Support Vector Machine – Concepts, Theory, Application to classifications

Kyungmin Kim

MATH F692 Mathematics for Machine Learning

3/03/2022

Presentation Outline

- What is Support Vector Machine (SVM)?
- Basic Concepts/ Theory regarding SVM and Classification
- Basic Python example
- Application of SVM to real-world examples
- SVM and optimization algorithm
- SVM and Perceptron comparison

What is Support Vector Machine?

- Introduced by Vapnik, 1995 paper as “Support vector networks”
- Supervised learning method for analyzing data for classification and regression.
- Getting Optimal way to separate data in the given number of class by labelled training data.
- It have generalizing capacity for using various application.
- It applies to both linear classification and non-linear case with kernel trick.

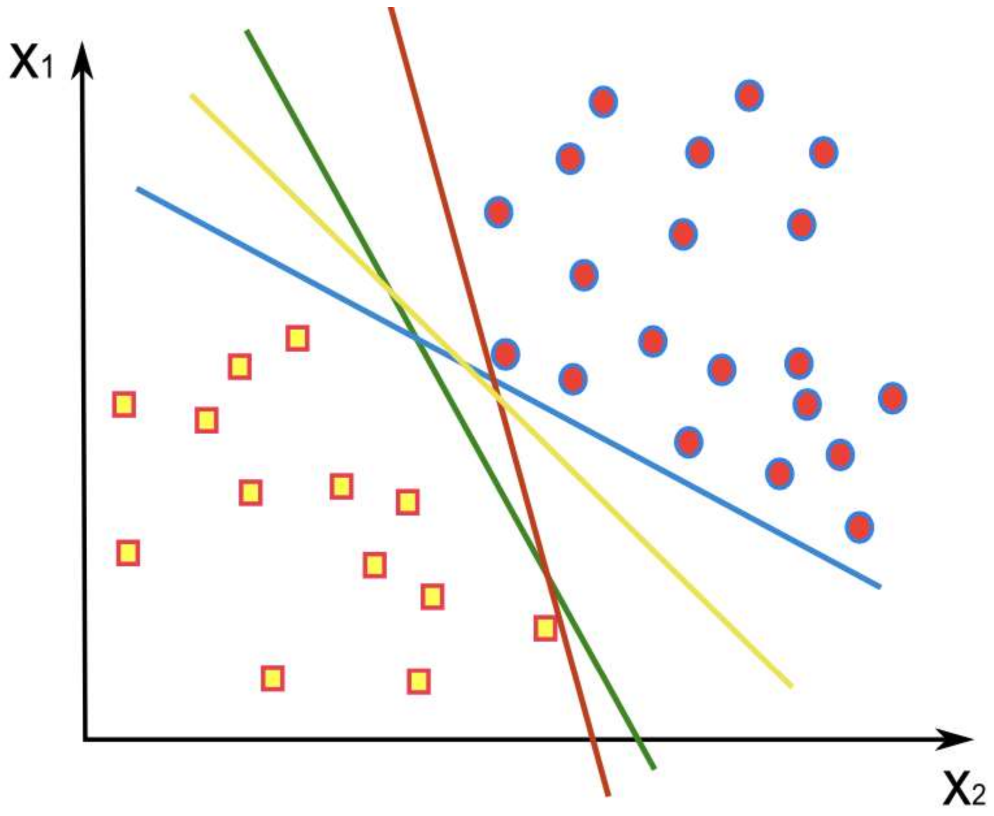
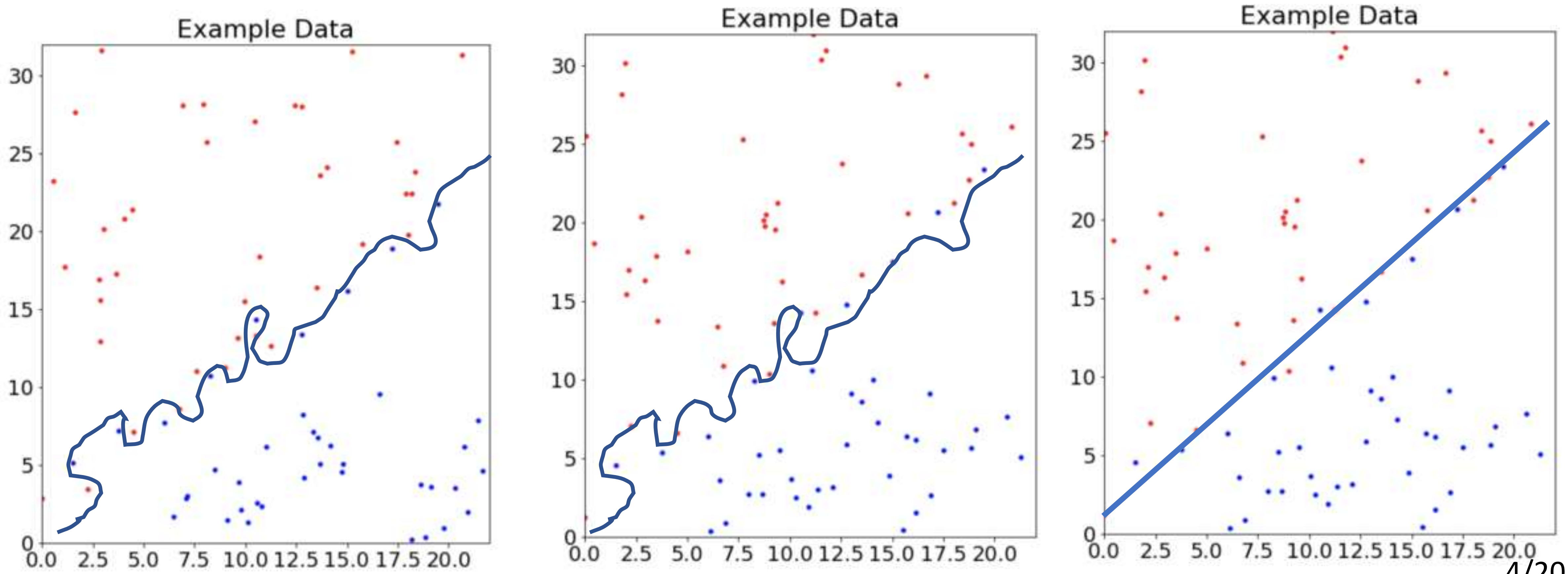


