Lecture 9: Large Margin Learning

TTIC 31020: Introduction to Machine Learning

Instructor: Kevin Gimpel

TTI-Chicago

October 29, 2019

Review: Perceptron algorithm

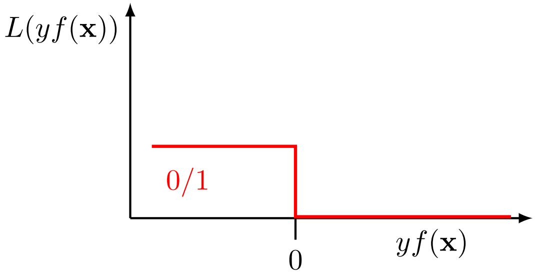
- Binary classification task: $\mathcal{Y} = \{\pm 1\}$
- Linear classifier: $h(\mathbf{x}) = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} + b)$
- Algorithm:
  initialize $\mathbf{w}^{(0)} = \mathbf{0}$, $b^{(0)} = 0$
  take one example $(\mathbf{x}_i, y_i)$ at a time
  if $y_i \left(\mathbf{w}^{(t)} \cdot \mathbf{x}_i + b^{(t)}\right) \leq 0$ (i.e., classifier was incorrect), update:

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} + y_i \mathbf{x}_i, \quad b^{(t+1)} := b^{(t)} + y_i$$

  otherwise (i.e., classifier was correct), do nothing
  stop when all data are classified correctly

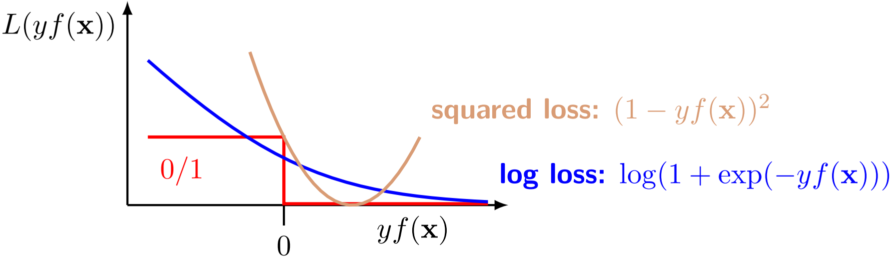
Loss functions for binary classification

- Recall that we really want to minimize 0/1 loss
- In plot below,
  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$
  $\mathcal{Y} = \{\pm 1\}$
  y is true class label
  L is "loss"



Loss functions for binary classification

- Linear regression for classification minimizes **squared loss**
- Logistic regression minimizes **log loss**
- In plot below, $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, $\mathcal{Y} = \{\pm 1\}$, y is true class label

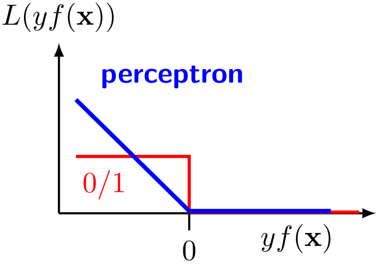


Perceptron loss

- A mistake driven algorithm: updates weights only when making a mistake on an example
- What loss does this minimize?

$$\text{loss} = \begin{cases} 0 & \text{if } yf(\mathbf{x}) > 0 \\ -yf(\mathbf{x}) & \text{if } yf(\mathbf{x}) \leq 0 \end{cases}$$

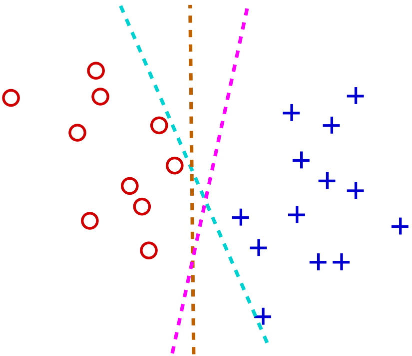
$$= \max(0, -yf(\mathbf{x}))$$



- "Perceptron" loss
- Continuous but non-smooth
- Perceptron performs descent on this loss
- **Subgradient** descent

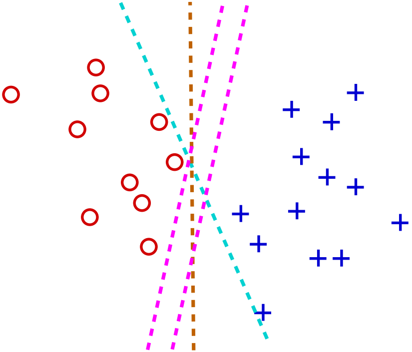
Optimal linear classifier

- Which decision boundary is better?



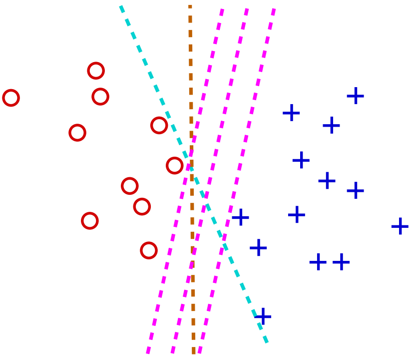
Optimal linear classifier

- Which decision boundary is better?



Optimal linear classifier

- Which decision boundary is better?



