ECE595 / STAT598: Machine Learning I
Lecture 21 Support Vector Machine: Soft & Kernel

Spring 2020
Stanley Chan

School of Electrical and Computer Engineering
Purdue University
Outline

Support Vector Machine

- Lecture 19 SVM 1: The Concept of Max-Margin
- Lecture 20 SVM 2: Dual SVM
- Lecture 21 SVM 3: Soft SVM and Kernel SVM

This lecture: Support Vector Machine: Soft and Kernel

- Soft SVM
  - Motivation
  - Formulation
  - Interpretation

- Kernel Trick
  - Nonlinearity
  - Dual Form
  - Kernel SVM
Linearly Not Separable

- the points can be linearly separated but there is a very narrow margin

- but possibly the large margin solution is better, even though one constraint is violated

http://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf
Soft Margin

- We want to allow data points to stay inside the margin.
- How about change
  \[ y_j(w^T x_j + w_0) \geq 1 \]
  to this one:
  \[ y_j(w^T x_j + w_0) \geq 1 - \xi_j, \quad \text{and} \quad \xi_j \geq 0. \]
- If \( \xi_j > 1 \), then \( x_j \) will be misclassified.
Soft Margin

- We can consider this problem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \mathbf{w} \|^2_2 \\
\text{subject to} & \quad y_j (\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1 - \xi_j, \\
& \quad \xi_j \geq 0, \quad \text{for} \quad j = 1, \ldots, n,
\end{align*}
\]

- But we need to control \( \xi \), for otherwise the solution will be \( \xi = \infty \).

- How about this:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \mathbf{w} \|^2_2 + C \| \mathbf{\xi} \|^2 \\
\text{subject to} & \quad y_j (\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1 - \xi_j, \\
& \quad \xi_j \geq 0, \quad \text{for} \quad j = 1, \ldots, n,
\end{align*}
\]

- Control the energy of \( \xi \).
Role of $C$

- If $C$ is big, then we enforce $\xi$ to be small.
- If $C$ is small, then $\xi$ can be big.
No Misclassification?

- You can have misclassification in soft SVM
- $\xi_j$ can be big for a few outliers

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \mathbf{w} \|^2_2 + C \| \xi \|^2 \\
\text{subject to} & \quad y_j (\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1 - \xi_j, \\
& \quad \xi_j \geq 0, \quad \text{for } j = 1, \ldots, N.
\end{align*}
\]
L1 Regularization

- Instead of $\ell_1$-norm, you can also do

$$\minimize_{w, w_0, \xi} \frac{1}{2} \|w\|_2^2 + C \|\xi\|_1$$

subject to $y_j(w^T x_j + w_0) \geq 1 - \xi_j$,

$\xi_j \geq 0$, for $j = 1, \ldots, N$.

- This enforces $\xi$ to be sparse.
- Only a few entries samples are allowed to live in the margin.
- The problem remains convex.
- So you can still use CVX to solve the problem.
Connection with Perceptron Algorithm

- In soft-margin SVM, \( \xi_j \geq 0 \) and \( y_j(w^T x_j + w_0) \geq 1 - \xi_j \) imply that
  \[
  \xi_j \geq 0, \quad \text{and} \quad \xi_j \geq 1 - y_j(w^T x_j + w_0).
  \]

- We can combine them to get
  \[
  \xi_j \geq \max\left\{ 0, \ 1 - y_j(w^T x_j + w_0) \right\}
  = \left[ 1 - y_j(w^T x_j + w_0) \right]_+
  \]

- So if we use SVM with \( \ell_1 \) penalty, then
  \[
  J(w, w_0, \xi) = \frac{1}{2} \|w\|_2^2 + C \sum_{j=1}^{N} \xi_j
  = \frac{1}{2} \|w\|_2^2 + C \sum_{j=1}^{N} \left[ 1 - y_j(w^T x_j + w_0) \right]_+
  \]
Connection with Perceptron Algorithm

- This means that the training loss is

\[ J(w, w_0) = \sum_{j=1}^{N} \left[ 1 - y_j(w^T x_j + w_0) \right]_+ + \frac{\lambda}{2} \|w\|_2^2, \]

if we define \( \lambda = \frac{1}{C} \).

- Now, you can make \( \lambda \to 0 \). This means \( C \to \infty \).

- Then,

\[ J(w, w_0) = \sum_{j=1}^{N} \left[ 1 - y_j(w^T x_j + w_0) \right]_+ = \sum_{j=1}^{N} \max\left\{ 0, 1 - y_j(w^T x_j + w_0) \right\} = \sum_{j=1}^{N} \max\left\{ 0, 1 - y_j g(x_j) \right\} \]
Connection with Perceptron Algorithm

- **SVM Loss:**

\[
J(w, w_0) = \sum_{j=1}^{N} \max \{0, 1 - y_j g(x_j)\}
\]

- **Perceptron Loss:**

\[
J(w, w_0) = \sum_{j=1}^{N} \max \{0, -y_j g(x_j)\}
\]

Therefore: SVM generalizes perceptron by allowing

\[
J(w, w_0) = \sum_{j=1}^{N} \max \{0, 1 - y_j g(x_j)\} + \frac{\lambda}{2} \|w\|_2^2.
\]

\(\|w\|_2^2\) regularizes the solution.
Comparing Loss functions

https://scikit-learn.org/dev/auto_examples/linear_model/plot_sgd_loss_functions.html
Outline

Support Vector Machine
- Lecture 19 SVM 1: The Concept of Max-Margin
- Lecture 20 SVM 2: Dual SVM
- Lecture 21 SVM 3: Soft SVM and Kernel SVM

