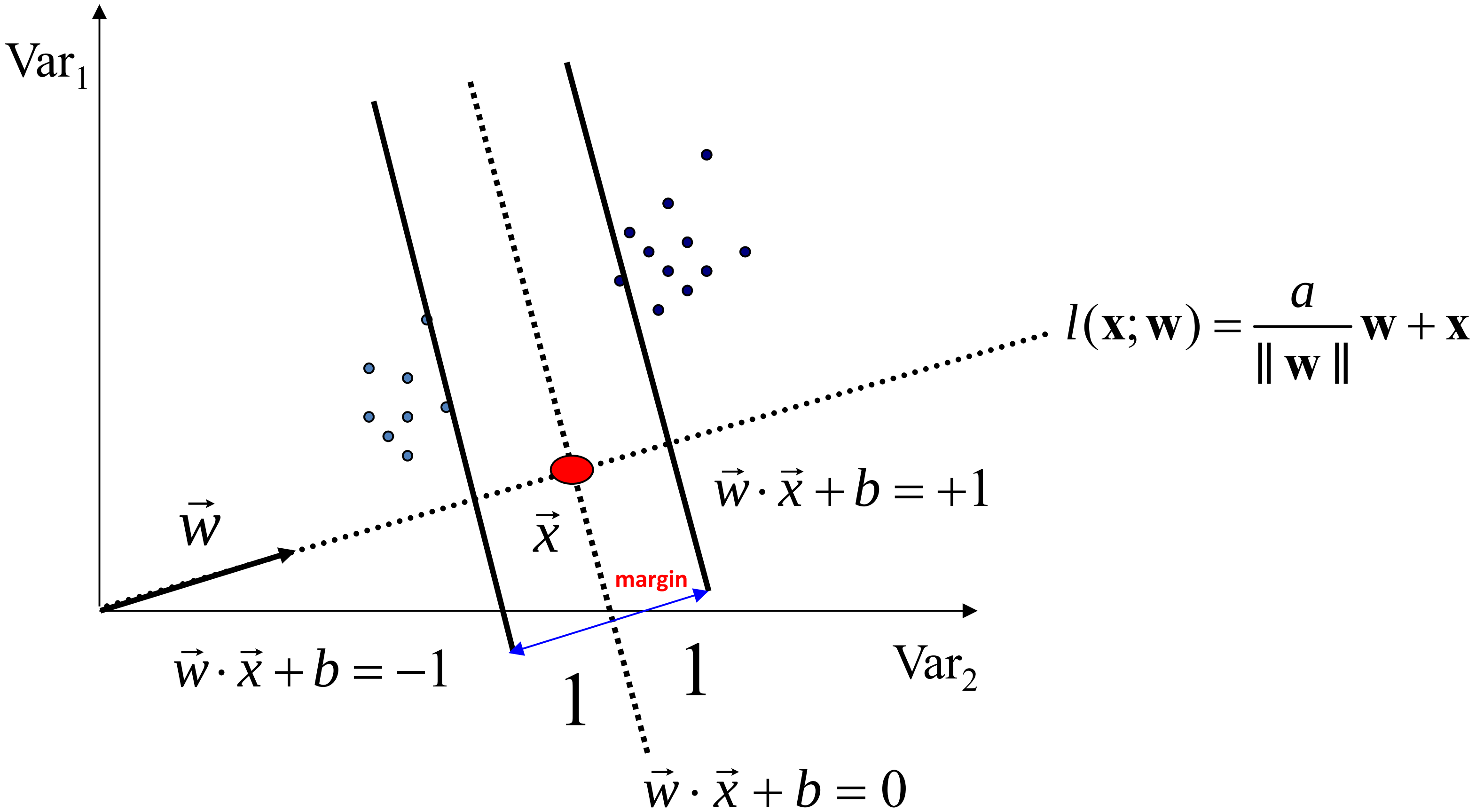
Hard Margin SVM 11 / 26
Optimal Hyperplane Determination I

- The Euclidean distance for a training datum **x** to the separating hyperplane parameterized by (**w** , b) is given by:

$$R(\mathbf{x};\mathbf{w},b)=\frac{|g(\mathbf{x})|}{\|\mathbf{w}\|}=\frac{|\langle\mathbf{w},\mathbf{x}\rangle+b|}{\|\mathbf{w}\|}$$

- Notice that w is orthogonal to the separating hyperplane.
- Line $l(\mathbf{x};\mathbf{w})$ goes through **x** being orthogonal to the separating hyperplane.

