Introduction to Machine Learning
Foundations and Applications

Paul J. Atzberger
University of California Santa Barbara
Support Vector Machines
Support Vector Machines: Motivations

Consider data: \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}, with features x, labels y.

Example: x ∈ \mathbb{R}^N, y ∈ \{-1, +1\}, with x=image, y = +1 → Apple, y = -1 → Orange.

Task: Find hyperplane that separates points x_i having different labels y_i.

Challenges:

What algorithms can be used to find hyperplanes from data?

Many hyperplanes are possible. Which may have the best generalization?

What if the data is not separable?

How do we precisely define “separation” and the classification task?

Support Vector Machines

SVM: Optimization Problem (Primal $\mathcal{P}$)

\[
\begin{align*}
\min_{w,b} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to} & \quad y_i (w \cdot x_i + b) \geq 1.
\end{align*}
\]

**Separable Case:** $S = \{(x_i, y_i)\}_{i=1}^m$, $\mathcal{H} = \{h \mid h(x) = \text{sign}(w^T x + b), \ w \in \mathbb{R}^N, \ b \in \mathbb{R}\}$.

**Definition:** The data is *separable* if there exists $h \in \mathcal{H}$ so that $h(x_i) = \text{sign}(w^T x_i + b) = y_i$, $i \in \{1, 2, \ldots, m\}$.

**Hyperplane:** $Q = \{x \mid w_0^T x + b_0 = 0\} = \{x \mid c^{-1}w_0^T x + c^{-1}b_0 = 0\}$.

**Require:** $\min_{x_i} |w^T x_i + b| = 1$ for the $w$ used for a given data set $S$.

**Result:** We will always have $y_i(w^T x + b) \geq 1$.

**Definition:** The *geometric margin* $\rho(x)$ of a point $x$ is the distance to the hyperplane $Q$.

Let $x^*$ be s.t. $-w^T x^* = b$, then $w^T x + b = w^T (x - x^*)$.

$\Rightarrow \left| \frac{w}{\|w\|} (x - x^*) \right| = \frac{|w^T x + b|}{\|w\|} = \rho(x)$.

**Consequence:** the closest data point $x_{i^*}$ has distance $\rho(x_{i^*}) = 1/\|w\|$.
Consider a data set \(\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}\), where \(x\) denotes features and \(y\) denotes labels.

**Example:** \(x \in \mathbb{R}^N, y \in \{-1, +1\}\), with \(x=\text{image}, y = +1 \rightarrow \text{Apple}, y = -1 \rightarrow \text{Orange}\).

Find hyperplane separating points \(x_i\) having different labels \(y_i\) and with the “greatest margin” (helps with generalization).

Find parameters \(w, b\) that optimize

\[
\min_{w, b} \frac{1}{2} w^T w \\
\text{subject to } y_i \left( w^T \phi(x_i) + b \right) \geq 1 \quad \text{(for now, } \phi(x_i) = x_i \text{)}
\]

This assumes data is separable. Minimizing \(w\) maximizes the margin. Classifier \(h(x) = \text{sign} \left( w^T \phi(x) + b \right)\)

What if data is not separable?
Summary: SVM Non-Separable Case

Case of data that is not separable?

**Find hyperplane** and with **biggest margin** that minimizes extent of misclassifications.

Introduce "slack variables" $\xi$ for the constraint.

**Find parameters** $w, b, \xi$ that optimize

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i \quad \xi_i \geq 0$ (for now, $\phi(x_i) = x_i$)

Tries to find hyperplane and margin that minimizes the total amount training data points violate the constraint.

**C is crucial regularization parameter** determining penalty for violating the constraint.

Insights into generalization using results from optimization (duality).

Can apply more generally using kernel methods.
Kernel Methods Overview
Support Vector Machines (Kernel Trick)

Data is often not separable.

Mapping points to higher dimensional spaces they can become separable.

