CS 4824/ECE 4424: Support Vector Machine

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Linear classifiers – multiple possibilities

- **Challenge**: How to pick the best classifier?
Pick the one with the largest margin!
Parameterizing the decision boundary

\[ w^T x + b > 0 \]

\[ w^T x + b < 0 \]

Labels \( y_i \in \{-1, +1\} \) — class
Maximizing the margin

- Margin = Distance of closest examples from the decision line/hyperplane

How to find the Max Margin = ?
Maximizing the margin

- Margin = Distance of closest examples from the decision line/hyperplane

$$\gamma = \frac{a}{\|w\|}$$

arg max \quad \frac{a}{\|w\|}

s.t. \quad (W^T X_j + b) y_j \geq a \forall j$$
Support Vector Machine

\[ \text{arg max}_{w,b} \, \frac{a}{\|w\|} \]
\[ s.t. (W^T X_j + b) y_j \geq a \, \forall j \]

\[ \text{arg min}_{w,b} \, W^T W \]
\[ s.t. (W^T X_j + b) y_j \geq a \, \forall j \]

Solve efficiently by quadratic programming (QP) - well studied

**Note:** a is arbitrary (can normalize equations by a)
Support Vector Machine

\[
\arg\max_{w,b} \frac{1}{\|w\|}
\]

\[
s.t. (W^T X_j + b) y_j \geq 1 \forall j
\]

\[
\arg\min_{w,b} W^T W
\]

\[
s.t. (W^T X_j + b) y_j \geq 1 \forall j
\]

Solve efficiently by quadratic programming (QP) - well studied

Note: a is arbitrary (can normalize equations by a)
SVM — primal and dual forms

Primal form: solve for \( w, b \)

\[
\begin{align*}
\text{arg min}_{w,b} & \quad w^T w \\
\text{s.t.} & \quad y_l (w^T X_l + b) \geq 1 \quad \forall l \in \text{training examples}
\end{align*}
\]

Classification for new \( X \): \( (w^T X + b) > 0 \)
SVM — primal and dual forms

**Primal form:** solve for \( w, b \)

\[
\begin{align*}
\arg\min_{w,b} & \quad W^TW \\
\text{s.t.} & \quad y_l(W^TX_l + b) \geq 1 \forall l \in \text{training examples}
\end{align*}
\]

Classification for new \( X \): \( (W^TX + b) > 0 \)

**Dual form:** solve for \( \alpha_1, \ldots, \alpha_n \)

\[
\arg\max_{\alpha_1,\ldots,\alpha_n} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{M} \alpha_j \alpha_k y_j y_k \langle X_j, X_k \rangle
\]

\[
\text{s.t.} \quad \alpha_l > 0 \forall l \in \text{training examples} \quad \sum_{l=1}^{M} \alpha_l y_l = 0
\]

Classification for new \( X \): \( \sum_{l \in SV's} \alpha_l y_l \langle X, X_l \rangle + b > 0 \)
Support Vectors

- The linear hyperplane is defined by “support vectors”
- Moving other points a little doesn’t effect the decision boundary
- Only need to store the support vectors to predict labels of new points

\[
\sum_{l \in \text{sv's}} \alpha_l y_l \langle x, x_l \rangle + b > 0
\]

\[
\sum_{l \in \text{sv's}} \alpha_l y_l \langle x, x_l \rangle + b < 0
\]
Kernel SVM — primal and dual forms

Primal form: solve for $w, b$

$$\arg \min_{w,b} w^T w$$
$$s.t. y_l(w^T \phi(X_l) + b) \geq 1 \forall l \in \text{training examples}$$

Classification for new $X$: $(w^T \phi(X) + b) > 0$

Dual form: solve for $\alpha_1, \ldots, \alpha_n$

$$\arg \max_{\alpha_1, \ldots, \alpha_n} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{M} \alpha_j \alpha_k y_j y_k K(X_j, X_k)$$
$$s.t. \alpha_l > 0 \forall l \in \text{training examples} \sum_{l=1}^{M} \alpha_l y_l = 0$$

Classification for new $X$: $\sum_{l \in SV's} \alpha_l y_l K(X, X_l) > + b > 0$

- Since the dual form depends only on inner products, we can apply the kernel trick to work in a (virtual) projected higher-dimensional space.
SVM Decision Surface using Gaussian Kernel

- Circled points are the *support vectors*: training examples with non-zero $\alpha_l$
- Points plotted in original 2-D space
- Contour lines show constant $\hat{f}(x)$

$$\hat{f}(x) = w^T \Phi(x) + b$$

$$\hat{f}(x) = b + \sum_{l=1}^{M} \alpha_l y_l \kappa(x, x_l) = b + \sum_{l=1}^{M} \alpha_l y_l \exp(-\|x - x_l\|^2/2\sigma^2)$$
SVM Summary

- **Objective**: maximize margin between decision surface and data

- Primal and dual formulations
  - dual represents classifier decision in terms of *support vectors*

- Kernel SVM’s
  - learn linear decision surface in high dimension space, working in original low dimension space

- SVM algorithm: Quadratic Program optimization
  - single global minimum