Hard Margin SVM 12 / 26
Optimal Hyperplane Determination II



Hard Margin SVM 13 / 26
Optimal Hyperplane Determination III

- |a| is the Euclidean distance from **x** to the hyperplane.
- $l(\mathbf{x};\mathbf{w})$ crosses the separating hyperplane at the point where $g(l(\mathbf{x};\mathbf{w}))=0$.

$$
\begin{aligned}
& g(l(x;w)) = 0 && \Leftrightarrow \\
& w^T l(x;w) + b = 0 && \Leftrightarrow \\
& w^T\left(\frac{a}{\|w\|}w + x\right) + b = 0 && \Leftrightarrow \\
& \frac{a}{\|w\|} w^T w + w^T x + b = 0 && \Leftrightarrow \\
& \frac{a}{\|w\|}\|w\|^2 = -w^T x - b && \Leftrightarrow \\
& a = -\frac{g(x)}{\|w\|} && \Leftrightarrow \\
& |a| = \frac{g(x)}{|\|w\|}
\end{aligned}
$$

Hard Margin SVM 14 / 26
Optimal Hyperplane Determination IV

- Let $\mathbf{x}^+$, $\mathbf{x}^-$ be two data points lying on the hyperplanes $g(\mathbf{x}) = +1$ and $g(\mathbf{x}) = -1$ respectively.
- Optimal hyperplane is determined by specifying (**w** , b) that maximize the quantity:

$$\gamma = \frac{1}{2}\{R(\mathbf{x}^+;\mathbf{w},b) + R(\mathbf{x}^-;\mathbf{w},b))\} = \frac{1}{\|\mathbf{w}\|}$$

- γ corresponds to the geometric margin.

Hard Margin SVM 15 / 26

- optimal separating hyperplane is obtained by maximizing the geometric margin.
- equivalent to minimizing the quantity: $f(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$ subject to the constraints:

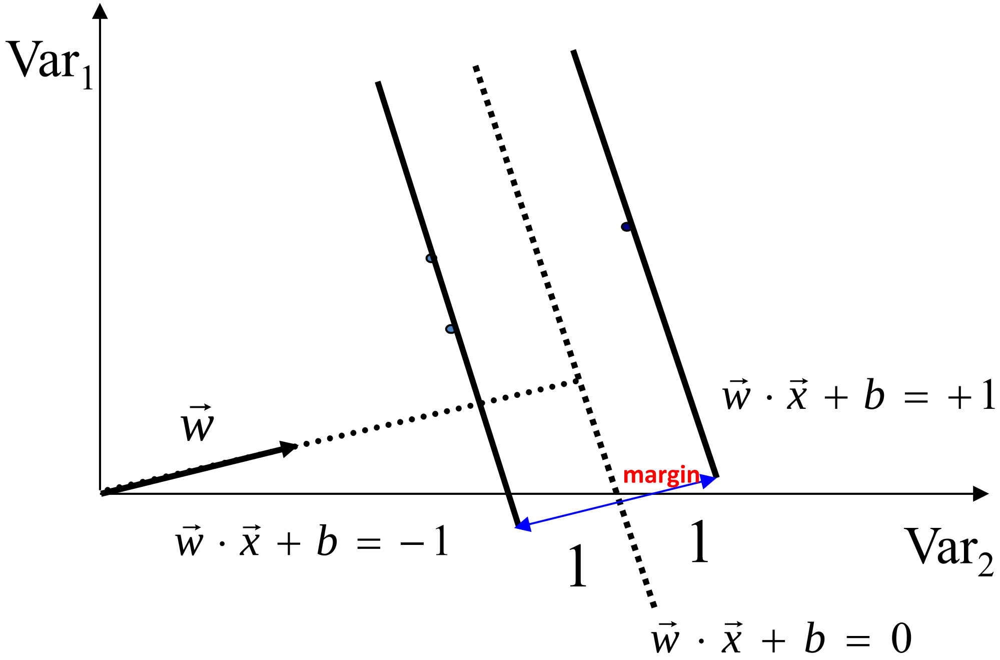
$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \forall i \in [l]$$

- The Euclidean norm ||**w**|| used to transform the optimization problem into a QP.
- The assumption of separability means that there exist (**w** , b) (feasible solutions) that satisfy the constraints.