- We will want to capture this intuition when learning linear classifiers

Linear classifiers

$$\widehat{y} = h(\mathbf{x}) \; = \; \mathrm{sign}\,(b + \mathbf{w} \cdot \mathbf{x})$$

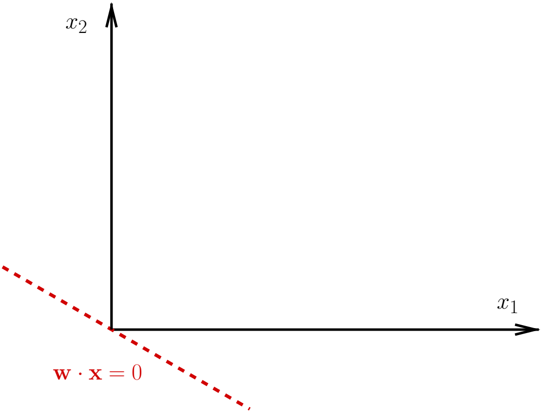
- Classifying using a linear decision boundary effectively reduces the data dimension to 1
- Need to find $\mathbf{w}$ (direction) and b (location) of the boundary

Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$

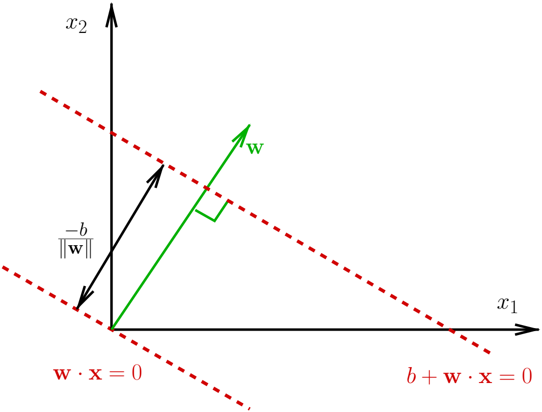
Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



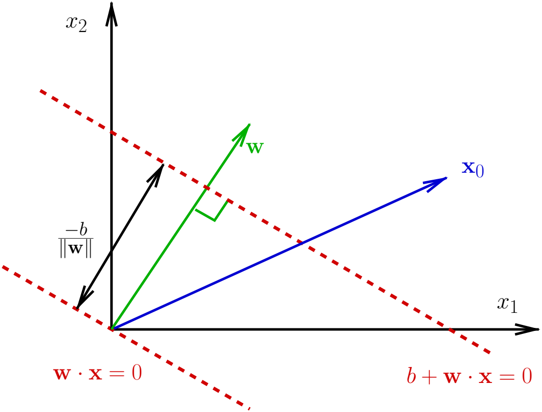
Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



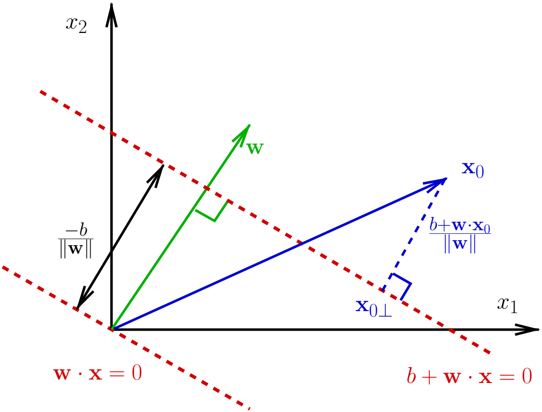
Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



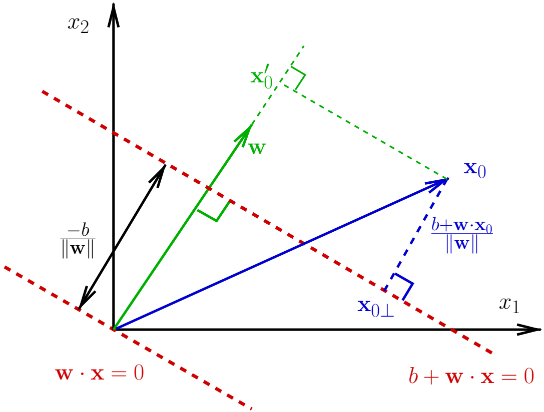
Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



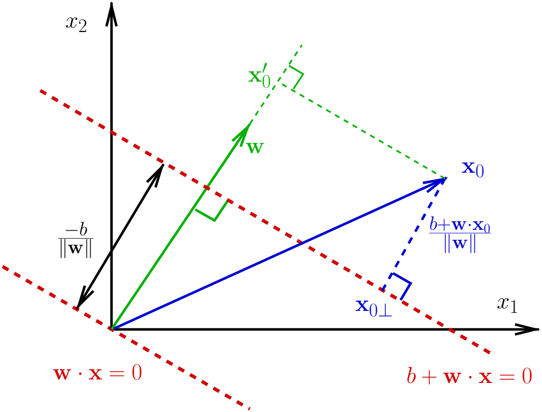
Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



Geometry of projections

- $\mathbf{w} \cdot \mathbf{x} = 0$: a line passing through the origin and **orthogonal** to $\mathbf{w}$
- $\mathbf{w} \cdot \mathbf{x} + b = 0$ shifts the line along $\mathbf{w}$



- Set up a new 1D coordinate system defined by projection of $\mathbf{x}$ onto the vector $\mathbf{w}$:
  $\mathbf{x} \ \rightarrow \ (b + \mathbf{w} \cdot \mathbf{x})/\|\mathbf{w}\|$
  (also see projections Jupyter notebook from last week)

