CS:E4830 Kernel Methods in Machine Learning

Lecture 6 : Kernel Support Vector Machines

Rohit Babbar

7th April, 2021

Some Announcements

- Assignment 2 will be released this week
- There will be no lecture next week
- Next upcoming session - Optional tutorial session tomorrow at 16:00

Course outline so far

- Introduction to kernel methods
  - Kernel Definition and examples
  - Reproducing Kernel Hilbert Spaces
  - Representer Theorem and Kernel Least Squares
- Introductory learning theory
  - Generalization
  - Empirical risk minimization
- Convex optimization Introduction
  - Convexity and Duality
- Algorithms - Supervised
  - *Kernel Support Vector Machines*
  - Logistic regression
  - Kernel methods for large-scale problems
- Algorithms - Unsupervised
  - PCA and k-means
  - Kernel variants

Supervised learning setup

Binary classification

- Input $\mathcal{X}$, can be in various forms such as images or text documents
- Output $\mathcal{Y} = \{-1, +1\}$ - binary classification. Other formulations include :
  - One-hot encoded binary vector for multi-class classification - Cifar10
  - Multi-label classification - Wikipedia
- Training set $S = (x_i, y_i)_{i=1}^{N}$ consists of samples that are sampled independently and identically from an unknown joint distribution P over $\mathcal{X} \times \mathcal{Y}$
- The goal is to build a classifier f to predict the label $\hat{y}$ for a test instance x.

Hinge Loss Function and Kernel SVM

- Kernel SVM solves the following optimization problem :

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i f(x_i)) + \lambda ||f||^2_{\mathcal{H}}$$

  where $\mathcal{H}$ is a reproducing kernel Hilbert space
- Hinge loss is a function $\mathbb{R} \mapsto \mathbb{R}_+$:

$$\ell_{hinge}(u) = max(1-u, 0) = \begin{cases} 0 & \text{if } u \geq 1 \\ 1-u & \text{otherwise} \end{cases}$$