Hard Margin SVM 16 / 26

- Optimization Problem:
  - quadratic objective function
  - inequality constraints defined by linear functions
- Even if the solutions are non-unique, the value of the objective function is unique.
- Non-uniqueness is not a problem for support vector machines.
- Advantage of SVMs over neural networks which have several local optima.

Hard Margin SVM 17 / 26



- Optimal Separating Hyperplane will remain the same even if it is computed by removing all the training patterns that satisfy the strict inequalities.
- Points on both sides of the separating hyperplane satisfying the corresponding equalities are called support vectors.

Hard Margin SVM 18 / 26

- Primal Optimization Problem of Hard Margin SVM:

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2$$
$$\text{s.t } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \forall i \in [l]$$

- Variables of the convex primal optimization problem are the parameters (**w** , b) defining the separating hyperplane.
- Variables = Dimensionality of the input space plus 1 which is n+1.
- When n is small, the solution can be obtained by QP technique.

Hard Margin SVM 19 / 26

- SVMs operate by mapping input space into high-dimensional feature spaces which in some cases may be of infinite dimensions.
- Solving the optimization problem is then too difficult to be addressed in its primal form.
- Natural solution is to re-express the optimization problem in its dual form.
- Variables in dual representation = Number of training data.

Hard Margin SVM 20 / 26

- Transform the original primal optimization problem into its dual by computing the Lagrangian function of the primal form.

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle - \sum_{i=1}^{l} a_i \{ y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \}$$

- $\mathbf{a} = [a_1 \ldots a_l]^T$ matrix of non-negative Lagrange multipliers.

Hard Margin SVM 21 / 26

- The dual problem is formulated as:

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} L(\mathbf{w}, b, \mathbf{a})$$
$$\text{s.t } a_i \geq 0, \forall i \in [l]$$

- Kuhn-Tucker Theorem: necessary and sufficient conditions for a normal point $(\mathbf{w}^*, b^*)$ to be an optimum is the existence of $\mathbf{a}^*$ such that:

Hard Margin SVM 22 / 26

$$\frac{\partial L(\mathbf{w}^*, b^*, \mathbf{a}^*)}{\partial \mathbf{w}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{l} a_i^* y_i \mathbf{x}_i \quad \text{(I)}$$

$$\frac{\partial L(\mathbf{w}^*, b^*, \mathbf{a}^*)}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{l} a_i^* y_i = 0 \quad \text{(II)}$$

$$a_i^* \{ y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1) \} = 0, \forall i \in [l] \quad \text{(III)}$$

Hard Margin SVM
Karush-Kuhn-Tucker
Complementarity Conditions

$$y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1) \geq 0, \forall i \in [l] \quad \text{(IV)}$$

$$a_i^* \geq 0, \forall i \in [l] \quad \text{(V)}$$

Hard Margin SVM 23 / 26

- Substituting (I),(II) in the original Lagrangian we get:

$$L(\mathbf{w}, b, \mathbf{a}) = \sum_{i=0}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

- The Dual Optimization Problem:

$$\max_{\mathbf{a}} \sum_{i=0}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{s.t} \sum_{i=1}^{l} a_i^* y_i = 0$$

$$\text{and } a_i \geq 0, \forall i \in [l]$$

Hard Margin SVM 24 / 26

- Dependence on original primal variables is removed.
- Dual formulation:
  - number of variables = number of the training patterns
  - concave quadratic programming problem
  - if a solution exists (linearly separable classification problem) then exists a global solution for $\mathbf{a}^*$.

Hard Margin SVM 25 / 26

- Karush-Kuhn-Tuck Complementarity Conditions:
  - for active constraints($\mathbf{a}_i^* = 0$) we have that:

    $$y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1) > 0$$

  - for inactive constraints ($\mathbf{a}_i^* > 0$) we have that:

    $$y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1) = 0$$

- Training data points $\mathbf{x}_i$ for which $\mathbf{a}_i^* > 0$ corresponds to support vectors lying on hyperplanes g(**x**) = +1 and g(**x**) = -1.

