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

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

Claim: if \( \| w \| = 1 \) then \( d(x, L) = |\langle w, x \rangle + b| \)
 Margin and Support Vectors

- Recall: a separating hyperplane is defined by $(w, b)$ s.t.
  \[ \forall i, \ y_i(\langle w, x_i \rangle + b) > 0 \]
- The margin of a separating hyperplane is the distance of the closest example to it: \[ \min_i |\langle w, x_i \rangle + b| \]
Margin and Support Vectors

- Recall: a separating hyperplane is defined by \((\mathbf{w}, b)\) s.t. 
  \[ \forall i, \ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0 \]
- 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**
**Hard-SVM**: Seek for the separating hyperplane with largest margin

\[
\text{argmax} \quad \min_{\|w\|=1} \min_{i \in [m]} |\langle w, x_i \rangle + b| \quad \text{s.t.} \quad \forall i, y_i(\langle w, x_i \rangle + b) > 0.
\]
Support Vector Machine (SVM)

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

\[
\argmax_{(w,b):\|w\|=1} \min_{i \in [m]} |\langle w, x_i \rangle + b| \quad \text{s.t.} \quad \forall i, \ y_i(\langle w, x_i \rangle + b) > 0 .
\]

- Equivalently:

\[
\argmax_{(w,b):\|w\|=1} \min_{i \in [m]} y_i(\langle w, x_i \rangle + b) .
\]
Support Vector Machine (SVM)

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

\[
\text{argmax } \min_{(w,b): \|w\|=1} \left| \langle w, x_i \rangle + b \right| \quad \text{s.t.} \quad \forall i, y_i(\langle w, x_i \rangle + b) > 0.
\]

Equivalently:

\[
\text{argmax } \min_{(w,b): \|w\|=1} y_i(\langle w, x_i \rangle + b).
\]

Equivalently:

\[
(w_0, b_0) = \text{argmin } \|w\|^2 \quad \text{s.t.} \quad \forall i, y_i(\langle w, x_i \rangle + b) \geq 1
\]
Support Vector Machine (SVM)

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

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

- Equivalently:

  \[
  \text{argmax}_{(w,b):\|w\|=1} \min_{i \in [m]} y_i(\langle w, x_i \rangle + b) .
  \]

- Equivalently:

  \[
  (w_0, b_0) = \text{argmin}_{(w,b)} \|w\|^2 \quad \text{s.t.} \quad \forall i, \ y_i(\langle w, x_i \rangle + b) \geq 1
  \]

- Observe: The margin of \( \left( \frac{w_0}{\|w_0\|}, \frac{b_0}{\|w_0\|} \right) \) is \( 1/\|w_0\| \) and is maximal margin
Margin-based Analysis

- **Margin is Scale Sensitive:**
  - if \((w, b)\) separates \((x_1, y_1), \ldots, (x_m, y_m)\) with margin \(\gamma\), then it separates \((2x_1, y_1), \ldots, (2x_m, y_m)\) with a margin of \(2\gamma\)
  - The margin depends on the scale of the examples
Margin-based Analysis

- **Margin is Scale Sensitive:**
  - if \((w, b)\) separates \((x_1, y_1), \ldots, (x_m, y_m)\) with margin \(\gamma\), then it separates \((2x_1, y_1), \ldots, (2x_m, y_m)\) with a margin of \(2\gamma\)
  - The margin depends on the scale of the examples

- **Margin of distribution:** We say that \(D\) is separable with a \((\gamma, \rho)\)-margin if exists \((w^*, b^*)\) s.t. \(\|w^*\| = 1\) and

\[
D(\{(x, y) : \|x\| \leq \rho \land y(\langle w^*, x \rangle + b^*) \geq 1\}) = 1.
\]
Margin-based Analysis

- **Margin is Scale Sensitive:**
  - if \((w, b)\) separates \((x_1, y_1), \ldots, (x_m, y_m)\) with margin \(\gamma\), then it separates \((2x_1, y_1), \ldots, (2x_m, y_m)\) with a margin of \(2\gamma\)
  - The margin depends on the scale of the examples

- **Margin of distribution:** We say that \(D\) is separable with a \((\gamma, \rho)\)-margin if exists \((w^*, b^*)\) s.t. \(\|w^*\| = 1\) and
  \[
  D(\{(x, y) : \|x\| \leq \rho \land y(\langle w^*, x \rangle + b^*) \geq 1\}) = 1.
  \]

