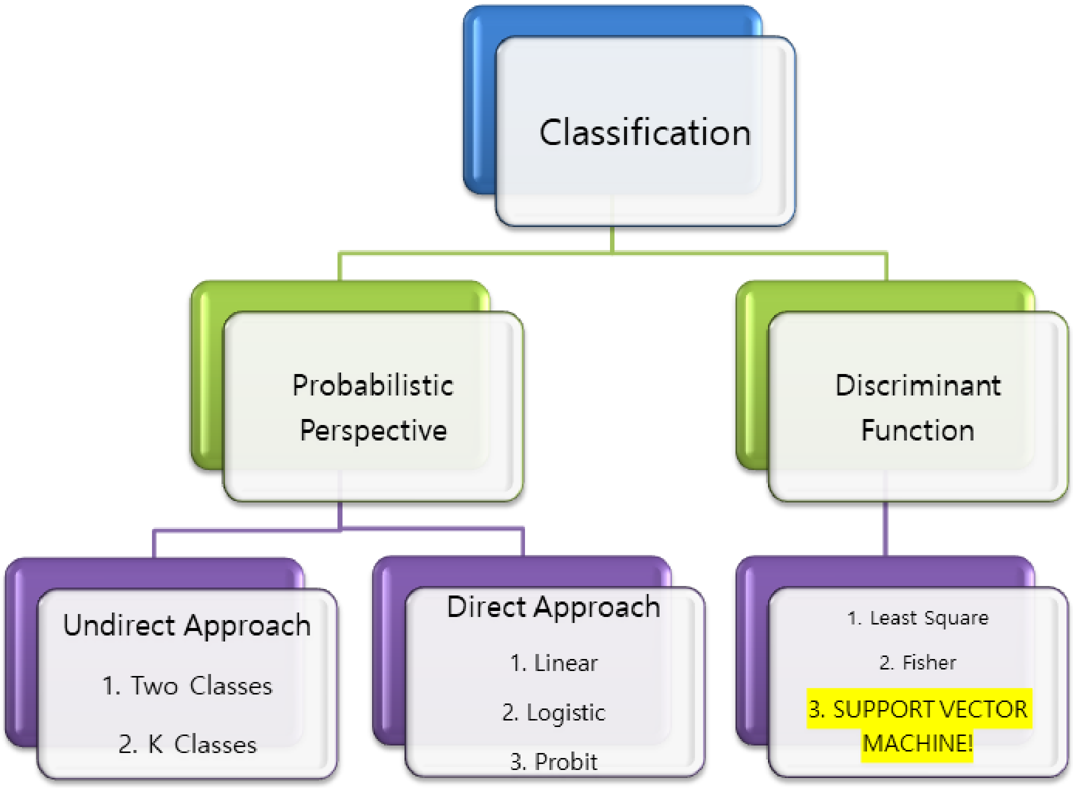
Week 4. Kernel and Support Vector Machine

Damian(Sunwoo) Lim/ ESC YONSEI UNIVERSITY

4/22/2020

1. Where are we?



SVM : No consideration of $p(C_k|X)$

2. Hard SVM without Expansion of Basis

2-1. Good Discriminant Function : Halfway Down!

Let's Figure Out so called "Good Linear Discriminant Function"

"Target" $t(=y)$: $t_n = 1or - 1$: labeled!