Hard Margin SVM 26 / 26

- Geometric margin (optimal hyperplane):

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

- Optimal Hyperplane:

$$\boxed{g(\mathbf{x}) = \sum_{i=1}^{l} a_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b^* = \sum_{i \in SV} a_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b^*}$$

- Optimal b parameter:

$$b^* = \frac{1}{n^+ + n^-} \{ (n^+ - n^-) - \sum_{i \in SV} \langle \mathbf{w}^*, \mathbf{x}_i \rangle \}$$

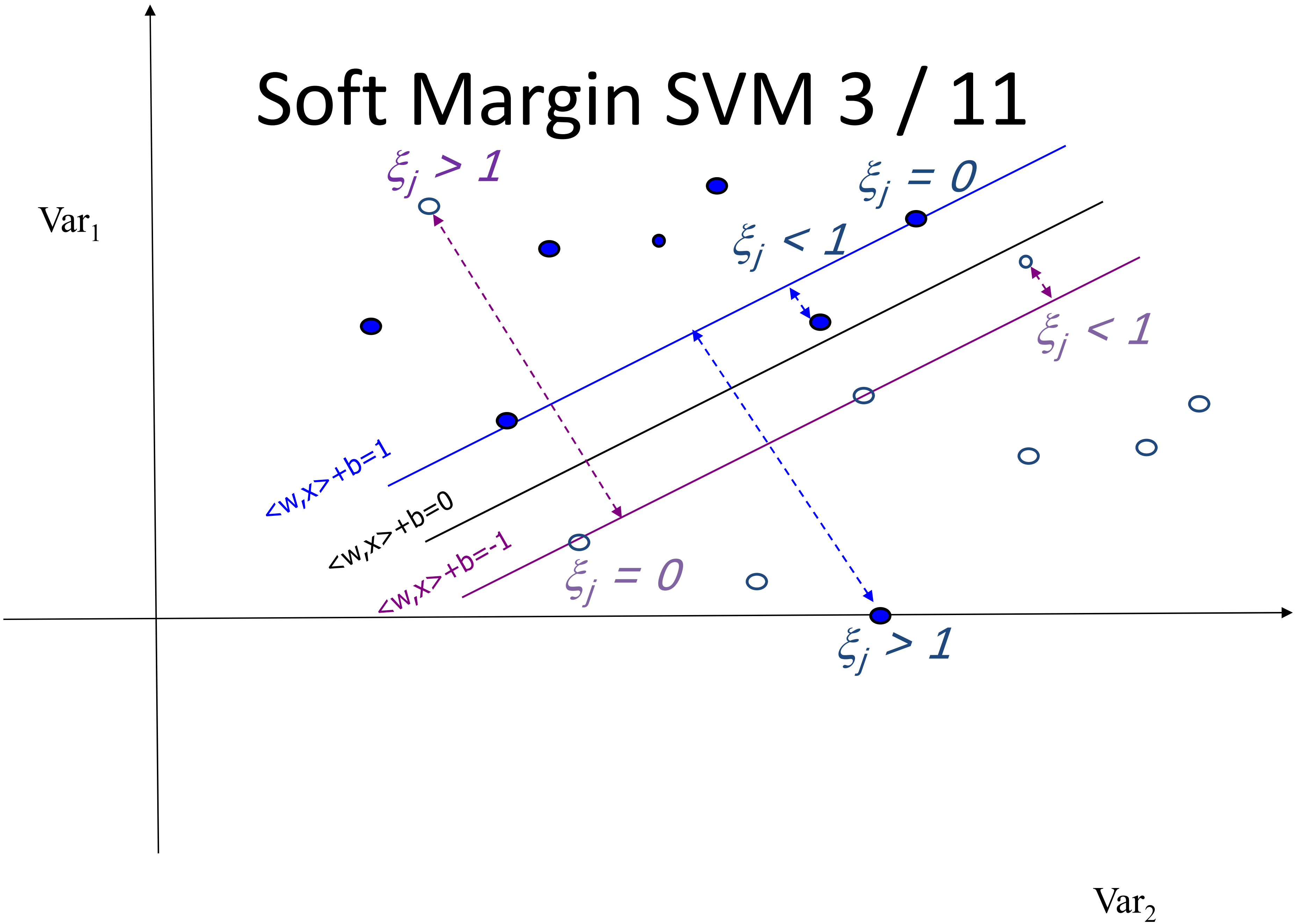
Soft Margin SVM 1 / 11

- Linearly inseparable data:
  - no feasible solution
  - optimization problem corresponding to Hard Margin Support Vector Machine unsolvable.
- Remedy: extension of Hard Margin paradigm by the so called Soft Margin Support Vector Machine.
- Key Idea: allow for some slight error represented by slack variables $\xi_i (\xi_i \geq 0)$.