Large margin classifier

- Distance from a *correctly* classified $(\mathbf{x}, y)$ to the boundary:

$$\frac{1}{\|\mathbf{w}\|} y \left(\mathbf{w} \cdot \mathbf{x} + b\right)$$

- Margin of the classifier on $X = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, assuming it achieves 100% accuracy: the distance to the closest point:

$$\min_i \frac{1}{\|\mathbf{w}\|} y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right)$$

- We are interested in a large margin classifier:

$$\underset{\mathbf{w}, b}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \right\}$$

y is 1 or -1

Optimal separating hyperplane

- So, we seek $\text{argmax}_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \right\}$
- We can set the margin to 1:

$$\min_i \, y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \, = \, 1$$

since we can rescale $\|\mathbf{w}\|$ and b appropriately

- Then, the optimization becomes:

$$\underset{\mathbf{w},b}{\text{argmax}} \qquad \frac{1}{\|\mathbf{w}\|} \qquad \text{s.t. } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \, \geq \, 1, \; \forall i = 1, \ldots, n$$

Optimal separating hyperplane

- So, we seek $\operatorname{argmax}_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \right\}$
- We can set the margin to 1:

$$\min_i \; y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \,=\, 1$$

since we can rescale $\|\mathbf{w}\|$ and b appropriately

- Then, the optimization becomes:

$$\begin{array}{lll} \underset{\mathbf{w},b}{\operatorname{argmax}} & \dfrac{1}{\|\mathbf{w}\|} & \text{s.t. } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \,\geq\, 1,\ \forall i = 1, \ldots, n \\ \Rightarrow \underset{\mathbf{w},b}{\operatorname{argmin}} & \|\mathbf{w}\|^2 & \qquad\text{\_\_\_\_\_\_\_\_ " " \_\_\_\_\_\_\_\_} \end{array}$$

Representer theorem

- Consider the optimization problem

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

- Theorem: the solution can be represented as

$$\mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

Representer theorem

- Consider the optimization problem

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \ \forall i$$

- Theorem: the solution can be represented as

$$\mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Note: obvious when $\mathbf{x} \in \mathbb{R}^d$ for $d < n$

Representer theorem

- Consider the optimization problem

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \ \forall i$$

- Theorem: the solution can be represented as

$$\mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Note: obvious when $\mathbf{x} \in \mathbb{R}^d$ for $d < n$
- Recall: this was the form of the perceptron boundary!
  what about logistic regression trained with [S]GD?

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^n \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \, \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^n \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, i.e., $\mathbf{w}_\perp \cdot \mathbf{x}_i = 0$ for all $i = 1, \ldots, n$

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^n \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, i.e., $\mathbf{w}_\perp \cdot \mathbf{x}_i = 0$ for all $i = 1, \ldots, n$
- For all $\mathbf{x}_i$ we have

$$\mathbf{w}^* \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i + \mathbf{w}_\perp \cdot \mathbf{x}_i =$$

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^{n} \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, i.e., $\mathbf{w}_\perp \cdot \mathbf{x}_i = 0$ for all $i = 1, \ldots, n$
- For all $\mathbf{x}_i$ we have

$$\mathbf{w}^* \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i + \mathbf{w}_\perp \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i$$

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^n \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, i.e., $\mathbf{w}_\perp \cdot \mathbf{x}_i = 0$ for all $i = 1, \ldots, n$
- For all $\mathbf{x}_i$ we have

$$\mathbf{w}^* \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i + \mathbf{w}_\perp \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i$$

therefore,

$$y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b) \geq 1 \quad \Rightarrow \quad y_i(\mathbf{w}_X \cdot \mathbf{x}_i + b) \geq 1$$

Representer theorem - proof II

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^*$$

Representer theorem - proof II

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^* = (\mathbf{w}_X + \mathbf{w}_\perp) \cdot (\mathbf{w}_X + \mathbf{w}_\perp)$$

Representer theorem - proof II

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^* = (\mathbf{w}_X + \mathbf{w}_\perp) \cdot (\mathbf{w}_X + \mathbf{w}_\perp) = \underbrace{\mathbf{w}_X \cdot \mathbf{w}_X}_{\|\mathbf{w}_X\|^2} + \underbrace{\mathbf{w}_\perp \cdot \mathbf{w}_\perp}_{\|\mathbf{w}_\perp\|^2},$$

since $\mathbf{w}_X \cdot \mathbf{w}_\perp = 0$.

Representer theorem - proof II

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^* = (\mathbf{w}_X + \mathbf{w}_\perp) \cdot (\mathbf{w}_X + \mathbf{w}_\perp) = \underbrace{\mathbf{w}_X \cdot \mathbf{w}_X}_{\|\mathbf{w}_X\|^2} + \underbrace{\mathbf{w}_\perp \cdot \mathbf{w}_\perp}_{\|\mathbf{w}_\perp\|^2},$$

  since $\mathbf{w}_X \cdot \mathbf{w}_\perp = 0$.
- Suppose $\mathbf{w}_\perp \neq \mathbf{0}$. Then, we have a solution $\mathbf{w}_X$ that satisfies all the constraints, and for which
  $\|\mathbf{w}_X\|^2 < \|\mathbf{w}_X\|^2 + \|\mathbf{w}_\perp\|^2 = \|\mathbf{w}^*\|^2$.

