Machine Learning (CSE 446): Support Vector Machines

Sham M Kakade

University of Washington
cse446-staff@cs.washington.edu

Quick Review: Kernels and Kernelized Perceptron

Kernels

A **kernel** function (implicitly) computes:

$$K(\mathbf{x}, \mathbf{v}) = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{v})$$

for some ϕ. Typically it is *cheap* to compute $K(\cdot, \cdot)$, and we never explicitly represent $\boldsymbol{\phi}(\mathbf{v})$ for any vector $\mathbf{v}$.

Some kernels:

$$\begin{array}{rl}
\text{linear} & K^{\text{linear}}(\mathbf{x}, \mathbf{v}) = \mathbf{x} \cdot \mathbf{v} \\
\text{quadratic} & K^{\text{quad}}(\mathbf{x}, \mathbf{v}) = (1 + \mathbf{x} \cdot \mathbf{v})^2 \\
\text{cubic} & K^{\text{cubic}}(\mathbf{x}, \mathbf{v}) = (1 + \mathbf{x} \cdot \mathbf{v})^3 \\
\text{polynomial} & K_p^{\text{poly}}(\mathbf{x}, \mathbf{v}) = (1 + \mathbf{x} \cdot \mathbf{v})^p \\
\text{radial basis function} & K_\gamma^{\text{rbf}}(\mathbf{x}, \mathbf{v}) = \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{v}\right\|_2^2\right) \\
\text{hyperbolic tangent} & \tilde{K}^{\text{tanh}}(\mathbf{x}, \mathbf{v}) = \tanh(1 + \mathbf{x} \cdot \mathbf{v}) \quad \text{(not a kernel)} \\
\text{all conjunctions} & K^{\text{all conj}}(\mathbf{x}, \mathbf{v}) = \prod_{j=1}^{d}(1 + x_j v_j) \quad \text{(for binary features)}
\end{array}$$

Perceptron Representer Theorem

At every stage of learning, there exist $\langle \alpha_1, \alpha_2, \ldots, \alpha_N \rangle$ such that

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n \cdot \mathbf{x}_n = \boldsymbol{\alpha}^\top \mathbf{X}$$

In other words, $\mathbf{w}$ is always in the span of the training data.

$\phi(\mathbf{x}_n)$ is Never Explicitly Computed!

$$\text{predict:} \quad \hat{y} = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot K(\mathbf{x}_i, \mathbf{x}_n) + b\right)$$

$$\text{update:} \quad \alpha_n^{(\text{new})} \leftarrow \alpha_n^{(\text{old})} + y_n$$

We only calculate inner products of such vectors.

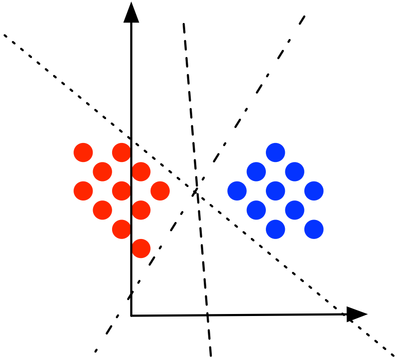
Kernelized Perceptron Learning Algorithm

Data: $D = \langle(\mathbf{x}_n, y_n)\rangle_{n=1}^{N}$, number of epochs E
Result: weights $\boldsymbol{\alpha}$ and bias b
initialize: $\boldsymbol{\alpha} = \mathbf{0}$ and $b = 0$;
for $e \in \{1, \ldots, E\}$ **do**
    for $n \in \{1, \ldots, N\}$, *in random order* **do**
        $\#$ predict
        $\hat{y} = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot K(\mathbf{x}_i, \mathbf{x}_n) + b\right)$;
        if $\hat{y} \neq y_n$ **then**
            $\#$ update
            $\alpha_n \leftarrow \alpha_n + y_n$;
            $b \leftarrow b + y_n$;
        end
    end
end
return $\boldsymbol{\alpha}, b$

Algorithm 1: 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.

