Optimization Methods In Machine Learning

Lecture 7-8: Support Vector Machines

Professor Katya Scheinberg

Lehigh University

Spring 2016

Outline

Outline

Review on Loss functions

Recall different type of loss functions:

- **0-1 loss function**

$$loss_{01}(h(x), y) = \begin{cases} 1 & \text{if} \;\; yh(x) < 0 \\ 0 & \text{if} \;\; yh(x) \geq 0. \end{cases}$$

- **Margin loss function**

$$loss_{m}(h(x), y) = \begin{cases} 1 & \text{if} \;\; yh(x) < 1 \\ 0 & \text{if} \;\; yh(x) \geq 1. \end{cases}$$

- **Logistic loss function**

$$loss_{g}(h(x), y) = log(1 + e^{-yh(x)}).$$

Hinge loss function

- The hinge loss function is used for "Maximum Margin Classification", most notably for "Support Vector Machines".
- For an intended output $y = \{+1, -1\}$ and a classifier $h(x)$, the hinge loss of the $h(x)$ is defined as:

$$loss_h(h(x), y) = \begin{cases} 0 & \text{if } \; yh(x) \geq 1 \\ 1 - yh(x) & \text{if } \; yh(x) < 1. \end{cases}$$

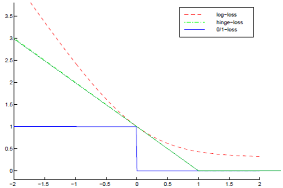


Figure: 0-1-loss upper bounded by log-loss and hinge-loss[1]

[1] Ben Taskar, Learning Structured Prediction Models:A Large Margin Approach, PhD Thesis,2004.

Properties of hinge loss function

- The hinge loss is a **convex** function, so many of the usual convex optimizers used in machine learning can work with it.
- Hinge loss function is **Lipschitz** with Lipschitz constant $L = 1$.
- Hinge loss is an upper bound on 0-1 loss.

Rademacher Complexity

- Rademacher Complexity is a concept for a particular size of sample set.
- It is upper-bounded by $\sqrt{\frac{W^2 X^2}{m}}$.

Where m is the sample size and X is the radius of the ball containing whole possible data (not just sample data), which is $X \geq \|\phi(x_i)\|$.

Note: We also want to keep W small!

Example

Recall logistic loss minimization:

$$\min \frac{1}{m}\sum_{i=1}^{m} loss_g(w^T\phi(x_i), y_i),$$
$$w: \ \|w\| \leq W.$$

- With $W = 10^6$ we will have:

$\hat{w} : \|\hat{w}\| = 10^6 \implies \hat{R}_g(\hat{w}) = 0.001 \implies R_g(w) \leq 0.001 + \sqrt{\frac{W^2X^2}{m}}$
Which means we need a sample size m as: $m = (100 \times 1000 \times 10^6)^2$.

- With $W = 100$ we will have:

$\hat{w} : \|\hat{w}\| = 100 \implies \hat{R}_g(\hat{w}) = 0.002 \implies R_g(w) \leq 0.002 + \sqrt{\frac{W^2X^2}{m}}$
Which means we need a sample size m as: $m = (100 \times 500 \times 100)^2$.

Outline

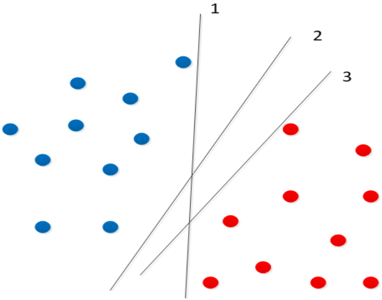
Linear Separators

- Minimizing following problem over w is the goal:

$$\min_{w} \sum_{i=1}^{m} loss_g(h(x_i), y_i) + \lambda \|w\|^2.$$

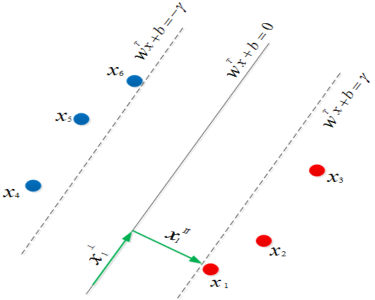
Note: The norm of w is not restricted.