Example:

Kernel: associated to map $\phi$ is an inner-product $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

Example: $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + r_i^2r_j^2$

More generally...
### Support Vector Machines (Kernel Trick)

**Data is often not separable.**

**Mapping points to higher dimensional spaces** they can become separable.

**Kernel:** associated to map $\phi$ is an inner-product $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

**Example:** $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + r_i^2r_j^2$
Support Vector Machines (Kernel Trick)

Data is often not separable but can become separable in higher dimensional spaces.

Inner-products can be replaced by kernel $\langle x_i, x_j \rangle \rightarrow K(x_i, x_j)$.

Kernel should be symmetric, positive definite, $L^2$: $\sum_{i,j=1}^{N} a_i a_j K(x_i, x_j) \geq 0$, $\int_X \int_X K(x, t)^2 d\mu(x) d\mu(t) < \infty$

Consider: $L_K : L^2_\mu(X) \rightarrow L^2_\mu(X)$, $L_K f(x) = \int_X K(x, t) f(t) d\mu(t)$ has countable set of non-negative eigenvalues $\{\lambda_k\}_{k=1}^{\infty}$.

**Theorem (Mercer 1909):** An $L^2$ kernel $K(x, t)$ that is symmetric positive definite can be represented as the product

$$K(x, t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(t)$$

Let $\Phi(x) = [\Phi_k(x)]$, then $K(x, t) = \langle \Phi(x), \Phi(t) \rangle$.

**Consequence:** $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$, so $k(v, x_i) = w^T \Phi(x_i)$ as appears in the SVM constraints.

**Note,** only action $K(x, y)$ is needed in SVM, so no need to map explicitly to feature space $\Phi(x)$. 
Kernel Methods: Hilbert-Schmidt and Mercer Theorem

**Theorem (Hilbert-Schmidt):** For $L_K$ a **self-adjoint compact operator** there is a countable complete orthonormal basis $\{\phi_i\}$ for $L^2_\mu(\mathbb{Z})$ so that $L_K\phi_i = \lambda_i \phi_i$ with $\lambda_i \to 0$.

**Theorem (Mercer 1909):** An $L^2$-kernel $K(x,t)$ that is **symmetric positive definite** can be represented as the product

$$K(x,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x)\phi_k(t)$$

with $\lambda_i > 0$, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots$, and $\lambda_i \to 0$.

**Remark:** We can interpret the **Mercer Theorem** as stating **there exists a non-linear transformation** $z = \Phi(x)$ related to the kernel $K$ as follows. Let $[\Phi(x)]_k = \sqrt{\lambda_k} \Phi_k(x)$ then

$$K(x,t) = \langle \Phi(x), \Phi(t) \rangle_{\ell^2}.$$ Also can represent using Reproducing Kernel Hilbert Space (RKHS) (later).

**Consequence:** This shows that **if a kernel is $L^2$ and symmetric positive definite** then we can interpret it as being the **inner-product** associated with some **non-linear transformation** $\Phi$ of the data! For instance, $\mathbb{R}^N \to \ell^2$ or later $\mathbb{R}^N \to RKHS$.

**Remark:** To compute the inner-product we do **not** need to use $\langle \Phi(x), \Phi(t) \rangle_{\ell^2}$ which could be expensive, instead we **only need to evaluate** kernel $K(x,t)$. **This called the kernel trick!**
Support Vector Machines (Kernel Trick)

**Kernels** provide sensitivity to different features of the data, $K(x,t) = \langle \Phi(x), \Phi(t) \rangle$.

**Popular Kernels:**

- **Linear:**
  
  $K(x, y) = \langle x, y \rangle$

- **Radial Basis Function (RBF):**

  $K(x, y) = \exp[-\gamma \|x - y\|^2]$

- **Polynomial (degree $d$):**

  $K(x, y) = (\gamma \langle x, y \rangle + r)^d$

Lots of other choices possible.
Optimization Theory
Optimization

Constrained Optimization Problem (Primal $\mathcal{P}$)

$$\begin{align*}
\min_{x \in \mathcal{X}} f(x) \\
\text{subject to } g_i(x) \leq 0.
\end{align*}$$

