Machine Learning (CSE 446): Support Vector Machines

Sham M Kakade
© 2018

University of Washington
cse446-staff@cs.washington.edu
Quick Review: Kernels and Kernelized Perceptron
Kernels

A kernel function (implicitly) computes:

\[ K(x, v) = \phi(x) \cdot \phi(v) \]

for some \( \phi \). Typically it is cheap to compute \( K(\cdot, \cdot) \), and we never explicitly represent \( \phi(v) \) for any vector \( v \).

Some kernels:

- **linear** \( K^{\text{linear}}(x, v) = x \cdot v \)
- **quadratic** \( K^{\text{quad}}(x, v) = (1 + x \cdot v)^2 \)
- **cubic** \( K^{\text{cubic}}(x, v) = (1 + x \cdot v)^3 \)
- **polynomial** \( K^{\text{poly}}_p(x, v) = (1 + x \cdot v)^p \)
- **radial basis function** \( K^{\text{rbf}}_{\gamma}(x, v) = \exp \left( -\gamma \|x - v\|_2^2 \right) \)
- **hyperbolic tangent** \( K^{\text{tanh}}(x, v) = \tanh(1 + x \cdot v) \) (not a kernel)
- **all conjunctions** \( K^{\text{all conj}}(x, v) = \prod_{j=1}^{d}(1 + x_j \cdot v_j) \) (for binary features)
At every stage of learning, there exist \( \langle \alpha_1, \alpha_2, \ldots, \alpha_N \rangle \) such that

\[
w = \sum_{n=1}^{N} \alpha_n \cdot x_n = \alpha^\top X
\]

In other words, \( w \) is always in the span of the training data.
\( \phi(x_n) \) is Never Explicitly Computed!

\[
\begin{align*}
\text{predict: } & \quad \hat{y} = \text{sign}\left( \sum_{i=1}^{N} \alpha_i \cdot K(x_i, x_n) + b \right) \\
\text{update: } & \quad \alpha_n^{(\text{new})} \leftarrow \alpha_n^{(\text{old})} + y_n
\end{align*}
\]

We only calculate inner products of such vectors.
Kernelized Perceptron Learning Algorithm

Data: $D = \langle (x_n, y_n) \rangle_{n=1}^{N}$, number of epochs $E$

Result: weights $\alpha$ and bias $b$

initialize: $\alpha = 0$ and $b = 0$;

for $e \in \{1, \ldots, E\}$ do

    for $n \in \{1, \ldots, N\}$, in random order do

        # predict

        $\hat{y} = \text{sign} \left( \sum_{i=1}^{N} \alpha_i \cdot K(x_i, x_n) + b \right)$;

        if $\hat{y} \neq y_n$ then

            # update

            $\alpha_n \leftarrow \alpha_n + y_n$;

            $b \leftarrow b + y_n$;

        end

    end

end

return $\alpha, b$

Algorithm 1: \textsc{KernelizedPerceptronTrain}
Back to linear models, for now . . .
Choosing a Hyperplane
Finding a Hyperplane with a Large Margin

The preference for a decision boundary with a **large margin** is an example of inductive bias.

\[
\begin{align*}
\gamma(w,b) = \max_{w,b} \min_n y_n \cdot (w \cdot x_n + b) \\
\text{s.t. } w \cdot x_n + b &\geq \varepsilon, \forall n : y_n = +1 \\
&\leq -\varepsilon, \forall n : y_n = -1
\end{align*}
\]

The constraints ensure that \( w \) and \( b \) form a *separating* hyperplane; the choice of \( \varepsilon > 0 \) is arbitrary.
Finding a Hyperplane with a Large Margin

The preference for a decision boundary with a **large margin** is an example of inductive bias.

\[
\max_{w, b} \gamma(w, b) \\
\text{s.t. } w \cdot x_n + b \geq \varepsilon, \forall n : y_n = +1 \\
w \cdot x_n + b \leq -\varepsilon, \forall n : y_n = -1
\]

The constraints ensure that \( w \) and \( b \) form a *separating* hyperplane; the choice of \( \varepsilon > 0 \) is arbitrary.
Finding a Hyperplane with a Large Margin

The preference for a decision boundary with a large margin is an example of inductive bias.