$$\max_{\mathbf{w},b} \overbrace{\min_n y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)}^{\gamma(\mathbf{w},b)}$$

$$\begin{aligned} \text{s.t. } & \mathbf{w} \cdot \mathbf{x}_n + b \geq \varepsilon, \forall n : y_n = +1 \\ & \mathbf{w} \cdot \mathbf{x}_n + b \leq -\varepsilon, \forall n : y_n = -1 \end{aligned}$$

The constraints ensure that $\mathbf{w}$ 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.

$$
\begin{aligned}
&\max_{\mathbf{w},b} \gamma(\mathbf{w}, b) \\
\text{s.t. } \; & \mathbf{w} \cdot \mathbf{x}_n + b \geq \varepsilon, \forall n : y_n = +1 \\
& \mathbf{w} \cdot \mathbf{x}_n + b \leq -\varepsilon, \forall n : y_n = -1
\end{aligned}
$$

The constraints ensure that $\mathbf{w}$ 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.

$$\begin{aligned}
&\min_{\mathbf{w},b} \frac{1}{\gamma(\mathbf{w},b)} \\
\text{s.t. } &\mathbf{w}\cdot\mathbf{x}_n + b \geq \varepsilon, \forall n : y_n = +1 \\
&\mathbf{w}\cdot\mathbf{x}_n + b \leq -\varepsilon, \forall n : y_n = -1
\end{aligned}$$

The constraints ensure that $\mathbf{w}$ 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.

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

The constraints ensure that $\mathbf{w}$ and b form a *separating* hyperplane; the choice of $\varepsilon > 0$ is arbitrary.

Finding a Hyperplane with a Large Margin

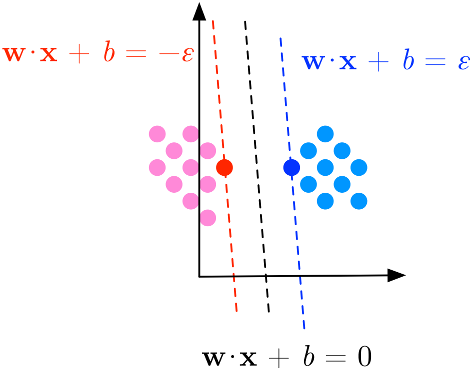
The preference for a decision boundary with a **large margin** is an example of inductive bias.

$$\min_{\mathbf{w},b} \frac{1}{\gamma(\mathbf{w},b)}$$
$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq \varepsilon, \forall n$$

The constraints ensure that $\mathbf{w}$ and b form a *separating* hyperplane; the choice of $\varepsilon > 0$ is arbitrary.

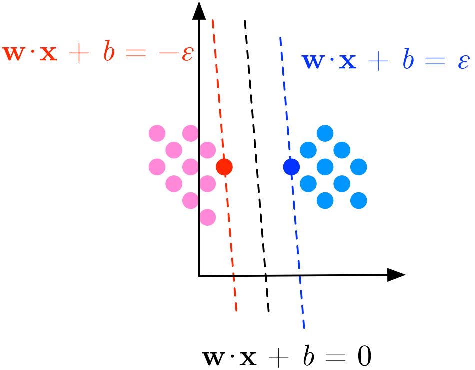
The perceptron looked for *some* $(\mathbf{w}, b)$ that satisfied the constraints; now we want the $(\mathbf{w}, b)$ that maximizes the margin!

Solving for $\gamma(\mathbf{w}, b)$



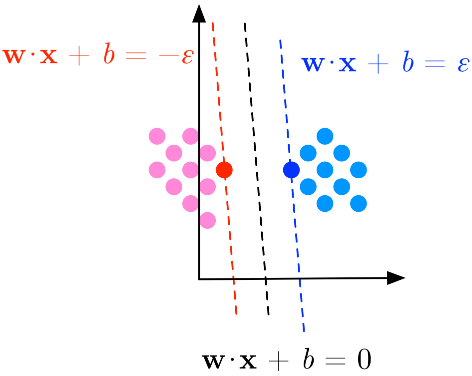
Let $\mathbf{x}_+$ be one training datapoint such that $\mathbf{w} \cdot \mathbf{x}_+ + b = \varepsilon$.
Let $\mathbf{x}_-$ be one training datapoint such that $\mathbf{w} \cdot \mathbf{x}_- + b = -\varepsilon$.