"Estimator" of t" : $y(x) = w^T x + b$

"Class 1" : y(x) = $w^T x + b > 0$: One side of the input space

"Class 2" : y(x) = $w^T x + b < 0$: The other side of the input space

"Discriminant Function" : $y(x) = w^t x + b = 0$: Discriminant Function

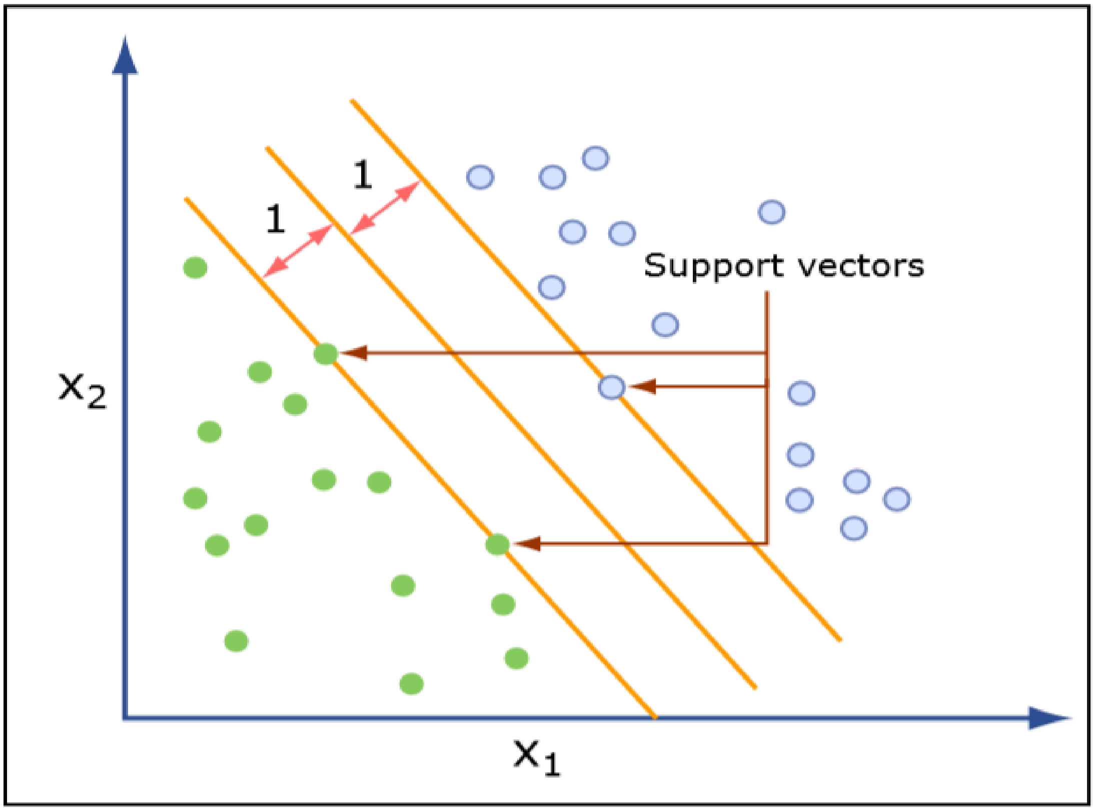
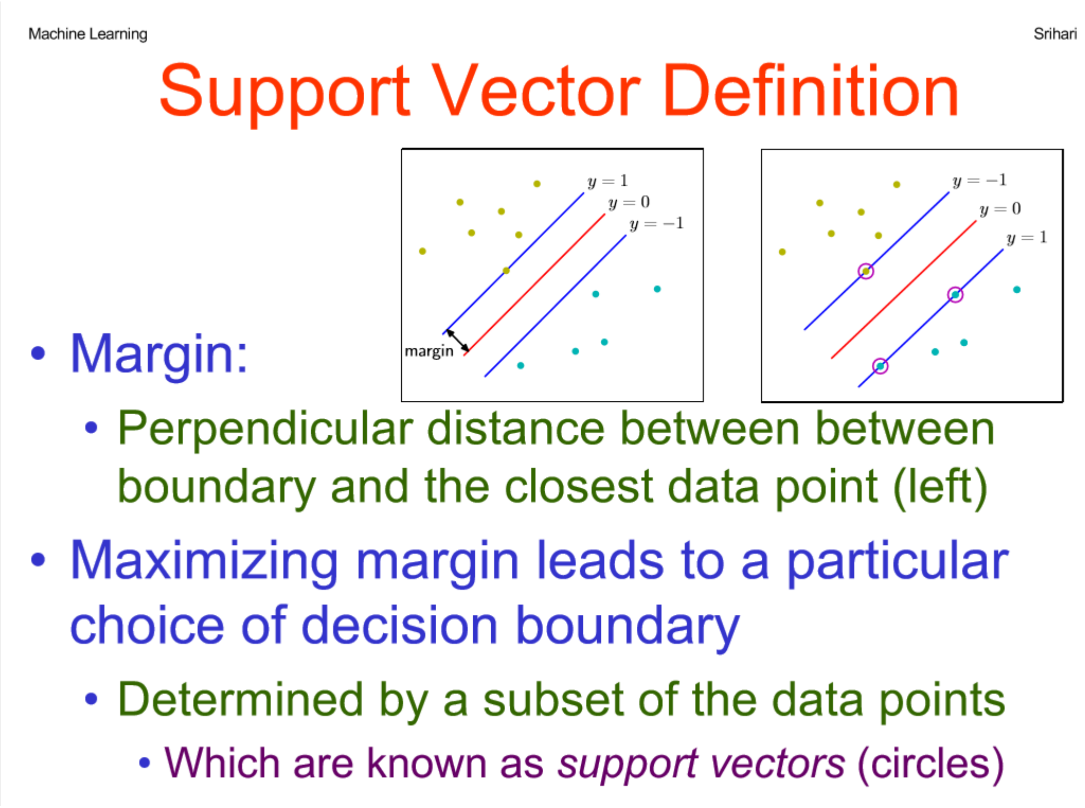