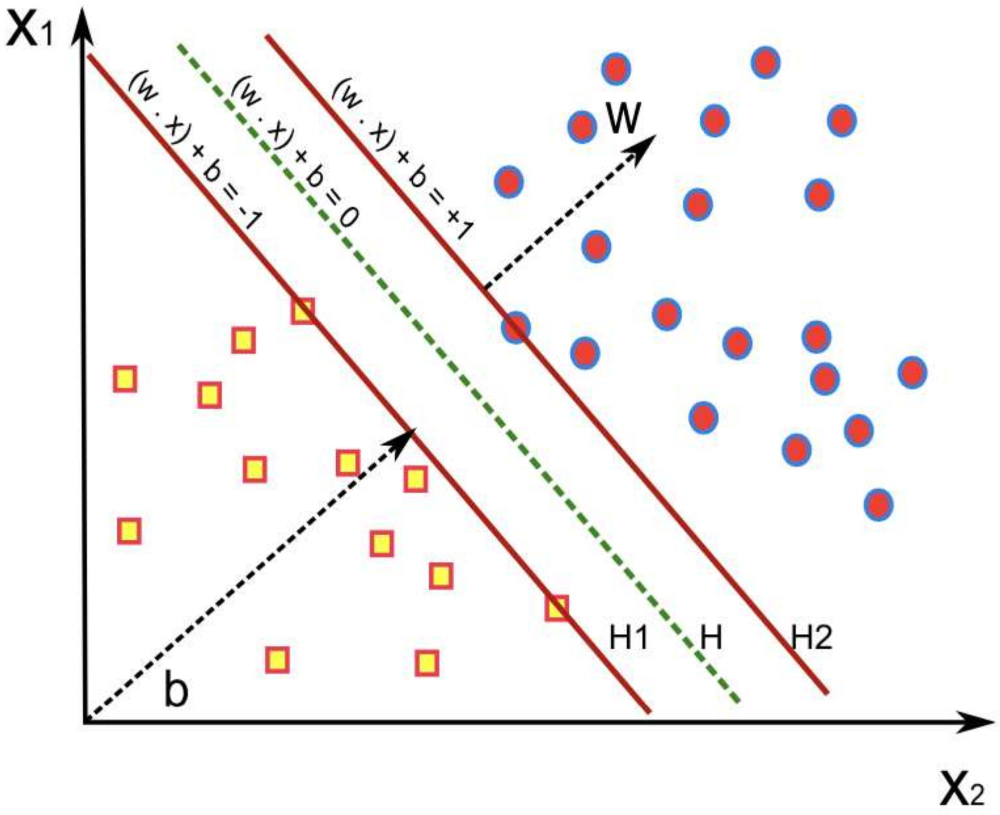
Fig. 1. Separation hyperplanes.

General objective of Classification

- Classification output should be based on training set, but it should not “memorize” training data set



Classification with linearly separable case



Optimizing the geometric margin means minimizing separating hyperplane

$$\langle w \cdot x^{+} \rangle + b = 1$$
$$\langle w \cdot x^{-} \rangle + b = -1$$

Geometric margin of x-y-x

$$\gamma_i = \frac{1}{2}\left(\left\langle \frac{w}{\|w\|} \cdot x^{+} \right\rangle - \left\langle \frac{w}{\|w\|} \cdot x^{-} \right\rangle\right)$$
$$= \frac{1}{2w}\left[\langle w \cdot x^{+} \rangle - \langle w \cdot x^{-} \rangle\right]$$
$$= \frac{1}{\|w\|}$$

w: optimal separation hyperplane
B: bias

Data lies in H1, H2 plane is “Support Vector”

Linearly Separable case and Lagrange Multiplier

For the Linearly Separable case

$S = [(x_1, y_1) \cdots (x_N, y_N)]$

Solution

$\min\langle w \cdot w\rangle = \|w\|^2$

Subject to : $y_i(\langle w \cdot x_i\rangle + b) \geq 1$

Then the maximal margin is given by $\gamma = \frac{1}{\|w\|}$

Change solution to dual problem using the Lagrange formula

$L(w, b, \alpha) = \frac{1}{2}\langle w \cdot w\rangle - \sum_{i=1}^{N} \alpha_i[y_i(\langle w \cdot x_i\rangle + b - 1)]$

α_i: Lagrange Multiplier

Lagrange Multiplier : Strategy point to find local minimum and maximum point in equally constraints