- Which of the linear separators is optimal?



Classification Margin

- Vectors closest to the hyperplane are **Support Vectors**.
- **Margin** of the separator is the distance between support vectors.



Maximum Margin Classification

$$\max \gamma$$

$$w^T x_1 \geq \gamma, \qquad w^T x_4 \leq -\gamma,$$

$$w^T x_2 \geq \gamma, \qquad w^T x_5 \leq -\gamma,$$

$$w^T x_3 \geq \gamma, \qquad w^T x_6 \leq -\gamma,$$

$\gamma \leq w^T x_1 + b = w^T x_1^{\parallel} + w^T x_1^{\perp} = \|w\| \|x_1^{\parallel}\|.$
$\gamma \leq -w^T x_4 - b = -w^T x_4^{\parallel} - w^T x_4^{\perp} = \|w\| \|x_4^{\parallel}\|.$

- $x_1^{\parallel}$ (from positive class) is colinear with w and has the same orientation, while $x_4^{\parallel}$ (from negative class) is collinear with w and has the opposite orientation. We have shown: $\gamma \leq \|w\| \|x_i^{\parallel}\| \ \forall i$.
- Maximizing γ while constraining $\|w\| \leq 1$ is equivalent to minimizing $\|w\|$ while constraining $\gamma \geq 1$.

Outline

Separable Case

- Mathematical formulation:

$$\min_{w,b} \|w\|^2,$$

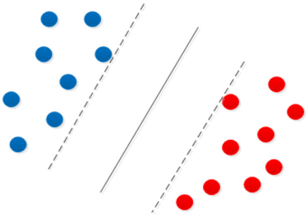
$$s.t \quad w^T x_i + b \geq 1, \qquad if \quad y_i = 1,$$

$$w^T x_i + b \leq -1, \quad if \quad y_i = -1.$$

- Which is equivalent to:

$$\min_{w,b} \|w\|^2,$$

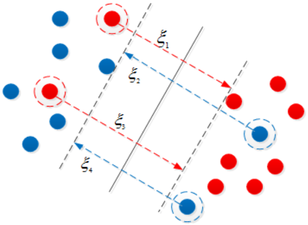
$$s.t \quad y_i(w^T x_i + b) \geq 1, \qquad \forall i = 1 \ldots m.$$



Non-separable Case

- **What if the points are not linearly separable?**

Error parameters ξ_i can be added to allow misclassification of difficult or noisy examples.



- Mathematical model after constraint relaxation and adding penalty to objective function:

$$\min_{w,b,\xi} \|w\|^2 + C\sum_{i=1}^{m}\xi_i,$$

$$\begin{aligned} s.t \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad && \forall i = 1...m, \\ & \xi_i \geq 0, && \forall i = 1...m. \end{aligned}$$

Hinge Loss vs. Misclassification Errors

- For any w and b, there is different feasible value for ξ, but we can optimize ξ_i separately for each sample point and make $\sum_{i=1}^{m} \xi_i$ minimum.
- The optimal value of ξ_i for any x_i and y_i is the following:

$$\xi_i = \max\{1 - y_i(w^T x_i + b), 0\}.$$

- This optimal value of ξ always is equal to to "hinge loss".

Unconstrained formulation

- Let the optimal solution be:

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{m} loss_h(w^T x_i, y_i) = \arg\min f(w).$$

- **Representer Theorem:** The optimal w, always is the linear combination of the data points x_i, for $i = 1...m$:

$$w^* = \sum_{i=1}^{m} \alpha_i x_i.$$

Proof of "Representer Theorem":

Consider $w^* = w^{\perp} + w^{\parallel}$, where:

$$w^{\parallel} : \quad w^{\parallel} = \sum_{i=1}^{m} \alpha_i x_i,$$

$$w^{\perp} : \quad {w^{\perp}}^T x_i = 0, \quad \forall i = 1...m.$$

Based on definition ($\|w^*\|^2 = \|w^{\parallel}\|^2 + \|w^{\perp}\|^2$) we have $\|w^*\|^2 \geq \|w^{\parallel}\|^2$.