Representer theorem - proof II

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^* = (\mathbf{w}_X + \mathbf{w}_\perp) \cdot (\mathbf{w}_X + \mathbf{w}_\perp) = \underbrace{\mathbf{w}_X \cdot \mathbf{w}_X}_{\|\mathbf{w}_X\|^2} + \underbrace{\mathbf{w}_\perp \cdot \mathbf{w}_\perp}_{\|\mathbf{w}_\perp\|^2},$$

  since $\mathbf{w}_X \cdot \mathbf{w}_\perp = 0$.
- Suppose $\mathbf{w}_\perp \neq \mathbf{0}$. Then, we have a solution $\mathbf{w}_X$ that satisfies all the constraints, and for which $\|\mathbf{w}_X\|^2 < \|\mathbf{w}_X\|^2 + \|\mathbf{w}_\perp\|^2 = \|\mathbf{w}^*\|^2$.
- This contradicts optimality of $\mathbf{w}^*$, hence $\mathbf{w}^* = \mathbf{w}_X$. QED

Support vectors

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \ \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

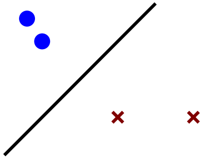
- What can we say if $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) > 1$?

Support vectors

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \ \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1$?

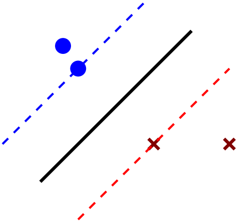
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition:



Support vectors

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1$?
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition: not change, once we found the boundary, we can throw some points.

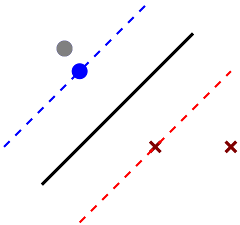


Support vectors

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \ \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1$?
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition:

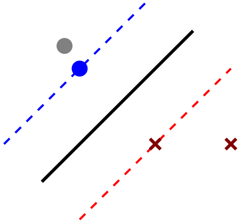
different from logistic regression



Support vectors

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) > 1$?



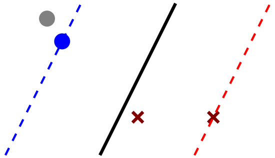
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition:

Support vectors

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1\ \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) > 1$?

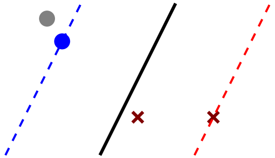
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition:



Support vectors

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \;\forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$$

- What can we say if $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) > 1$?
- Consider removing $(\mathbf{x}_i, y_i)$ from the data; how will the solution change?
- Intuition:



- Training examples with $\beta_i \neq 0$ are the **support vectors** for the decision boundary; those are the examples that determine the solution

Non-separable data: slack variables

- Not linearly separable data: we can no longer satisfy $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ for all i.
- We introduce **slack variables** to satisfy margin constraints

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0, \qquad \xi_i \geq 0$$

- We want ξ_i to capture the *minimum* amount we need to fix:

get rid of the constraint

$$\xi_i = \max\{0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)\}$$

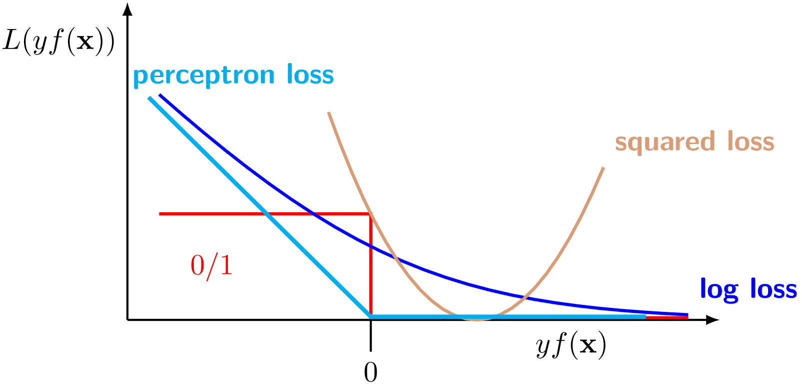
this is negative if we separate it wrong

note: ξ_i is really a function of $\mathbf{w}$, b

- Our objective now: minimize $\|\mathbf{w}\|$ with **minimum constraint violation**

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\}$$

Loss functions for binary classification



Loss in SVM

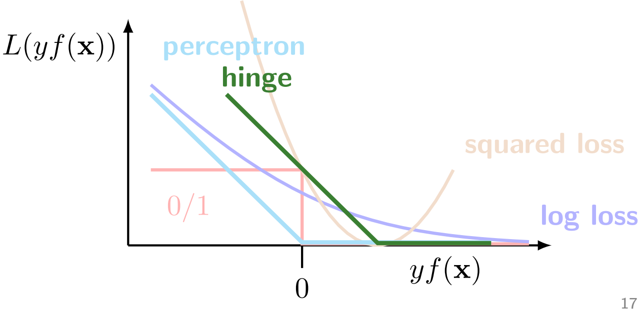
no constraint here anymore

$$\min_{\mathbf{w},b}\left\{ \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{regularizer}} + \underbrace{C\sum_{i=1}^{n}\xi_i(\mathbf{w},b)}_{\text{loss}} \right\}$$