\[
\min_{w,b} \frac{1}{\gamma(w,b)}
\]  

s.t. \( w \cdot x_n + b \geq \varepsilon, \forall n : y_n = +1 \)  
\( w \cdot x_n + b \leq -\varepsilon, \forall n : y_n = -1 \)

The constraints ensure that \( w \) and \( b \) form a separating hyperplane; the choice of \( \varepsilon > 0 \) is arbitrary.
Finding a Hyperplane with a Large Margin

The preference for a decision boundary with a large margin is an example of inductive bias.

\[
\min_{w,b} \frac{1}{\gamma(w, b)} \\
\text{s.t. } y_n \cdot (w \cdot x_n + b) \geq \varepsilon, \forall n
\]

The constraints ensure that \( w \) and \( b \) form a separating hyperplane; the choice of \( \varepsilon > 0 \) is arbitrary.
Finding a Hyperplane with a Large Margin

The preference for a decision boundary with a large margin is an example of inductive bias.

\[
\begin{align*}
\min_{w,b} & \frac{1}{\gamma(w, b)} \\
\text{s.t. } & y_n \cdot (w \cdot x_n + b) \geq \varepsilon, \forall n
\end{align*}
\]

The constraints ensure that \(w\) and \(b\) form a separating hyperplane; the choice of \(\varepsilon > 0\) is arbitrary.

The perceptron looked for some \((w, b)\) that satisfied the constraints; now we want the \((w, b)\) that maximizes the margin!
Solving for $\gamma(w, b)$

Let $x_+$ be one training datapoint such that $w \cdot x_+ + b = \varepsilon$.
Let $x_-$ be one training datapoint such that $w \cdot x_- + b = -\varepsilon$. 
Solving for $\gamma(w, b)$

$$w \cdot x + b = 0$$

$$w \cdot x + b = -\varepsilon$$

$$w \cdot x + b = \varepsilon$$

$$\gamma(w, b) = \text{distance}(x_+, [w \cdot x + b = 0]) + \text{distance}(x_-, [w \cdot x + b = 0])$$
Solving for $\gamma(w, b)$

\[ w \cdot x + b = -\varepsilon \]
\[ w \cdot x + b = \varepsilon \]
\[ w \cdot x + b = 0 \]

\[ \gamma(w, b) = \frac{|w \cdot x_+ + b|}{\|w\|_2} + \frac{|w \cdot x_- + b|}{\|w\|_2} = \frac{2\varepsilon}{\|w\|_2} \]
"Hard Margin SVM"

\[
\begin{align*}
\min_{\mathbf{w}, b} & \quad \frac{1}{\gamma(\mathbf{w}, b)} \\
\text{s.t.} & \quad y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq \varepsilon, \forall n
\end{align*}
\]

\[
\begin{align*}
\min_{\mathbf{w}, b} & \quad \frac{1}{2\varepsilon} \left\| \mathbf{w} \right\|_2^2 \\
\text{s.t.} & \quad y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq \varepsilon, \forall n
\end{align*}
\]
Relaxing the Constraints

Feasible set:

\[ \{(w, b) : y_n \cdot (w \cdot x_n + b) \geq 1, \forall n\} \]

It’s quite plausible that the feasible set will be empty.
Relaxing the Constraints

Feasible set:

\[ \{(w, b) : y_n \cdot (w \cdot x_n + b) \geq 1, \forall n\} \]

It's quite plausible that the feasible set will be empty.

Solution: add some “slack” for every instance \( n \).

\[
\begin{align*}
\min_{w, b, \zeta} & \quad \frac{1}{2} \|w\|_2^2 + C \sum_{n=1}^{N} \zeta_n \\
\text{s.t.} & \quad y_n \cdot (w \cdot x_n + b) \geq 1 - \zeta_n, \forall n \\
& \quad \zeta_n \geq 0, \forall n
\end{align*}
\]
Slack

\[ w \cdot x + b = 0 \]
"Soft-Margin SVM"

\[
\begin{align*}
\min_{\mathbf{w}, b, \zeta} & \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \zeta_n \\
\text{s.t.} & \quad y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n \\
& \quad \zeta_n \geq 0, \forall n
\end{align*}
\]

(\(C\) is a hyperparameter.)
“Soft-Margin SVM”

\[
\min_{w,b,\zeta} \left\| w \right\|_2^2 + C \sum_{n=1}^{N} \zeta_n \\
\text{s.t.} \quad y_n \cdot (w \cdot x_n + b) \geq 1 - \zeta_n, \forall n \\
\zeta_n \geq 0, \forall n
\]

(C is a hyperparameter.)

