Introduction to Machine Learning (67577)
Lecture 8

Shai Shalev-Shwartz

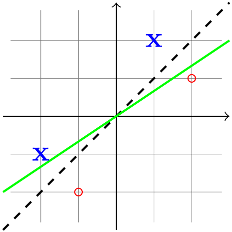
School of CS and Engineering,
The Hebrew University of Jerusalem

Support Vector Machines and Kernel Methods

Outline

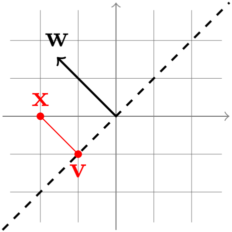
1. Support Vector Machines
   - Margin
   - hard-SVM
   - soft-SVM
   - Solving SVM using SGD
2. Kernels
   - Embeddings into feature spaces
   - The Kernel Trick
   - Examples of kernels
   - SGD with kernels
   - Duality

Which separating hyperplane is better ?



- Intuitively, dashed black is better

Margin



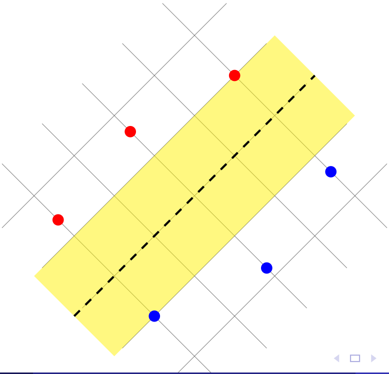
- Given hyperplane defined by $L = \{\mathbf{v} : \langle \mathbf{w}, \mathbf{v} \rangle + b = 0\}$, and given $\mathbf{x}$, the distance of $\mathbf{x}$ to L is

$$d(\mathbf{x}, L) = \min\{\|\mathbf{x} - \mathbf{v}\| : \mathbf{v} \in L\}$$

- Claim: if $\|\mathbf{w}\| = 1$ then $d(\mathbf{x}, L) = |\langle \mathbf{w}, \mathbf{x} \rangle + b|$

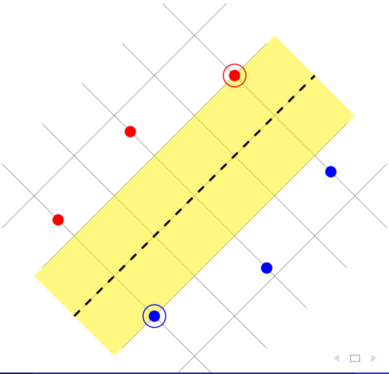
Margin and Support Vectors

- Recall: a separating hyperplane is defined by $(\mathbf{w}, b)$ s.t. $\forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$
- The margin of a separating hyperplane is the distance of the closest example to it: $\min_i |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$



Margin and Support Vectors

- Recall: a separating hyperplane is defined by $(\mathbf{w}, b)$ s.t. $\forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$
- The margin of a separating hyperplane is the distance of the closest example to it: $\min_i |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$
- The closest examples are called support vectors



Support Vector Machine (SVM)

- Hard-SVM: Seek for the separating hyperplane with largest margin

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \ \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0 \ .$$

Support Vector Machine (SVM)

- Hard-SVM: Seek for the separating hyperplane with largest margin

$$\underset{(\mathbf{w},b):\|\mathbf{w}\|=1}{\operatorname{argmax}} \ \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0 \ .$$

- Equivalently:

$$\underset{(\mathbf{w},b):\|\mathbf{w}\|=1}{\operatorname{argmax}} \ \min_{i\in[m]} \ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \ .$$

Support Vector Machine (SVM)

- Hard-SVM: Seek for the separating hyperplane with largest margin

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \; \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0 \; .$$

- Equivalently:

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \; \min_{i\in[m]} \; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \; .$$

- Equivalently:

$$(\mathbf{w}_0, b_0) = \operatorname*{argmin}_{(\mathbf{w},b)} \|\mathbf{w}\|^2 \text{ s.t. } \forall i,\; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1$$