Soft Margin SVM 2 / 11

- Introduction of slack variables yields that the original inequalities will be reformulated as:
  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \forall i \in [l]$
- Utilization of slack variables guarantees the existence of feasible solutions for the reformulated optimization problem.

Soft Margin SVM 3 / 11



Soft Margin SVM 4 / 11

- Optimal Separating Hyperplane correctly classifies all training patterns $\mathbf{x}_i$ for which: $0 < \xi_i < 1$ even if they do not have the maximum margin.
- Optimal Separating Hyperplane fails to correctly classify those training patterns for which: $\xi_i > 1$.

Soft Margin SVM 5 / 11

- Primal optimization problem of Soft Margin SVM introduces a tradeoff parameter C between maximizing margin and minimizing the sum of slack variables.
- Margin: directly influences generalization ability of the classifier.
- Sum of Slack Variables: quantifies the empirical risk of the classifier.

Soft Margin SVM 6 / 11

- Primal Optimization Problem of Soft Margin SVM:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i$$
$$\text{s.t. } y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i, \forall i \in [l]$$
$$\text{and } \xi_i \geq 0, \forall i \in [l]$$

- Lagrangian:

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle - \sum_{i=1}^{l} a_i y_i(\langle \mathbf{w},\mathbf{x}_i\rangle - b\sum_{i=1}^{l} a_i y_i + \sum_{i=1}^{l}\{C - a_i - \beta_i\}\xi_i$$
$$\mathbf{a} = [a_1 \ldots a_l]^T a_i \geq 0, \text{ i } \in [\text{l}]$$
$$\beta = [\beta_1 \ldots \beta_l]^T \beta_i \geq 0, \text{ i } \in [\text{l}]$$

Soft Margin SVM 7 / 11

- The dual problem is formulated as:

$$\max_{\mathbf{a},\beta} \min_{\mathbf{w},b,\xi} L(\mathbf{w}, b, \mathbf{a})$$
$$\text{s.t } a_i \geq 0, \forall i \in [l]$$
$$\text{and } \beta_i \geq 0, \forall i \in [l]$$

- Kuhn-Tucker Theorem: necessary and sufficient conditions for a normal point $(\mathbf{w}^*, b^*, \xi^*)$ to be an optimum is the existence of $(\mathbf{a}^*, \beta^*)$ such that:

Soft Margin SVM 8 / 11

$$\frac{\partial L(\mathbf{w}^*, b^*, \xi^*, \mathbf{a}^*, \beta^*)}{\partial \mathbf{w}} = \mathbf{0} \implies \mathbf{w}^* = \sum_{i=1}^{l} a_i^* y_i \mathbf{x}_i \quad \text{(I)}$$

$$\frac{\partial L(\mathbf{w}^*, b^*, \xi^*, \mathbf{a}^*, \beta^*)}{\partial \xi} = \mathbf{0} \implies C - a_i^* - \beta_i^* = 0, \forall i \in [l]. \quad \text{(II)}$$

$$\frac{\partial L(\mathbf{w}^*, b^*, \xi^*, \mathbf{a}^*, \beta^*)}{\partial b} = 0 \implies \sum_{i=1}^{l} a_i^* y_i = 0 \quad \text{(III)}$$

$$a_i^* \{ y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1 + \xi_i) \} = 0, \forall i \in [l] \quad \text{(IV)}$$

$$\beta_i \xi_i = 0, \forall i \in [l] \quad \text{(V)}$$

KKT Complementarity Conditions

$$y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1 + \xi_i) \geq 0, \forall i \in [l] \quad \text{(VI)}$$

$$a_i^* \geq 0, \forall i \in [l] \quad \text{(VII)}$$

$$\beta_i^* \geq 0, \forall i \in [l] \quad \text{(VIII)}$$

Soft Margin SVM 9 / 11

- Equations (II),(VII) and (VIII) may be combined as: $0 \leq a_i^* \leq C$.
- Substituting (I),(II) and (III) in the original Lagrangian we get:

$$L(\mathbf{w}, b, \mathbf{a}) = \sum_{i=0}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