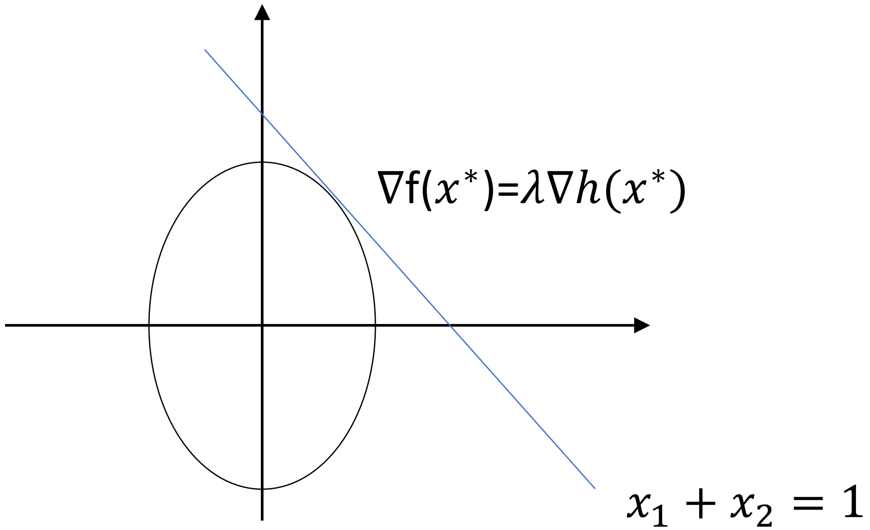
Minimize $2x_1^2 + x_2^2$

Subject to : $x_1 + x_2 = 1$

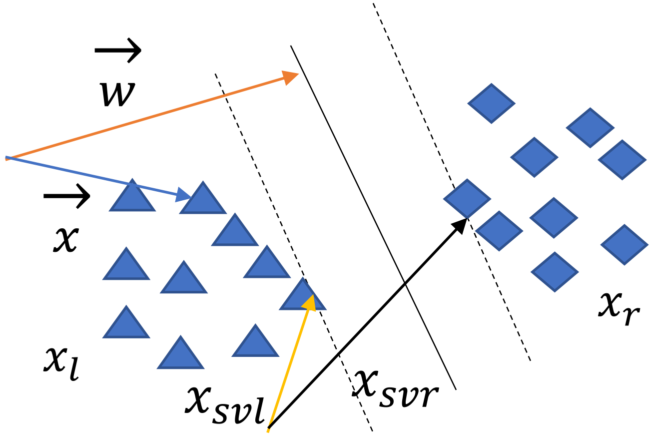
$\frac{df}{dx_1} = 4x_1$, $\frac{df}{dx_2} = 2x_2$

$L(x_1, x_2, \lambda) = 2x_1^2 + x_2^2 + \lambda(1 - x_1 - x_2)$

$\lambda = \frac{4}{3}$ -> get solution of $x_1 = \frac{1}{3}, x_2 = \frac{2}{3}$



Example of Lagrange Multiplier and support vector machine



$\vec{w} \cdot \vec{x} + b \geq 0$

$\vec{w} \cdot \vec{x_r} + b \geq 1; \vec{w} \cdot \vec{x_l} + b \leq 1$

$y_i = \begin{cases} -1\ for\ x_r \\ +1.\ for\ x_l \end{cases}$ For generalize the equation

$y_i \cdot (\vec{w} \cdot \vec{x_r} + b)$-1≥ 0

Width of Dashed line $\frac{2}{|w|}$

Hyperplane between two dataset

Minimize $\frac{1}{2}|w|^2$

Lagrangian we want to minimize

$$\text{L} = \frac{1}{2}|w|^2 - \sum_i \alpha_i \left(y_{svi} \cdot \left(\vec{w} \cdot \vec{x}_{svi} + b \right) - 1 \right)$$

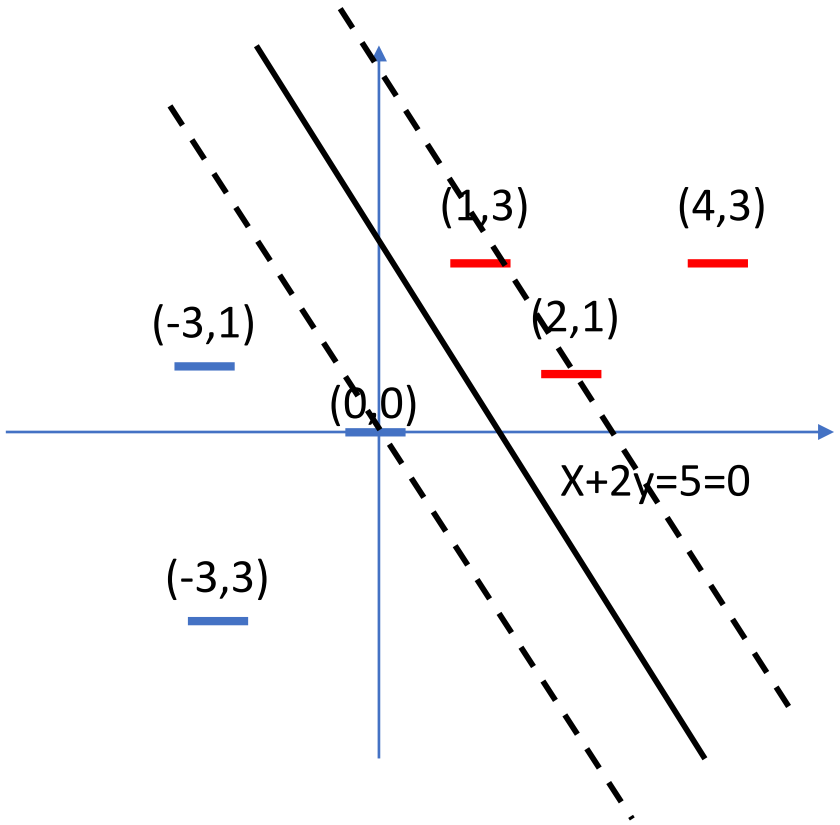
α_i: *Lagrangian Multiplier*

Expanding the expression

$$\text{L} = \frac{1}{2}|w|^2 - \sum_i \alpha_i \left(y_{svi} \cdot \left(\vec{w} \cdot \vec{x}_{svi} + b \right) \right) + \alpha_i$$

$$\text{L} = \sum_i \alpha - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \overrightarrow{x_{SV}}\, \overrightarrow{x_{SV}}$$

Example of Lagrange Multiplier and support vector machine



Point (0,0), (2,1) and (1,3) closest to the separating hyperplane, and work as "support vector"

$$\alpha_i = 1\ for\ (0,0)$$
$$\alpha_i = 0.5\ for\ (1,3),(2,1)$$

Point far away from plane (e.g. (-3,3)) are not work as support vector

Inequaility constraints and Karush-Kuhn-Tucker Conditions

Karush-Kuhn-Tucker condition (KKT)

- Condition to obtain an optimal solution to a general optimization problem.