- The loss is measured as margin constraint violation

$$\sum_{i=1}^{n}\xi_i(\mathbf{w},b)$$

- This surrogate loss is known as **hinge loss**



Loss in SVM

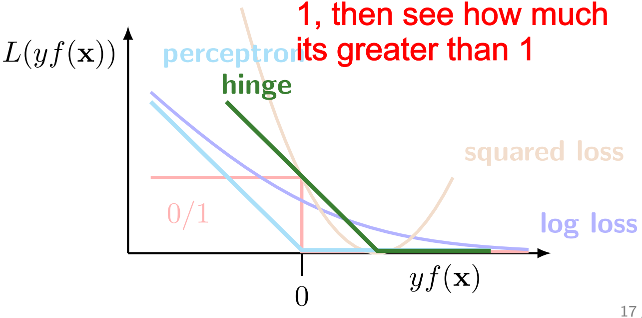
$$\min_{\mathbf{w},b}\left\{\underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{regularizer}} + \underbrace{C\sum_{i=1}^{n}\xi_i(\mathbf{w},b)}_{\text{loss}}\right\}$$

- The loss is measured as margin constraint violation

$$\sum_{i=1}^{n}\xi_i(\mathbf{w},b) = \sum_{i=1}^{n}\max\{0, 1-y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\}$$

no longer greater than 1, then see how much its greater than 1

- This surrogate loss is known as **hinge loss**

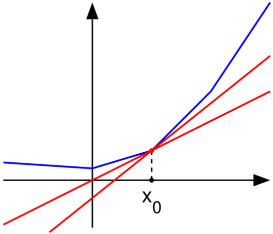


SVM via gradient descent

- With the notation $[\cdot]_+ = \max\{0, \cdot\}$, setting $\lambda = 1/C$:

$$\text{primal:} \qquad \min_{\mathbf{w},b} \left\{ \frac{\lambda}{2}\|\mathbf{w}\|^2 \; + \; \sum_{i=1}^{n} \left[1 - y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right)\right]_+ \right\}$$

- Traditional tactic (next time): write the **dual**, solve using QP
- Alternative: optimize regularized ERM directly, via gradient descent
- Problem: hinge loss is not differentiable at $y(\mathbf{w} \cdot \mathbf{x} + b) = 1$
- Solution: *sub*gradient descent
- Subgradient of convex function [Wikipedia]:



Review: subgradient $\mathbf{g}$

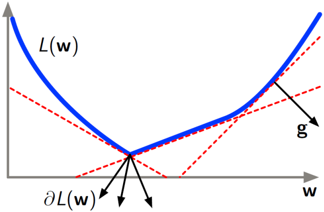


Figure: A. Vedaldi

- Subgradient of L at $\mathbf{w}$ is any $\mathbf{g}$ s.t.

  $$\forall \mathbf{w}' : \quad L(\mathbf{w}') \geq L(\mathbf{w}) + \mathbf{g} \cdot (\mathbf{w}' - \mathbf{w})$$

  i.e., $\mathbf{g}$ defines a tight linear lower bound on L at $\mathbf{w}$

- Subdifferential of L at $\mathbf{w}$:
  $\partial L(\mathbf{w}) = \{\mathbf{g} : \mathbf{g}$ is a subgradient of L at $\mathbf{w}\}$
- If L is differentiable at $\mathbf{w}$ then $\partial L(\mathbf{w}) = \{\nabla L(\mathbf{w})\}$

SVM via subgradient descent

primal: $$\min_{\mathbf{w},b}\Big\{\frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{n}\underbrace{\max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\}}_{L_i(\mathbf{w},b)}\Big\}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) \;=\; \begin{cases} \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) < 1 : \\ \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 : \end{cases}$$

SVM via subgradient descent

primal: $$\min_{\mathbf{w},b}\Big\{\frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{n}\underbrace{\max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\}}_{L_i(\mathbf{w},b)}\Big\}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) \;=\; \begin{cases} \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) < 1 : & -\, y_i\mathbf{x}_i \\ \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 : & \end{cases}$$

SVM via subgradient descent

primal: $$\min_{\mathbf{w},b}\left\{\frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{n}\underbrace{\max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\}}_{L_i(\mathbf{w},b)}\right\}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

like gradient here

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) \;=\; \begin{cases} \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) < 1 : & -y_i\mathbf{x}_i \\ \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 : & 0 \end{cases}$$

SVM via subgradient descent

primal: $$\min_{\mathbf{w},b}\Big\{\frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{n}\underbrace{\max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\}}_{L_i(\mathbf{w},b)}\Big\}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) \;=\; \begin{cases} \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) < 1: & -\,y_i\mathbf{x}_i \\ \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1: & 0 \end{cases}$$

- Bias term b updated similarly

SVM via subgradient descent

primal: $$\min_{\mathbf{w},b}\Big\{\frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{n}\underbrace{\max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\}}_{L_i(\mathbf{w},b)}\Big\}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, b) \;=\; \begin{cases} \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) < 1: & -\,y_i\mathbf{x}_i \\ \text{if } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1: & 0 \end{cases}$$

- Bias term b updated similarly
- Remember to add gradient of the regularizer!
- If current $\mathbf{w}$ classifies $(\mathbf{x}_i, y_i)$ correctly with large enough margin, i.e., $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$, that example contributes nothing to update; does it resemble another algorithm we have seen?

