# Lecture 20: Support Vector Machines

# Outline

- Discriminative learning of classifiers.
  - Learning a decision boundary.
  - Issue: generalization.
- Linear Support Vector Machine (SVM) classifier.
  - Margin and generalization.
  - Training of linear SVM.

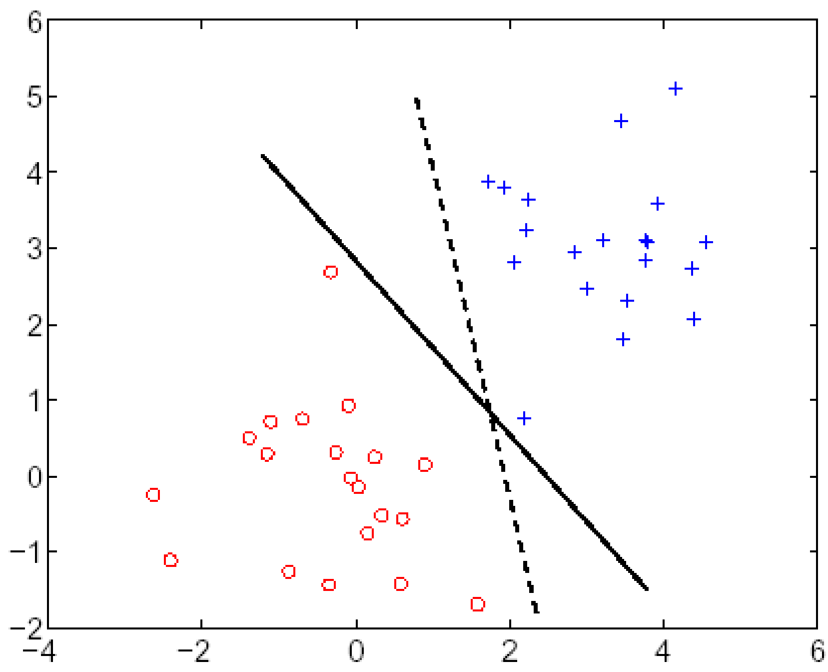
# Linear Classification

- Binary classification problem: we assign labels $y \in \{-1, 1\}$ to input data $\mathbf{x}$.

- Linear classifier: $y = \mathbf{sign}(\mathbf{w} \cdot \mathbf{x} + w_0)$ and its decision surface is a hyperplane defined by $\mathbf{w} \cdot \mathbf{x} + w_0 = 0$.

- Linearly separable: we can find a linear classifier so that all the training examples are classified correctly.

$$y_i[\mathbf{w} \cdot \mathbf{x}_i + w_0] > 0, \qquad \forall i = 1, ..., n$$



# Perceptrons

- Find line that separates input patterns so that output $o = +1$ on one side, $o = -1$ on other, and these match target values $y$

$$o(\mathbf{x}) = \mathbf{sign}(\mathbf{w} \cdot \mathbf{x} + w_0) =? y(\mathbf{x})$$

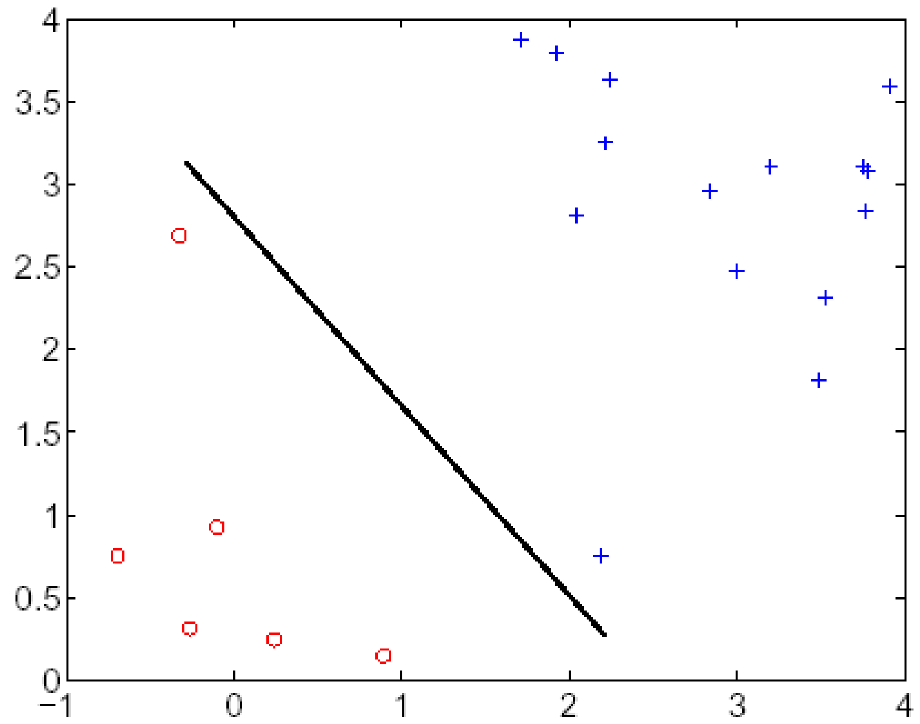
rewrite – for every training example $i$:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) > 0$$