Minmize f(w), w$\in\Omega$

$$g_i(w) \le 0, i = 1, \dots, k$$
$$h_i(w) = 0, \mathrm{i} = 1, \dots, \mathrm{m}$$

Necessity and sufficient conditions for a normal point w^* to be optimal are the existence of α, β such that

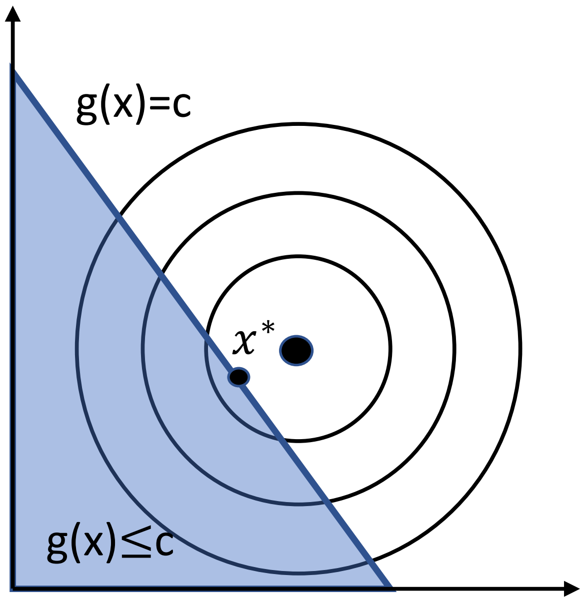
$$\frac{\partial L(\alpha^*, \beta^*, w^*)}{\partial w} = 0$$
$$\frac{\partial L(\alpha^*, \beta^*, w^*)}{\partial \beta} = 0$$
$$\alpha_i g_i(w^*) = 0$$
$$g_i(w^*) \le 0$$
$$\alpha_i^* \ge 0$$

From KKT condition if training set linearly separable

$$\|w\|^2 = \langle w^* \cdot w^* \rangle = \left(\sum \alpha_i\right)^{-1/2}$$

L(x) is Lagrangean Function

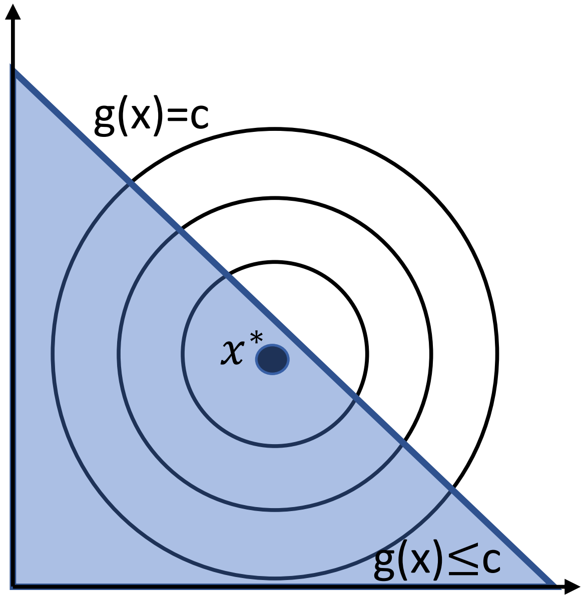
L(x)=f(x)-$\alpha_i(g_i(x) - c)$



If g(x^*)=c, $L_i'(x^*)$=0

$\alpha_i \ge 0$ in this case

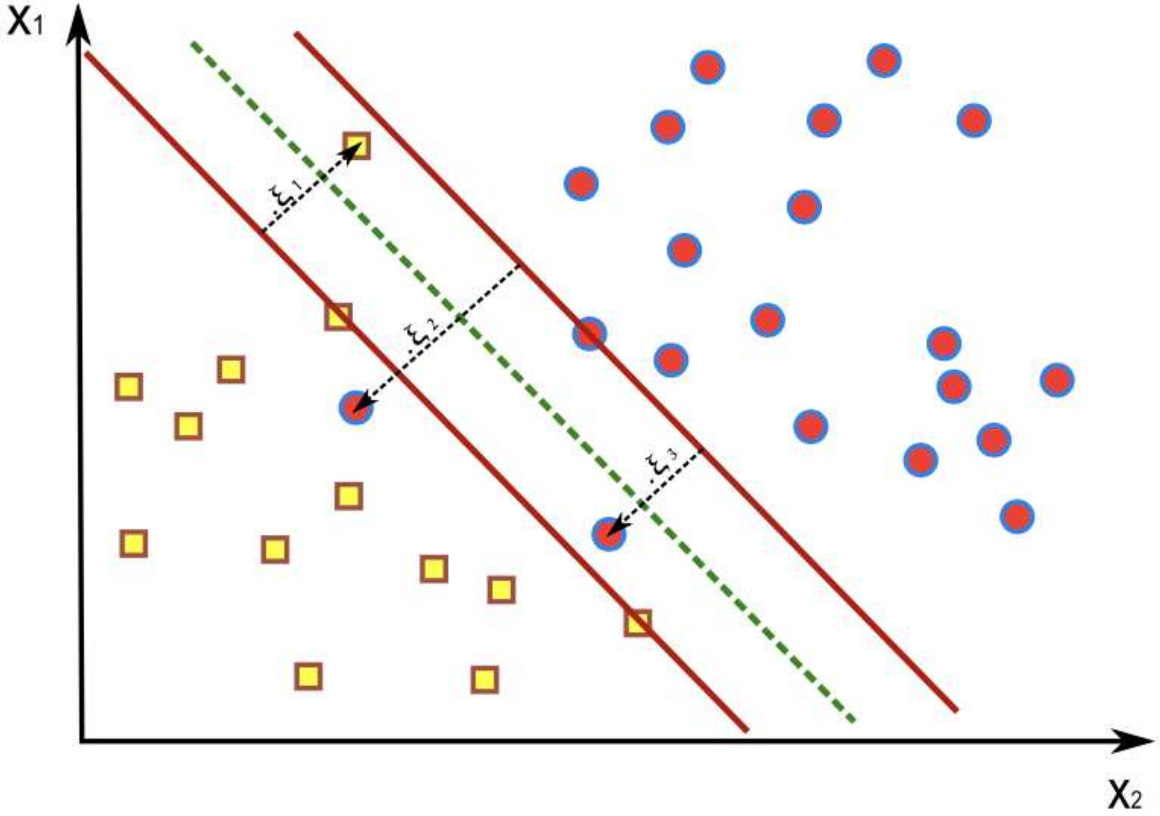
If $\alpha_i < 0$ small decrease in x increase the value of f



