Lecture 5: Support Vector Machines (hard-SVM)

Introduction to Learning and Analysis of Big Data
Linear predictors: Intermediate summary

- Linear predictors are very popular, because
  - If the sample size is $\Theta(d)$ (e.g. 10 times the dimensions), the training error and the true error will probably be similar
  - For many natural problems, there are linear predictors with low error.

- Computing the ERM for a linear predictor is NP-hard.

- But in the realizable/separable case, there are efficient algorithms:
  - Using Linear Programming;
  - The Batch Perceptron algorithm.
Linear predictors: Problems and Solutions

- Two problems:
  - In many cases the dimension is huge: we don’t have $\Theta(d)$ examples.
  - In many cases the problem is not separable.

- One solution: **Support Vector Machine** (SVM).

- We will first study SVM for separable problems.
Which linear predictor to choose?

- ERM: Find a linear predictor which has lowest error on the sample.
- Separable case: ERM finds a predictor with zero training error.
- Which one should the algorithm choose?

Intuitively: choose the one with the largest margin.

We expect this to work better for new unseen examples.
The large-margin algorithm

Large-margin algorithm
Select the separating linear predictor with maximal margin on sample.

- This is an ERM algorithm.
- Can this algorithm work better than selecting an arbitrary separator?
- In other words, can we have less overfitting?
- **Overfitting**: When true error is much larger than training error.
- **Sample complexity**: The sample size which is needed to prevent overfitting.
- The sample complexity of an arbitrary ERM is $\Theta(d)$.
- The sample complexity of the large-margin algorithm can be much better!
- It depends on the **margin of the distribution**.
The margin

- The sample complexity of the large-margin algorithm depends on the margin of the distribution.
- Recall margin definition:

\[ R \triangleq \max_i \|x_i\|, \]
\[
\text{For } w \text{ such that } \text{err}(h_w, S) = 0, \quad \gamma(w) \triangleq \frac{1}{R} \min_{i \leq m} \frac{|\langle w, x_i \rangle|}{\|w\|}.
\]

- Maximal margin on \( S \): \( \gamma_S \triangleq \max \{ \gamma(w) \mid w \in \mathbb{R}^d, \text{err}(h_w, S) = 0 \} \).
- This is the sample margin, it depends on \( S \).
- What is the distribution margin?
- Need to define “\( R \)” and “\( \gamma(w) \)” that do not depend on a sample.
The distribution margin

Sample margin

- \( R := \max_i \|x_i\|, \)
- For \( w \) such that \( \text{err}(h_w, S) = 0 \), \( \gamma(w) := \frac{1}{R} \min_{i \leq m} \frac{|\langle w, x_i \rangle|}{\|w\|}. \)
- Let \( \mathcal{D} \) be a distribution over \( \mathcal{X} \times \mathcal{Y}, \mathcal{X} = \mathbb{R}^d. \)

Distribution margin

- \( R_D := \min \{ r \in \mathbb{R} | \mathbb{P}_{(X,Y) \sim \mathcal{D}}[\|X\| \leq r] = 1 \}. \)
- For \( w \) such that \( \text{err}(h_w, \mathcal{D}) = 0 \),
  \[ \gamma_D(w) := \max \left\{ \gamma \in \mathbb{R} \left| \mathbb{P}_{(X,Y) \sim \mathcal{D}} \left[ \frac{1}{R_D} \frac{|\langle w, X \rangle|}{\|w\|} \geq \gamma \right] = 1 \right\} . \]
- The distribution margin of \( \mathcal{D} \):
  \[ \gamma_* := \max \{ \gamma_D(w) | w \in \mathbb{R}^d, \text{err}(w, \mathcal{D}) = 0 \}. \]
Large margin reduces sample complexity

Theorem

The sample complexity of the large-margin algorithm is $O\left(\frac{1}{\gamma_*^2}\right)$.