**Definition:** The **Lagrangian** $\mathcal{L}$ of $\mathcal{P}$ is

$$
\mathcal{L}(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) \rightarrow \mathcal{L}(x, \alpha) = f(x) + \alpha \cdot g(x), \text{ where } \alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{bmatrix}, \quad g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_M(x) \end{bmatrix}.
$$

**Definition:** A **saddle-point** $(x^*, \alpha^*)$ of the Lagrangian $\mathcal{L}$ is a point satisfying

$$
\mathcal{L}(x^*, \alpha) \leq \mathcal{L}(x^*, \alpha^*) \leq \mathcal{L}(x, \alpha^*), \text{ holding for } \forall x \in \mathcal{X}, \alpha \geq 0.
$$

**Theorem:** For constrained optimization problem $\mathcal{P}$, a **saddle-point** $(x^*, \alpha^*)$ of the Lagrangian $\mathcal{L}$ is a solution of $\mathcal{P}$. 
Optimization

Constrained Optimization Problem (Primal $\mathcal{P}$)

\[
\begin{align*}
\min_{x \in \mathcal{X}} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0.
\end{align*}
\]

Lagrangian

\[
\mathcal{L}(x, \alpha) = f(x) + \alpha \cdot g(x)
\]

Theorem

For constrained optimization problem $\mathcal{P}$, a saddle-point $(x^*, \alpha^*)$ of the Lagrangian $\mathcal{L}$ is a solution of $\mathcal{P}$.

**Proof:** From $\mathcal{L}(x^*, \alpha) \leq \mathcal{L}(x^*, \alpha^*) \Rightarrow \alpha \cdot g(x^*) \leq \alpha^* \cdot g(x^*), \forall \alpha \geq 0, \Rightarrow g(x^*) \leq 0.$

If $g_i(x^*) > 0$ then take $\alpha_i$ large so that $\alpha_i \cdot g_i(x^*) > c_i$ for any given $c_i \in \mathbb{R}$. Let $c = \alpha^* \cdot g(x^*)$.

Furthermore, $\alpha^* \cdot g(x^*) = 0$. Consider $\alpha \to 0$ then $0 \leq \alpha^* \cdot g(x^*) \leq 0, \Rightarrow \alpha^* \cdot g(x^*) = 0$.

From $\mathcal{L}(x^*, \alpha^*) \leq \mathcal{L}(x, \alpha^*), \forall x \Rightarrow f(x^*) \leq f(x) + \alpha^* \cdot g(x)$ for all $x$ s.t. $g(x) \leq 0$.

We have $f(x^*) \leq f(x)$ so $(x^*, \alpha^*)$ solves $\mathcal{P}$. ■.
Optimization

**Definition:** Strong Constraint Qualification (Slater’s Condition)

\[ \exists \bar{x} \in \text{interior}(\mathcal{X}), \quad \forall i \in \{1, 2, \ldots, m\}, \quad g_i(\bar{x}) < 0. \]

**Definition:** Weak Constraint Qualification (Weak Slater’s Condition)

\[ \exists \bar{x} \in \text{interior}(\mathcal{X}), \quad \forall i \in \{1, 2, \ldots, m\}, \quad (g_i(\bar{x}) < 0) \lor (g_i(\bar{x}) = 0 \land g_i(\bar{x}) = a\bar{x} + b \text{ (affine)}). \]

**Theorem:** (when saddle point is necessary w/ strong slater)

Let \( f, g \) be convex functions with strong slater condition holding.
If \( x^* \) is a solution to \( P \) then \( \exists \alpha^* \geq 0 \) s.t. \( (x^*, \alpha^*) \) satisfies the saddle condition for \( \mathcal{L} \).