If g(x^*)<c, $f_i'(x^*)$=0

α_i does not enter the condition and can set α_i = 0

Soft Margin Hyperplane



For linearly separable case

$$y_i(\langle w \cdot x_i \rangle + b) \geqslant 1$$

Allow ζ term to make data linearly separable

$$y_i(\langle w^T \cdot x_i \rangle + b) \geqslant 1 - \xi_i \quad \forall i$$

Width of margin can be determined by penalty parameter C

$$\min_{w,b,\xi_i} \langle w \cdot w \rangle + C \sum_{i=1}^{l} \xi_i^2$$

$$\text{s.t.} : y_i(\langle w \cdot x_i \rangle + b) \geqslant 1 - \xi_i$$

$$\xi_i \geqslant 0$$

Subject to

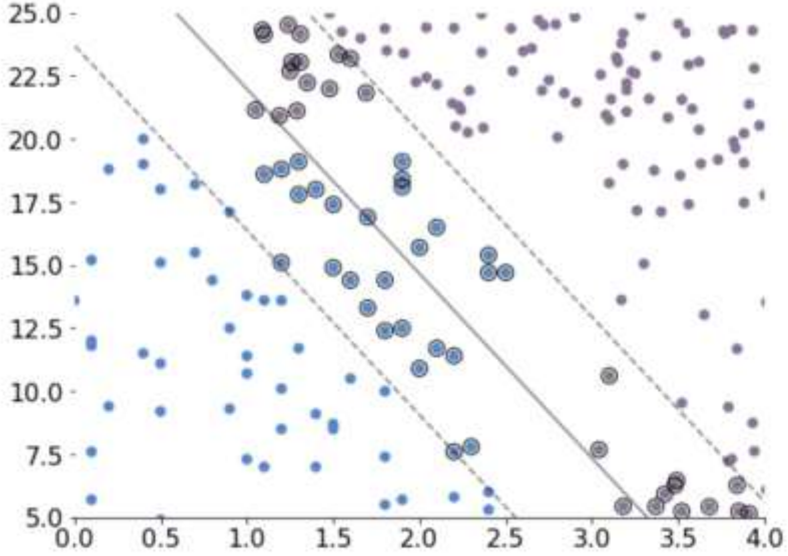
$$\langle w \cdot x_i \rangle + b \geqslant +1 - \xi_i, y_i = +1, \xi_i \geqslant 0$$
$$\langle w \cdot x_i \rangle + b \leqslant -1 + \xi_i, y_i = -1, \xi_i \geqslant 0$$

Python example of Soft Margin Hyperplane

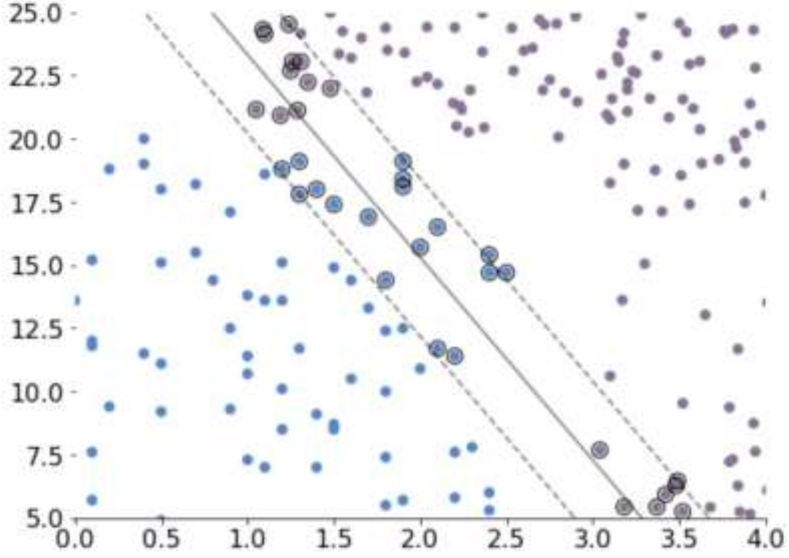
```python
model = svm.SVC(kernel='linear',C=100)
model.fit(x_train, y_train)
```

Degree of “soft margin” determined by regularization factor C

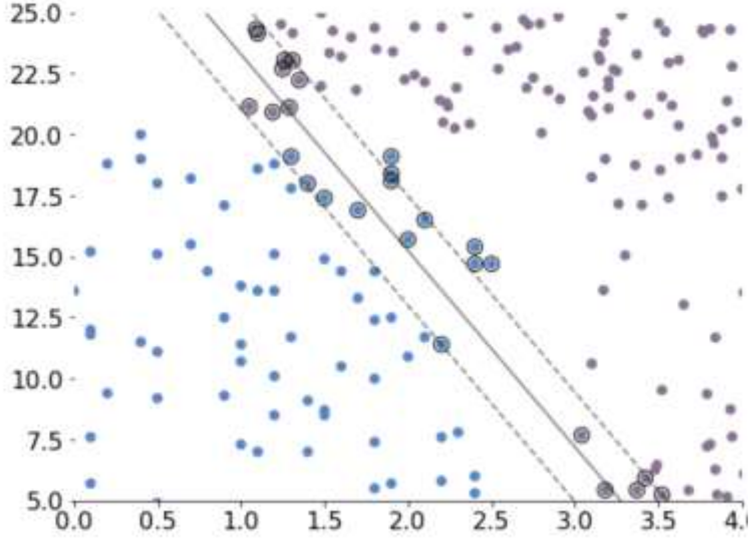
C= 0.1



C= 1.0



C= 100



Kernel function

Basic Idea :
Input vector x∈ R^n project to higher dimensional feature space

$$x \in \mathbb{R}^n \rightarrow \Phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_n(x)]^T \in \mathbb{R}^f$$

Hypothesis considered to be

$$f(x) = \sum_{i=1}^{l} w_i \phi_i(x) + b$$

For Kernel function K, each x,z∈K

$$K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle$$

Kernel function must obey following property

1. $x \cdot x = 0$ only if $x = 0$
2. $x \cdot x > 0$ otherwise
3. $x \cdot y = y \cdot x$
4. $(\alpha x \cdot y) = \alpha(x \cdot y)$
5. $(z + x) \cdot y = (z \cdot y) + (x \cdot y)$