Hinge Loss and Others

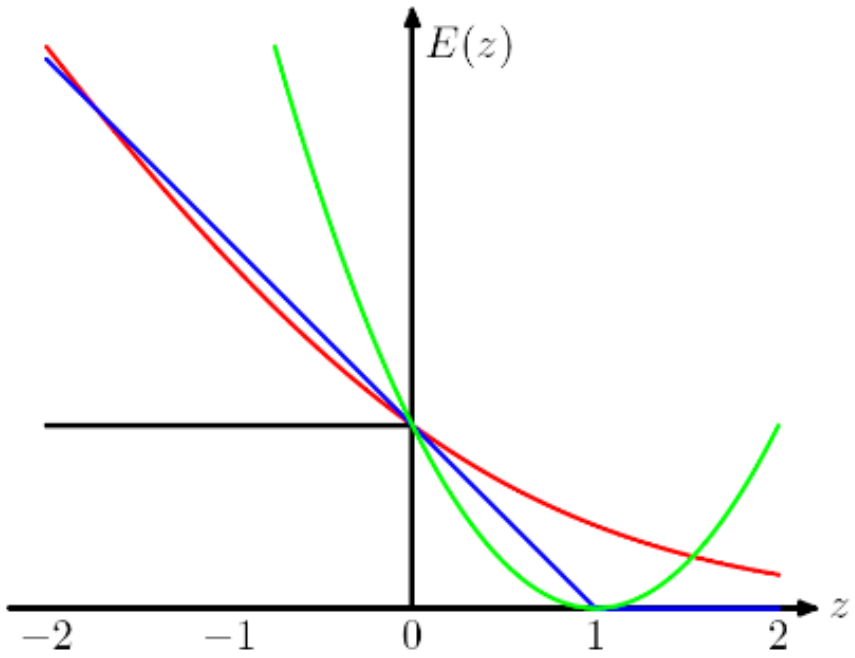


Figure: $z = yf(x)$ in the above graph

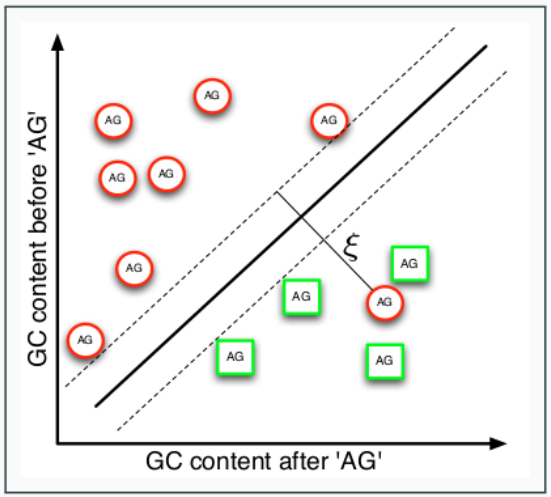
Convex Upper Bounds on 0-1 loss

- Hinge Loss (in blue) is given by $\max(1 - yf(x), 0)$
- Logistic Loss is given by $\frac{1}{\log 2} \log(1 + \exp(-yf(x)))$

Soft-Margin SVM (Cortes & Vapnik, 1995)

The soft-margin SVM allows non-separable data by using the relaxed the margin constraints

$$\begin{aligned} &Minimize \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \\ &\quad w.r.t \text{ variables } \mathbf{w}, \boldsymbol{\xi} \\ &\text{Subject to } y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i \\ &\quad \text{for all } i = 1, \ldots, m. \\ &\quad \xi_i \geq 0, \text{for all } i = 1, \ldots, m. \end{aligned}$$



- The sum (or average) of slack variables appear as a penalty in the objective
- The coefficient $C > 0$ controls the balance between model complexity (low C) and empirical error (high C)

12

From Representer Theorem

- For the following optimization

$$f_{\mathcal{H}} := \arg\min_{f} \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i, f(x_i)) + \lambda\theta(||f||^2_{\mathcal{H}})$$

where $\ell_{hinge}(.,.)$ is the hinge loss function and $\theta : [0, \infty) \mapsto \mathbb{R}$ is non-decreasing function, and $\mathcal{H}$ is an RKHS

- Even though the above problem is potentially an infinite dimensional optimization problem, **Representer Theorem** states its solution can be expressed in the following form

$$f(.) = \sum_{j=1}^{N} \alpha_j k(., x_j)$$

where $\alpha_i \in \mathbb{R}$, i.e. it is linear combination of kernel evaluations at training points

Implications of Representer Theorem

- Representer Theorem allows us to look for the solutions of the following form:

$$f(.) = \sum_{j=1}^{N} \alpha_i k(., x_j)$$

- Implications
  - The desired function just involves kernel computation on **training points only** via $k(., x_i)$ in the above solution
  - It reduces the problem of finding $f \in \mathcal{H}$ which could be infinite dimensional to a finite dimensional problem
    - We just need to find the coefficients of the finite linear combination $\alpha_1, \ldots, \alpha_N$
  - Also, we can reforumlate the original objective function in the new form (more in next slides)

Reformulating the Objective using Representer Theorem - I

- Recall the original objective :

$$f_{\mathcal{H}} := \arg\min_{f} \frac{1}{n} \sum_{i=1}^{N} \ell(y_i, f(x_i)) + \lambda\theta(||f||_{\mathcal{H}})$$

- For the i-th **training point**,

$$f_{\mathcal{H}}(x_i) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j k(x_i, x_j) = [K\boldsymbol{\alpha}]_i$$

which is the i-th element of the matrix-vector product $K\boldsymbol{\alpha}$

Reformulating the Objective using Representer Theorem - II

- Rewriting the regularization term :

$$
\begin{aligned}
||f||_{\mathcal{H}}^2 &= \langle f(.), f(.) \rangle \\
&= \left\langle \sum_{i=1}^{N} \alpha_i k(., x_i), \sum_{i=1}^{N} \alpha_i k(., x_i) \right\rangle \text{ (using representer theorem)} \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(x_i, x_j) \text{ (evaluation of dot product)} \\
&= \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \text{ (writing in matrix notation)}
\end{aligned}
$$

Reformulating the Objective using Representer Theorem - III

Using the above substitutions, the original (potentially intractable) objective

$$f_{\mathcal{H}} := \arg\min_{f} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(x_i)) + \lambda\theta(||f||_{\mathcal{H}})$$

translates to an equivalent (tractable) form below

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, [K\boldsymbol{\alpha}]_i) + \lambda\theta(\boldsymbol{\alpha}^T K\boldsymbol{\alpha})$$

SVM Problem Refomulation

- Using Representer theorem, the problem can be reformulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i) + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\}$$

SVM Problem Refomulation

- Using Representer theorem, the problem can be reformulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i) + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\}$$

- The above optimization problem is convex (Why?)

