An introduction to Support Vector Machine

Liyu Gong

Department of Statistics
University of Kentucky

02/15/2018

Outline

Machine Learning is An Optimization Problem

- Structural model: candidate decision function set
- Error model: criterion to be optimized
- Algorithm: computable way to do the optimization

Linear Regression: An Example

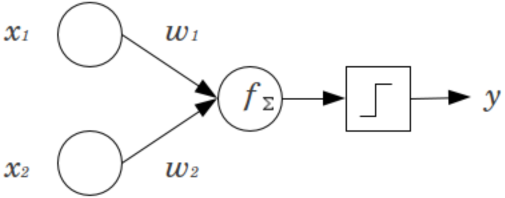
- Structural model: $y(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{x}$
- Error model: $\sum(t_i - y(x_i))^2$
- Algorithm: closed form solution or gradient descent

Perceptron: structural model

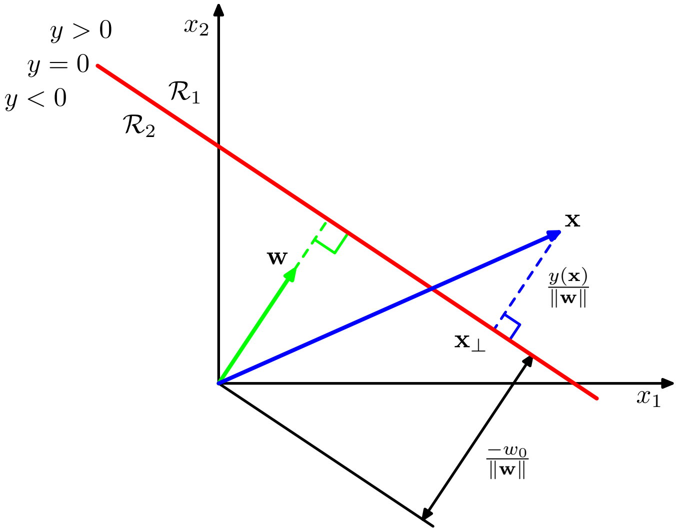
$$y(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w}^T\boldsymbol{x} + b)$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & : x \geq 0 \\ -1 & : x < 0 \end{cases}$$