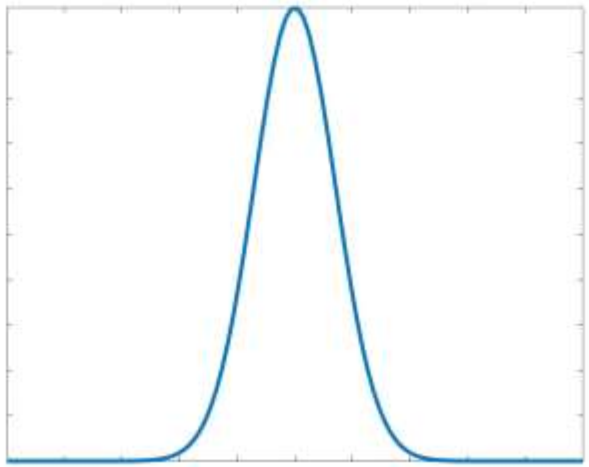
Polynomial Kernel : K(x_i, x_j)=($x_i \cdot x_j + 1$)^p

Gaussian Kernel : K$\left(x_i, x_j\right) = e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$

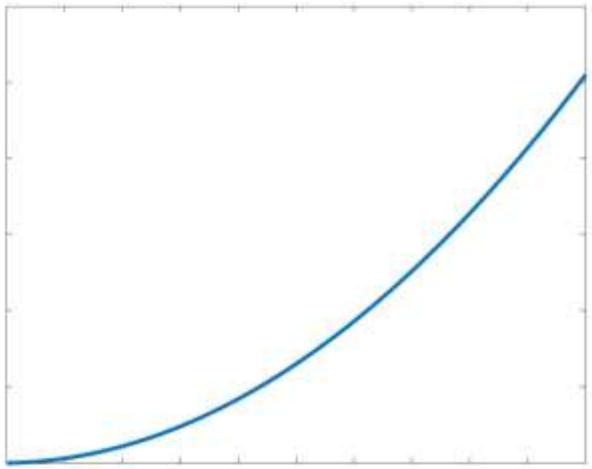
RBF Kernel : K$\left(x_i, x_j\right)$=$e^{-\gamma(xi - xj)^2}$

Sigmoid kernel : K(x_i, x_j)=tanh($\eta x_i \cdot x_j + \iota$

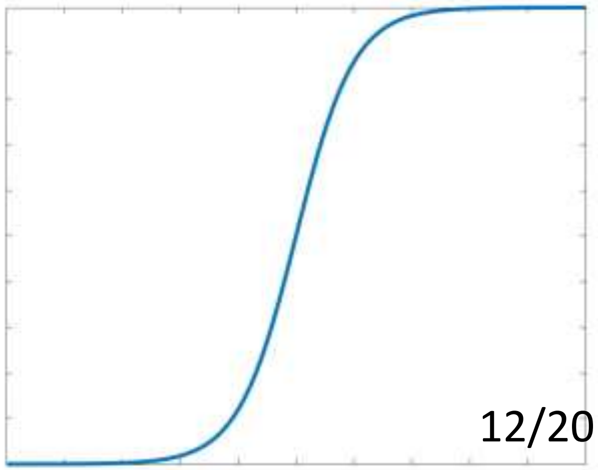
RBF Kernel Function



Polynomial



Sigmoid Kernel Function



Mercer's Theorem

Condition of a function to be kernel

A symmetric continuous function

$$K : [a,b] \times [a,b] \rightarrow \mathbb{R}$$

K is positive semi-definite if and only if

$$\sum_{i=1}^{n}\sum_{j=1}^{n} K(x_i, x_j) c_i c_j \geq 0$$

For all finite sequence of points $x_1, x_2 \ldots, x_N$
And all choice of real number $c_1, c_2, \ldots, c_N$

Associated to K is a linear operator on function defined by

$$[T_{K\varphi}](x) = \int_a^b K(x,s)\varphi(s)ds$$

For Mercer's theorem, any symmetric feature that maps a feature space of x times x, it is square integrable on its domain and satisfies its integral.

Mercer's condition

Given a finite input space X = $\{x_1, x_2 \ldots, x_n\}$ and real-valued function K=$(K(x_i \cdot x_j))_{i,j=1}^{n}$

$$\iint g(x)K(x,y)g(y)dxdy \geq 0$$

Example

A positive constant function K(x,y)=c

$\iint g(x)cg(y)dxdy$= c$\int g(x)dx \int g(y)dy$ =c$(\int g(x)dx)^2$

Gaussian kernel function

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

$$e^{-\frac{1}{2\sigma^2}(x_i - x_j)^2} = e^{\frac{-x_i^2 - x_j^2}{2\sigma^2}}(1 + \frac{2x_i x_j}{1!} + \frac{(2x_j x_j)^2}{2!} + \cdots)$$

= $\phi(x_i)^T \phi(x_j)$

Weakness of Support Vector Machine

- Excessive computation cost in large datasets because kernel matrix grow quadratic form with the size of data
- SVM designed to solve binary problems – multi-classification case are necessary to change as multiple binary classification problem.
- Difficult to get optimal separation hyperplane for an SVM trained with imbalance data.