SVM Problem Refomulation

- Using Representer theorem, the problem can be reformulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i) + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\}$$

- The above optimization problem is convex (Why?)
- However, it is non-smooth optimization problem (Why?)

SVM Problem Refomulation

- Using Representer theorem, the problem can be reformulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i) + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\}$$

- The above optimization problem is convex (Why?)
- However, it is non-smooth optimization problem (Why?)

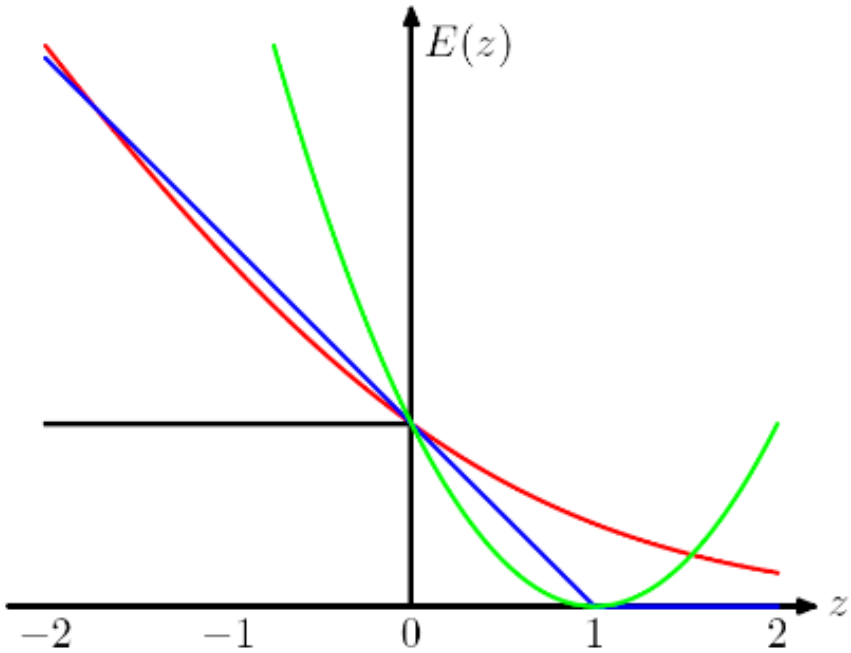


Figure: $\ell_{\text{hinge}}(.)$ is the function in blue, $z = yf(x)$ in the above graph

Another Equivalent Reformulation

- The optimization problem on the previous slide is equivalent (even though not immediately obvious) to the following, if we re-write it in terms of slack variables $\boldsymbol{\xi}_i \in \mathbb{R}$ for $i = 1, \ldots, N$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N, \boldsymbol{\xi} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\} \text{ such that } \boldsymbol{\xi}_i \geq \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i)$$

Another Equivalent Reformulation

- The optimization problem on the previous slide is equivalent (even though not immediately obvious) to the following, if we re-write it in terms of slack variables $\boldsymbol{\xi}_i \in \mathbb{R}$ for $i = 1, \ldots, N$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N, \boldsymbol{\xi} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\} \text{ such that } \boldsymbol{\xi}_i \geq \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i)$$

- In the above formulation, the objective is smooth but not the constraints

Another Equivalent Reformulation

- The optimization problem on the previous slide is equivalent (even though not immediately obvious) to the following, if we re-write it in terms of slack variables $\boldsymbol{\xi}_i \in \mathbb{R}$ for $i = 1, \ldots, N$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N, \boldsymbol{\xi} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\} \text{ such that } \boldsymbol{\xi}_i \geq \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i)$$

- In the above formulation, the objective is smooth but not the constraints
- Recall the definition of hinge loss from first slide

$$\ell_{hinge}(u) = max(1 - u, 0) \iff \begin{cases} 0 & \text{if } u \geq 1 \\ 1 - u & \text{otherwise} \end{cases}$$