Support Vector Machine (SVM)

- Hard-SVM: Seek for the separating hyperplane with largest margin

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \; \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0 \;.$$

- Equivalently:

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \; \min_{i\in[m]} \; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \;.$$

- Equivalently:

$$(\mathbf{w}_0, b_0) = \operatorname*{argmin}_{(\mathbf{w},b)} \|\mathbf{w}\|^2 \text{ s.t. } \forall i,\; y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1$$

- Observe: The margin of $\left(\frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}, \frac{b_0}{\|\mathbf{w}_0\|}\right)$ is $1/\|\mathbf{w}_0\|$ and is maximal margin

Margin-based Analysis

- Margin is Scale Sensitive:
  - if $(\mathbf{w}, b)$ separates $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with margin γ, then it separates $(2\mathbf{x}_1, y_1), \ldots, (2\mathbf{x}_m, y_m)$ with a margin of 2γ
  - The margin depends on the scale of the examples

Margin-based Analysis

- Margin is Scale Sensitive:
  - if $(\mathbf{w}, b)$ separates $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with margin γ, then it separates $(2\mathbf{x}_1, y_1), \ldots, (2\mathbf{x}_m, y_m)$ with a margin of 2γ
  - The margin depends on the scale of the examples
- Margin of distribution: We say that $\mathcal{D}$ is separable with a (γ, ρ)-margin if exists $(\mathbf{w}^\star, b^\star)$ s.t. $\|\mathbf{w}^\star\| = 1$ and

$$\mathcal{D}(\{(\mathbf{x}, y) : \|\mathbf{x}\| \leq \rho \wedge y(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star) \geq 1\}) = 1 \ .$$

Margin-based Analysis

- Margin is Scale Sensitive:
  - if $(\mathbf{w}, b)$ separates $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with margin γ, then it separates $(2\mathbf{x}_1, y_1), \ldots, (2\mathbf{x}_m, y_m)$ with a margin of 2γ
  - The margin depends on the scale of the examples
- Margin of distribution: We say that $\mathcal{D}$ is separable with a (γ, ρ)-margin if exists $(\mathbf{w}^\star, b^\star)$ s.t. $\|\mathbf{w}^\star\| = 1$ and

$$\mathcal{D}(\{(\mathbf{x}, y) : \|\mathbf{x}\| \le \rho \wedge y(\langle \mathbf{w}^\star, \mathbf{x}\rangle + b^\star) \ge 1\}) = 1 \ .$$

- Theorem: If $\mathcal{D}$ is separable with a (γ, ρ)-margin then the sample complexity of hard-SVM is

$$m(\epsilon, \delta) \ \le \ \frac{8}{\epsilon^2}\left(2(\rho/\gamma)^2 + \log(2/\delta)\right)$$

Margin-based Analysis

- Margin is Scale Sensitive:
  - if $(\mathbf{w}, b)$ separates $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with margin γ, then it separates $(2\mathbf{x}_1, y_1), \ldots, (2\mathbf{x}_m, y_m)$ with a margin of 2γ
  - The margin depends on the scale of the examples
- Margin of distribution: We say that $\mathcal{D}$ is separable with a (γ, ρ)-margin if exists $(\mathbf{w}^\star, b^\star)$ s.t. $\|\mathbf{w}^\star\| = 1$ and

$$\mathcal{D}(\{(\mathbf{x}, y) : \|\mathbf{x}\| \le \rho \wedge y(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star) \ge 1\}) = 1 \ .$$

- Theorem: If $\mathcal{D}$ is separable with a (γ, ρ)-margin then the sample complexity of hard-SVM is

$$m(\epsilon, \delta) \ \le \ \frac{8}{\epsilon^2} \left(2(\rho/\gamma)^2 + \log(2/\delta)\right)$$