- **Theorem:** If \(D\) is separable with a \((\gamma, \rho)\)-margin then the sample complexity of hard-SVM is
  \[
  m(\epsilon, \delta) \leq \frac{8}{\epsilon^2} \left(2(\rho/\gamma)^2 + \log(2/\delta)\right)
  \]
Margin-based Analysis

- **Margin is Scale Sensitive:**
  - if \((w, b)\) separates \((x_1, y_1), \ldots, (x_m, y_m)\) with margin \(\gamma\), then it separates \((2x_1, y_1), \ldots, (2x_m, y_m)\) with a margin of \(2\gamma\)
  - The margin depends on the scale of the examples

- **Margin of distribution:** We say that \(D\) is separable with a \((\gamma, \rho)\)-margin if exists \((w^*, b^*)\) s.t. \(\|w^*\| = 1\) and

\[
D(\{(x, y) : \|x\| \leq \rho \land y(\langle w^*, x \rangle + b^*) \geq 1\}) = 1.
\]

- **Theorem:** If \(D\) is separable with a \((\gamma, \rho)\)-margin then the sample complexity of hard-SVM is

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

- Unlike VC bounds, here the sample complexity depends on \(\rho/\gamma\) 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

\[
\arg\min_{\mathbf{w}, b, \xi} \left( \lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)
\]

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

\[
\ell_{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}, y)) = \max \{0, 1 - y (\langle \mathbf{w}, \mathbf{x} \rangle + b)\}.
\]
Soft-SVM

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

\[
\arg\min_{w,b,\xi} \left( \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)
\]

s.t. \( \forall i, \ y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \) and \( \xi_i \geq 0 \)

- Can be written as regularized loss minimization:

\[
\arg\min_{w,b} \left( \lambda \|w\|^2 + L^{hinge}_{S}((w, b)) \right)
\]

where we use the hinge loss

\[
\ell^{hinge}((w, b), (x, y)) = \max\{0, 1 - y(\langle w, x \rangle + b)\}.
\]
The Homogenous Case

- Recall: by adding one more feature to $x$ 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

\[ \text{Observe: hinge-loss, } \mathbf{w} \mapsto \max \{ 0, 1 - y \langle \mathbf{w}, \mathbf{x} \rangle \}, \text{ is } \| \mathbf{x} \| \text{-Lipschitz} \]

Assume that \( D \) 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 \( u \),

\[ \mathbb{E}_{S \sim D} [L_{\text{hinge}}(A(S))] \leq L_{\text{hinge}}(u) + \lambda \| u \|_2^2 + 2\rho^2 \]

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

For every \( B > 0 \), if we set \( \lambda = \sqrt{\frac{8}{\rho^2} B^2} \), then:

\[ \mathbb{E}_{S \sim D} [L_{0-1}(A(S))] \leq \min_{\| \mathbf{w} \| \leq B} L_{\text{hinge}}(\mathbf{w}) + \sqrt{\frac{8}{\rho^2} B^2} \]
Sample complexity of soft-SVM

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

$$
\mathbb{E}_{S \sim \mathcal{D}^m} [L^\text{hinge}_\mathcal{D}(A(S))] \leq L^\text{hinge}_\mathcal{D}(u) + \lambda\|u\|^2 + \frac{2\rho^2}{\lambda m}.
$$
Sample complexity of soft-SVM

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

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

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

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

$$
\mathbb{E}_{S \sim D^m}[L^{\text{hinge}}_D(A(S))] \leq L^{\text{hinge}}_D(u) + \lambda \|u\|^2 + \frac{2\rho^2}{\lambda m}.
$$