- Using above, the N constraints ($\boldsymbol{\xi}_i \geq \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i)$) can be replaced by $2N$ constraints to make the problem smooth as follows :

$$\boldsymbol{\xi}_i \geq \ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i) \iff \begin{cases} \boldsymbol{\xi}_i \geq 0 \\ \boldsymbol{\xi}_i \geq 1 - y_i [K\boldsymbol{\alpha}]_i \end{cases}$$

SVM Primal Formulation

SVM Primal Formulation

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N,\boldsymbol{\xi}\in\mathbb{R}^N} \left\{ \frac{1}{n}\sum_{i=1}^{N} \boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right\}$$

such that

$$\begin{cases} 1 - y_i [K\alpha]_i - \boldsymbol{\xi}_i \leq 0 & \text{for } i = 1, \ldots, N \\ -\boldsymbol{\xi}_i \leq 0 & \text{for } i = 1, \ldots, N \end{cases}$$

Putting in the standard convex optimization problem framework where the inequality constraints should in the less than ($\leq$) form

Lagrangian

- The Lagrangian of the problem is :

$$L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} + \sum_{i=1}^{N} \boldsymbol{\mu}_i [1 - y_i [K\boldsymbol{\alpha}]_i - \boldsymbol{\xi}_i] - \sum_{i=1}^{N} \boldsymbol{\nu}_i \boldsymbol{\xi}_i$$

  - $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^N$, $\boldsymbol{\mu} \geq 0$ and $\boldsymbol{\nu} \geq 0$
- Note that constraints have moved to the Lagrangian.

Computing derivatives w.r.t ξ

- The lagrangian of the problem is :

$$L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\xi}_i + \lambda \boldsymbol{\alpha}^T K \boldsymbol{\alpha} + \sum_{i=1}^{N}\boldsymbol{\mu}_i[1 - y_i[K\boldsymbol{\alpha}]_i - \boldsymbol{\xi}_i] - \sum_{i=1}^{N}\boldsymbol{\nu}_i\boldsymbol{\xi}_i$$

Lagrangian wrt ξ

- $L(\alpha, \xi, \mu, \nu)$ is a linear function in ξ. What is its minimum value

Computing derivatives w.r.t ξ

- The lagrangian of the problem is :

$$L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\xi}_i + \lambda\boldsymbol{\alpha}^T K\boldsymbol{\alpha} + \sum_{i=1}^{N}\boldsymbol{\mu}_i[1 - y_i[K\boldsymbol{\alpha}]_i - \boldsymbol{\xi}_i] - \sum_{i=1}^{N}\boldsymbol{\nu}_i\boldsymbol{\xi}_i$$

Lagrangian wrt ξ

- $L(\alpha, \xi, \mu, \nu)$ is a linear function in ξ. What is its minimum value
- Its minimum value is $-\infty$, except when it is constant,

$$\nabla_{\boldsymbol{\xi}} L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{\mathbf{1}}{N} - \boldsymbol{\mu} - \boldsymbol{\nu} = \mathbf{0}$$

equivalently,

$$\frac{\mathbf{1}}{N} = \boldsymbol{\mu} + \boldsymbol{\nu}$$

Computing derivatives w.r.t α

- The lagrangian of the problem is :

$$L(\alpha, \xi, \mu, \nu) = \frac{1}{N}\sum_{i=1}^{N}\xi_i + \lambda\alpha^T K\alpha + \sum_{i=1}^{N}\mu_i[1 - y_i[K\alpha]_i - \xi_i] - \sum_{i=1}^{N}\nu_i\xi_i$$

Lagrangian wrt α

- $L(\alpha, \xi, \mu, \nu)$ is a convex quadratic function in α. To find the optimal value, set the gradient to $\mathbf{0}$ (the zero vector) :

$$\nabla_\alpha L = \mathbf{0}$$

- The optimal solution α^* is given by

$$\alpha_i^* = \frac{y_i\mu_i}{2\lambda}$$

Lagrange Dual Function and Dual Problem

Lagrange Dual Function

- The Lagrange dual function as obtained by substituting the optimal values (as obtained in previous two slides) is given by :

$$q(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^N, \boldsymbol{\xi} \in \mathbb{R}^N} L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu})$$

$$= \begin{cases} \sum_{i=1}^{N} \boldsymbol{\mu}_i - \frac{1}{4\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \boldsymbol{\mu}_i \boldsymbol{\mu}_j K(x_i, x_j) & \text{if } \boldsymbol{\mu} + \boldsymbol{\nu} = \frac{\mathbf{1}}{N} \\ -\infty & \text{otherwise} \end{cases}$$