Solving for $\gamma(\mathbf{w}, b)$



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

Solving for $\gamma(\mathbf{w}, b)$



$$\gamma(\mathbf{w}, b) = \frac{|\mathbf{w}\cdot\mathbf{x}_{+} + b|}{\|\mathbf{w}\|_2} + \frac{|\mathbf{w}\cdot\mathbf{x}_{-} + b|}{\|\mathbf{w}\|_2} = \frac{2\varepsilon}{\|\mathbf{w}\|_2}$$

"Hard Margin SVM"

$$\min_{\mathbf{w},b} \frac{1}{\gamma(\mathbf{w},b)}$$
$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq \varepsilon, \forall n$$

$$\min_{\mathbf{w},b} \frac{1}{2\varepsilon} \|\mathbf{w}\|_2^2$$
$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq \varepsilon, \forall n$$

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|_2^2$$
$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1, \forall n$$

Relaxing the Constraints

Feasible set:

$$\{(\mathbf{w}, b) : y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1, \forall n\}$$

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

Relaxing the Constraints

Feasible set:

$$\{(\mathbf{w}, b) : y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1, \forall n\}$$

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

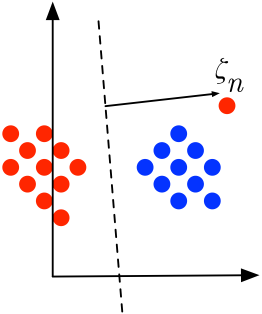
Solution: add some "slack" for every instance n.

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

$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n$$

$$\zeta_n \geq 0, \forall n$$

Slack



$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

"Soft-Margin SVM"

$$
\min_{\mathbf{w},b,\boldsymbol{\zeta}} \quad \overbrace{\|\mathbf{w}\|_2^2}^{\text{large margin}} \quad + C \overbrace{\sum_{n=1}^{N} \zeta_n}^{\text{small slack}}
$$

$$
\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n
$$

$$
\zeta_n \geq 0, \forall n
$$

(C is a hyperparameter.)

"Soft-Margin SVM"

$$
\begin{aligned}
\min_{\mathbf{w},b,\boldsymbol{\zeta}} \quad & \overbrace{\|\mathbf{w}\|_2^2}^{\text{large margin}} + C\overbrace{\sum_{n=1}^{N} \zeta_n}^{\text{small slack}} \\
\text{s.t.} \quad & y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n \\
& \zeta_n \geq 0, \forall n
\end{aligned}
$$

(C is a hyperparameter.)

Claim: solving this problem is equivalent to minimizing the hinge loss, with L_2 regularization. Choosing C equates to choosing λ (the regularization strength).

Solving for ζ_n (in terms of $\mathbf{w}$, b, $\mathbf{x}_n$, and y_n)

Three possibilities:

- $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1$: constraint is satisfied; penalty pushes ζ_n to zero
- $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) < 1$: set $\zeta_n = 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)$ to satisfy the constraint
  - If $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) > 0$, 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:

$$\zeta_n = \begin{cases} 0 & \text{if } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 \\ 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) & \text{otherwise} \end{cases}$$

Hinge loss (from A4):

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

Optimal Slack Values are Hinge Losses

From the last slide:

$$\zeta_n = \begin{cases} 0 & \text{if } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 \\ 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) & \text{otherwise} \end{cases}$$

Hinge loss (from A4):

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

Unconstrained loss minimization problem:

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_2^2 + \sum_{n=1}^{N} L_n^{(\text{hinge})}(\mathbf{w}, b)$$

What have we learned?

What have we learned?

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

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?

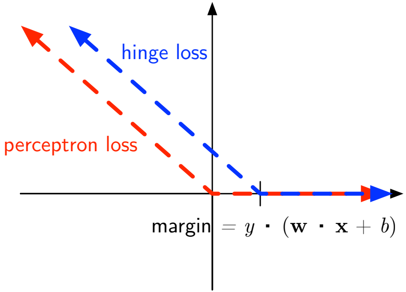
- New motivation for L_2 regularization: "small norm $\Leftrightarrow$ large margin" (among separating hyperplanes)
- New motivation for hinge loss: "separate data if possible, minimize slack if you can't"
- New insight about perceptron:

$$
\begin{aligned}
L_n^{(\text{perceptron})}(\mathbf{w}, b) &= \max\{0, \quad - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\} \\
L_n^{(\text{hinge})}(\mathbf{w}, b) &= \max\{0, 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\}
\end{aligned}
$$

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:

$$L_n^{(\text{perceptron})}(\mathbf{w}, b) = \max\{0, \quad - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\}$$
$$L_n^{(\text{hinge})}(\mathbf{w}, b) = \max\{0, 1 - y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\}$$



But why are they called "support vector machines"?