Claim: solving this problem is equivalent to minimizing the hinge loss, with \( L_2 \) regularization. Choosing \( C \) equates to choosing \( \lambda \) (the regularization strength).
Solving for $\zeta_n$ (in terms of $w$, $b$, $x_n$, and $y_n$)

Three possibilities:

- $y_n \cdot (w \cdot x_n + b) \geq 1$: constraint is satisfied; penalty pushes $\zeta_n$ to zero
- $y_n \cdot (w \cdot x_n + b) < 1$: set $\zeta_n = 1 - y_n \cdot (w \cdot x_n + b)$ to satisfy the constraint
  - If $y_n \cdot (w \cdot x_n + b) > 0$, this is a “margin” mistake, and $\zeta_n < 1$.
  - Otherwise, this is an actual mistake, and $\zeta_n \geq 1$. 
Optimal Slack Values are Hinge Losses

From the last slide:

\[ \zeta_n = \begin{cases} 
0 & \text{if } y_n \cdot (w \cdot x_n + b) \geq 1 \\
1 - y_n \cdot (w \cdot x_n + b) & \text{otherwise} 
\end{cases} \]

Hinge loss (from A4):

\[ L_n^{(\text{hinge})}(w, b) = \max\{0, 1 - y_n \cdot (w \cdot x_n + b)\} \]
Optimal Slack Values are Hinge Losses

From the last slide:

\[ \zeta_n = \begin{cases} 
0 & \text{if } y_n \cdot (w \cdot x_n + b) \geq 1 \\
1 - y_n \cdot (w \cdot x_n + b) & \text{otherwise}
\end{cases} \]

Hinge loss (from A4):

\[ L_n^{(hinge)}(w, b) = \max\{0, 1 - y_n \cdot (w \cdot x_n + b)\} \]

Unconstrained loss minimization problem:

\[ \min_{w, b} \|w\|_2^2 + \sum_{n=1}^{N} L_n^{(hinge)}(w, b) \]
What have we learned?

▶ New motivation for $L_2$ regularization: "small norm ⇔ large margin" (among separating hyperplanes)

▶ New motivation for hinge loss: "separate data if possible, minimize slack if you can't"

▶ New insight about perceptron:

\[
L(\text{perceptron})_n(w, b) = \max\left\{0, 1 - y_n \cdot (w \cdot x_n + b)\right\}
\]

\[
L(\text{hinge})_n(w, b) = \max\left\{0, 1 - y_n \cdot (w \cdot x_n + b)\right\}
\]
What have we learned?

- New motivation for $L_2$ regularization: “small norm $\Leftrightarrow$ large margin” (among separating hyperplanes)

\[ L(\text{perceptron})_n(w, b) = \max \{0, 1 - y_n \cdot (w \cdot x_n + b)\} \]

\[ L(\text{hinge})_n(w, b) = \max \{0, 1 - y_n \cdot (w \cdot x_n + b)\} \]
What have we learned?

- New motivation for $L_2$ regularization: “small norm $\Leftrightarrow$ large margin” (among separating hyperplanes)
- New motivation for hinge loss: “separate data if possible, minimize slack if you can’t”
What have we learned?

- New motivation for $L_2$ regularization: “small norm $\Leftrightarrow$ large margin” (among separating hyperplanes)
- New motivation for hinge loss: “separate data if possible, minimize slack if you can’t”
- New insight about perceptron:

$$L_n^{(\text{perceptron})}(\mathbf{w}, b) = \max\{0, -y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\}$$

$$L_n^{(\text{hinge})}(\mathbf{w}, b) = \max\{0, 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\}$$
What have we learned?