Lagrange Dual Problem

- The Lagrange dual problem is

$$\max q(\boldsymbol{\nu}, \boldsymbol{\mu}) \text{ such that } \boldsymbol{\mu} \geq 0, \boldsymbol{\nu} \geq 0$$

Closer Look At The Dual Problem

- The Lagrange dual problem is

$$\max q(\boldsymbol{\nu}, \boldsymbol{\mu}) \text{ such that } \boldsymbol{\mu} \geq 0, \boldsymbol{\nu} \geq 0$$

- If $0 \leq \boldsymbol{\mu}_i \leq 1/N$ for all i, then the dual function takes finite values. Also, the value of $\boldsymbol{\nu}_i$ is fixed at $\boldsymbol{\nu}_i = 1/N - \boldsymbol{\mu}_i$ in this case.
- The dual problem is therefore given by

$$\max_{\mathbf{0} \leq \boldsymbol{\mu} \leq \mathbf{1}/N} \sum_{i=1}^{N} \boldsymbol{\mu}_i - \frac{1}{4\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \boldsymbol{\mu}_i \boldsymbol{\mu}_j K(x_i, x_j)$$

Rewriting in terms of Primal Variables

Dual problem (from previous slide)

$$\max_{\mathbf{0} \leq \boldsymbol{\mu} \leq \mathbf{1}/N} \sum_{i=1}^{N} \boldsymbol{\mu}_i - \frac{1}{4\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \boldsymbol{\mu}_i \boldsymbol{\mu}_j K(x_i, x_j)$$

Since the primal variable $\boldsymbol{\alpha}$ and the dual variable $\boldsymbol{\mu}$ are related by $\boldsymbol{\alpha}_i = \frac{\boldsymbol{\mu}_i y_i}{2\lambda}$, it can be written in the form of primal variables as follows

writing in terms of primal variable $\boldsymbol{\alpha}$

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^N} 2 \sum_{i=1}^{N} \boldsymbol{\alpha}_i y_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j K(x_i, x_j)$$

such that

$$0 \leq y_i \boldsymbol{\alpha}_i \leq \frac{1}{2\lambda N} \text{ for } i = 1, \ldots, N$$

The above can be solved using a standard Quadratic program solver.

Back to Representer Theorem

- Once we have the $\boldsymbol{\alpha}$ from the previous slide, the decision function in RKHS is constructed using representer theorem:

$$f(.) = \sum_{j=1}^{N} \boldsymbol{\alpha}_i k(., x_j)$$

- However, if N is large, this might still be computationally expensive for prediction
- In order to tackle this, the complementarity conditions (next) motivate the idea of *support vectors*, which we see next.

Complementarity conditions at the optimum

(*Check corresponding slide from previous Lecture on Complementary Slackness*)

- These are given by the product of the dual variables and the corresponding constraint as follows :

$$\boldsymbol{\mu}_i[y_i f(x_i) + \boldsymbol{\xi}_i - 1] = 0$$
$$\boldsymbol{\nu}_i \boldsymbol{\xi}_i = 0$$

- Recalling that $\boldsymbol{\mu}_i = \frac{2\lambda \boldsymbol{\alpha}_i}{y_i}$ and $\boldsymbol{\nu}_i = \frac{\mathbf{1}}{N} - \frac{2\lambda \boldsymbol{\alpha}_i}{y_i}$
- In terms of the primal variable $\boldsymbol{\alpha}$, it is given by

$$\boldsymbol{\alpha}_i[y_i f(x_i) + \boldsymbol{\xi}_i - 1] = 0$$
$$\left(\boldsymbol{\alpha}_i - \frac{y_i}{2\lambda N}\right) \boldsymbol{\xi}_i = 0$$

Interpreting the Complementarity Conditions

Complementarity Conditions

$$\alpha_i [y_i f(x_i) + \xi_i - 1] = 0$$

$$\left(\alpha_i - \frac{y_i}{2\lambda N}\right) \xi_i = 0$$

Interpreting the Complementarity Conditions

Complementarity Conditions

$$\alpha_i[y_i f(x_i) + \xi_i - 1] = 0$$
$$\left(\alpha_i - \frac{y_i}{2\lambda N}\right)\xi_i = 0$$