On the other hand:

$${w^*}^T x_i = {w^{\parallel}}^T x_i + {w^{\perp}}^T x_i \implies {w^*}^T x_i = {w^{\parallel}}^T x_i.$$

Which implies:

$$loss({w^*}^T x_i, y_i) = loss({w^{\parallel}}^T x_i, y_i).$$

$w^{\parallel}$ and w^* have the same loss, but the norm of $w^{\parallel}$ is less than the norm of w^*, which means w^* can not be the minimizer!

So, for the optimal w^*, we have $w^* = w^{\parallel} = \sum_{i=1}^{m} \alpha_i x_i$.

Lagrange Duality and Optimality

- Lagrangian with multipliers α and u is the following:

$$L(w, b, \xi, \alpha, u) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i(w^Tx_i + b) - 1 + \xi_i\right) - u_i\xi_i.$$

- KKT conditions:

- Derivatives:

$$\nabla_w L(w, b, \xi, \alpha, u) = w - \sum_{i=1}^{m}\alpha_i y_i x_i = 0 \implies w = \sum_{i=1}^{m}\alpha_i y_i x_i$$

Note: This is exactly the result of “Representer Theorem”.

$$\nabla_\xi L(w, b, \xi, \alpha, u) = C - \alpha_i - u_i = 0 \quad \forall i \implies u_i = C - \alpha_i \quad \forall i$$

$$\nabla_b L(w, b, \xi, \alpha, u) = \sum_{i=1}^{m}\alpha_i y_i = 0 \quad \forall i$$

$$\alpha_i \geq 0 \quad u_i \geq 0 \implies 0 \leq \alpha_i \leq C \quad \forall i$$

Solving Optimization Problem

- Complementary slackness for inequality constraints:

$$\begin{aligned} y_i(w^T x_i + b) \geq 1 - \xi_i \quad &\longleftrightarrow \quad \alpha_i, \\ \xi_i \geq 0 \quad &\longleftrightarrow \quad C - \alpha_i, \end{aligned}$$

- $\xi_i = 0$ and $\alpha_i = 0 \implies y_i(w^T x_i + b) \geq 1$: the point is classified up the margin.
- $\xi_i = 0$ and $\alpha_i > 0 \implies y_i(w^T x_i + b) = 1$: the point is on the margin (it's support vector).
- $\xi_i > 0$ and $\alpha_i > 0 \implies y_i(w^T x_i + b) \geq 1 - \xi_i$: the point is misclassified and is up the margin.

Note1: Since $w = \sum_{i=1}^{m} \alpha_i y_i x_i$, we can omit points with zero α (which are on the "correct" side of the margin and not on the margin), and have exactly the same solution.

Note2: The points on the margin (and those violating the margin) are support vectors and only these vectors matter; other training vectors are ignorable.

Equivalent representations

- By using the definition of $w = \sum_{i=1}^{m} \alpha_i y_i x_i$, we will have the following optimization problem:

$$\min_{\alpha,\xi} \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j (y_j x_j)^T (y_i x_i) + C \sum_{i=1}^{m} \xi_i,$$

$$s.t \;\; (\sum_{j=1}^{m} \alpha_j y_j x_j)^T y_i x_i \geq 1 - \xi_i, \qquad \forall i = 1...m,$$

$$\xi_i \geq 0, \qquad \forall i = 1...m.$$

Equivalent representations

- By defining matrix $Q \subseteq \mathbb{R}^{m\times m}$ as $Q_{ij} = (y_i y_j)x_i^T x_j$, we will have:

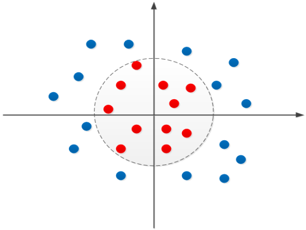
$$
\begin{aligned}
&\min_{\alpha,b,\xi} \ \frac{1}{2}\alpha^T Q\alpha + C\sum_{i=1}^{m}\xi_i, \\
s.t \ \ & Q\alpha + y_i b \geq 1 - \xi_i, \qquad \forall i = 1...m, \\
& \xi_i \geq 0, \qquad\qquad\qquad\quad \forall i = 1...m.
\end{aligned}
$$