**Theorem:** (when saddle point is necessary w/ weak slater)

Let \( f, g \) be convex and differentiable functions with weak slater condition holding.
If \( x^* \) is a solution to \( P \) then \( \exists \alpha^* \geq 0 \) s.t. \( (x^*, \alpha^*) \) satisfies the saddle condition for \( \mathcal{L} \).
Optimization:

**Theorem: Karuch-Kuhn-Tucker (KKT) Conditions**

Let \( f, g_i \) be **convex** and **differentiable** functions where the **weak constraint qualification** is satisfied. A \( \bar{x} \) is a solution to the **constrained optimization problem** \( \mathcal{P} \) if and only if \( \exists \bar{\alpha} \geq 0 \) s.t.

\[
\nabla_x \mathcal{L}(\bar{x}, \bar{\alpha}) = \nabla_x f + \bar{\alpha} \cdot \nabla_x g(\bar{x}) = 0.
\]

\[
\nabla_{\alpha} \mathcal{L}(\bar{x}, \bar{\alpha}) = g(\bar{x}) \leq 0.
\]

\[
\bar{\alpha} \cdot g(\bar{x}) = \sum_{i=1}^{m} \bar{\alpha}_i g_i(\bar{x}) = 0.
\]

**SVM Separable Case (KKT):** \( \mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{m} \alpha_i [y_i (w \cdot x_i + b) - 1] \).

\[
\nabla_w \mathcal{L} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0, \quad \Rightarrow \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i.
\]

\[
\nabla_b \mathcal{L} = - \sum_{i=1}^{m} \alpha_i y_i = 0, \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0.
\]

\( \forall i, \alpha_i [y_i (w \cdot x_i + b) - 1] = 0, \quad \Rightarrow \quad \alpha_i = 0 \vee y_i (w \cdot x_i + b) = 1. \)
SVM
Dual Formulations
Optimization:

**Definition:** Dual Function
\[ \forall \alpha \geq 0, \quad F(\alpha) = \inf_{x \in \mathcal{X}} \mathcal{L}(x, \alpha). \]

**Definition:** Dual Optimization Problem \( \mathcal{P}^* \)
\[
\begin{align*}
\max_{\alpha \in \mathbb{R}^m} & \quad F(\alpha) \\
\text{subject to} & \quad \alpha_i \geq 0.
\end{align*}
\]

**SVM Separable Case (Dual Form):** \( \mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i [y_i (w \cdot x_i + b) - 1] \).

From KKT we have \( w = \sum_{i=1}^m \alpha_i y_i x_i \) and \( \sum_{i=1}^m \alpha_i y_i = 0 \). This gives

**Dual Function:** \( F(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \)

**Dual Optimization Problem \( \mathcal{P}^* \):**
\[
\begin{align*}
\max_{\alpha} & \quad \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\
\text{subject to} & \quad \alpha_i \geq 0 \land \sum_{i=1}^m \alpha_i y_i = 0, \quad \forall i \in \{1, 2, \ldots, m\}
\end{align*}
\]
SVN Non-Separable Case (Primal):
\[
\begin{align*}
\min_{w,b,\xi} & \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} \xi_i^p \\
\text{subject to} & \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \land \xi_i \geq 0, \ i \in [1,m]
\end{align*}
\]

Lagrangian:
\[
\mathcal{L}(w,b,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{m} \beta_i \xi_i
\]

KKT Conditions:
\[
\begin{align*}
\nabla_w \mathcal{L} &= w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \quad \implies \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i \\
\nabla_b \mathcal{L} &= - \sum_{i=1}^{m} \alpha_i y_i = 0 \quad \implies \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \\
\nabla_{\xi_i} \mathcal{L} &= C - \alpha_i - \beta_i = 0 \quad \implies \quad \alpha_i + \beta_i = C \quad \implies \quad \alpha_i \leq C \beta_i \leq C \\
\forall i, \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] &= 0 \quad \implies \quad \alpha_i = 0 \lor y_i(w \cdot x_i + b) = 1 - \xi_i \\
\forall i, \beta_i \xi_i &= 0 \quad \implies \quad \beta_i = 0 \lor \xi_i = 0.
\end{align*}
\]

Dual Function \( F(\alpha, \beta) \):
\[
\mathcal{L} = \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i x_i^2 - \sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{m} \alpha_i y_i b + \sum_{i=1}^{m} \alpha_i 
\]

\[
\mathcal{L} = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)
\]

same as separable case!
SVM Dual Formulation

SVM Non-Separable Case:

KKT Conditions:

\[ \nabla_w \mathcal{L} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i \]
\[ \nabla_b \mathcal{L} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \]
\[ \nabla_{\xi_i} \mathcal{L} = C - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C \]
\[ \forall i, \alpha_i [y_i (w \cdot x_i + b) - 1 + \xi_i] = 0 \quad \Rightarrow \quad \alpha_i = 0 \lor y_i (w \cdot x_i + b) = 1 - \xi_i \]
\[ \forall i, \alpha_i \xi_i = 0 \quad \Rightarrow \quad \beta_i = 0 \lor \xi_i = 0. \]

Dual Function \( F(\alpha, \beta) \):

\[ \mathcal{L} = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \]

Inner-products of \( (x_i \cdot x_j) \) only appear. Kernel Method: \( \hat{x} = \phi(x) \) holds for \( \hat{x}_i \cdot \hat{x}_j = \phi(x_i) \cdot \phi(x_j) = k(x_i, x_j) \).

Dimension of dual problem is \( m \). Primal problem has dimension \( N \).

Regularization \( C \) in primal problem \( \mathcal{P} \) becomes constraint in dual problem \( \mathcal{P}^* \).

Provides alternative ways to solve the optimization problem.

SVM Non-Separable Case (Dual \( \mathcal{P}^* \)):

\[ \max_\alpha \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \]
subject to: \( 0 \leq \alpha_i \leq C \land \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [1, m] \).
Generalization Error Bounds for Support Vector Machines
Example: Learning separating hyperplane in $\mathbb{R}^N$ (related to SVM). For data $\{(x_i, y_i)\}$ with $x_i \in \mathbb{R}^N$ and $y_i \in \{-1, 1\}$. Ideally, find $w$, $b$ so that $\text{sign}(w^T x_i + b) = y_i$.

Hypothesis class:
$\mathcal{H} = \{ h : h(x) = \text{sign}(w^T x + b) \text{ with } w \in \mathbb{R}^N, b \in \mathbb{R} \}$.

What is the $VCdim(\mathcal{H})$?

Claim: $VCdim(\mathcal{H}) = N + 1$

In separable case we have bound on generalization error ($pr > 1 - \delta$)

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{2(N+1) \log \frac{em}{N+1}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

We can do even better in bounding sampling complexity using special structure of SVM.

We want bounds independent of feature dimension $N$ so can handle large $N$ or even $N = \infty$. 

VC-Dimension: Hyperplanes

Image Database

Linear Classifier

features

features
SVM: Generalization Error

**Definition:** The geometric margin of a data point $x$ is

$$\rho(x) = \frac{y(w \cdot x + b)}{||w||} \quad y \in \{-1, 1\} \text{ depending on side of hyperplane}$$

**Definition:** The margin of linear classifier $h(x) = \text{sign}(w \cdot x + b)$ for data set $S = (x_1, x_2, \ldots, x_m)$ is

$$\rho = \min_{1 \leq i \leq m} \frac{y_i(w \cdot x_i + b)}{||w||}$$

**Definition:** We define a marginal loss function using

$$\Phi_\rho(x) = \begin{cases} 0 & \text{if } \rho \leq x \\ 1 - x/\rho & \text{if } 0 \leq x \leq \rho \\ 1 & \text{if } x \leq 0. \end{cases}$$

**Definition:** We define empirical marginal loss as

$$\hat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^{m} \Phi_\rho(y_i h(x_i))$$
**Theorem (Margin bound for binary classification):** For any fixed $\rho > 0$ and $\delta > 0$, we have with probability $1 - \delta$ that the generalization error for marginal loss function is

$$R(h) \leq \hat{R}_\rho(h) + \frac{2}{\rho} R_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

$$R(h) \leq \hat{R}_\rho(h) + \frac{2}{\rho} \hat{R}_S(H) + 3 \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$

**Key idea:** obtain bounds using the Rademacher Complexity of $\mathcal{H} = \{h: h(x) = \text{sign}(w^T x + b) \text{ with } w \in \mathbb{R}^N, b \in \mathbb{R}\}$.