- We can adjust weights $\{\mathbf{w}, w_0\}$ by Perceptron learning rule, which guarantees to converge to the correct solution in the *linear separable* case.

- Problem: which solution will has the best generalization?

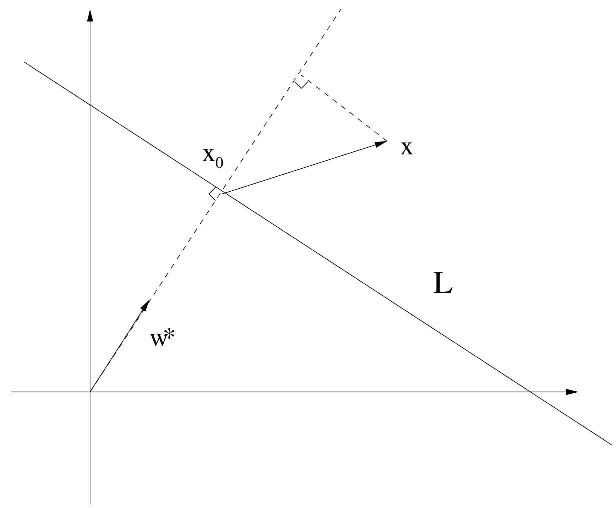
# Geometrical View of Linear Classifiers




- Margin: minimal gap between classes and decision boundary.
- Answer: The linear decision surface with the maximal *margin*.

# Geometric Margin

- Some Vector Algebra:




– Any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ lying in $L$, we have $\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 0$, which implies $\mathbf{w}^* = \mathbf{w}/||\mathbf{w}||$ is the unit vector normal to the surface of $L$.

– Any point $\mathbf{x}_0$ in $L$, $\mathbf{w} \cdot \mathbf{x}_0 = -w_0$.

– The signed distance of $\mathbf{x}$ to $L$ is given by

$$\mathbf{w}^* \cdot (\mathbf{x} - \mathbf{x}_0) = \frac{1}{||\mathbf{w}||}(\mathbf{w} \cdot \mathbf{x} + w_0)$$

- Geometric margin of $(\mathbf{x}_i, y_i)$ w.r.t $L$: $\gamma_i = y_i \frac{1}{||\mathbf{w}||}(\mathbf{w} \cdot \mathbf{x}_i + w_0)$.

- Geometric margin of $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$ w.r.t $L$: $\min_i \gamma_i$.

# Linear SVM Classifier

- Linear SVM maximizes the geometric margin of training dataset:

$$\begin{aligned} \max_{\mathbf{w},w_0} \quad & C \\ s.t. \quad & y_i \frac{1}{||\mathbf{w}||}(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq C, \quad i = 1,...,n \end{aligned} \tag{1}$$

- For any solution satisfying the constraints, any positively scaled multiple satisfies them too. So arbitrarily setting $||\mathbf{w}|| = 1/C$, we can formulate linear SVM as: ($\min ||x|| \Leftrightarrow \min 1/2||x||^2$)

$$\begin{aligned} \min_{\mathbf{w},w_0} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ s.t. \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1, \quad i = 1,...,n \end{aligned} \tag{2}$$

- With this setting, we define a margin around the linear decision boundary with thickness $1/||\mathbf{w}||$.