- If $\alpha_i = 0$, then the second constraint is active : $\xi_i = 0$. This implies $y_i f(x_i) \geq 1$ (Why?). (*These are correctly classified points with some margin*)

Interpreting the Complementarity Conditions

Complementarity Conditions

$$\boldsymbol{\alpha}_i[y_i f(x_i) + \boldsymbol{\xi}_i - 1] = 0$$
$$\left(\boldsymbol{\alpha}_i - \frac{y_i}{2\lambda N}\right)\boldsymbol{\xi}_i = 0$$

- If $\boldsymbol{\alpha}_i = 0$, then the second constraint is active : $\boldsymbol{\xi}_i = 0$. This implies $y_i f(x_i) \geq 1$ (Why?). (*These are correctly classified points with some margin*)
- If $0 < y_i\boldsymbol{\alpha}_i < \frac{1}{2\lambda N}$, then both the constraints are active, i.e., $\boldsymbol{\xi}_i = 0$ and $y_i f(x_i) + \xi_i - 1 = 0$. This leads to $y_i f(x_i) = 1$. (*These are correctly classified points right at the margin*)

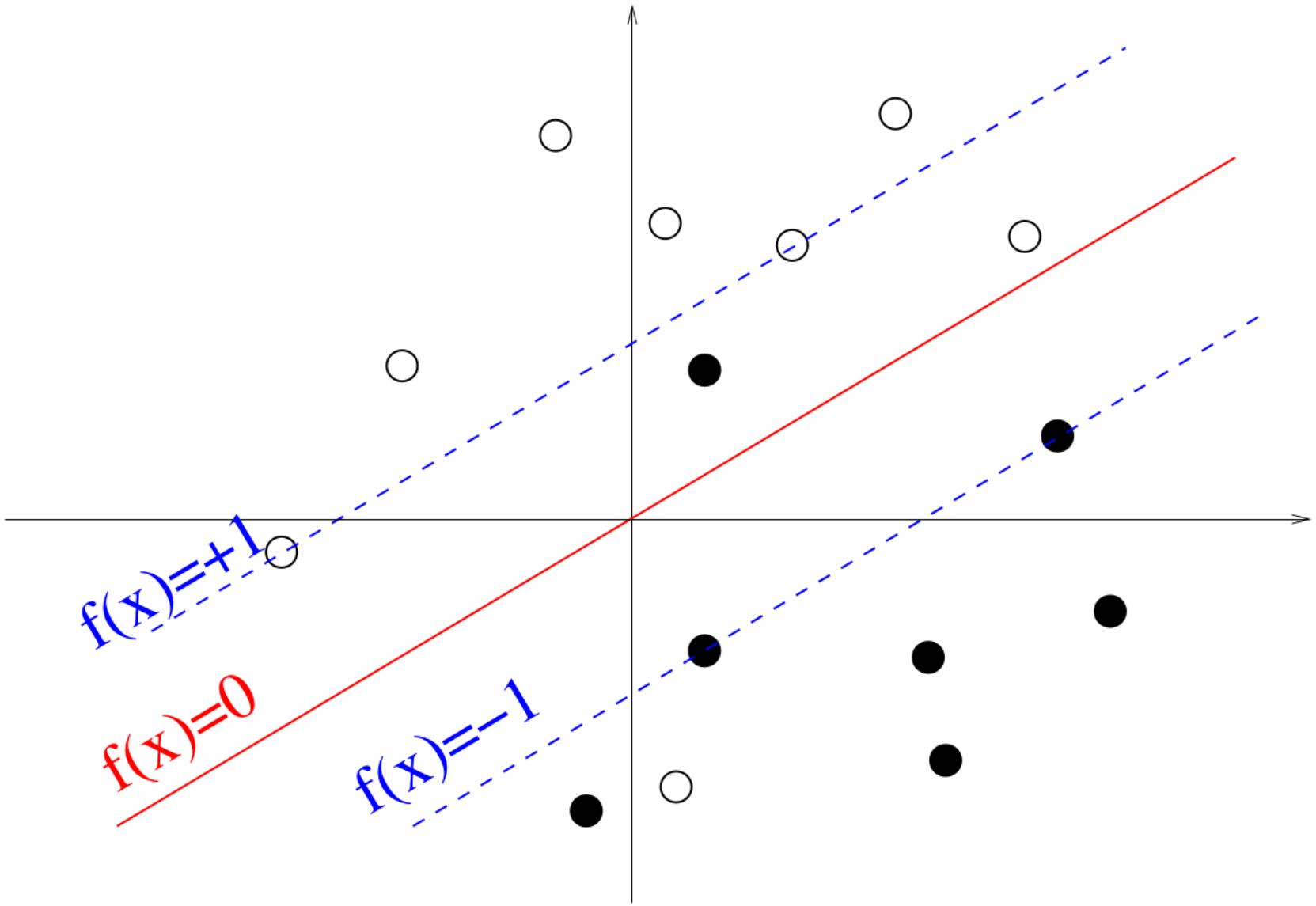
Interpreting the Complementarity Conditions