**Notational convention:** Suppress the $b$ term by using $\bar{x} = [\bar{x}^T 1]^T, \bar{w} = [w^T b]^T, \mathcal{H} = \{h: h(\bar{x}) = \text{sign}(\bar{w}^T \bar{x}) \text{ with } \bar{w} \in \mathbb{R}^{N+1}\}$. 

**SVM: Generalization Error**
SVM: Generalization Error

Theorem: (Rademacher Complexity of Constrained Hyperplanes for Bounded Data $S$)

Let $S \subseteq \{x : \|x\| \leq r\}$ be a sample of size $m$ and let $H = \{h \mid h(x) = \text{sign}(w \cdot x) \mid \|w\| \leq \Lambda\}$. The Rademacher complexity satisfies

$$\hat{R}_S(H) \leq \sqrt{\frac{r^2 \Lambda^2}{m}}.$$

Proof: $\hat{R}_S(H) = \frac{1}{m} E_\sigma \left[ \sup_{\|w\| \leq \Lambda} \sum_{i=1}^{m} \sigma_i w \cdot x_i \right] = \frac{1}{m} E_\sigma \left[ \sup_{\|w\| \leq \Lambda} w \cdot \sum_{i=1}^{m} \sigma_i x_i \right]$

$$\leq \frac{\Lambda}{m} E_\sigma \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\| \right] \leq \frac{\Lambda}{m} E_\sigma \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|^2 \right]^{1/2}$$

$$\leq \frac{\Lambda}{m} E_\sigma \left[ \sum_{i,j=1}^{m} \sigma_i \sigma_j x_i \cdot x_j \right]^{1/2} \leq \frac{\Lambda}{m} \left[ \sum_{i=1}^{m} \|x_i\|^2 \right]^{1/2} \leq \frac{\Lambda \sqrt{m r^2}}{m} = \sqrt{\frac{r^2 \Lambda^2}{m}} \quad \blacksquare.$$

Cauchy-Swartz Lemma:

$$a \cdot b \leq \|a\| \|b\|.$$

Jensen Inequality:

$$\phi(E[X]) \leq E[\phi(X)].$$

$$(E[X])^2 \leq E[X^2].$$

Rademacher Random Variables:

$$E[\sigma_i \sigma_j] = E[\sigma_i] E[\sigma_j] = 0.$$

Talagrand’s Lemma:

Let $\Phi : \mathbb{R} \to \mathbb{R}$ be an $\ell$-Lipschitz function $\|\Phi(x) - \Phi(y)\| \leq \ell|x - y|$, then

$$\hat{R}_S(\Phi \circ H) \leq \ell \hat{R}_S(H).$$
**SVM: Generalization Error**

**Theorem (Margin bound for binary classification):** For any $\rho > 0$ and $\delta > 0$, we have with probability $1 - \delta$ that the generalization error for marginal loss function is

$$R(h) \leq \hat{R}_\rho(h) + \frac{2}{\rho} \mathcal{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

$$R(h) \leq \hat{R}_\rho(h) + \frac{2}{\rho} \mathcal{R}_S(H) + 3 \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$

**Key idea:** obtain bounds using the Rademacher Complexity of $\mathcal{H} = \{h: h(x) = \text{sign}(w^T x + b) \text{ with } w \in \mathbb{R}^N, b \in \mathbb{R}\}$.