- We can also solve the dual form of this optimization problem:

$$
\begin{aligned}
&\min_{\alpha} \ \frac{1}{2}\alpha^T Q\alpha - e^T\alpha, \\
s.t \ \ & y^T\alpha = 0, \qquad \forall i = 1...m, \\
& 0 \leq \alpha \leq C, \quad \forall i = 1...m.
\end{aligned}
$$

Kernel Methods and Nonlinear classification

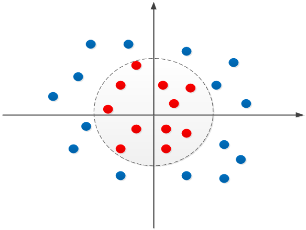
- We often need to deal with nonlinear pattern in data,
- Classes may not be separable by a linear boundary,



- In this case linear SVM is not powerful enough to separate classes accurately.

Kernel Methods and Nonlinear classification

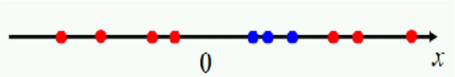
- We often need to deal with nonlinear pattern in data,
- Classes may not be separable by a linear boundary,



- In this case linear SVM is not powerful enough to separate classes accurately.
- Kernels make linear models work in nonlinear setting by mapping data to higher dimensions (via changing the feature representation).
- Linear model can be applied in new feature space.

Classifying Non-Linearly Separable Data via Kernel Trick(Example 1)

- Consider a binary classification problem [2] such that each example is represented by a single feature x.

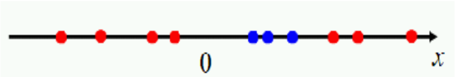


- No linear boundary can separate these data points,

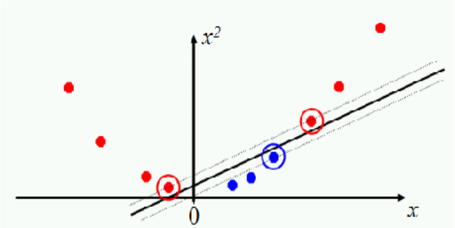
[2] http://nlp.stanford.edu/IR-book/html/htmledition/nonlinear-svms-1.html

Classifying Non-Linearly Separable Data via Kernel Trick(Example 1)

- Consider a binary classification problem [2] such that each example is represented by a single feature x.



- No linear boundary can separate these data points,
- By mapping $x \longrightarrow \{x, x^2\}$, each point has two features x and x^2.

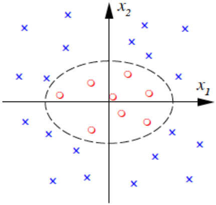


- Data points became linearly separable in the new feature space.

[2] http://nlp.stanford.edu/IR-book/html/htmledition/nonlinear-svms-1.html

Classifying Non-Linearly Separable Data via Kernel Trick(Example 2)

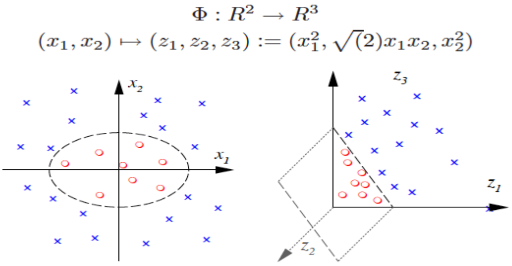
- Consider another problem such that each example is defined by two features $\{x_1, x_2\}$.



- No linear separator can classify these data points.
- Now by defining new feature space $\{x_1, x_2\} \longrightarrow \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$, each point has three features.

Classifying Non-Linearly Separable Data via Kernel Trick(Example 2)

- Consider another problem such that each example is defined by two features $\{x_1, x_2\}$.



