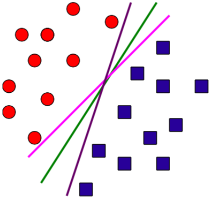
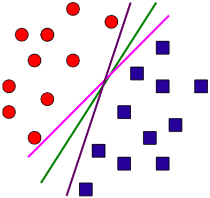
The Best Hyperplane Separator?

- Perceptron finds one of the many possible hyperplanes separating the data
  - .. if one exists
- Of the many possible choices, which one is the best?



The Best Hyperplane Separator?

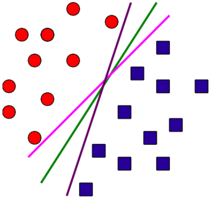
- Perceptron finds one of the many possible hyperplanes separating the data
  - .. if one exists
- Of the many possible choices, which one is the best?



- Intuitively, we want the hyperplane having the maximum margin

The Best Hyperplane Separator?

- Perceptron finds one of the many possible hyperplanes separating the data
  - .. if one exists
- Of the many possible choices, which one is the best?



- Intuitively, we want the hyperplane having the maximum margin
- Large margin leads to good generalization on the test data
  - We will see this formally when we cover Learning Theory

Support Vector Machine (SVM)

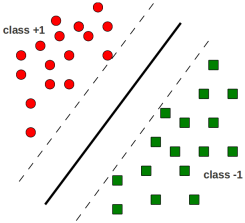
- Probably the most popular/influential classification algorithm
- Backed by solid theoretical groundings (Vapnik and Cortes, 1995)

Support Vector Machine (SVM)

- Probably the most popular/influential classification algorithm
- Backed by solid theoretical groundings (Vapnik and Cortes, 1995)
- A hyperplane based classifier (like the Perceptron)

Support Vector Machine (SVM)

- Probably the most popular/influential classification algorithm
- Backed by solid theoretical groundings (Vapnik and Cortes, 1995)
- A hyperplane based classifier (like the Perceptron)
- *Additionally* uses the Maximum Margin Principle
  - Finds the hyperplane with *maximum separation margin* on the training data



Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b

Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$

Support Vector Machine

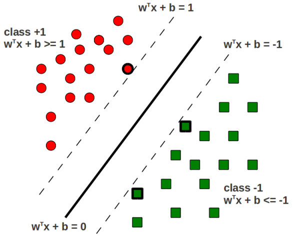
- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin

Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)

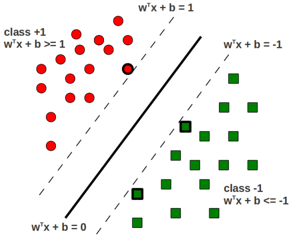
Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)



Support Vector Machine

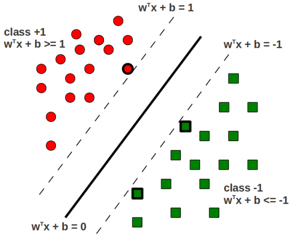
- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)



- Assume the hyperplane is such that
  - $\mathbf{w}^T\mathbf{x}_n + b \geq 1$ for $y_n = +1$

Support Vector Machine

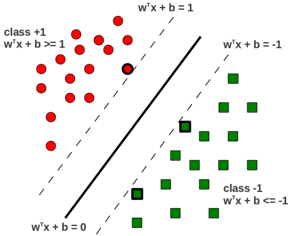
- A hyperplane based linear classifier defined by **w** and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn **w** and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)



- Assume the hyperplane is such that
  - $\mathbf{w}^T\mathbf{x}_n + b \geq 1$ for $y_n = +1$
  - $\mathbf{w}^T\mathbf{x}_n + b \leq -1$ for $y_n = -1$

Support Vector Machine

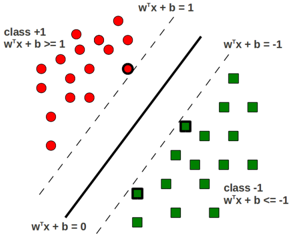
- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)



- Assume the hyperplane is such that
  - $\mathbf{w}^T\mathbf{x}_n + b \geq 1$ for $y_n = +1$
  - $\mathbf{w}^T\mathbf{x}_n + b \leq -1$ for $y_n = -1$
  - Equivalently, $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$

Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)

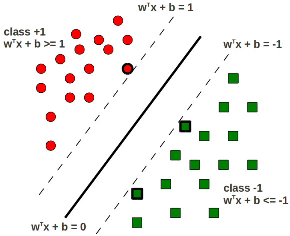


- Assume the hyperplane is such that
  - $\mathbf{w}^T\mathbf{x}_n + b \geq 1$ for $y_n = +1$
  - $\mathbf{w}^T\mathbf{x}_n + b \leq -1$ for $y_n = -1$
  - Equivalently, $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$
    $\Rightarrow \min_{1\leq n\leq N} |\mathbf{w}^T\mathbf{x}_n + b| = 1$
  - The hyperplane's margin:

$$\gamma = \min_{1\leq n\leq N} \frac{|\mathbf{w}^T\mathbf{x}_n + b|}{||\mathbf{w}||}$$

Support Vector Machine

- A hyperplane based linear classifier defined by $\mathbf{w}$ and b
- Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$
- **Given:** Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- **Goal:** Learn $\mathbf{w}$ and b that achieve the maximum margin
- For now, assume the entire training data is correctly classified by $(\mathbf{w}, b)$
  - Zero loss on the training examples (non-zero loss case later)

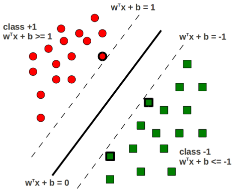


- Assume the hyperplane is such that
  - $\mathbf{w}^T\mathbf{x}_n + b \geq 1$ for $y_n = +1$
  - $\mathbf{w}^T\mathbf{x}_n + b \leq -1$ for $y_n = -1$
  - Equivalently, $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$
    $\Rightarrow \min_{1\leq n\leq N} |\mathbf{w}^T\mathbf{x}_n + b| = 1$
  - The hyperplane's margin:

$$\boxed{\gamma = \min_{1\leq n\leq N} \frac{|\mathbf{w}^T\mathbf{x}_n + b|}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}}$$

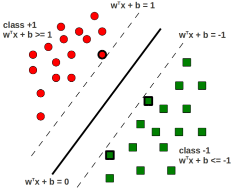
Support Vector Machine: The Optimization Problem

- We want to maximize the margin $\gamma = \frac{1}{||\mathbf{w}||}$



Support Vector Machine: The Optimization Problem

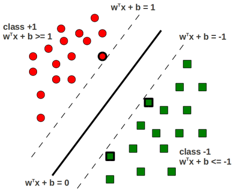
- We want to maximize the margin $\gamma = \frac{1}{||\mathbf{w}||}$



- Maximizing the margin γ = minimizing $||\mathbf{w}||$ (the norm)

Support Vector Machine: The Optimization Problem

- We want to maximize the margin $\gamma = \frac{1}{||\mathbf{w}||}$

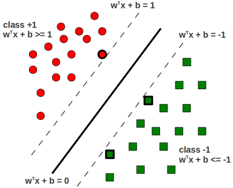


- Maximizing the margin γ = minimizing $||\mathbf{w}||$ (the norm)
- Our optimization problem would be:

$$\boxed{\begin{array}{ll} \text{Minimize} \quad f(\mathbf{w}, b) = \dfrac{||\mathbf{w}||^2}{2} & \\ \text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, & n = 1, \ldots, N \end{array}}$$

Support Vector Machine: The Optimization Problem

- We want to maximize the margin $\gamma = \frac{1}{||\mathbf{w}||}$



- Maximizing the margin γ = minimizing $||\mathbf{w}||$ (the norm)
- Our optimization problem would be:

$$\boxed{\begin{array}{ll} \text{Minimize} \quad f(\mathbf{w}, b) = \dfrac{||\mathbf{w}||^2}{2} & \\ \text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, & n = 1, \ldots, N \end{array}}$$

- This is a Quadratic Program (QP) with N linear inequality constraints

Large Margin = Good Generalization

- Large margins intuitively mean good generalization
- We can give a slightly more formal justification to this

Large Margin = Good Generalization

- Large margins intuitively mean good generalization
- We can give a slightly more formal justification to this
- Recall: Margin $\gamma = \frac{1}{||\mathbf{w}||}$
- Large margin $\Rightarrow$ small $||\mathbf{w}||$