**Notational convention:** Suppress the $b$ term by using $\bar{x} = \begin{bmatrix} x \end{bmatrix}, \bar{w} = \begin{bmatrix} w^T \end{bmatrix}, \mathcal{H} = \{h: h(\bar{x}) = \text{sign}(\bar{w}^T \bar{x}) \text{ with } \bar{w} \in \mathbb{R}^{N+1}\}$.
Theorem (Margin bound for binary classification): For any fixed $\rho > 0$ and $\delta > 0$, we have with probability $1 - \delta$ that the generalization error for marginal loss function is

$$R(h) \leq \hat{R}_\rho(h) + 2\sqrt{\frac{r^2\Lambda^2/\rho^2}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

We have the **marginal loss** is bounded by **hinge loss** $\Phi_1(x) \leq \max(1 - x, 0)$

Theorem: For any fixed $\delta > 0$, we have with probability $1 - \delta$ that the generalization error for marginal loss function is

$$R(h) \leq \frac{1}{m} \sum_{i=1}^{m} \xi_i + 2\sqrt{\frac{r^2\Lambda^2}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

Key Result: The SVM objective function $\rightarrow$ makes small the RHS bound!

Trade-off: make **slack variables small** while making **margin $\rho = 1/||w||$ large. Allows for $N = \infty$.**

Regularization: Make $||w||^2 \leq \Lambda^2$ small $\rightarrow$ serves as **regularization term** (controlled by $\Lambda = \Lambda(C^{-1})$).
Example of SVM
Classifying Apples & Oranges
Support Vector Machines (Apples and Oranges Training Data)

Apple Image Set

Orange Image Set

Average Apple

Average Orange
Supervised Learning (Apples and Oranges)

What features to use to distinguish apples and oranges?

- Natural to use the colors of the objects in the images.
- However, many different color spaces can be used (RGB, HLS, ...)
- Does the choice matter?

Red, Green, Blue: [Pixel (RGB)]
Hue, Luminance, Saturation: [Pixel (HLS)]
Support Vector Machines (Apples and Oranges Training Data)

RGB Training Data

HLS Training Data

Hue and Saturation Features (hue shift 0.5)
SVM Performance

Linear: $K(x, y) = \langle x, y \rangle$

RBF: $K(x, y) = \exp\left[-\gamma \|x - y\|^2\right]

Polynomial: $K(x, y) = (\gamma \langle x, y \rangle + r)^d$

SVM Results:

RGB Features

HLS Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roundness</td>
<td>0.8</td>
</tr>
<tr>
<td>Sweetness</td>
<td>0.9</td>
</tr>
<tr>
<td>Redness</td>
<td>0.1</td>
</tr>
<tr>
<td>Greenness</td>
<td>0.3</td>
</tr>
</tbody>
</table>
SVM Performance

Importance of features used?
Importance of regularization C?
How does training set generalize?

SVM Results:

<table>
<thead>
<tr>
<th>Item</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Images</td>
<td>Roundness</td>
</tr>
<tr>
<td></td>
<td>Sweetness</td>
</tr>
<tr>
<td></td>
<td>Redness</td>
</tr>
<tr>
<td></td>
<td>Greenness</td>
</tr>
<tr>
<td>Orange Images</td>
<td>Roundness</td>
</tr>
<tr>
<td></td>
<td>Sweetness</td>
</tr>
<tr>
<td></td>
<td>Redness</td>
</tr>
<tr>
<td></td>
<td>Greenness</td>
</tr>
</tbody>
</table>
SVM Example

Apples vs Oranges vs Blueberries
Support Vector Machines (Apples and Oranges and Blueberries)

How might we train on more than two data sets? Three data sets: Apples, Oranges, and Blueberries.
SVM Multi-class Classification
**Support Vector Machines (Multi-Class Case)**

Many classification involve multiple classes.

**Problem:** From data x learn for k classes $C_1, C_2, ..., C_k$ a classifying function $f(x) \rightarrow y \in \{C_1, C_2, ..., C_k\}$.

**Binary classification:** $f(x) \rightarrow y \in \{-1, 1\}$ which corresponds to $k = 2$ classes $C_1 = -1, C_2 = 1$.

**Multi-class classification:** $f(x) \rightarrow y \in \{1, 2, ..., k\}$ corresponds to $k$ classes $C_1 = 1, C_2 = 2, ..., C_k = k$.

How can we extend linear classifier methods to handle multiple classes?

**Two common approaches:**

**One vs All (OvA):** Reduce to a collection of $k$ binary classification problems to determine one category labeled +1 vs rest of the data labeled -1. Pick classification with the greatest margin.