Back to the "Soft-Margin SVM"

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \overbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}^{\text{large margin}} + C \overbrace{\sum_{n=1}^{N} \zeta_n}^{\text{small slack}}$$

$$\text{s.t. } y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n$$

$$\zeta_n \geq 0, \forall n$$

Back to the "Soft-Margin SVM"

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \overbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}^{\text{large margin}} + \overbrace{C\sum_{n=1}^{N}\zeta_n}^{\text{small slack}}$$

$$\text{s.t. } y_n \cdot (\mathbf{w}\cdot\mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n$$

$$\zeta_n \geq 0, \forall n$$

Lagrangian:

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \max_{\boldsymbol{\alpha}\geq\mathbf{0}} \max_{\boldsymbol{\beta}\geq\mathbf{0}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\zeta_n - \beta_n \cdot \overbrace{\zeta_n}^{\text{nonnegativity}} - \alpha_n \cdot \overbrace{(y_n \cdot (\mathbf{w}\cdot\mathbf{x}_n + b) - 1 + \zeta_n)}^{\text{separation-with-slack constraint}}$$

Back to the "Soft-Margin SVM"

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \quad \overbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}^{\text{large margin}} + C \overbrace{\sum_{n=1}^{N} \zeta_n}^{\text{small slack}}$$

$$\text{s.t.} \quad y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 - \zeta_n, \forall n$$

$$\zeta_n \geq 0, \forall n$$

Lagrangian:

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \max_{\boldsymbol{\alpha}\geq\mathbf{0}} \max_{\boldsymbol{\beta}\geq\mathbf{0}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N} \zeta_n - \beta_n \cdot \overbrace{\zeta_n}^{\text{nonnegativity}} - \alpha_n \cdot \overbrace{(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b) - 1 + \zeta_n)}^{\text{separation-with-slack constraint}}$$

$$\min_{\mathbf{w},b,\boldsymbol{\zeta}} \max_{\boldsymbol{\alpha}\geq\mathbf{0}} \max_{\boldsymbol{\beta}\geq\mathbf{0}} F(\mathbf{w}, b, \boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Solve for $\mathbf{w}$ (in terms of $\boldsymbol{\alpha}, \mathbf{x}_{1:N}, y_{1:N}$)

Gradient with respect to $\mathbf{w}$:

$$\nabla_{\mathbf{w}} F = \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot \mathbf{x}_i \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot \mathbf{x}_i$$

Solve for $\mathbf{w}$ (in terms of $\boldsymbol{\alpha}, \mathbf{x}_{1:N}, y_{1:N}$)

Gradient with respect to $\mathbf{w}$:

$$\nabla_{\mathbf{w}} F = \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot \mathbf{x}_i \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot \mathbf{x}_i$$

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 $\boldsymbol{\beta}$ and rearrange terms (see pp. 149–151), we get:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (\mathbf{x}_n \cdot \mathbf{x}_i) - \sum_{n=1}^{N} \alpha_n$$

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

The Dual Form of Soft-Margin SVMs

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

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (\mathbf{x}_n \cdot \mathbf{x}_i) - \sum_{n=1}^{N} \alpha_n$$

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

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 β and rearrange terms (see pp. 149–151), we get:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{N} \alpha_n \cdot \alpha_i \cdot y_n \cdot y_i \cdot (\mathbf{x}_n \cdot \mathbf{x}_i) - \sum_{n=1}^{N} \alpha_n$$

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

This is a **quadratic** problem with “bound” constraints.

Note that now it is possible to kernelize, replacing $\mathbf{x}_n \cdot \mathbf{x}_i$ with $K(\mathbf{x}_n, \mathbf{x}_i)$.

But why are they called "support vector machines"?