- New motivation for $L_2$ regularization: “small norm $\iff$ large margin” (among separating hyperplanes)
- New motivation for hinge loss: “separate data if possible, minimize slack if you can’t”
- New insight about perceptron:

$$L_n^{(\text{perceptron})}(w, b) = \max \{0, -y_n \cdot (w \cdot x_n + b)\}$$

$$L_n^{(\text{hinge})}(w, b) = \max \{0, 1 - y_n \cdot (w \cdot x_n + b)\}$$

margin $= y \cdot (w \cdot x + b)$
But why are they called “support vector machines”? 
Back to the “Soft-Margin SVM”

$$\min_{w,b,\zeta} \frac{1}{2} \|w\|^2_2 + C \sum_{n=1}^{N} \zeta_n$$

s.t. $y_n \cdot (w \cdot x_n + b) \geq 1 - \zeta_n, \forall n$

$\zeta_n \geq 0, \forall n$
Back to the “Soft-Margin SVM”

\[
\min_{w,b,\zeta} \frac{1}{2} \|w\|_2^2 + C \sum_{n=1}^{N} \zeta_n \\
\text{s.t. } y_n \cdot (w \cdot x_n + b) \geq 1 - \zeta_n, \forall n \\
\zeta_n \geq 0, \forall n
\]

Lagrangian:

\[
\min \max_{w,b,\zeta} \max_{\alpha \geq 0, \beta \geq 0} \frac{1}{2} \|w\|_2^2 + C \sum_{n=1}^{N} \zeta_n - \beta_n \cdot \zeta_n - \alpha_n \cdot (y_n \cdot (w \cdot x_n + b) - 1 + \zeta_n)
\]

large margin \hspace{1cm} \text{small slack}

\[
\text{nonnegativity } \hspace{1cm} \text{separation-with-slack constraint}
\]
Back to the “Soft-Margin SVM”

\[
\begin{align*}
\min_{w,b,\zeta} & \quad \frac{1}{2} \|w\|^2_2 + C \sum_{n=1}^{N} \zeta_n \\
\text{s.t.} & \quad y_n \cdot (w \cdot x_n + b) \geq 1 - \zeta_n, \forall n \\
& \quad \zeta_n \geq 0, \forall n
\end{align*}
\]

Lagrangian:

\[
\begin{align*}
\min_{w,b,\zeta} & \quad \frac{1}{2} \|w\|^2_2 + C \sum_{n=1}^{N} \zeta_n - \beta_n \cdot \zeta_n - \alpha_n \cdot (y_n \cdot (w \cdot x_n + b) - 1 + \zeta_n) \\
\max_{\alpha \geq 0} & \quad \max_{\beta \geq 0} F(w, b, \zeta, \alpha, \beta)
\end{align*}
\]
Solve for $w$ (in terms of $\alpha, x_{1:N}, y_{1:N}$)

Gradient with respect to $w$:

$$\nabla_w F = w - \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i$$
Solve for $w$ (in terms of $\alpha, x_{1:N}, y_{1:N}$)

Gradient with respect to $w$:

$$\nabla_w F = w - \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i$$

This should immediately remind you of the kernelized perceptron, which was based on a very similar claim about the weights.
The Dual Form of Soft-Margin SVMs

After a series of mechanical steps that eliminate $b$ and $\beta$ and rearrange terms (see pp. 149–151), we get:

$$
\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (x_n \cdot x_i) - \sum_{n=1}^{N} \alpha_n
$$

s.t. $0 \leq \alpha_n \leq C, \forall n$
After a series of mechanical steps that eliminate $b$ and $\beta$ and rearrange terms (see pp. 149–151), we get:

$$\min_\alpha \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (x_n \cdot x_i) - \sum_{n=1}^{N} \alpha_n$$

s.t. $0 \leq \alpha_n \leq C$, $\forall n$

This is a **quadratic** problem with “bound” constraints.
The Dual Form of Soft-Margin SVMs

After a series of mechanical steps that eliminate $b$ and $\beta$ and rearrange terms (see pp. 149–151), we get:

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (x_n \cdot x_i) - \sum_{n=1}^{N} \alpha_n$$

s.t. $0 \leq \alpha_n \leq C$, $\forall n$

This is a quadratic problem with “bound” constraints.

Note that now it is possible to kernelize, replacing $x_n \cdot x_i$ with $K(x_n, x_i)$. 
But why are they called “support vector machines”? 