# Solution to Linear SVM

- We can convert the contrained minimization to an unconstrained optimization problem by representing the constraints as penality terms:

$$\min_{\mathbf{w}, w_0} \quad \frac{1}{2}||\mathbf{w}||^2 + \texttt{penality term}$$

- For data $(\mathbf{x}_i, y_i)$, use the following penality term:

$$\left\{ \begin{array}{l} 0, y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 \\ \infty, \texttt{otherwise} \end{array} \right\} = \max_{\alpha_i \geq 0} \alpha_i (1 - y_i [w_0 + \mathbf{w} \cdot \mathbf{x}_i])$$

- Rewrite the minimization problem

$$\begin{aligned} & \min_{\mathbf{w}, w_0} \{ \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \max_{\alpha_i \geq 0} \alpha_i (1 - y_i [w_0 + \mathbf{w} \cdot \mathbf{x}_i]) \} \qquad (3) \\ = \ & \min_{\mathbf{w}, w_0} \max_{\{\alpha_i \geq 0\}} \{ \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i [w_0 + \mathbf{w} \cdot \mathbf{x}_i]) \} \end{aligned}$$

- $\{\alpha_i\}$'s are called the *Lagrange multipliers*.

# Solution to Linear SVM (cont'd)

- We can swap 'max' and 'min':

$$\begin{aligned}
&\min_{\mathbf{w},w_0} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i(1 - y_i[w_0 + \mathbf{w} \cdot \mathbf{x}_i])\} \quad (4)\\
= &\max_{\{\alpha_i \geq 0\}} \min_{\mathbf{w},w_0} \underbrace{\{\frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i(1 - y_i[w_0 + \mathbf{w} \cdot \mathbf{x}_i])\}}_{J(\mathbf{w},w_0;\alpha)}
\end{aligned}$$

- We first minimize $J(\mathbf{w}, w_0; \alpha)$ w.r.t $\{\mathbf{w}, w_0\}$ for any fixed setting of the Lagrange multipliers:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}, w_0; \alpha) &= \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \quad (5)\\
\frac{\partial}{\partial w_0} J(\mathbf{w}, w_0; \alpha) &= -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (6)
\end{aligned}$$

# Solution to Linear SVM (cont'd)

- Substitute (5) and (6) back to $J(\mathbf{w}, w_0; \alpha)$:

$$\max_{\{\alpha_i \geq 0\}} \min_{\mathbf{w}, w_0} \underbrace{\{\frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i[w_0 + \mathbf{w} \cdot \mathbf{x}_i])\}}_{J(\mathbf{w}, w_0; \alpha)} \tag{7}$$

$$= \max_{\substack{\alpha_i \geq 0 \\ \sum_i \alpha_i y_i = 0}} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)\}$$

- Finally, we transform the original linear SVM training to a quadratic programming problem (7), which has the unique optimal solution.

- We can find the optimal setting of the Lagrange multipliers $\{\widehat{\alpha}_i\}$, then solve the optimal weights $\{\widehat{\mathbf{w}}, \widehat{w}_0\}$.

- Essentially, we transform the primal problem to its dual form. Why should we do this?

# Summary of Linear SVM

- Binary and linear separable classfication.

- Linear classifier with maximal margin.

- Training SVM by maximizing

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

subject to $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$.

- Weights $\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$.

- Only a small subset of $\hat{\alpha}_i$'s will be nonzero and the corresponding data $\mathbf{x}_i$'s are called *support vectors*.

- Prediction on a new example $\mathbf{x}$ is the sign of

$$\hat{w}_0 + \mathbf{x} \cdot \hat{\mathbf{w}} = \hat{w}_0 + \mathbf{x} \cdot (\sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i) = \hat{w}_0 + \sum_{i \in SV} \hat{\alpha}_i y_i (\mathbf{x} \cdot \mathbf{x}_i)$$