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

$$
\mathbb{E}_{S \sim D^m}[L^{0-1}_D(A(S))] \leq \min_{w: \|w\| \leq B} L^{\text{hinge}}_D(w) + \sqrt{\frac{8\rho^2B^2}{m}}.
$$
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 $\left(\frac{\rho}{\gamma}\right)^2$ 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 $L_{\text{hinge}} D(w^\star)$ while in the fundamental theorem we have $L_0 D(w^\star)$. This is an additional prior knowledge on the problem, namely, that $L_{\text{hinge}} D(w^\star)$ is not much larger than $L_0 D(w^\star)$. 
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, $\rho^2 B^2$
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)
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, $\rho^2B^2$
- Sometimes $d \gg \rho^2B^2$ (as we saw in the previous lecture)
- No contradiction to the fundamental theorem, since here we bound the error of the algorithm using $L_{D}^{\text{hinge}}(w^*)$ while in the fundamental theorem we have $L_{D}^{0-1}(w^*)$
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 $L_{D}^{\text{hinge}}(w^*)$ while in the fundamental theorem we have $L_{D}^{0-1}(w^*)$
- This is an additional prior knowledge on the problem, namely, that $L_{D}^{\text{hinge}}(w^*)$ is not much larger than $L_{D}^{0-1}(w^*)$. 
Solving SVM using SGD

SGD for solving Soft-SVM

goal: Solve $\arg\min_w \left( \frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y\langle w, x_i \rangle\} \right)$

parameter: $T$

initialize: $\theta^{(1)} = 0$

for $t = 1, \ldots, T$

Let $w^{(t)} = \frac{1}{\lambda t} \theta^{(t)}$

Choose $i$ uniformly at random from $[m]$

If $(y_i \langle w^{(t)}, x_i \rangle < 1)$

Set $\theta^{(t+1)} = \theta^{(t)} + y_i x_i$

Else

Set $\theta^{(t+1)} = \theta^{(t)}$

output: $\bar{w} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)}$
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

\[ x \rightarrow (x, x^2) \]
The following sample in $\mathbb{R}^1$ is not separable by halfspaces.

But, if we map $x \rightarrow (x, x^2)$ it is separable by halfspaces.
Embeddings into feature spaces

The general approach:

- Define $\psi : \mathcal{X} \rightarrow \mathcal{F}$, where $\mathcal{F}$ is some feature space (formally, we require $\mathcal{F}$ to be a subset of a Hilbert space)
- Train a halfspace over $(\psi(x_1), y_1), \ldots, (\psi(x_m), y_m)$
Embeddings into feature spaces

The general approach:

- Define \( \psi : X \rightarrow F \), where \( F \) is some feature space
  (formally, we require \( F \) to be a subset of a Hilbert space)
- Train a halfspace over \( (\psi(x_1), y_1), \ldots, (\psi(x_m), y_m) \)

Questions:
- How to choose \( \psi \) ?
- 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
  
  $$p(x) = \sum_{j=0}^{k} w_j x^j$$

- More generally, a degree $k$ multivariate polynomial from $\mathbb{R}^n$ to $\mathbb{R}$ can be written as

  $$\sum_{J \in \mathbb{N}^n} \sum_{r \leq k} w_J r \prod_{i=1}^{r} x^J_i$$

  As before, we can rewrite $p(x) = \langle w, \psi(x) \rangle$ where now $\psi: \mathbb{R}^n \to \mathbb{R}^d$ is such that for every $J \in \mathbb{N}^n$, $r \leq k$, the coordinate of $\psi(x)$ associated with $J$ is the monomial $\prod_{i=1}^{r} x^J_i$. 
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 $\langle w, \psi(x) \rangle$ where $\psi(x) = (1, x, x^2, \ldots, x^k)$
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 $\langle w, \psi(x) \rangle$ where $\psi(x) = (1, x, x^2, \ldots, x^k)$
- More generally, a degree $k$ multivariate polynomial from $\mathbb{R}^n$ to $\mathbb{R}$ can be written as
  \[ p(x) = \sum_{J \in [n]^r : r \leq k} w_J \prod_{i=1}^{r} x_{J_i} \]
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 \( \langle w, \psi(x) \rangle \) where \( \psi(x) = (1, x, x^2, \ldots, x^k) \)
- More generally, a degree $k$ multivariate polynomial from $\mathbb{R}^n$ to $\mathbb{R}$ can be written as
  \[ p(x) = \sum_{J \in [n]^r : r \leq k} w_J \prod_{i=1}^{r} x_{J_i} \]
- As before, we can rewrite $p(x) = \langle w, \psi(x) \rangle$ 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 $\psi(x)$ associated with $J$ is the monomial $\prod_{i=1}^{r} x_{J_i}$. 
The Kernel Trick

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

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

- We will see that sometimes, it is easy to calculate $K(x, x')$ efficiently, without applying $\psi$ at all.