Perceptron vs. SVM

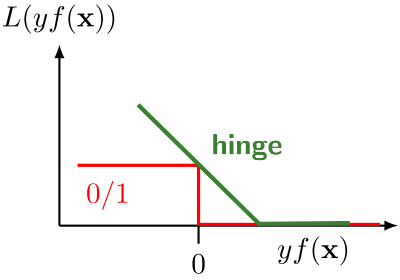
- Update in SVM (ignoring bias and regularizer):

$$\mathbf{w} \;:=\; \mathbf{w} + \eta \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 1 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 \end{cases}$$

- Update in perceptron (ignoring bias, no regularizer):

$$\mathbf{w} := \mathbf{w} + \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 0 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 0 \end{cases}$$

- What are the differences?



Perceptron vs. SVM

- Update in SVM (ignoring bias and regularizer):

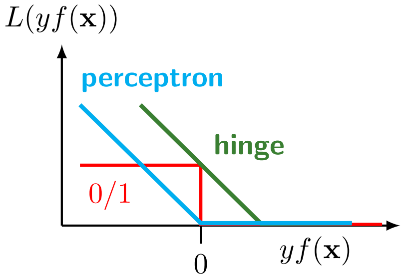
$$\mathbf{w} := \mathbf{w} + \eta \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 1 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 \end{cases}$$

- Update in perceptron (ignoring bias, no regularizer):

is doing subgradient on similar loss

$$\mathbf{w} := \mathbf{w} + \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 0 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 0 \end{cases}$$

- What are the differences?



Perceptron vs. SVM

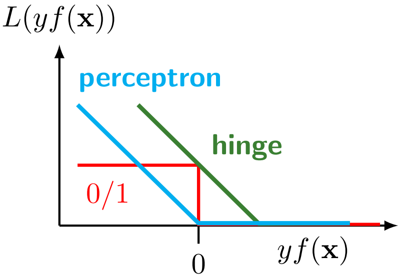
- Update in SVM (ignoring bias and regularizer):

$$\mathbf{w} := \mathbf{w} + \eta \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 1 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 \end{cases}$$

- Update in perceptron (ignoring bias, no regularizer):

$$\mathbf{w} := \mathbf{w} + \begin{cases} y_i \mathbf{x}_i & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) < 0 \\ 0 & \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 0 \end{cases}$$

- What are the differences?
- Margin size
- Learning rate
- Regularization



Maximum margin decision boundary

- Can refine the representer theorem form for the optimal $\mathbf{w}$

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \qquad \alpha_i \geq 0$$

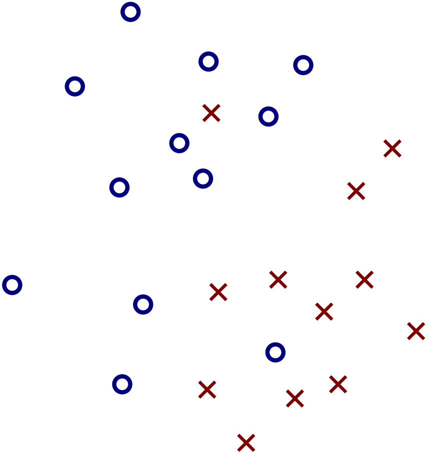
(why? consider the updates in gradient descent) last slide

- Support vectors: $(\mathbf{x}_i, y_i)$ with $\alpha_i > 0$, so

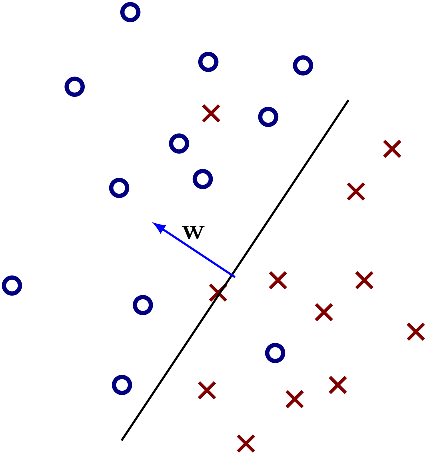
$$\mathbf{w}^* = \sum_{i:\alpha_i > 0} \alpha_i y_i \mathbf{x}_i$$

- b is set by making the margin equidistant to two classes.
- We can compute $\mathbf{w}, b$ and discard the SVs