Linear Discriminant Plane



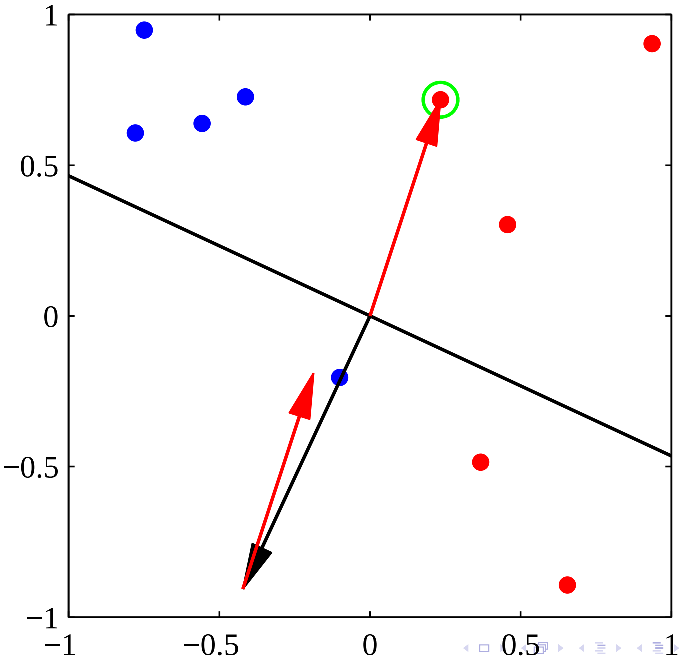
Error Model and Algorithm

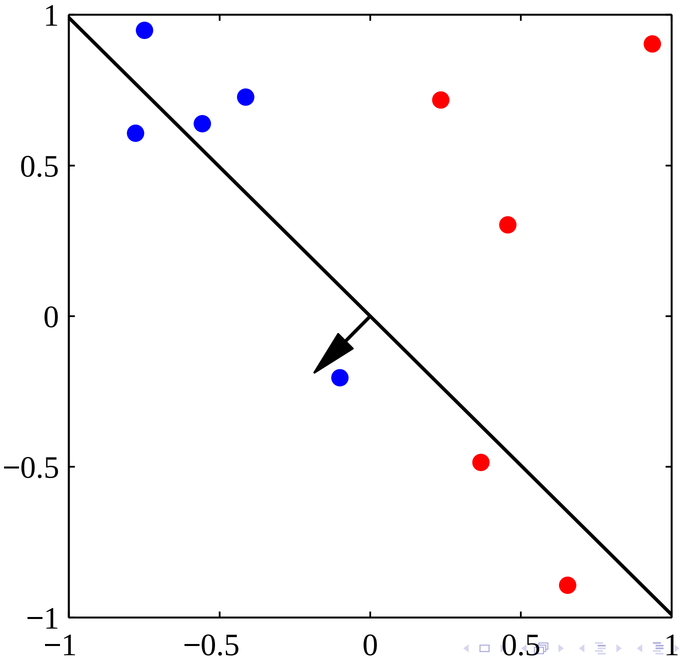
- perceptron criterion

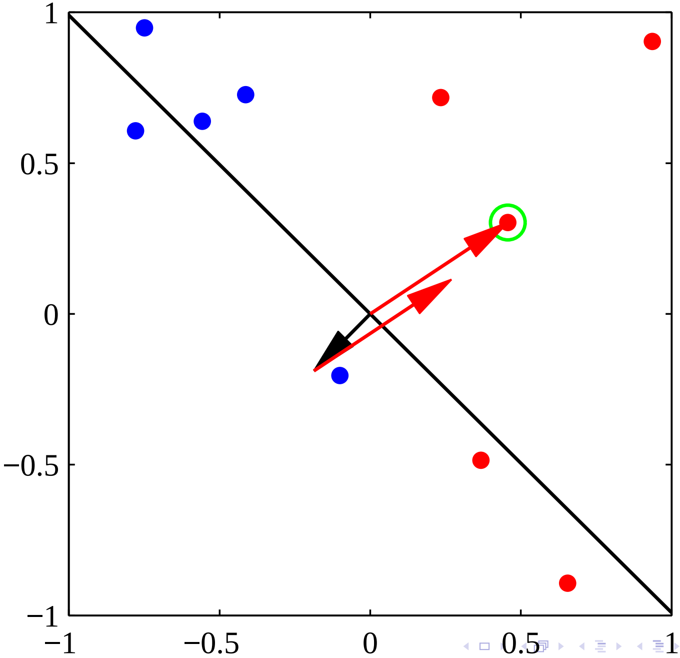
$$E_P(\boldsymbol{w}) = -\sum_{n\in\mathcal{M}} \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_n t_n$$

- stochastic gradient descent

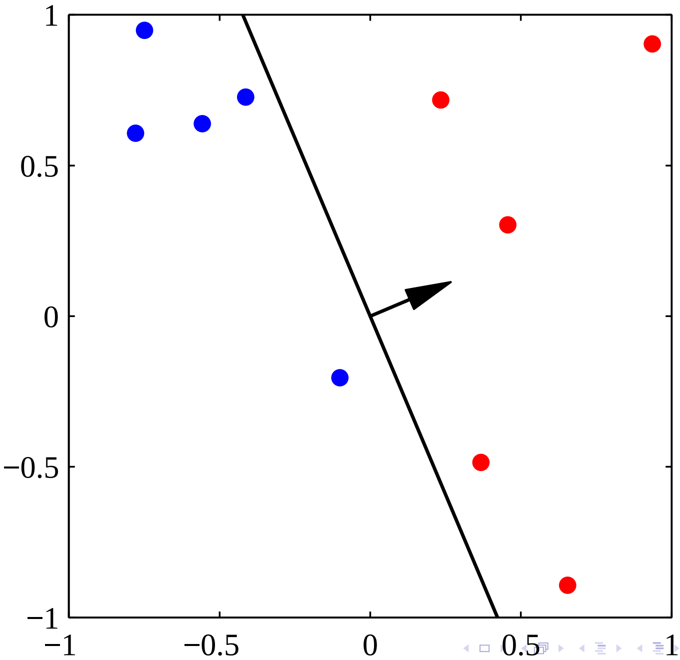
$$\begin{aligned}\tilde{\boldsymbol{w}}^{\tau+1} &= \boldsymbol{w}^{\tau} - \eta \nabla E_P(\boldsymbol{w}) \\ &= \boldsymbol{w}^{\tau} + \eta \boldsymbol{x}_n t_n\end{aligned}$$


