March 2012

**Motivation**

- **Examples:** \( \{ \langle x \cdot -1 \rangle = 0 \} \)
  - \( \langle x \cdot -1 \rangle = 0 \)

**Linear Separators**

- **Hypothesis space:**
  - Linear separability:
    - \( \{0 = q + (x \cdot m) : x\} = \mathcal{H} \)
- \( \langle x \cdot m \rangle \), \( \langle 0 \rangle \)

**Decision boundary**

- Decision boundary is a solution for:
  - \( 0 = q + (x \cdot m) \)
- \( H \)

**Linear separator**

- \( H \)

**Examples:**

- \( \langle x \cdot -1 \rangle = 0 \)
  - \( \langle x \cdot -1 \rangle = 0 \)

**Assaf Glazer**
Problem Definition

For each example \( x_j \) (in some order)

\[ w + \eta y_j x_j = 0 \]

For each example \( x_j \) (in some order)

Which hypothesis is preferred?

Support Vector Machines

The Separate Case

Perceptron [Rosenblatt 62]

\[ 1 + \eta = 1 \]

\[ 1^x = 1^m \]

\[ \eta = 1^m \]

\[ 0 > 1^x (1^m) \]

For each example \( x_j \) (in some order)
The Minimal Distance Between Positive and Negative Examples

Support Vector Machines - The Separate Case

Maximal Margin

Point-Plane Distance
Convex Function

Lagrange Multipliers

Support Vector Machines
Support Vector Machines

The Separate Case

Primal objective function (min):

1. Setting the derivatives $w, b$ equal to zero:

$$\alpha = \frac{1}{2} \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1)$$

2. Calculating $w$:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

3. Final decision function:

$$f(x) = \text{sign}(w^T x + b)$$

The Non-Separate Case

Primal objective function (min):

1. Setting the derivatives $w, b$ equal to zero:

$$\alpha = \frac{1}{2} \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1)$$

2. Calculating $w$:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

3. Final decision function:

$$f(x) = \text{sign}(w^T x + b)$$
Support Vector Machines

The Non-Separate Case

Primal objective function (min):

\[ \sum_{i=1}^{n} \left( \alpha_i - y_i (x_i \cdot w) + b \right) \]

Subject to:

\[ 0 \leq \alpha_i \leq C \]

Dual problem is (max):

\[ \sum_{i=1}^{n} \alpha_i = \alpha \]

The dual problem is (max):

\[ \sum_{i=1}^{n} \alpha_i \left( y_i (x_i \cdot w) - b \right) + \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \]

Subject to:

\[ 0 \leq \alpha_i \leq C \]

Calculating \( w \), \( b \):

Final decision function:

\[ f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot w) + b \right) \]

The Non-Separate Case

Tuning C Parameter

Support Vector Machines

Train

Test

Train

Test

Tuning C Parameter

Support Vector Machines

Large C

Small C

Use an exponentially growing sequence of values

Course and fine search are recommended

Tuning the C parameter with cross-validation

Support Vector Machines

Tuning C Parameter

Support Vector Machines

Train

Test

Train

Test

Tuning C Parameter

Support Vector Machines

Final decision function:

\[ f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot w) + b \right) \]

Calculating \( w \), \( b \):

\[ \frac{1}{q + \frac{q}{q}} = q \]

\[ 1 - \frac{q}{q} \text{ max } = q \]

\[ 1 + \frac{q}{q} \text{ min } = q \]

\[ 0 \leq 1 - (q + x \cdot w) \frac{z}{q} \]

\[ \frac{z}{q} = \frac{x}{z} \]

\[ \text{sign} = (x) \cdot f \]
Generalization Vs. Overfitting

Too overfitted?

Too general?

Optimal?

Non-Linear SVM

Kernel functions are used to calculate inner products in $F$.