Large Margin = Good Generalization

- Large margins intuitively mean good generalization
- We can give a slightly more formal justification to this
- Recall: Margin $\gamma = \frac{1}{||\mathbf{w}||}$
- Large margin $\Rightarrow$ small $||\mathbf{w}||$
- Small $||\mathbf{w}||$ $\Rightarrow$ regularized/simple solutions (w_i's don't become too large)
- Simple solutions $\Rightarrow$ good generalization on test data

Large Margin = Good Generalization

- Large margins intuitively mean good generalization
- We can give a slightly more formal justification to this
- Recall: Margin $\gamma = \frac{1}{||\mathbf{w}||}$
- Large margin $\Rightarrow$ small $||\mathbf{w}||$
- Small $||\mathbf{w}||$ $\Rightarrow$ regularized/simple solutions (w_i's don't become too large)
- Simple solutions $\Rightarrow$ good generalization on test data
- Want to see an even more formal justification? :-)

Large Margin = Good Generalization

- Large margins intuitively mean good generalization
- We can give a slightly more formal justification to this
- Recall: Margin $\gamma = \frac{1}{||\mathbf{w}||}$
- Large margin $\Rightarrow$ small $||\mathbf{w}||$
- Small $||\mathbf{w}||$ $\Rightarrow$ regularized/simple solutions (w_i's don't become too large)
- Simple solutions $\Rightarrow$ good generalization on test data
- Want to see an even more formal justification? :-)
  - Wait until we cover Learning Theory!

Solving the SVM Optimization Problem

- Our optimization problem is:

$$\boxed{\begin{array}{ll} \text{Minimize} \quad f(\mathbf{w}, b) = \dfrac{||\mathbf{w}||^2}{2} & \\ \text{subject to} \quad 1 \leq y_n(\mathbf{w}^T\mathbf{x}_n + b), & \qquad n = 1, \ldots, N \end{array}}$$

Solving the SVM Optimization Problem

- Our optimization problem is:

$$\boxed{\begin{array}{l} \text{Minimize} \quad f(\mathbf{w}, b) = \dfrac{||\mathbf{w}||^2}{2} \\ \text{subject to} \quad 1 \leq y_n(\mathbf{w}^T\mathbf{x}_n + b), \qquad n = 1, \ldots, N \end{array}}$$

- Introducing Lagrange Multipliers α_n ($n = \{1, \ldots, N\}$), one for each constraint, leads to the **Lagrangian**:

$$\boxed{\begin{array}{l} \text{Minimize} \quad L(\mathbf{w}, b, \alpha) = \dfrac{||\mathbf{w}||^2}{2} + \displaystyle\sum_{n=1}^{N} \alpha_n\{1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)\} \\ \text{subject to} \quad \alpha_n \geq 0; \quad n = 1, \ldots, N \end{array}}$$

Solving the SVM Optimization Problem

- Our optimization problem is:

$$\boxed{\begin{array}{ll} \text{Minimize} \quad f(\mathbf{w}, b) = \dfrac{||\mathbf{w}||^2}{2} & \\ \text{subject to} \quad 1 \leq y_n(\mathbf{w}^T \mathbf{x}_n + b), & n = 1, \ldots, N \end{array}}$$

- Introducing Lagrange Multipliers α_n ($n = \{1, \ldots, N\}$), one for each constraint, leads to the **Lagrangian**:

$$\boxed{\begin{array}{l} \text{Minimize} \quad L(\mathbf{w}, b, \alpha) = \dfrac{||\mathbf{w}||^2}{2} + \displaystyle\sum_{n=1}^{N} \alpha_n \{1 - y_n(\mathbf{w}^T \mathbf{x}_n + b)\} \\ \text{subject to} \quad \alpha_n \geq 0; \quad n = 1, \ldots, N \end{array}}$$

- We can now solve this Lagrangian
  - i.e., optimize $L(\mathbf{w}, b, \alpha)$ w.r.t. $\mathbf{w}$, b, and α
  - .. making use of the Lagrangian Duality theory..

Next class..

- Solving the SVM optimization problem
- Allowing misclassified training examples (**non-zero loss**)
- Introduction to kernel methods (nonlinear SVMs)