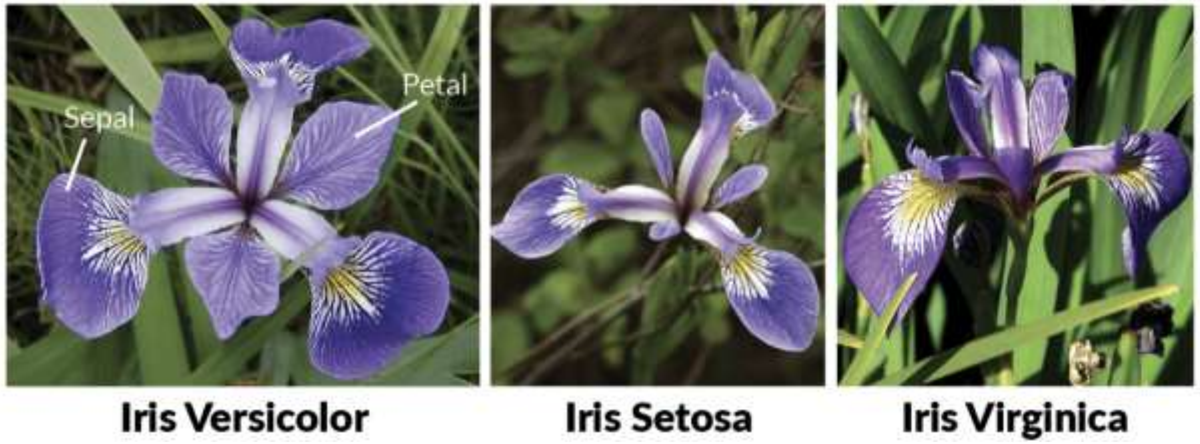
➢ Including synthetic minority data to improve classification accuracy (Koknar-Tezel and Latecki 2009)

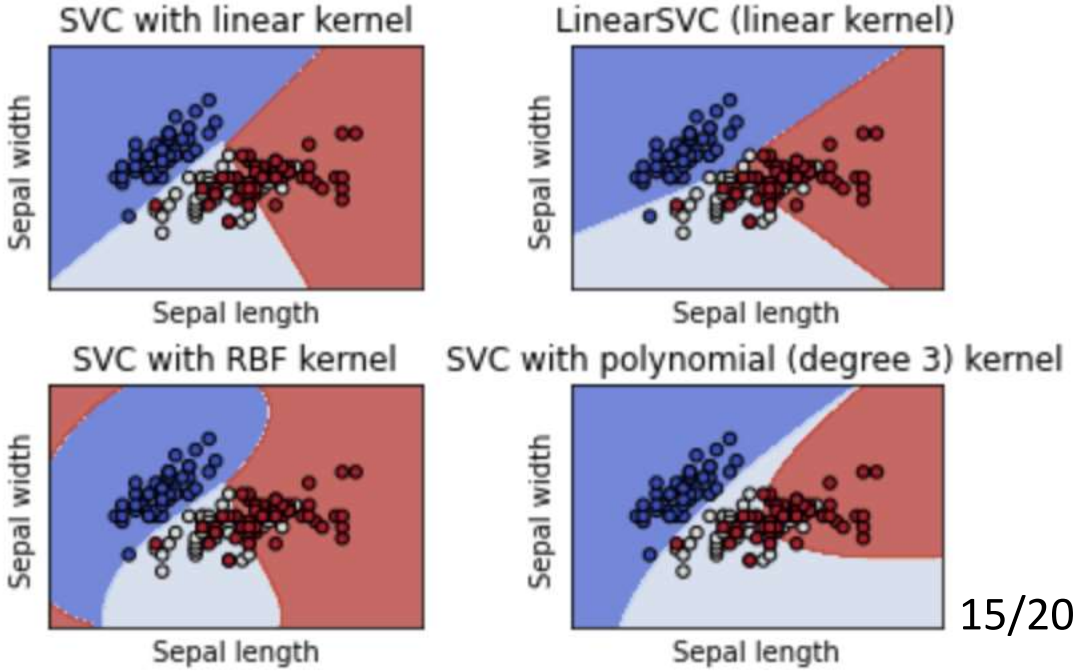
➢ "Weighted" SVM algorithm to overcome imbalance data set (Du and Chen 2005)

Python classification by Support Vector machine

```
models = (
    svm.SVC(kernel="linear", C=C),
    svm.LinearSVC(C=C, max_iter=10000),
    svm.SVC(kernel="rbf", gamma=0.7, C=C),
    svm.SVC(kernel="poly", degree=3, gamma="auto", C=C),
)
models = (clf.fit(X, y) for clf in models)
```

Three species of IRIS





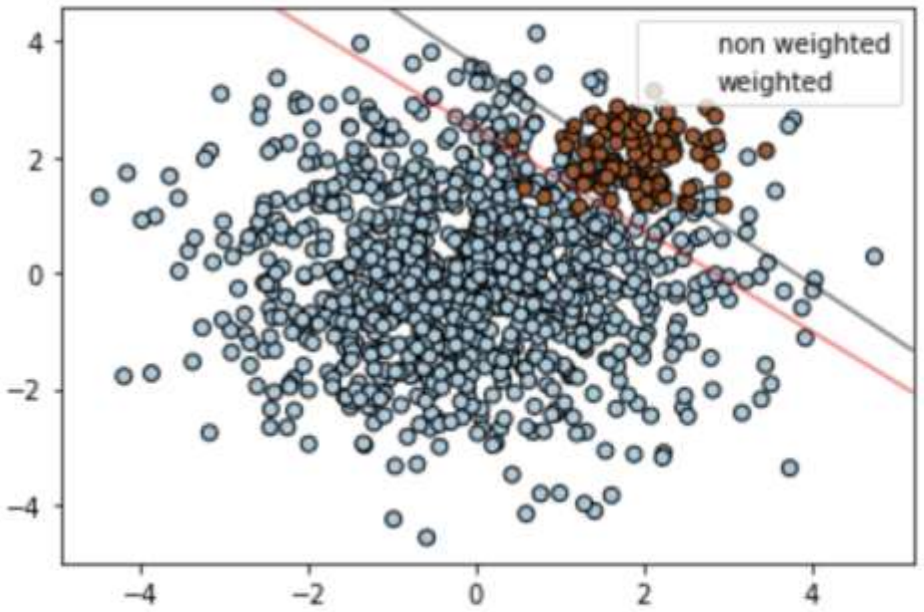
Source : https://scikit-learn.org/stable/modules/svm.html

Support Vector Machine with Imbalanced dataset

Get different Hyperplane by weighting imbalanced data.

```
# fit the model and get the separating hyperplane
clf = svm.SVC(kernel="linear", C=1.0)
clf.fit(X, y)

# fit the model and get the separating hyperplane
# using weighted classes
wclf = svm.SVC(kernel="linear", class_weight={1: 10})
wclf.fit(X, y)
```