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1



1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

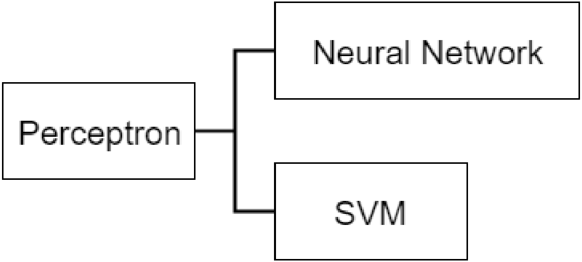
Drawbacks and Insights

- Converges in finite steps only if there exist an exact solution
- It is a linear model
- There are multiple solutions, which one is selected depends on the initial value of and the order of training set
- If $\tilde{\boldsymbol{w}}$ is initialized as a linear combination of $\tilde{\boldsymbol{x}}_n$, then the resultant $\tilde{\boldsymbol{w}}$ is also a linear combination of $\tilde{\boldsymbol{x}}$, which means the final solution depends on $\boldsymbol{x}_n$ through their inner product only.
- Only a subset of the training data contributes to the decision function


perceptron


MARK I PERCEPTRON

Overcome the Problems of Perceptron



- Problem 1: Non-Linearity
  - Neural Network: multiple layer perceptron
  - SVM: Kernels
- Problem 2: No unique solution
  - Neural Network: I don't care ...
  - SVM: maximum margin classification hpyerplane

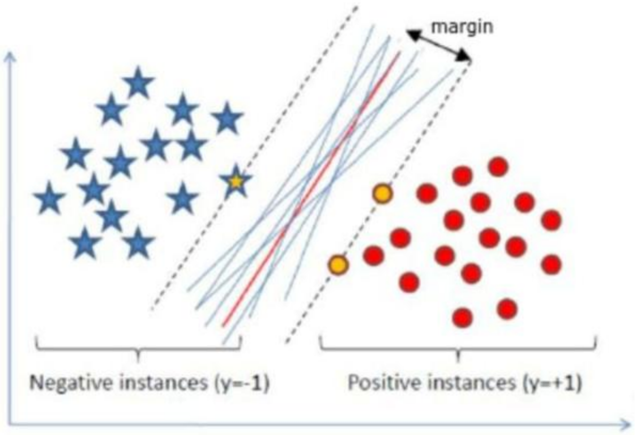


Figure: Multiple possible solutions, which one is better?

Maximum Margin Decision Hyperplane

- Recall the distance of a point to the hyperplane is

$$\frac{t_n y(\boldsymbol{x}_n)}{\|\boldsymbol{w}\|} = \frac{t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b)}{\|\boldsymbol{w}\|}$$

- To find maximum margin hyperplane, we just need to find the solution for

$$\arg\max_{\boldsymbol{w},b}\left\{\frac{1}{\|\boldsymbol{w}\|}\min_n\left[t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b)\right]\right\}$$

MMD: quadratic programming

Note that when we scaling $\boldsymbol{w}$ and b simutaneously, the distance is unchanged. Therefore, the previous problem is equivalvelent to