- Dual optimization problem:

$$\max_{\mathbf{a}} \sum_{i=0}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{s.t} \sum_{i=1}^{l} a_i^* y_i = 0 \text{and} a_i \geq 0, \forall i \in [l]$$

$$\text{and} \beta_i \geq 0, \forall i \in [l]$$

Soft Margin SVM 10 / 11

- Karush-Kuhn-Tuck Complementarity Conditions:
  - active constraints: $a_i^* = 0 \Rightarrow \beta_i = C \neq 0 \Rightarrow \xi_i = 0$ corresponding training patterns $\mathbf{x}_i$ are correctly classified.
  - inactive constraints:
    - (unbounded support vectors) $0 < a_i^* < C \Rightarrow \beta_i \neq 0 \Rightarrow \xi_i = 0 \Rightarrow y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) = 1$
    - (bounded support vectors) $a_i^* = C \Rightarrow \beta_i = 0 \Rightarrow \xi_i \neq 0 \Rightarrow y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1 + \xi_i = 0$

Soft Margin SVM 11 / 11

- Geometric margin (optimal hyperplane):

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

- Optimal b parameter:

$$b^* = \frac{1}{n_u^+ + n_u^-}\{(n_u^+ - n_u^-) - \sum_{i \in SV_u} \langle \mathbf{w}^*, \mathbf{x}_i \rangle\}$$

- Optimal ξ_i parameters:

$$\xi_i^* = \max(0, 1 - y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle) + b^*)$$

- Optimal Hyperplane:

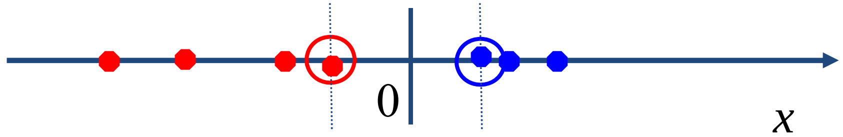
$$g(\mathbf{x}) = \sum_{i=1}^{l} a_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b^* = \sum_{i \in SV} a_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b^*$$

Linear SVMs Overview

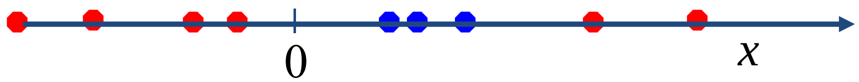
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors as they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products.

Mapping Data to High Dimensional Feature Spaces (1 / 4)

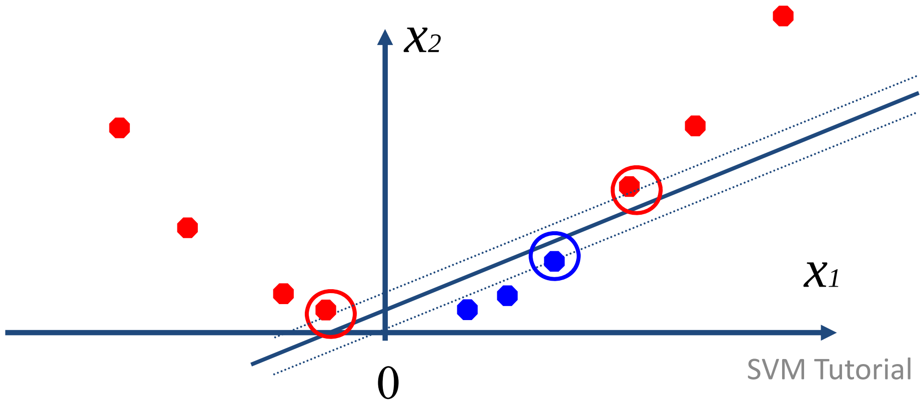
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