- We will not prove this here.
- Compare to ERM:
  - ERM selects any linear predictor with lowest training error.
  - Sample complexity of $O(d)$.
- For a separable sample: There are many ERM solutions, but only one large-margin solution.
- Large-margin is also an ERM algorithm!
- So large-margin sample complexity is $O(\min(d, \frac{1}{\gamma_*^2}))$.
- A surprising similarity:
  - Recall: The Perceptron requires $O\left(\frac{1}{\gamma_*^2}\right)$ updates
  - Indeed, the perceptron actually requires only $O\left(\frac{1}{\gamma_*^2}\right)$ training examples.
- How to implement the large-margin algorithm?
The Hard-SVM algorithm

- **Hard-SVM**: Find the separator with the largest *sample margin*.
- **Hard** := Works only on a separable sample.
- **SVM** := Support-Vector Machines.
- What are Support Vectors?
Support vectors

- Support Vectors: a set of sample points that determine the maximal-margin separator.
- The support vectors are all **on the margin**.
- In $\mathbb{R}^d$, any $d$ support vectors suffice to determine the separator.
- There could be more than $d$ support vectors.
- There could be less than $d$ support vectors.
- There is only one maximal-margin separator.
The Hard-SVM algorithm

**Hard-SVM**

**input**  A separable training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

**output**  $\hat{w} \in \mathbb{R}^d$ such that $\forall i \leq m, h_w(x_i) = y_i$.

1: Find $\hat{w}$ that solves the following problem:

$$\begin{align*}
\text{Minimize}_{w \in \mathbb{R}^d} & \quad \|w\|^2 \\
\text{s.t.} & \quad \forall i, y_i \langle w, x_i \rangle \geq 1.
\end{align*}$$

2: Return $\hat{w}$.

- We will see soon how to do the minimization.
- Does this find the maximal-margin separator?
The Hard-SVM algorithm

- Hard-SVM minimization problem: Minimize $\|w\|^2$ s.t. $\forall i, y_i \langle w, x_i \rangle \geq 1$.
- **Claim:** Hard-SVM returns $\hat{w}$ which is a maximal-margin separator.
- In other words: For any $w \in \mathbb{R}^d$ such that $\text{err}(h_w, S) = 0$, $\gamma(w) \leq \gamma(\hat{w})$.
- **Proof:**
  - For the solution $\hat{w}$, we have $\forall i, y_i \langle \hat{w}, x_i \rangle \geq 1$.
  - The margin of $\hat{w}$ is $\gamma(\hat{w}) = \frac{1}{R} \min \frac{|\langle \hat{w}, x_i \rangle|}{\|\hat{w}\|} \geq \frac{1}{R \|\hat{w}\|}$.
  - Let $v \in \mathbb{R}^d$ that satisfies the constraints, such that $y_i \langle v, x_i \rangle = 1$ for some $i$.
  - Then $\gamma(v) = \frac{1}{R \|v\|}$.
  - We have $\|v\| \geq \|\hat{w}\|$. Therefore $\gamma(v) = \frac{1}{R \|v\|} \leq \frac{1}{R \|\hat{w}\|} \leq \gamma(\hat{w})$.
  - What about $w' \in \mathbb{R}^d$ that does not satisfy the constraints?
    - If $\text{err}(h_{w'}, S) > 0$ it is not a separator.
    - If $\text{err}(h_{w'}, S) = 0$, define $v' := \frac{w'}{\min_i |\langle w', x_i \rangle|}$.
    - Then $v'$ satisfies the properties of $v$ above.
    - So $\gamma(v') \leq \gamma(\hat{w})$.
    - Also, $\gamma(w') = \gamma(v')$.
    - So $\gamma(w') \leq \gamma(\hat{w})$. □
The Hard-SVM algorithm

- Hard-SVM minimization problem: Minimize $\|w\|^2$ s.t. $\forall i, y_i \langle w, x_i \rangle \geq 1$.
- How can we solve this problem?
- Is this a Linear Program?

### Quadratic Program

A QP is a problem of the following form:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} w^T \cdot H \cdot w + \langle u, w \rangle$$

subject to $Aw \geq v$.

- $w \in \mathbb{R}^d$: a vector we wish to find.
- $u \in \mathbb{R}^d$, $H \in \mathbb{R}^{d \times d}$, $v \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times d}$.
- The values of $u, H, v, A$ define the specific quadratic program.
- $H$ must be a **positive definite matrix** (real positive eigenvalues).
- QPs can be solved efficiently 🎈
- Many solvers are available 😊.
- In Matlab: $w = \text{quadprog}(H,u,-A,-v)$. 