**One vs One (OvO):** Reduce to a collection of $\binom{k}{2} = k(k - 1)/2$ binary classification problems to determine one category labeled +1 vs one other category labeled -1. Consider each classifier as a voter and pick class with the most number of votes.

Above heuristics do not always work well in practice. **Alternatives:** optimization formulations (more expensive).
Support Vector Machines (Multi-Class Classification)

Optimization of Maximum Margin (OMM): Classes $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, with $C_\ell = \ell$, data $(x_i, y_i)$ with $y_i \in \mathcal{C}$.

$$\min_{w, \xi} \frac{1}{2} \sum_{i=1}^{k} \|w_i\|^2 + C \sum_{i=1}^{m} \xi_i$$
subject: $\forall i \in [1, m], \forall l \in \mathcal{C} \setminus \{y_i\}$

$w_{y_i} \cdot \Phi(x_i) \geq w_l \cdot \Phi(x_i) + 1 - \xi_i, \quad \xi_i \geq 0$

Classifier obtained: $h(x) = \arg\max_{l \in \mathcal{C}} w_l \cdot \Phi(x_i)$, where $\Phi(x)$ is transformation of the data.

Dual Optimization Problem (Generalization):

$$\max_{\alpha \in \mathbb{R}^m \times k} \sum_{i=1}^{m} \alpha_i \cdot e_{y_i} - \frac{1}{2} \sum_{i=1}^{m} (\alpha_i \cdot \alpha_j) K(x_i, x_j)$$
subject to: $\forall i \in [1, m], (0 \leq \alpha_i y_i \leq C) \land (\forall j \neq y_i, \alpha_i y_j \leq 0) \land (\alpha_i \cdot 1 = 0)$

Generalization Bounds: $R(h) = \mathbb{E}_{x \sim D} [h(x) \neq f(x)]$ $\mathcal{H} = \{h(x) = \arg\max_{l \in \mathcal{C}} w_l \cdot \Phi(x) | W = (w_1, w_2, ..., w_k)^T, \sum_{l=1}^{k} \|w_l\|^2 \leq \Lambda^2 \}$

$$R(h) \leq \frac{1}{m} \sum_{i=1}^{m} \xi_i + 2k^2 \sqrt{\frac{\log \frac{1}{\delta}}{2m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$
(for any $\delta > 0$ holds with probability $1 - \delta$)

where $\xi_i = \max \{1 - [w_{y_i} \cdot \Phi(x_i) - \max_{y' \neq y_i} w_{y'} \cdot \Phi(x_i)], 0\}$ for all $i \in [1, m]$
Support Vector Machines (Apples and Oranges and Blueberries)

How might we train on more than two data sets?
Three data sets: Apples, Oranges, and Blueberries.

Apple Image Set
Orange Image Set
Blueberry Image Set

Average Apple
Average Orange
Average Blueberry
Supervised Learning (SVM Results: Apples, Oranges and Blueberries)

How does SVM distinguish between three data sets?
Importance of features used.
Importance of regularization C.
How does training set generalize?

SVM Results:

RGB Features

HLS Features

Linear Kernel

Green

Red

C = 1e-4.0

Saturation

Hue
Supervised Learning (SVM Results: Apples, Oranges and Blueberries)

SVM Results:

Linear: $K(x, y) = \langle x, y \rangle$

RBF: $K(x, y) = \exp[-\gamma \|x - y\|^2]$

Polynomial: $K(x, y) = (\gamma \langle x, y \rangle + r)^d$

RGB Features

HLS Features
Supervised Learning (SVM Results: Apples, Oranges and Blueberries)

How does SVM distinguish between three data sets?
Importance of features used.
Importance of regularization $C$.
How does training set generalize?

SVM Results:
- RGB Features
- HLS Features

Linear Kernel
- $C = 1e+6.9$

Linear Kernel II
- $C = 1e+6.9$

Radial Basis Function
- $C = 1e+6.9$

Polynomial Kernel
- $C = 1e+6.9$