- But, is this enough?
The Representer Theorem

Theorem

Consider any learning rule of the form

\[ w^* = \underset{w}{\text{argmin}} \left( f \left( \langle w, \psi(x_1) \rangle, \ldots, \langle w, \psi(x_m) \rangle \right) + \lambda \| w \|^2 \right), \]

where \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) is an arbitrary function. Then, \( \exists \alpha \in \mathbb{R}^m \) such that

\[ w^* = \sum_{i=1}^{m} \alpha_i \psi(x_i). \]
The Representer Theorem

**Theorem**

Consider any learning rule of the form

\[ w^* = \arg\min_w \left( f (\langle w, \psi(x_1) \rangle, \ldots, \langle w, \psi(x_m) \rangle) + \lambda \| w \|_2^2 \right), \]

where \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) is an arbitrary function. Then, \( \exists \alpha \in \mathbb{R}^m \) such that

\[ w^* = \sum_{i=1}^m \alpha_i \psi(x_i). \]

**Proof.**

We can rewrite \( w^* \) as

\[ w^* = \sum_{i=1}^m \alpha_i \psi(x_i) + u, \]

where \( \langle u, \psi(x_i) \rangle = 0 \) for all \( i \). Set \( w = w^* - u \). Observe, \( \| w^* \|_2^2 = \| w \|_2^2 + \| u \|_2^2 \), and for every \( i \),

\[ \langle w, \psi(x_i) \rangle = \langle w^*, \psi(x_i) \rangle. \]

Hence, the objective at \( w \) equals the objective at \( w^* \) minus \( \lambda \| u \|_2^2 \). By optimality of \( w^* \), \( u \) must be zero. \( \square \)
Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$w = \sum_i \alpha_i \psi(x_i)$$
Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$ w = \sum_i \alpha_i \psi(x_i) $$

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

$$ \langle w, \psi(x_i) \rangle = \langle \sum_j \alpha_j \psi(x_j), \psi(x_i) \rangle = \sum_{j=1}^{m} \alpha_j \langle \psi(x_j), \psi(x_i) \rangle = (G\alpha)_i $$
Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$w = \sum_i \alpha_i \psi(x_i)$$

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

$$\langle w, \psi(x_i) \rangle = \langle \sum_j \alpha_j \psi(x_j), \psi(x_i) \rangle = \sum_{j=1}^m \alpha_j \langle \psi(x_j), \psi(x_i) \rangle = (G\alpha)_i$$

and

$$\|w\|^2 = \langle \sum_j \alpha_j \psi(x_j), \sum_j \alpha_j \psi(x_j) \rangle = \sum_{i,j=1}^m \alpha_i \alpha_j \langle \psi(x_i), \psi(x_j) \rangle = \alpha^\top G\alpha.$$
Implications of Representer Theorem

By representer theorem, optimal solution can be written as

$$w = \sum_{i} \alpha_{i} \psi(x_{i})$$

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

$$\langle w, \psi(x_{i}) \rangle = \langle \sum_{j} \alpha_{j} \psi(x_{j}), \psi(x_{i}) \rangle = \sum_{j=1}^{m} \alpha_{j} \langle \psi(x_{j}), \psi(x_{i}) \rangle = (G\alpha)_{i}$$

and

$$\|w\|^{2} = \langle \sum_{j} \alpha_{j} \psi(x_{j}), \sum_{j} \alpha_{j} \psi(x_{j}) \rangle = \sum_{i,j=1}^{m} \alpha_{i} \alpha_{j} \langle \psi(x_{i}), \psi(x_{j}) \rangle = \alpha^{\top} G\alpha .$$

So, we can optimize over $\alpha$

$$\arg\min_{\alpha \in \mathbb{R}^{m}} \left( f(G\alpha) + \lambda \alpha^{\top} G\alpha \right)$$
The Kernel Trick

- Observe: the Gram matrix, $G$, only depends on inner products, and therefore can be calculated using $K$ alone.