Kontorovich and Sabato (BGU) 
Lecture 5 
13 / 17
### Solving Hard-SVM with Quadratic Programming

<table>
<thead>
<tr>
<th>Quadratic program</th>
<th>Hard-SVM minimization problem</th>
</tr>
</thead>
</table>
| \[
\begin{align*}
\text{minimize}_{w \in \mathbb{R}^d} & \quad \frac{1}{2} w^T \cdot H \cdot w + \langle u, w \rangle \\
\text{subject to} & \quad A w \geq v.
\end{align*}
\] | \[
\begin{align*}
\text{Minimize } & \|w\|^2 \text{ s.t. } \forall i, y_i \langle w, x_i \rangle \geq 1.
\end{align*}
\] |

- Define \( \text{Id} := \) The identity matrix (diagonal with 1’s on the diagonal.)
- Expressing \( \|w\|^2 \) as \( \frac{1}{2} w^T \cdot H \cdot w \):
  \[
  \|w\|^2 = \sum_{i=1}^{d} w(i)^2 = (w(1), w(2), \ldots, w(d)) \cdot \text{Id} \cdot \begin{pmatrix} w(1) \\ w(2) \\ \vdots \\ w(d) \end{pmatrix}.
  \]
- So \( H = 2 \cdot \text{Id} \) and \( u = (0, \ldots, 0). \)
- \( A \) and \( v \) are as in the Perceptron: \( v = (1, \ldots, 1), \) row \( i \) of the matrix \( A \) is \( y_i x_i \equiv (y_i \cdot x_i(1), \ldots, y_i \cdot x_i(d)). \)
Hard-SVM: Recap

- Hard-SVM works on separable problems.
- It finds the linear predictor with the maximal margin on the training sample.
- It requires only $O(\min(d, \frac{1}{\gamma^2}))$ training examples.
  - Recall: A general ERM for linear predictors requires $O(d)$ examples.
- Hard-SVM can be solved efficiently with a quadratic program.
Example: margin versus dimension

- Hard-SVM improves the sample complexity over ERM if $\frac{1}{\gamma_*^2} \ll d$.
- Example: Classifying news items — is it about economics?
- Representation: **Bag of Words**.
  - $X = \mathbb{R}^d$, each coordinate represents a word in English.
  - $x = a$ binary vector, $x(i) = \mathbb{I}[\text{word } i \text{ exists in the document}]$.
- $d =$ number of words in the English language.
- We will show that under reasonable assumptions, the sample complexity of hard-SVM is smaller than a general ERM.
Example: margin versus dimension

- Recall:

\[ R_D := \min\{r \in \mathbb{R} \mid \mathbb{P}(X,Y) \sim \mathcal{D}[\|X\| \leq r] = 1\}. \]

\[ \text{err}(h_w, \mathcal{D}) = 0 \Rightarrow \gamma_D(w) := \max \left\{ \gamma \in \mathbb{R} \mid \mathbb{P}(X,Y) \sim \mathcal{D} \left[ \frac{1}{R_D} \frac{|\langle w, X \rangle|}{\|w\|} \geq \gamma \right] = 1 \right\}. \]

\[ \gamma_* := \max\{\gamma_D(w) \mid w \in \mathbb{R}^d, \text{err}(w, \mathcal{D}) = 0\}. \]

- There are \( d \) coordinates (possible words, say \( d = 100,000 \))

- Suppose:
  - At most \( n \) different words in each document (say \( n = 1,000 \))
  - There is a separating \( w_* \) which is **sparse**: Only a small number of coordinates \( k \) have non-zero weights. (say \( k = 10 \)).
  - And all non-zero coordinates are either \(-1\) or \(1\) in \( w_* \).

- Then:
  - For all possible documents \( x \), \( |\langle w_*, x \rangle| \geq 1 \)
  - \( \|w_*\| = \sqrt{\sum_{i=1}^{d} w_*^2(i)} \leq \sqrt{k} \)
  - \( R_D \leq \sqrt{n} \)
  - \( \gamma(w_*) \geq \frac{1}{\sqrt{nk}} \)

- We conclude that \( \frac{1}{\gamma_*^2} \ll d \) if \( nk \ll d \).

- So, large-margin learning will require a smaller sample than a general ERM.