Image by MIT OpenCourseWare.

2-2. Hard Margin



Machine Learning Srihari

Support Vector Definition

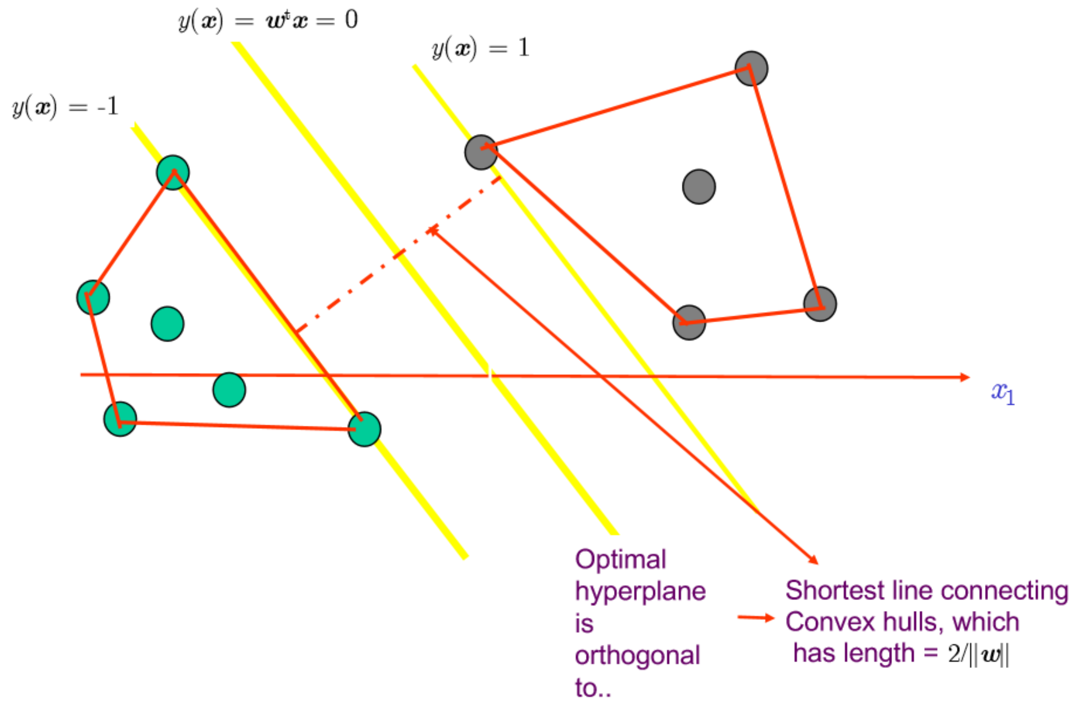
- Margin:
  - Perpendicular distance between between boundary and the closest data point (left)
- Maximizing margin leads to a particular choice of decision boundary
  - Determined by a subset of the data points
    - Which are known as *support vectors* (circles)

Previously, we've set the boundary as y(x) = 0 so that

- **Class1** : y(x) = $w^T x + b > 0$: One side of the input space
- **Class2** : y(x) = $w^T x + b < 0$: The other side of the input space.

  But, when we build models, we consider the Margin so that we standardize…

- $C_1: w^T x + b \geq 1, C_2: w^T x + b \leq$ -1. Think of 1 and -1 as standardization.
- **Support Vector** : Vectors that satisfy **Equality.**



Optimal hyperplane is orthogonal to.. → Shortest line connecting Convex hulls, which has length = $2/\|\boldsymbol{w}\|$

Like this, we can think as **margin** as

Distance of $w^T x + b = 0$ and $w^T x + b = 1$,

Distance of $w^T x + b = 0$ and $w^T x + b = -1$.