- Unlike VC bounds, here the sample complexity depends on ρ/γ instead of d

Soft-SVM

- Hard-SVM assumes that the data is separable

Soft-SVM

- Hard-SVM assumes that the data is separable
- What if it's not? We can relax the constraint to yield soft-SVM

$$\operatorname*{argmin}_{\mathbf{w},b,\boldsymbol{\xi}} \left(\lambda \|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \right)$$

$$\text{s.t. } \forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

Soft-SVM

- Hard-SVM assumes that the data is separable
- What if it's not? We can relax the constraint to yield soft-SVM

$$\operatorname*{argmin}_{\mathbf{w},b,\boldsymbol{\xi}} \left(\lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$
$$\text{s.t. } \forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

- Can be written as regularized loss minimization:

$$\operatorname*{argmin}_{\mathbf{w},b} \left(\lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}((\mathbf{w}, b)) \right)$$

where we use the hinge loss

$$\ell^{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\langle \mathbf{w}, \mathbf{x} \rangle + b)\} \ .$$

The Homogenous Case

- Recall: by adding one more feature to $\mathbf{x}$ with the constant value of 1 we can remove the bias term
- However, this will yield a slightly different algorithm, since now we'll effectively regularize the bias term, b, as well
- This has little effect on the sample complexity, and simplify the analysis and algorithmic, so from now on we omit b

Sample complexity of soft-SVM

- Observe: soft-SVM = RLM

Sample complexity of soft-SVM

- Observe: soft-SVM = RLM
- Observe: the hinge-loss, $\mathbf{w} \mapsto \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle\}$, is $\|\mathbf{x}\|$-Lipschitz

Sample complexity of soft-SVM

- Observe: soft-SVM = RLM
- Observe: the hinge-loss, $\mathbf{w} \mapsto \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle\}$, is $\|\mathbf{x}\|$-Lipschitz
- Assume that $\mathcal{D}$ is s.t. $\|\mathbf{x}\| \leq \rho$ with probability 1

Sample complexity of soft-SVM

- Observe: soft-SVM = RLM
- Observe: the hinge-loss, $\mathbf{w} \mapsto \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$, is $\|\mathbf{x}\|$-Lipschitz
- Assume that $\mathcal{D}$ is s.t. $\|\mathbf{x}\| \leq \rho$ with probability 1
- Then, we obtain a convex-Lipschitz loss, and by the results from previous lecture, for every $\mathbf{u}$,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{\text{hinge}}(A(S))] \;\leq\; L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda\, m} \; .$$

Sample complexity of soft-SVM

- Observe: soft-SVM = RLM
- Observe: the hinge-loss, $\mathbf{w} \mapsto \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$, is $\|\mathbf{x}\|$-Lipschitz
- Assume that $\mathcal{D}$ is s.t. $\|\mathbf{x}\| \le \rho$ with probability 1
- Then, we obtain a convex-Lipschitz loss, and by the results from previous lecture, for every $\mathbf{u}$,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{\mathrm{hinge}}(A(S))] \;\le\; L_{\mathcal{D}}^{\mathrm{hinge}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda\, m}\ .$$

- Since the hinge-loss upper bounds the 0-1 loss, the right hand side is also an upper bound on $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{0-1}(A(S))]$

Sample complexity of soft-SVM

- Observe: soft-SVM $=$ RLM
- Observe: the hinge-loss, $\mathbf{w} \mapsto \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$, is $\|\mathbf{x}\|$-Lipschitz
- Assume that $\mathcal{D}$ is s.t. $\|\mathbf{x}\| \le \rho$ with probability 1
- Then, we obtain a convex-Lipschitz loss, and by the results from previous lecture, for every $\mathbf{u}$,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}^{\mathrm{hinge}}(A(S))] \ \le \ L_{\mathcal{D}}^{\mathrm{hinge}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda\, m} \ .$$

- Since the hinge-loss upper bounds the 0-1 loss, the right hand side is also an upper bound on $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{0-1}(A(S))]$
- For every $B > 0$, if we set $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$ then:

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}^{0-1}(A(S))] \ \le \ \min_{\mathbf{w}:\|\mathbf{w}\|\le B} L_{\mathcal{D}}^{\mathrm{hinge}}(\mathbf{w}) + \sqrt{\frac{8\rho^2 B^2}{m}} \ .$$

Margin/Norm vs. Dimensionality

- The VC dimension of learning halfspaces depends on the dimension, d

Margin/Norm vs. Dimensionality

- The VC dimension of learning halfspaces depends on the dimension, d
- Therefore, the sample complexity grows with d

Margin/Norm vs. Dimensionality

- The VC dimension of learning halfspaces depends on the dimension, d
- Therefore, the sample complexity grows with d
- In contrast, the sample complexity of SVM depends on $(\rho/\gamma)^2$, or equivalently, $\rho^2 B^2$

Margin/Norm vs. Dimensionality

- The VC dimension of learning halfspaces depends on the dimension, d
- Therefore, the sample complexity grows with d
- In contrast, the sample complexity of SVM depends on $(\rho/\gamma)^2$, or equivalently, $\rho^2 B^2$
- Sometimes $d \gg \rho^2 B^2$ (as we saw in the previous lecture)

- The VC dimension of learning halfspaces depends on the dimension, d
- Therefore, the sample complexity grows with d
- In contrast, the sample complexity of SVM depends on $(\rho/\gamma)^2$, or equivalently, $\rho^2 B^2$
- Sometimes $d \gg \rho^2 B^2$ (as we saw in the previous lecture)
- No contradiction to the fundamental theorem, since here we bound the error of the algorithm using $L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}^\star)$ while in the fundmental theorem we have $L_{\mathcal{D}}^{0-1}(\mathbf{w}^\star)$

Margin/Norm vs. Dimensionality

- The VC dimension of learning halfspaces depends on the dimension, d
- Therefore, the sample complexity grows with d
- In contrast, the sample complexity of SVM depends on $(\rho/\gamma)^2$, or equivalently, $\rho^2 B^2$
- Sometimes $d \gg \rho^2 B^2$ (as we saw in the previous lecture)
- No contradiction to the fundamental theorem, since here we bound the error of the algorithm using $L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}^\star)$ while in the fundmental theorem we have $L_{\mathcal{D}}^{0-1}(\mathbf{w}^\star)$
- This is an additional prior knowledge on the problem, namely, that $L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}^\star)$ is not much larger than $L_{\mathcal{D}}^{0-1}(\mathbf{w}^\star)$.

Solving SVM using SGD

SGD for solving Soft-SVM

goal: Solve $\operatorname{argmin}_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^m \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}_i\rangle\}\right)$
parameter: T
initialize: $\boldsymbol{\theta}^{(1)} = \mathbf{0}$
for $t = 1, \ldots, T$
  Let $\mathbf{w}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$
  Choose i uniformly at random from $[m]$
  If $(y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle < 1)$
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + y_i\mathbf{x}_i$
  Else
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$
output: $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^T \mathbf{w}^{(t)}$

Outline

1. Support Vector Machines
   - Margin
   - hard-SVM
   - soft-SVM
   - Solving SVM using SGD

2. Kernels
   - Embeddings into feature spaces
   - The Kernel Trick
   - Examples of kernels
   - SGD with kernels
   - Duality

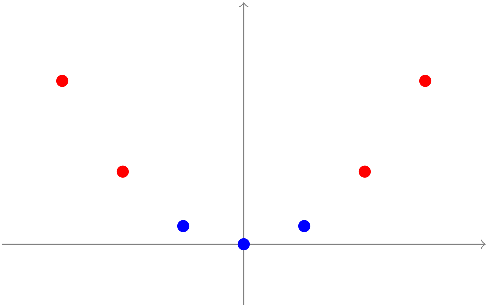
Embeddings into feature spaces

- The following sample in $\mathbb{R}^1$ is not separable by halfspaces

Embeddings into feature spaces

- The following sample in $\mathbb{R}^1$ is not separable by halfspaces

- But, if we map $x \to (x, x^2)$ it is separable by halfspaces



Embeddings into feature spaces

The general approach:

- Define $\psi : \mathcal{X} \to \mathcal{F}$, where $\mathcal{F}$ is some feature space (formally, we require $\mathcal{F}$ to be a subset of a Hilbert space)
- Train a halfspace over $(\psi(\mathbf{x}_1), y_1), \ldots, (\psi(\mathbf{x}_m), y_m)$

Embeddings into feature spaces

The general approach:

- Define $\psi : \mathcal{X} \to \mathcal{F}$, where $\mathcal{F}$ is some feature space (formally, we require $\mathcal{F}$ to be a subset of a Hilbert space)
- Train a halfspace over $(\psi(\mathbf{x}_1), y_1), \ldots, (\psi(\mathbf{x}_m), y_m)$

Questions:

- How to choose ψ ?
- If F is high dimensional we face
  - statistical challenge — can be tackled using margin
  - computational challenge — can be tackled using kernels

Choosing a mapping

- In general, requires prior knowledge
- In addition, there are some generic mappings that enrich the class of halfspaces, e.g. polynomial mappings

Polynomial mappings

- Recall, a degree k polynomial over a single variable is $p(x) = \sum_{j=0}^{k} w_j x^j$

Polynomial mappings

- Recall, a degree k polynomial over a single variable is $p(x) = \sum_{j=0}^{k} w_j x^j$
- Can be rewritten as $\langle \mathbf{w}, \psi(x) \rangle$ where $\psi(\mathbf{x}) = (1, x, x^2, \ldots, x^k)$

Polynomial mappings

- Recall, a degree k polynomial over a single variable is $p(x) = \sum_{j=0}^{k} w_j x^j$
- Can be rewritten as $\langle \mathbf{w}, \psi(x) \rangle$ where $\psi(\mathbf{x}) = (1, x, x^2, \ldots, x^k)$
- More generally, a degree k multivariate polynomial from $\mathbb{R}^n$ to $\mathbb{R}$ can be written as

$$p(\mathbf{x}) = \sum_{J \in [n]^r : r \le k} w_J \prod_{i=1}^{r} x_{J_i} .$$

Polynomial mappings

- Recall, a degree k polynomial over a single variable is $p(x) = \sum_{j=0}^{k} w_j x^j$
- Can be rewritten as $\langle \mathbf{w}, \psi(x) \rangle$ where $\psi(\mathbf{x}) = (1, x, x^2, \ldots, x^k)$
- More generally, a degree k multivariate polynomial from $\mathbb{R}^n$ to $\mathbb{R}$ can be written as

$$p(\mathbf{x}) = \sum_{J \in [n]^r : r \leq k} w_J \prod_{i=1}^{r} x_{J_i} \ .$$

- As before, we can rewrite $p(\mathbf{x}) = \langle \mathbf{w}, \psi(\mathbf{x}) \rangle$ where now $\psi : \mathbb{R}^n \to \mathbb{R}^d$ is such that for every $J \in [n]^r$, $r \leq k$, the coordinate of $\psi(\mathbf{x})$ associated with J is the monomial $\prod_{i=1}^{r} x_{J_i}$.

The Kernel Trick

- A kernel function for a mapping ψ is a function that implements inner product in the feature space, namely,

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$$

- We will see that sometimes, it is easy to calculate $K(\mathbf{x}, \mathbf{x}')$ efficiently, without applying ψ at all
- But, is this enough ?

The Representer Theorem

Theorem

Consider any learning rule of the form

$$\mathbf{w}^\star = \operatorname*{argmin}_{\mathbf{w}} \left(f\left(\langle \mathbf{w}, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \mathbf{w}, \psi(\mathbf{x}_m)\rangle\right) + \lambda \|\mathbf{w}\|^2 \right) \quad ,$$

where $f : \mathbb{R}^m \to \mathbb{R}$ *is an arbitrary function. Then,* $\exists \boldsymbol{\alpha} \in \mathbb{R}^m$ *such that* $\mathbf{w}^\star = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$.