- No linear separator can classify these data points.
- Now by defining new feature space $\{x_1, x_2\} \longrightarrow \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$, each point has three features and points are linearly separable in this new feature space. [3]

[3] http://courses.cs.ut.ee/2011/graphmining/Main/KernelMethodsForGraphs

Feature Mapping

- Consider following **quadratic mapping** for a $d-$dimensional point $x = \{x_1, x_2, ..., x_d\}$, such that each new feature uses a pair of original feature,

$$\phi : x \longrightarrow \{x_1^2, x_2^2, ..., x_d^2, x_1x_2, x_1x_3, ..., x_1x_d, ..., x_{d-1}x_d\},$$

Feature Mapping

- Consider following **quadratic mapping** for a $d-$dimensional point $x = \{x_1, x_2, ..., x_d\}$, such that each new feature uses a pair of original feature,
$$\phi : x \longrightarrow \{x_1^2, x_2^2, ..., x_d^2, x_1x_2, x_1x_3, ..., x_1x_d, ..., x_{d-1}x_d\},$$
- In general computing mapping can be expensive and inefficient,
- On the other hand since after this mapping the number of features blow up, using the mapped representation may be inefficient too!
- By using Kernel Trick we can avoid both of these issues, since the mapping does not have to be explicitly computed, and working with new feature space remain efficient.

Kernels as Highe Dimensional Feature Space

- Consider two examples $x = \{x_1, x_2\}$ and $z = \{z_1, z_2\}$,
- Assume we have the kernel function k as $k(x, z) = (x^T z)^2$, then we have

$$\begin{aligned} k(x, z) &= (x^T z)^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2} z_1 z_2, z_2^2) \\ &= \phi(x)^T \phi(z). \end{aligned}$$

Kernels as Highe Dimensional Feature Space

- Consider two examples $x = \{x_1, x_2\}$ and $z = \{z_1, z_2\}$,
- Assume we have the kernel function k as $k(x, z) = (x^T z)^2$, then we have

$$\begin{aligned} k(x, z) &= (x^T z)^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2} z_1 z_2, z_2^2) \\ &= \phi(x)^T \phi(z). \end{aligned}$$

- The kernel function k, implicitly defines a mapping ϕ to a higher dimensional space as

$$\phi(x) = \{x_1^2, \sqrt{2} x_1 x_2, x_2^2\}.$$

Kernels as inner products in high-dimensional feature space

- We **do not have to** define or compute this mapping,
- Defining kernel function $k(x, z)$ in a certain way, produce the higher dimensional mapping ϕ,
- Note that the kernel function $k(x, z)$ computes the dot product $\phi(x)^T \phi(z)$, (so we don't need to do this expensive computation explicitly).
- All kernel functions have these properties.

The Kernel Matrix

- Kernel function k defines the kernel matrix K over the data,
- Given m examples $\{x_1, ..., x_m\}$, the $(i, j)-$th entry of matrix K is defined as:

$$K_{ij} = k(x_i, x_j) = \phi(x_i)^T \phi(x_j),$$

- Matrix K is a symmetric matrix.
- Matrix K is a positive definite matrix, except for a few exceptions

Kernel's Examples

The following are the most popular kernels,

- Linear Kernel:
$$k(x, z) = x^T z$$
- Quadratic Kernel:
$$k(x, z) = (x^T z)^2 \quad \text{or} \quad (1 + x^T z)^2$$
- Polynomial Kernel of degree d:
$$k(x, z) = (x^T z)^d \quad \text{or} \quad (1 + x^T z)^d$$
- Radial Basis Function (RBF) Kernel:
$$k(x, z) = \exp(-\frac{\|x - z\|^2}{2\sigma}).$$

- Kernel hyperparametres (e.g. d, σ) chosen via cross-validation.

Kernelized SVM Training

- By using $\phi(x)$ instead of x, the first representation(primal model) of the model will change to the following:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i, \\ s.t \quad & y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \qquad \forall i = 1...m, \\ & \xi_i \geq 0, \qquad \forall i = 1...m. \end{aligned}$$

- The change in second representation(dual problem) will be in substituting dot product $x_i^T x_j$ by,

$$K_{ij} = k(x_i, x_j) = \phi(x_i)^T\phi(x_j).$$

Note: Kernelized SVM learns a linear separator in new feature space which corresponds to a non-linear separator in original feature space.

Refereces

You can find more detail in following materials

- https://www.cs.utah.edu/~piyush/teaching/15-9-slides.pdf