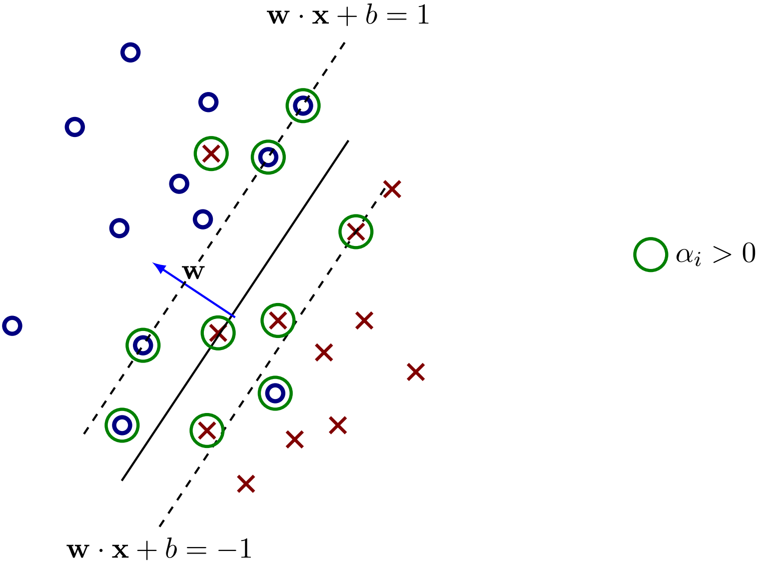
SVM geometry



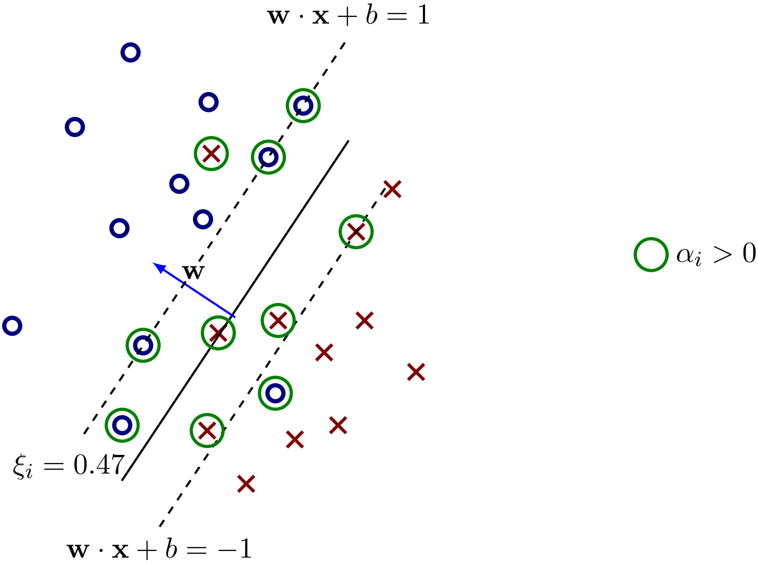
SVM geometry



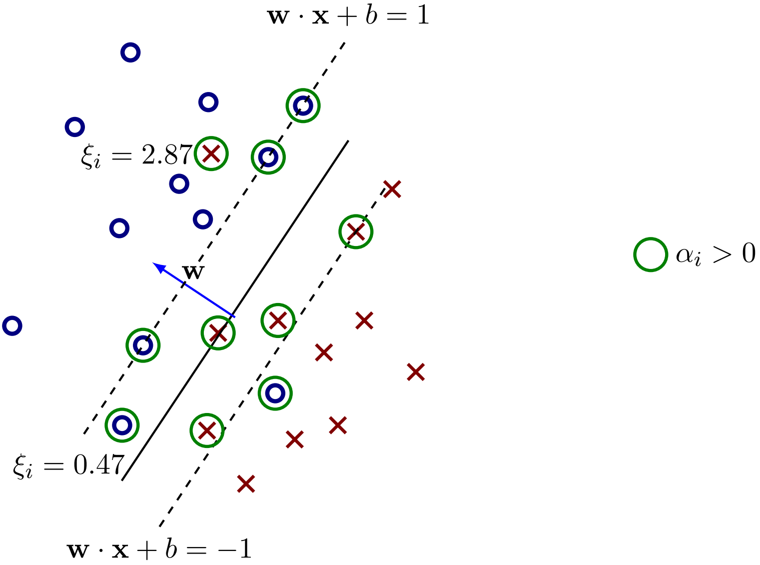
SVM geometry



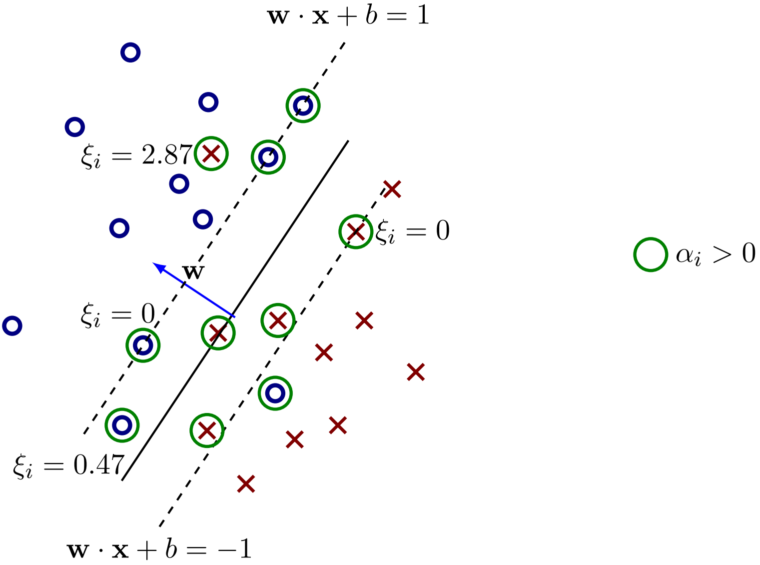
SVM geometry



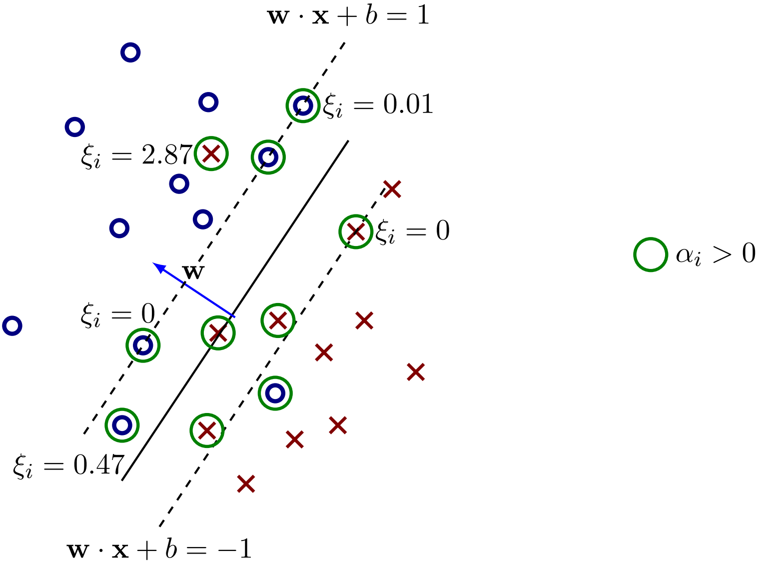
SVM geometry



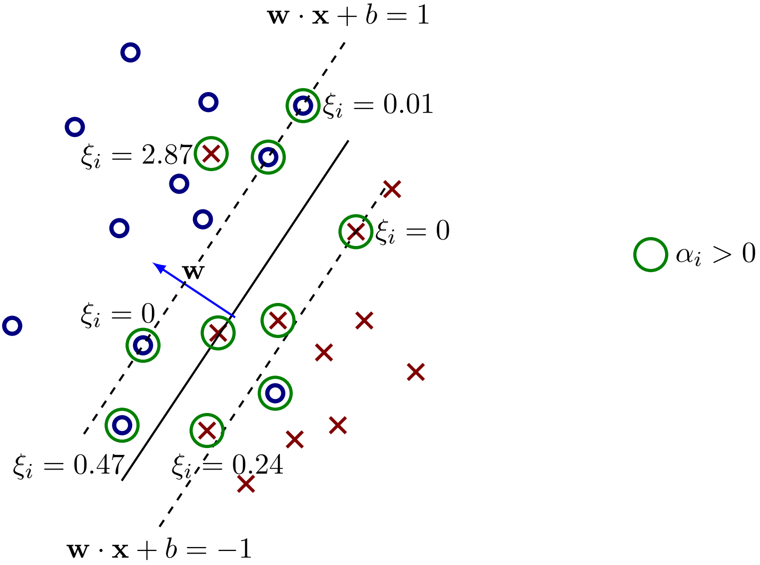
SVM geometry



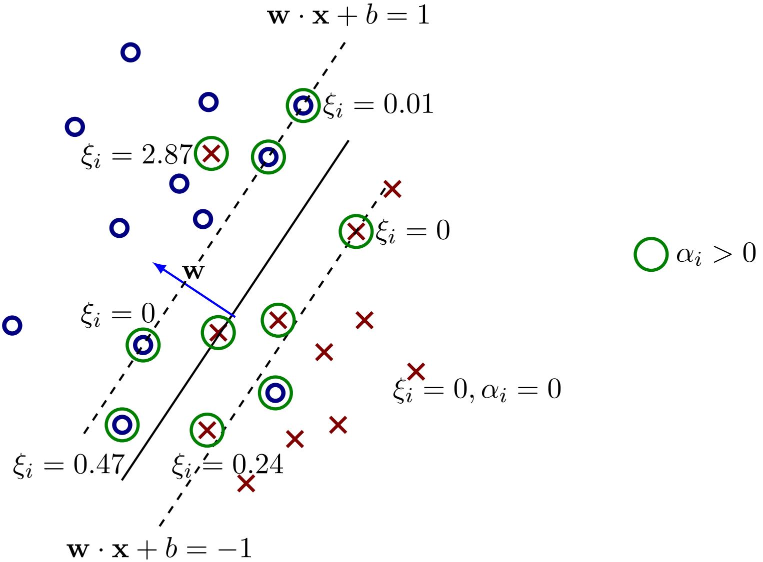
SVM geometry



SVM geometry



SVM geometry



Closer look at support vectors

$$\mathbf{w} \;=\; \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i.$$

- Given a test example $\mathbf{x}$, it is classified by

$$\widehat{y} \;=\; \operatorname{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right)$$

Closer look at support vectors

$$\mathbf{w} \;=\; \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i.$$

- Given a test example $\mathbf{x}$, it is classified by

$$\begin{aligned}\widehat{y} \;&=\; \operatorname{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right) \\ &=\; \operatorname{sign}\left((\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i) \cdot \mathbf{x} + b \right)\end{aligned}$$

Closer look at support vectors

$$\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i.$$

- Given a test example $\mathbf{x}$, it is classified by

$$\begin{aligned}
\widehat{y} &= \operatorname{sign}(\mathbf{w} \cdot \mathbf{x} + b) \\
&= \operatorname{sign}\left(\left(\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i\right) \cdot \mathbf{x} + b \right) \\
&= \operatorname{sign}\left(\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b \right)
\end{aligned}$$

- The classifier is based on the expansion in terms of dot products of $\mathbf{x}$ with support vectors.

Dot product and SVMs

- SVMs rely on dot product

$$\widehat{y} = \operatorname{sign}\left(\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

- Intuition: dot product measures similarity between examples
- A vector $\mathbf{x}$ corresponds to direction $\mathbf{x}/\|\mathbf{x}\|$ and magnitude $\|\mathbf{x}\|$; direction is determined by *relative* strength of "loading" of features (dimensions of $\mathbf{x}$)
- Common interpretation: direction captures meaning
- Normalization often used with SVMs: scale each training example to unit norm before training

$$\mathbf{x}_i \rightarrow \mathbf{x}_i/\|\mathbf{x}_i\|$$

make sure to scale test examples too!