The Representer Theorem

Theorem

Consider any learning rule of the form

$$\mathbf{w}^\star = \operatorname*{argmin}_{\mathbf{w}} \left(f\left(\langle \mathbf{w}, \psi(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{w}, \psi(\mathbf{x}_m) \rangle \right) + \lambda \|\mathbf{w}\|^2 \right) \quad ,$$

where $f : \mathbb{R}^m \to \mathbb{R}$ *is an arbitrary function. Then,* $\exists \boldsymbol{\alpha} \in \mathbb{R}^m$ *such that* $\mathbf{w}^\star = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$.

Proof.

We can rewrite $\mathbf{w}^\star$ as $\mathbf{w}^\star = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i) + \mathbf{u}$, where $\langle \mathbf{u}, \psi(\mathbf{x}_i) \rangle = 0$ for all i. Set $\mathbf{w} = \mathbf{w}^\star - \mathbf{u}$. Observe, $\|\mathbf{w}^\star\|^2 = \|\mathbf{w}\|^2 + \|\mathbf{u}\|^2$, and for every i, $\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \langle \mathbf{w}^\star, \psi(\mathbf{x}_i) \rangle$. Hence, the objective at $\mathbf{w}$ equals the objective at $\mathbf{w}^\star$ minus $\lambda \|\mathbf{u}\|^2$. By optimality of $\mathbf{w}^\star$, $\mathbf{u}$ must be zero. $\square$

Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$\mathbf{w} = \sum_i \alpha_i \psi(\mathbf{x}_i)$$

Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$\mathbf{w} = \sum_i \alpha_i \psi(\mathbf{x}_i)$$

Denote by G the matrix s.t. $G_{i,j} = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$. We have that for all i

$$\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \langle \sum_j \alpha_j \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = \sum_{j=1}^m \alpha_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = (G\boldsymbol{\alpha})_i$$

Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$\mathbf{w} = \sum_i \alpha_i \psi(\mathbf{x}_i)$$

Denote by G the matrix s.t. $G_{i,j} = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$. We have that for all i

$$\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \langle \sum_j \alpha_j \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = \sum_{j=1}^m \alpha_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = (G\boldsymbol{\alpha})_i$$

and

$$\|\mathbf{w}\|^2 = \langle \sum_j \alpha_j \psi(\mathbf{x}_j), \sum_j \alpha_j \psi(\mathbf{x}_j) \rangle = \sum_{i,j=1}^m \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle = \boldsymbol{\alpha}^\top G \boldsymbol{\alpha} \,.$$

Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$\mathbf{w} = \sum_i \alpha_i \psi(\mathbf{x}_i)$$

Denote by G the matrix s.t. $G_{i,j} = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$. We have that for all i

$$\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \langle \sum_j \alpha_j \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = \sum_{j=1}^{m} \alpha_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle = (G\boldsymbol{\alpha})_i$$

and

$$\|\mathbf{w}\|^2 = \langle \sum_j \alpha_j \psi(\mathbf{x}_j), \sum_j \alpha_j \psi(\mathbf{x}_j) \rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle = \boldsymbol{\alpha}^\top G \boldsymbol{\alpha} \,.$$

So, we can optimize over $\boldsymbol{\alpha}$

$$\operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left(f\left(G\boldsymbol{\alpha}\right) + \lambda\, \boldsymbol{\alpha}^\top G \boldsymbol{\alpha} \right)$$

The Kernel Trick

- Observe: the Gram matrix, G, only depends on inner products, and therefore can be calculated using K alone
- Suppose we found $\boldsymbol{\alpha}$, then, given a new instance,

$$\langle \mathbf{w}, \psi(\mathbf{x}) \rangle = \langle \sum_j \psi(\mathbf{x}_j), \psi(\mathbf{x}) \rangle = \sum_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}) \rangle = \sum_j K(\mathbf{x}_j, \mathbf{x})$$