This lecture: Support Vector Machine: Soft and Kernel
- Soft SVM
  - Motivation
  - Formulation
  - Interpretation
- Kernel Trick
  - Nonlinearity
  - Dual Form
  - Kernel SVM
The Kernel Trick

- A trick to turn linear classifier to nonlinear classifier.
- Dual SVM

$$\begin{align*}
\text{maximize} \quad & - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{j=1}^{n} \lambda_j \\
\text{subject to} \quad & \sum_{j=1}^{n} \lambda_j y_j = 0.
\end{align*}$$

- Kernel Trick

$$\begin{align*}
\text{maximize} \quad & - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \Phi(x_i)^T \Phi(x_j) + \sum_{j=1}^{n} \lambda_j \\
\text{subject to} \quad & \sum_{j=1}^{n} \lambda_j y_j = 0.
\end{align*}$$

- You have to do this in dual. Primal is hard. See next slide.
The Kernel Trick

- Define

\[ K(x_i, vx_j) = \Phi(x_i)^T \Phi(x_j). \]

- The matrix \( Q \) is

\[
Q = \begin{bmatrix}
y_1 y_1 x_1^T x_1 & \ldots & y_1 y_N x_1^T x_N \\
y_2 y_1 x_2^T x_1 & \ldots & y_2 y_N x_2^T x_N \\
\vdots & \ddots & \vdots \\
y_N y_1 x_N^T x_1 & \ldots & y_N y_N x_N^T x_N \\
\end{bmatrix}
\]

- By Kernel Trick:

\[
Q = \begin{bmatrix}
y_1 y_1 K(x_1, x_1) & \ldots & y_1 y_N K(x_1, x_N) \\
y_2 y_1 K(x_2, x_1) & \ldots & y_2 y_N K(x_2, x_N) \\
\vdots & \ddots & \vdots \\
y_N y_1 K(x_N, x_1) & \ldots & y_N y_N K(x_N, x_N) \\
\end{bmatrix}
\]
The inner product $\Phi(x_i)^T \Phi(x_j)$ is called a **kernel**

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j).$$

- Second-Order Polynomial kernel

$$K(u, v) = (u^T v)^2.$$

- Degree-Q Polynomial kernel

$$K(u, v) = (\gamma u^T v + c)^Q.$$

- Gaussian Radial Basis Function (RBF) Kernel

$$K(u, v) = \exp \left\{ -\frac{\|u - v\|^2}{2\sigma^2} \right\}.$$
SVM with Second Order Kernel

Boxed samples = Support vectors.
Radial Basis Function takes the form of

\[ K(u, v) = \exp \left\{ -\gamma \| u - v \|^2 \right\} . \]

- Typical \( \gamma \in [0, 1] \).
- \( \gamma \) too big: Over-fit.
Non-Linear Transform for RBF?

- Let us consider scalar $u \in \mathbb{R}$.

\[
K(u, v) = \exp\{-(u - v)^2\}
\]
\[= \exp\{-u^2\} \exp\{2uv\} \exp\{-v^2\}
\]
\[= \exp\{-u^2\} \left( \sum_{k=0}^{\infty} \frac{2^k u^k v^k}{k!} \right) \exp\{-v^2\}
\]
\[= \exp\{-u^2\} \left( 1, \sqrt{\frac{2^1}{1!}} u, \sqrt{\frac{2^2}{2!}} u^2, \sqrt{\frac{2^3}{3!}} u^3, \ldots, \right)^T
\]
\[
\times \left( 1, \sqrt{\frac{2^1}{1!}} v, \sqrt{\frac{2^2}{2!}} v^2, \sqrt{\frac{2^3}{3!}} v^3, \ldots, \right) \exp\{-v^2\}
\]

- So $\Phi$ is

\[
\Phi(x) = \exp\{-x^2\} \left( 1, \sqrt{\frac{2^1}{1!}} x, \sqrt{\frac{2^2}{2!}} x^2, \sqrt{\frac{2^3}{3!}} x^3, \ldots, \right)
\]
So You Need

Example. Radial Basis Function

\[ K(u, v) = \exp \left\{ -\gamma \| u - v \|^2 \right\}. \]

The non-linear transform is:

\[ \Phi(x) = \exp\{-x^2\} \left( 1, \sqrt{\frac{2^1}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \sqrt{\frac{2^3}{3!}}x^3, \ldots, \right) \]

- You need infinite dimensional non-linear transform!
- But to compute the kernel \( K(u, v) \) you do not need \( \Phi \).
- Another Good thing about Dual SVM: You can do infinite dimensional non-linear transform.
- Cost of computing \( K(u, v) \) is bottleneck by \( \| u - v \|^2 \).
Is RBF Always Better than Linear?

- Noisy dataset: Linear works well.
- RBF: Over fit.
Testing with Kernels

- Recall:
  \[ w^* = \sum_{n=1}^{N} \lambda_n^* y_n x_n. \]

- The hypothesis function is
  \[
  h(x) = \text{sign} \left( w^*^T x + w_0^* \right)
  = \text{sign} \left( \sum_{n=1}^{N} \lambda_n^* y_n x_n \right)^T x + w_0^*
  = \text{sign} \left( \sum_{n=1}^{N} \lambda_n^* y_n (x_n^T x + w_0^*) \right).
  
- Now you can replace \( x_n^T x \) by \( K(x_n, x) \).
Support Vector Machine

- Mustafa, *Learning from Data*, e-Chapter
- Duda-Hart-Stork, *Pattern Classification*, Chapter 5.5
- Chris Bishop, *Pattern Recognition*, Chapter 7.1
- UCSD Statistical Learning
  - [http://www.svcl.ucsd.edu/courses/ece271B-F09/](http://www.svcl.ucsd.edu/courses/ece271B-F09/)