Complementarity Conditions

$$\boldsymbol{\alpha}_i[y_i f(x_i) + \boldsymbol{\xi}_i - 1] = 0$$
$$\left(\boldsymbol{\alpha}_i - \frac{y_i}{2\lambda N}\right)\boldsymbol{\xi}_i = 0$$

- If $\boldsymbol{\alpha}_i = 0$, then the second constraint is active : $\boldsymbol{\xi}_i = 0$. This implies $y_i f(x_i) \geq 1$ (Why?). (*These are correctly classified points with some margin*)
- If $0 < y_i\boldsymbol{\alpha}_i < \frac{1}{2\lambda N}$, then both the constraints are active, i.e., $\boldsymbol{\xi}_i = 0$ and $y_i f(x_i) + \xi_i - 1 = 0$. This leads to $y_i f(x_i) = 1$. (*These are correctly classified points right at the margin*)
- If $\boldsymbol{\alpha}_i = \frac{y_i}{2\lambda N}$, then the second constraint is not active ($\boldsymbol{\xi}_i \geq 0$) but the first one is active : $y_i f(x_i) + \boldsymbol{\xi}_i = 1$. This implies that $y_i f(x_i) \leq 1$. (*These points may be correctly or incorrectly classified points*)

Decision Hyperplanes



Pictorial Depiction for α values

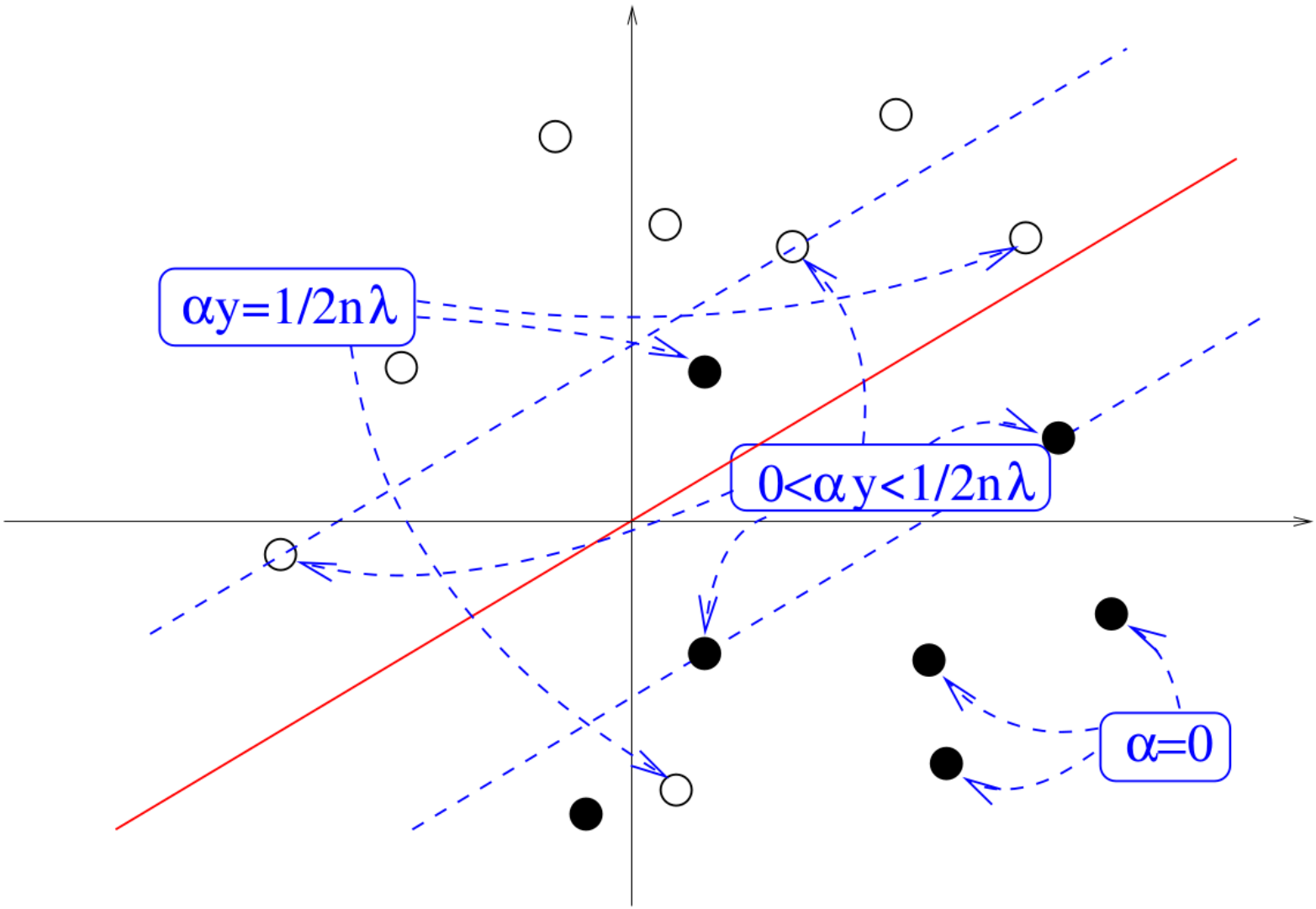


Figure: n in the figure above corresponds to N in our notation

Support Vectors

- From Representer theorem, the function evaluation at any $x \in \mathcal{X}$ (the input space) is given by

$$f(x) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i k(x_i, x) = \sum_{i \in SV} \boldsymbol{\alpha}_i k(x_i, x)$$

  where SV is the set of support vectors i.e. those training points for which $\boldsymbol{\alpha}_i \neq 0$
- Hence the name Support Vector Machines
- The above sparsity of $\boldsymbol{\alpha} \in \mathbb{R}^N$ can be used for
  - Faster prediction since once needs to go over only the support vectors

Another variant - C-SVM

Sometimes, instead of the regularization parameter λ, the SVM problem is written in the following form :