- That is, we can do training and prediction using K alone

Representer Theorem for SVM

Soft-SVM:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left(\lambda \boldsymbol{\alpha}^T G \boldsymbol{\alpha} + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i (G\boldsymbol{\alpha})_i\} \right)$$

Representer Theorem for SVM

Soft-SVM:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left(\lambda \boldsymbol{\alpha}^T G \boldsymbol{\alpha} + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i (G\boldsymbol{\alpha})_i\} \right)$$

Hard-SVM

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \boldsymbol{\alpha}^T G \boldsymbol{\alpha} \quad \text{s.t.} \quad \forall i, \; y_i (G\boldsymbol{\alpha})_i \geq 1$$

Polynomial Kernels

- The k degree polynomial kernel is defined to be

$$K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^k \; .$$

Polynomial Kernels

- The k degree polynomial kernel is defined to be

$$K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^k \ .$$

- Exercise: show that if we define $\psi : \mathbb{R}^n \to \mathbb{R}^{(n+1)^k}$ s.t. for $J \in \{0, 1, \ldots, n\}^k$ there is an element of $\psi(\mathbf{x})$ that equals to $\prod_{i=1}^k x_{J_i}$, then

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle \ .$$

Polynomial Kernels

- The k degree polynomial kernel is defined to be

$$K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^k \ .$$

- Exercise: show that if we define $\psi : \mathbb{R}^n \to \mathbb{R}^{(n+1)^k}$ s.t. for $J \in \{0, 1, \ldots, n\}^k$ there is an element of $\psi(\mathbf{x})$ that equals to $\prod_{i=1}^k x_{J_i}$, then

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle \ .$$

- Since ψ contains all the monomials up to degree k, a halfspace over the range of ψ corresponds to a polynomial predictor of degree k over the original space.

Polynomial Kernels

- The k degree polynomial kernel is defined to be

$$K(\mathbf{x},\mathbf{x}') = (1 + \langle \mathbf{x},\mathbf{x}'\rangle)^k \ .$$

- Exercise: show that if we define $\psi : \mathbb{R}^n \to \mathbb{R}^{(n+1)^k}$ s.t. for $J \in \{0,1,\ldots,n\}^k$ there is an element of $\psi(\mathbf{x})$ that equals to $\prod_{i=1}^k x_{J_i}$, then

$$K(\mathbf{x},\mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}')\rangle \ .$$

- Since ψ contains all the monomials up to degree k, a halfspace over the range of ψ corresponds to a polynomial predictor of degree k over the original space.
- Observe that calculating $K(\mathbf{x},\mathbf{x}')$ takes $O(n)$ time while the dimension of $\psi(\mathbf{x})$ is n^k

Gaussian kernel (RBF)

Let the original instance space be $\mathbb{R}$ and consider the mapping ψ where for each non-negative integer $n \geq 0$ there exists an element $\psi(x)_n$ which equals to $\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n$. Then,

$$\begin{aligned}\langle \psi(x), \psi(x') \rangle &= \sum_{n=0}^{\infty} \left(\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n \right) \left(\frac{1}{\sqrt{n!}} e^{-\frac{(x')^2}{2}} (x')^n \right) \\ &= e^{-\frac{x^2+(x')^2}{2}} \sum_{n=0}^{\infty} \left(\frac{(xx')^n}{n!} \right) = e^{-\frac{(x-x')^2}{2}} .\end{aligned}$$

Gaussian kernel (RBF)

Let the original instance space be $\mathbb{R}$ and consider the mapping ψ where for each non-negative integer $n \geq 0$ there exists an element $\psi(x)_n$ which equals to $\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n$. Then,

$$\begin{aligned} \langle \psi(x), \psi(x') \rangle &= \sum_{n=0}^{\infty} \left(\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n \right) \left(\frac{1}{\sqrt{n!}} e^{-\frac{(x')^2}{2}} (x')^n \right) \\ &= e^{-\frac{x^2+(x')^2}{2}} \sum_{n=0}^{\infty} \left(\frac{(xx')^n}{n!} \right) = e^{-\frac{(x-x')^2}{2}} . \end{aligned}$$