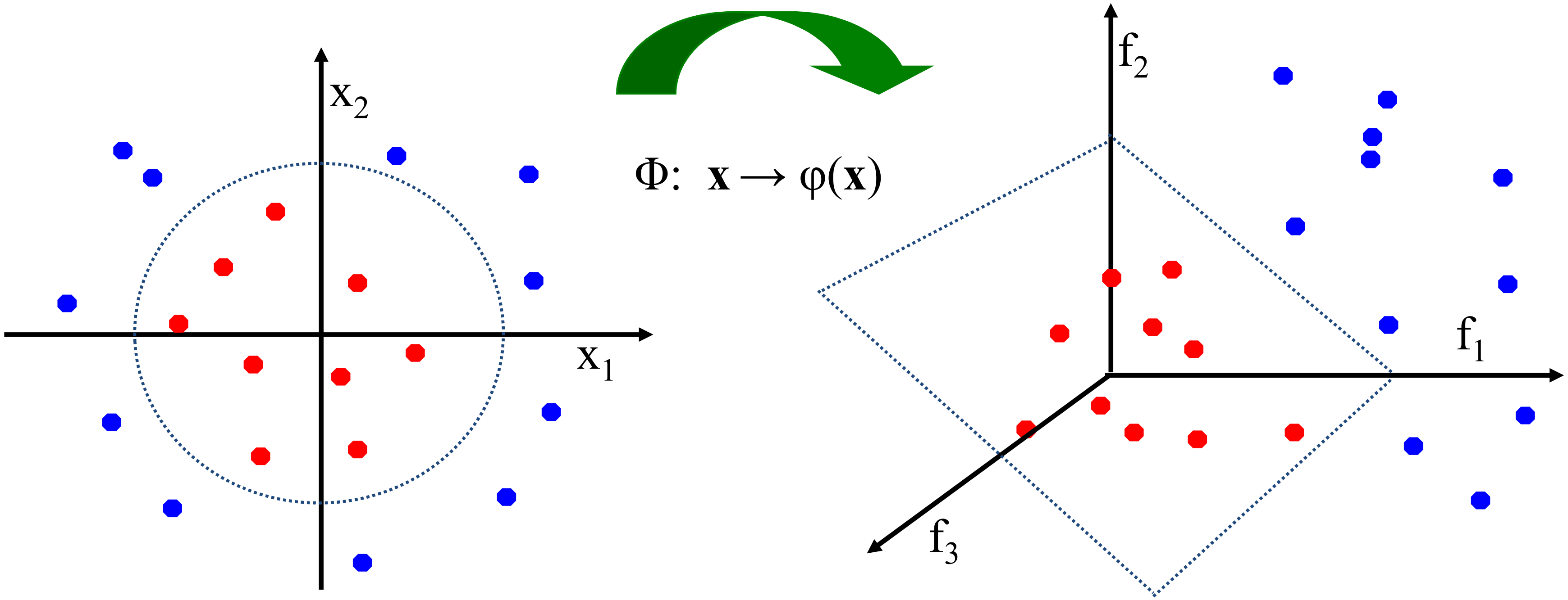


- How about… mapping data to a higher-dimensional space:



Mapping Data to High Dimensional Feature Spaces (2 / 4)

- General idea: the original input space can always be mapped to some higher dimensional feature space where the training set is separable.



Mapping Data to High Dimensional Feature Spaces (3 / 4)

- Find function Φ(x) to map to a different space, then SVM formulation becomes:

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$s.t.\ \ y_i(< w, \Phi(x) > +b) \geq 1 - \xi_i, \forall x_i$$

$$\xi_i \geq 0$$

- Data appear as Φ(x), weights *w* are now weights in the new space.
- Explicit mapping expensive if Φ(x) is very high dimensional.
- Solving the problem without explicitly mapping the data is desirable.

Mapping Data to High Dimensional Feature Spaces (4 / 4)

- Original SVM formulation
  - n inequality constraints
  - n positivity constraints
  - n number of ξ constraints

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$
$$s.t.\ \ y_i(w \cdot \Phi(x) + b) \geq 1 - \xi_i, \forall x_i$$
$$\xi_i \geq 0$$

- Dual formulation
  - one equality constraint
  - *n* positivity constraints
  - *n* number of α variables (Lagrange multipliers)
  - NOTICE: Data only appear as $<\Phi(x_i) , \Phi(x_j)>$

$$\min_{a_i} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j < \Phi(x_i) \cdot \Phi(x_j) > - \sum_i \alpha_i$$
$$s.t.\ \ \mathrm{C} \geq \alpha_{\mathrm{i}} \geq 0, \forall x_i$$
$$\sum_i \alpha_i y_i = 0$$

Kernel Trick (1/ 2)

- The linear classifier relies on inner product between vectors $K(\mathbf{x}_l, \mathbf{x}_j) = <\mathbf{x}_l, \mathbf{x}_j>$.
- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \phi(x)$, the inner product becomes:

$$K(\mathbf{x}_l, \mathbf{x}_j) = <\phi(\mathbf{x}_l), \phi(\mathbf{x}_j)>.$$

- A kernel function is some function that corresponds to an inner product in some expanded feature space.
- We can find a function such that:
  - $K(<x_i, x_j>) = <\Phi(x_i), \Phi(x_j)>$, i.e., the image of the inner product of the data is the inner product of the images of the data.