- Suppose we found $\alpha$, then, given a new instance,

\[
\langle \mathbf{w}, \psi(x) \rangle = \langle \sum_j \psi(x_j), \psi(x) \rangle = \sum_j \langle \psi(x_j), \psi(x) \rangle = \sum_j K(x_j, x)
\]

- That is, we can do training and prediction using $K$ alone.
Representer Theorem for SVM

Soft-SVM:

\[
\min_{\alpha \in \mathbb{R}^m} \left( \lambda \alpha^T G \alpha + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i (G \alpha)_i\} \right)
\]
Representer Theorem for SVM

Soft-SVM:

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

Hard-SVM

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

The $k$ degree polynomial kernel is defined to be

$$K(x, x') = (1 + \langle x, x' \rangle)^k.$$
The $k$ degree polynomial kernel is defined to be

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

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 $\psi(x)$ that equals to $\prod_{i=1}^k x_{J_i}$, then

$$K(x, x') = \langle \psi(x), \psi(x') \rangle .$$
The $k$ degree polynomial kernel is defined to be

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

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

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

Since $\psi$ contains all the monomials up to degree $k$, a halfspace over the range of $\psi$ corresponds to a polynomial predictor of degree $k$ over the original space.
Polynomial Kernels

- The $k$ degree polynomial kernel is defined to be

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

- 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 $\psi(x)$ that equals to $\prod_{i=1}^k x_{J_i}$, then

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

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

- Observe that calculating $K(x, x')$ takes $O(n)$ time while the dimension of $\psi(x)$ is $n^k$. 


Shai Shalev-Shwartz (Hebrew U)
IML Lecture 8
SVM 23 / 31
Gaussian kernel (RBF)

Let the original instance space be $\mathbb{R}$ and consider the mapping $\psi$ 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 $K(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$. Can learn any polynomial...
Gaussian kernel (RBF)

Let the original instance space be $\mathbb{R}$ and consider the mapping $\psi$ 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

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

Let the original instance space be $\mathbb{R}$ and consider the mapping $\psi$ 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

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

Can learn any polynomial ...
Lemma (Mercer’s conditions)

A symmetric function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ implements an inner product in some Hilbert space if and only if it is positive semidefinite; namely, for all $x_1, \ldots, x_m$, the Gram matrix, $G_{i,j} = K(x_i, x_j)$, is a positive semidefinite matrix.
Implementing soft-SVM with kernels

- We can use a generic convex optimization algorithm on the $\alpha$ problem.
- Alternatively, we can implement the SGD algorithm on the original $w$ problem, but observe that all the operations of SGD can be implemented using the kernel alone.
SGD for Solving Soft-SVM with Kernels

**parameter:** $T$

**Initialize:** $\beta^{(1)} = 0 \in \mathbb{R}^m$

for $t = 1, \ldots, T$

Let $\alpha^{(t)} = \frac{1}{\lambda t} \beta^{(t)}$

Choose $i$ uniformly at random from $[m]$

For all $j \neq i$ set $\beta^{(t+1)}_j = \beta^{(t)}_j$

If $(y_i \sum_{j=1}^m \alpha_j^{(t)} K(x_j, x_i) < 1)$

Set $\beta^{(t+1)}_i = \beta^{(t)}_i + y_i$

Else

Set $\beta^{(t+1)}_i = \beta^{(t)}_i$

**Output:** $\bar{w} = \sum_{j=1}^m \bar{\alpha}_j \psi(x_j)$ where $\bar{\alpha} = \frac{1}{T} \sum_{t=1}^T \alpha^{(t)}$
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:

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad \forall i, y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$$
Hard-SVM can be rewritten as:

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

- Hard-SVM can be rewritten as:

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

- Let's flip the order of \( \min \) and \( \max \). This can only decrease the objective value, so we obtain the weak duality inequality:

\[
\min_{\mathbf{w}} \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left( \frac{1}{2} \| \mathbf{w} \|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right) \geq \\
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \min_{\mathbf{w}} \left( \frac{1}{2} \| \mathbf{w} \|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right)
\]
Hard-SVM can be rewritten as:

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

Let's flip the order of \(\min\) and \(\max\). This can only decrease the objective value, so we obtain the weak duality inequality:

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

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

The dual problem:

\[
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \min_w \left( \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle w, x_i \rangle) \right)
\]
Duality

- The dual problem:

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

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

$$w = \sum_{i=1}^m \alpha_i y_i x_i$$
Duality

- The dual problem:

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

- 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

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

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