Source : https://scikit-learn.org/stable/modules/svm.html

```
# get the separating hyperplane
Z = clf.decision_function(xy).reshape(XX.shape)

# plot decision boundary and margins
a = ax.contour(XX, YY, Z, colors="k", levels=[0], alpha=0.5, linestyles=["-"])

# get the separating hyperplane for weighted classes
Z = wclf.decision_function(xy).reshape(XX.shape)

# plot decision boundary and margins for weighted classes
b = ax.contour(XX, YY, Z, colors="r", levels=[0], alpha=0.5, linestyles=["-"])
```

SVM application to Volcano-seismicity classification

LP : Volcanic Long Period events (associated with resonance of fluid-filled crack)

VT : Volcano tectonic Earthquake (associated with brittle fracture of rock)

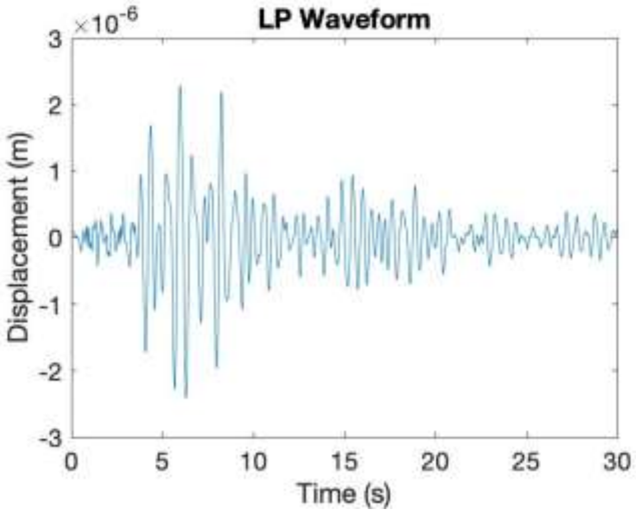
TC : Tectonic Earthquake

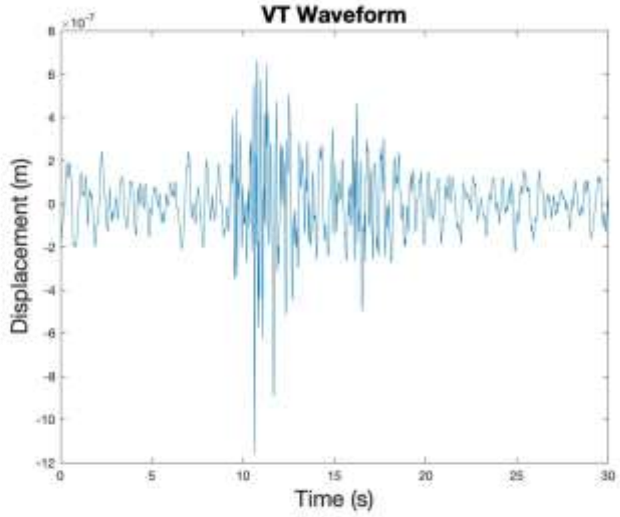
OT : Other signal (e.g. melting glacier, storm etc)

(Curilem et al. 2014)

Characterizing Different waveform features of LP and VT waveform for testing -> Auto-classification SVM (Malfante et al. 2018)

Different Type of Signal from Great Sitkin





Optimization algorithm

What is appropriate optimization algorithm for solving SVM?

First-Order Algorithm

- Gradient Descent
- Momentum
- Adagrad
- RMSProp

Second-Order Algorithm

- Newton's method
- Secant Method
- Quasi-Newton Method

For Non-differential Object Function

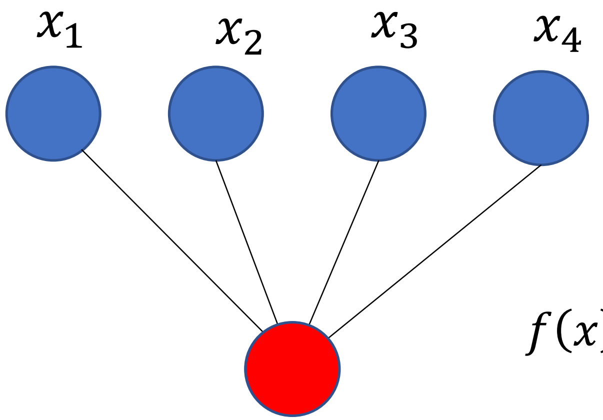
- Direct Algorithm
- Stochastic Algorithm
- Population Algorithm

Direct algorithm : Navigate the pattern search as they navigate the space using geometric shape or decision (e.g. pattern)

Stochastic Algorithm: Algorithm that make use of randomness of search procedure

Population Algorithm: Algorithm maintaining a pool of candidate solution are used sample, explore an optima.

Single layer Perceptron and compared to SVM



$$f(x) = \sigma(\sum_{i=1}^{n} w_i x_i + b)$$

The output of the neuron is a linear combination of the inputs

Argument X:= $\{x_1,\ldots x_m\} \subset$ x (data)
Y:= $\{x_1,\ldots x_m\} \subset \{\pm 1\}$ (label)
Function(w,b) = Perceptron(X,Y)
  initialize w,b=0
  repeat
    Pick (x_i, y_i) from data
    if yi(w*xi+b)≤ 0 Then
      $w' = w + y_i x_i$
      $b' = \mathrm{b} + y_i$
Until yi(w*xi+b) >0 for all i

Source : http://alex.smola.org/teaching/pune2007/pune_3.pdf

Perceptron : Update classification for each iteration, different from SVM

SVM : Get the optimal "hyperplane" between classification data. For perceptron, the classification cannot be optimal out of data

Single-layer perceptron : Form of SVM?

Summary

- Support Vector Machine is useful for learning method of classification
- It aims to get optimal hyperplane to classify data in feature space.
- Hyperplane between different category of data can be get from linearly separate case.
- Soft Margin of Hyperplane, Kernel trick(method) is used for more complex real-world data
- Appropriate optimization algorithm is needed for SVM and classification problem
- SVM can be interpreted as Single-layer perceptron