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

w.r.t

$$t_n(\boldsymbol{w}^T\boldsymbol{x}+b) \geq 1, \qquad n = 1, \ldots, N.$$

Equivalent Error Model

$$\sum_{n=1}^{N} E_\infty(y(\boldsymbol{x}_n)t_n - 1) + \lambda\|\boldsymbol{w}\|^2$$

where

$$E_\infty(z) = \begin{cases} 0 & z \geq 0 \\ \infty & 0 \end{cases}$$

MMDH: Lagrange Multipliers

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n \left[t_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - 1 \right]$$

take the derivative with respect to w and b, and set them to zero, we have

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{x}_n$$

$$0 = \sum_{n=1}^{N} a_n t_n$$

KKT conditions

$$a_n \geq 0$$

$$t_n y(\boldsymbol{x}_n) - 1 \geq 0$$

$$a_n \{ t_n y(\boldsymbol{x}_n) - 1 \} = 0$$

MMDH: Dual Representation

Eliminating $\boldsymbol{w}$ and b from $L(\boldsymbol{w}, b, \boldsymbol{a})$, the problem becomes to maximize

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{n=1}^{N} a_n a_m t_n t_m \boldsymbol{x}_n^T \boldsymbol{x}_m$$

with respect to a subject to the constraints

$$a_n \geq 0, \qquad n = 1, \ldots, N,$$
$$\sum_{n=1}^{N} a_n t_n = 0.$$

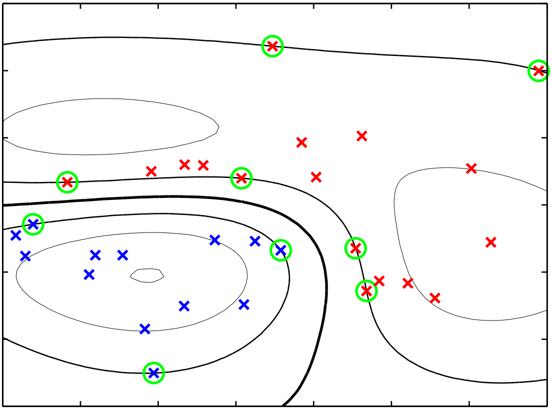
After solve the problem, the predict decision function could be written as

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n t_n \boldsymbol{x}^T \boldsymbol{x}_n + b$$

Kernel Trick

- The problem depends on $\boldsymbol{x_n}$ through the inner product $\boldsymbol{x}_n^T \boldsymbol{x}_m$
- If we can calculate the inner product of another feature space, we do not need to transform the feature actually
- Kernel function: compute the inner product of some transformed feature space using the current feature space
- The decision function will become $y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}, \boldsymbol{x}_n)$
- Requirements for kernel function
  - symmetry
  - positive semi-definite
- Methods to find kernel
  - Construct a function, prove its a kernel
  - Combine existing kernels

Overfitting

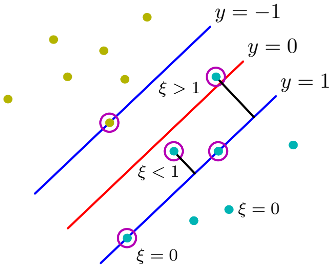


hinge error function

$$\sum_{n=1}^{N} [1 - y_n t_n]_+ + \lambda \|\boldsymbol{w}\|^2$$

- By not penalize misclassified points with ∞, we allow some points to be misclassified by the decision plane
- We also allow some points to fall into the decision margin

Soft Margin



We minimize

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\boldsymbol{w}\|^2$$

with subject to

$$t_n y(\boldsymbol{x}_n) \geq 1 - \xi_n, \qquad n = 1, \ldots, N$$

Soft Margin: Dual Representation

We minimize

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

with subject to

$$0 \leq a_n \leq C$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

Multiple Classes Classification

- No essential solution for multiple classes
- Needs high level strategies
- One-versus-the-Rest
  - Use K binary classifiers for K classes problem
- One-verus-One (pairwise)
  - Use $\frac{K(K-1)}{2}$ binary classifiers for K classes problem

SVR: Error Model

For regularized least square, we minimize

$$\frac{1}{2}\sum_{n=1}^{N}(y_n - t_n)^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$$