$$\min_{f \in \mathcal{H}} \left\{ C \sum_{i=1}^{N} (\ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i))^2 + \frac{1}{2} ||f||_{\mathcal{H}}^2 \right\}$$

Another variant - C-SVM

Sometimes, instead of the regularization parameter λ, the SVM problem is written in the following form :

$$\min_{f \in \mathcal{H}} \left\{ C \sum_{i=1}^{N} (\ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i))^2 + \frac{1}{2} ||f||_{\mathcal{H}}^2 \right\}$$

- This is equivalent to the original formulation on the first slide($\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i f(x_i)) + \lambda ||f||_{\mathcal{H}}^2$) with $C = \frac{1}{2N\lambda}$

Another variant - C-SVM

Sometimes, instead of the regularization parameter λ, the SVM problem is written in the following form :

$$\min_{f \in \mathcal{H}} \left\{ C \sum_{i=1}^{N} (\ell_{hinge}(y_i [K\boldsymbol{\alpha}]_i))^2 + \frac{1}{2} ||f||_{\mathcal{H}}^2 \right\}$$

- This is equivalent to the original formulation on the first slide($\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell_{hinge}(y_i f(x_i)) + \lambda ||f||_{\mathcal{H}}^2$) with $C = \frac{1}{2N\lambda}$
- Using the Lagrangian formulation, the dual can be written as

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^N} 2 \sum_{i=1}^{N} \boldsymbol{\alpha}_i y_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j K(x_i, x_j)$$

such that

$$0 \leq y_i \boldsymbol{\alpha}_i \leq C \text{ for } i = 1, \ldots, N \text{ (also called box constraints)}$$

References

- Most of the material for this lecture is based on a similar course by Julien Mairal's at ENS Paris
- Further details (with somewhat different notation) on SVMs - JST & Christianini book, Chapter 7