Kernel Trick (2/ 2)

- Then, we do not need to explicitly map the data into the high-dimensional space to solve the optimization problem (for training)
- How do we classify without explicitly mapping the new instances? Turns out:
  - Optimal Hyperplane: $g(\mathbf{x}) = \sum_{i=1}^{l} a_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* = \sum_{i \in SV} a_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*$
  - Optimal b parameter: $b^* = \frac{1}{n_u^+ + n_u^-} \{(n_u^+ - n_u^-) - \sum_{i \in SV_u} K(\mathbf{w}^*, \mathbf{x}_i)\}$
  - Optimal ξ parameter: $\xi_i^* = \max(0, 1 - y_i(K(\mathbf{w}^*, \mathbf{x}_i)) + b^*)$

Kernels (1 / 5)
Examples I

- 2D input space mapped to 3D feature space:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (< \mathbf{x}_i, \mathbf{x}_j >)^2 \Rightarrow \phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{pmatrix}$$ where $x, y \in R^2$

$$(x \cdot y)^2 = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right)^2 = \left(\begin{bmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ \sqrt{2}\, y_1 y_2 \\ y_2^2 \end{bmatrix} \right)$$

$$= (\varphi(x) \cdot \varphi(y)) = k(x, y)$$

Kernels (2 / 5)
Examples II

2D input space mapped to 6D feature space:

$\mathbf{x}=[x_1 \;\; x_2]$; let $K(\mathbf{x}_i, \mathbf{x}_j)=(1 + \langle\mathbf{x}_l, \mathbf{x}_j\rangle)^2$,

Need to show that $K(\mathbf{x}_i, \mathbf{x}_\mathbf{j}) = \langle \boldsymbol{\phi}(\mathbf{x}_l), \boldsymbol{\phi}(\mathbf{x}_j)\rangle$:

$K(\mathbf{x}_i, \mathbf{x}_\mathbf{j})=(1 + \langle\mathbf{x}_i, \mathbf{x}_j\rangle)^2=$

$1+ x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} =$

$[1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}] =$

$= \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle$

where $\boldsymbol{\phi}(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

Kernels (3 / 5)

- Which functions are kernels?
- For some functions $\boldsymbol{K}(\mathbf{x_i}, \mathbf{x_j})$ checking that $\boldsymbol{K}(\mathbf{x_i}, \mathbf{x_j}) = <\varphi(\mathbf{x_i}), \varphi(\mathbf{x_j})>$ can be easy.
- Is there a mapping $\Phi(x)$ for any symmetric function $K(x, z)$? No
- The SVM dual formulation requires calculation $K(x_i, x_j)$ for each pair of training instances. The array $G_{ij} = K(x_i, x_j)$ is called the Gram matrix.

Kernels (4 / 5)

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_l)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_l)$
…	…	…	…	…
$K(\mathbf{x}_l,\mathbf{x}_1)$	$K(\mathbf{x}_l,\mathbf{x}_2)$	$K(\mathbf{x}_l,\mathbf{x}_3)$	…	$K(\mathbf{x}_l,\mathbf{x}_l)$

- There is a feature space $\Phi(x)$ when the Kernel is such that *G* is always semi-positive definite (Mercer Theorem)
  - A symmetric matrix ***A*** is said to be ***positive semi-definite*** if, for any non 0 vector ***x*** : $x^T A x \geq 0$

Kernels (5 / 5)

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j) = \langle \mathbf{x}_i,\mathbf{x}_j \rangle$
  - Mapping Φ: **x** → **ϕ**(**x**), where **ϕ**(**x**) is **x** itself.
- Polynomial of power *p*: $K(\mathbf{x}_i,\mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^p$
  - Mapping Φ: **x** → **ϕ**(**x**), where **ϕ**(**x**) has $\binom{n+p}{p}$ dimensions.
- Gaussian (radial-basis function): $\mathrm{K}(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j}) = e^{-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}}$
  - Mapping Φ: **x** → **ϕ**(**x**), where **ϕ**(**x**) is *infinite-dimensional.*

Conclusions

Neural Networks

- Hidden Layers map to lower dimensional spaces
- Search space has multiple local minima
- Training is expensive
- Classification extremely efficient
- Requires number of hidden units and layers
- Very good accuracy in typical domains

SVMs

- Kernel maps to a very-high dimensional space
- Search space has a unique minimum
- Training is extremely efficient
- Classification extremely efficient
- Kernel and cost the two parameters to select
- Very good accuracy in typical domains
- Extremely robust