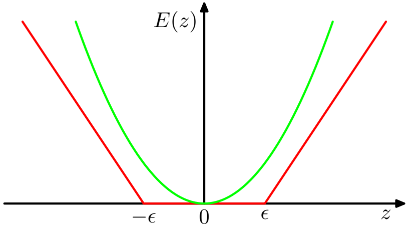
We replace the quadratic term with an ϵ-insenstive error function

$$E_\epsilon\left(y(\boldsymbol{x}) - t\right) = \begin{cases} 0 & \text{if} \quad |y(\boldsymbol{x}) - t| < \epsilon \\ |y(\boldsymbol{x}) - t| - \epsilon & \text{otherwise} \end{cases}$$

Then we minimize

$$E_\epsilon\left(y(\boldsymbol{x}_n) - t_n\right) + \frac{1}{2}\|\boldsymbol{w}\|^2$$

SVR: Error Model Illustration



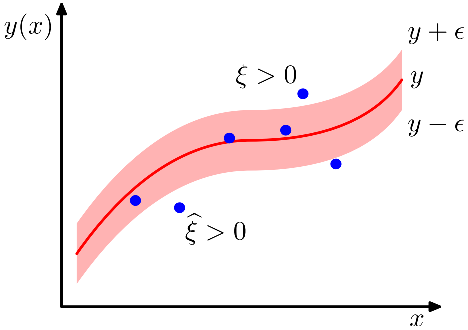
SVR: Optimization Problem

By introducing $\xi_n \geq 0$ and $\hat{\xi} \geq 0$, we can reformulate the problem as minimize

$$C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\boldsymbol{w}\|$$

with subject to

$$t_n \leq y(\boldsymbol{x}_n) + \epsilon + \xi_n$$
$$t_n \leq y(\boldsymbol{x}_n) - \epsilon - \hat{\xi}_n$$


$y(x)$
$y + \epsilon$
$\xi > 0$
y
$y - \epsilon$
$\hat{\xi} > 0$
x

SVR: Dual Representation

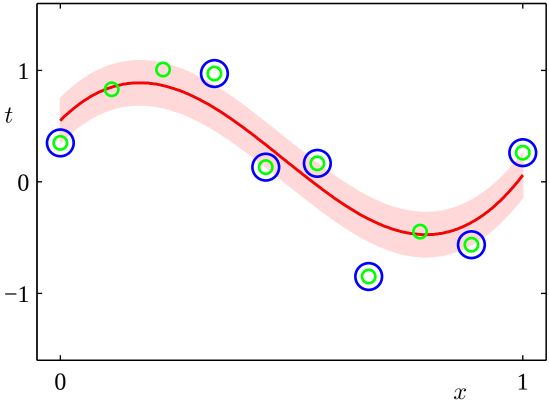
By manipulation using Lagrange multipliers, we can get the dual representation as to minimize

$$\begin{aligned}\tilde{L}(\boldsymbol{a}, \hat{\boldsymbol{a}}) = & -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \hat{a}_n)(a_m - \hat{a}_m)k(\boldsymbol{x}_n, \boldsymbol{x}_m) \\ & -\epsilon\sum_{n=1}^{N}(a_n + \hat{a}_n) + \sum_{n=1}^{N}(a_n - \hat{a}_n)t_n\end{aligned}$$

with subject to

$$0 \leq a_n \leq C$$
$$0 \leq \hat{a}_n \leq C$$

SVR: Example



Summary

- Error model of SVC: hinge function
- Error model of SVR: ϵ-insenstive error function
- Kernel tricks to extend a linear model to nonlinear one
- SVM is a kind of sparse kernel methods
- Well formulated optimization problem, thus can gurantte to find the global optimal
- practical tips
  - Always try to normalize $\boldsymbol{x}$
  - The parameter C, γ and ϵ should be sampled in log-space if we need to do a grid search

References

- Bishop: Pattern recognition and Machine Learning
- Vapnik: Statistical Learning Theory
- Vapnik: The Nature of Statistical Learning Theory

Figure: Vladimir Vapnik