SVM decision function in $F$:

$\mathbf{f}(x) = \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i) \Phi(x) + b$

SVM decision function in $R^n$:

$\mathbf{f}(x) = \sum_{i=1}^{l} \alpha_i y_i x_i x + b$

Types of kernel function:

- Linear: $\mathbf{K}(x, x') = x \cdot x'$
- Polynomial: $\mathbf{K}(x, x') = (x \cdot x' + c)^d$
- Gaussian: $\mathbf{K}(x, x') = \exp\left(-\frac{(x - x')^2}{\gamma}\right)$
- Tanh: $\mathbf{K}(x, x') = \tanh(x \cdot x' + \gamma)$

Non-Linear SVM

Motivation

Motivation
Support Vector Machines

Tuning Parameters

- "Grid Search" procedure is used to find the $C$ and $\gamma$ parameters (for Gaussian kernels).
- Use an exponentially growing sequences of values.
- "Course and fine" search are recommended.

Train %
Test %

Support Vector Machines

Technical Considerations

- Normalization with respect to the train data.
- Transform nominal features into numeric ones using binary encoding.
- Handling missing values.
- "Grid Search" procedure is used to find the $C$ and $\gamma$ parameters.
Support Vector Machines

Karush-Kuhn-Tucker (KKT) Conditions

Necessary and sufficient conditions for the SVM optimization problem:

\[ \begin{align*}
\alpha_i &= 0 \quad \text{if} \quad 0 < \alpha_i < C \\
\alpha_i &= C \quad \text{if} \quad \left( (x) f(x) \right) \leq 0
\end{align*} \]

Support Vectors (SVs)

What \( \alpha \) coefficients can be used for?

- Finding instances that contribute to the classification decision
  (all SVs with \( 0 < \alpha_i \))

- Finding instances on the decision boundary (unbounded SVs with \( 0 < \alpha_i < C \))

- Feature selection using Linear SVM, Prefer feature \( i \) over feature \( j \)

Risk Minimization

Classification loss function:

\[ \begin{align*}
\text{The expected risk:} & \quad \mathbb{E} \left[ L(y, f(x)) \right] \\
\text{The empirical risk:} & \quad \frac{1}{l} \sum_{i=1}^{l} L(y_i, f(x_i))
\end{align*} \]

With probability \( 1 - \eta \), the following bound holds:

\[ \left( \frac{1}{l} \right)^{\frac{1}{2}} \log \frac{1}{\eta} \Rightarrow \mathbb{E} \left[ L(y, f(x)) \right] \leq \mathbb{E} \left[ L(y, f(x)) \right] + \alpha \]
VC Dimension

The VC-dim of $H$ in $\mathbb{R}^2$ is 3

VC Dimension

Theorem

\[ 1 + p = (\text{VC-dim}(H)) \]

Binary classification problems are considered

A property of the hypothesis space $(H)$

VC Dimension

Definition (VC-dim):

If 1 points can be labeled in 2 ways and for each labeling a member of $H$ can be found which correctly assign those labels, we say that set of points is shattered by $H$. The VC dimension for $H$ is the maximum number of points that can be shattered by $H$. The VC dimension (VC-dim) is:

\[ \text{VC-dim}(H) = d + 1 \]
Platt Confidence Value

Support Vector Machines

Platt Distribution:
Maximun log-likelihood criteria:
\[ (p)^{\sigma d} - 1 = (1 - \alpha) d \]
\[ \frac{(p)^{\sigma d} + 1}{1} = (p)^{\sigma d} = (1 + \alpha) d \]

Sigmoid function:
\[ \frac{(q + px) \exp + 1}{1} = (p)^{\sigma} d \]

Linear Vs. Non-linear

OR
Support Vector Machines

Multiclass Classification Methods

One versus all

\[ c = \arg \max_{y \in \{1, \ldots, C\}} \sum_{i=1}^{C} \alpha_i y_i \]

Tournament

Support Vector Machines

Multiclass Probabilities

One versus all

\[ (d, y, z, L) \]

Tournament
Support Vector Machines

Summary

Pros

• High learning power
• Generalization capabilities
• Built-in feature selection
• High learning power

Cons

• Sensitivity to missing values
• Sensitivity to noise

Reference

Burges, Christopher J. C.: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2:121-167, 1998