More generally, the Gaussian kernel is defined to be

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\,\sigma}} .$$

Gaussian kernel (RBF)

Let the original instance space be $\mathbb{R}$ and consider the mapping ψ where for each non-negative integer $n \geq 0$ there exists an element $\psi(x)_n$ which equals to $\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n$. Then,

$$\langle \psi(x), \psi(x') \rangle = \sum_{n=0}^{\infty} \left(\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n \right) \left(\frac{1}{\sqrt{n!}} e^{-\frac{(x')^2}{2}} (x')^n \right)$$
$$= e^{-\frac{x^2+(x')^2}{2}} \sum_{n=0}^{\infty} \left(\frac{(xx')^n}{n!} \right) = e^{-\frac{(x-x')^2}{2}} .$$

More generally, the Gaussian kernel is defined to be

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\,\sigma}} .$$

Can learn any polynomial ...

Characterizing Kernel Functions

Lemma (Mercer's conditions)

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ implements an inner product in some Hilbert space if and only if it is positive semidefinite; namely, for all $\mathbf{x}_1, \ldots, \mathbf{x}_m$, the Gram matrix, $G_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, is a positive semidefinite matrix.

Implementing soft-SVM with kernels

- We can use a generic convex optimization algorithm on the α problem
- Alternatively, we can implement the SGD algorithm on the original $\mathbf{w}$ problem, but observe that all the operations of SGD can be implemented using the kernel alone

SGD with kernels for soft-SVM

SGD for Solving Soft-SVM with Kernels

parameter: T
Initialize: $\boldsymbol{\beta}^{(1)} = \mathbf{0} \in \mathbb{R}^m$
for $t = 1, \ldots, T$
  Let $\boldsymbol{\alpha}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\beta}^{(t)}$
  Choose i uniformly at random from $[m]$
  For all $j \neq i$ set $\beta_j^{(t+1)} = \beta_j^{(t)}$
  If $(y_i \sum_{j=1}^m \alpha_j^{(t)} K(\mathbf{x}_j, \mathbf{x}_i) < 1)$
    Set $\beta_i^{(t+1)} = \beta_i^{(t)} + y_i$
  Else
    Set $\beta_i^{(t+1)} = \beta_i^{(t)}$
Output: $\bar{\mathbf{w}} = \sum_{j=1}^m \bar{\alpha}_j \psi(\mathbf{x}_j)$ where $\bar{\boldsymbol{\alpha}} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{\alpha}^{(t)}$

Duality

- Historically, many of the properties of SVM have been obtained by considering a *dual* problem
- It is not a must, but can be helpful
- We show how to derive a dual problem to Hard-SVM:

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } \forall i, y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$$

Duality

- Hard-SVM can be rewritten as:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right)$$

Duality

- Hard-SVM can be rewritten as:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

- Lets flip the order of $\min$ and $\max$. This can only decrease the objective value, so we obtain the weak duality inequality:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right) \geq$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

Duality

- Hard-SVM can be rewritten as:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

- Lets flip the order of $\min$ and $\max$. This can only decrease the objective value, so we obtain the weak duality inequality:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right) \geq$$

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

- In our case, there's also strong duality (i.e., the above holds with equality)

Duality

- The dual problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m}\alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

Duality

- The dual problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

- We can solve analytically the inner optimization and obtain the solution

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i$$

Duality

- The dual problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\mathbf{w},\mathbf{x}_i\rangle)\right)$$

- We can solve analytically the inner optimization and obtain the solution

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

- Plugging it back, yields

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\frac{1}{2}\left\|\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i\right\|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i\langle\sum_j \alpha_j y_j \mathbf{x}_j,\mathbf{x}_i\rangle)\right) .$$

Summary

- Margin as additional prior knowledge
- Hard and Soft SVM
- Kernels