We will maximize two Margins. In fact, no matter where we set the discriminant function $w^T x + b = k$, k bounded between -1 and 1, the sum of margins is the same.

Instead, we are maximizing the Sum of Squares of two Margins so that we are setting the discriminant function HALFWAY DOWN.

2-3. Maximum Hard Margin Classifier

2-3-1. Problem Setting

- Till now, **Input Space** was 2 dimensional, **Discriminant Function** was 1 dimensional.
- Genearlize! If **Input Space is p dimensional, Discriminant is a (p-1) dimensional hyperplane!**
- Thus, we have to figure out how to set the discriminant function so that **the distance between Discriminant Function and Support Vector is maximized!**
- **Spoiler Alert: Maximize the distance between the Discriminant Function and Support Vectors.**
- A Brief Review of distance of a point and a plane (Linear Algebra)

Result : A formula including Absolute Value.

2-3-2. Get rid of the Absolute Value

Here, there is an **Absolute Value** at the numerator! (not differentiable) Need Brilliance to get rid of Absolute Value to make it a **Differentiable Optimization Problem** !

Solution : multiply by $t_n \in (-1,1)$

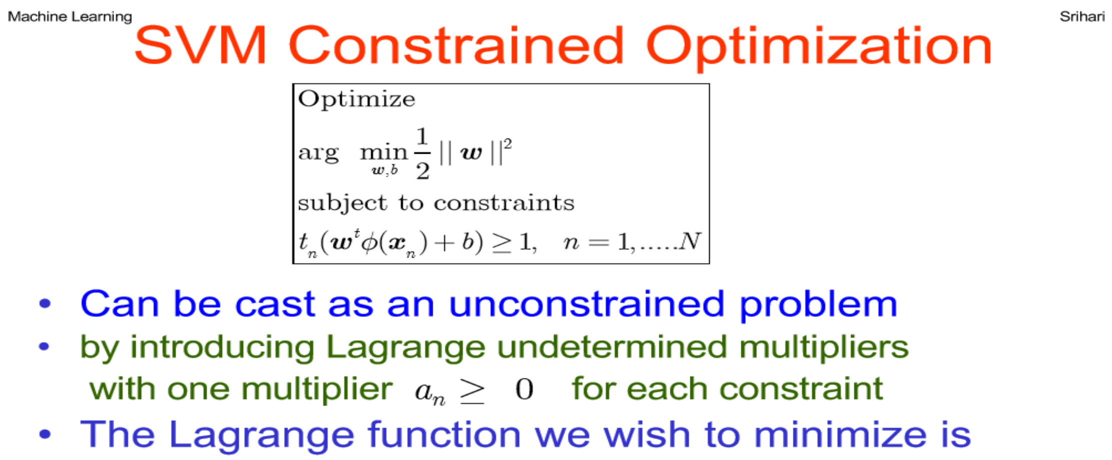
- $t_n = 1$ (adult, conservative,..) : Want $w^T X_n + b \geq 1$ ← For perfect Sample classification
- $t_n = -1$ (children, democratic, ..) : Want $w^T X_n + b \leq 1$ ← For perfect Sample classification
- $t_n(w^T(x_n) + b) \geq 1, \forall n \in (1,2,\ldots,N)$
- $\therefore Margin = min \frac{t_n(w^T(x_n)+b)}{\|w\|}$
- So, the Maximum Margin Solution is : $argmax_{w,b} min \frac{t_n(w^T(x_n)+b)}{\|w\|}$

2-3-3. Simplify this argmax solution

- Still hard to calculate. Brilliance Required again!
- 2 Jargons that help solve this problem : *Active*, *Inactive*
- Active points : $t_n(w^T(x_n) + b) = 1$ = Support Vectors
- Inactive points : $t_n(w^T(x_n) + b) > 1$ = Non- Support Vectors
- In each Class, at least one Active Point! → At least two Active Points total!
- Why is it important? Active Points minimize $t_n(w^T(x_n) + b)$ as merely 1 !

2-3-4. So?

- $\therefore Margin = \frac{1}{\|w\|}$
- ∴ Constrained Optimization Problem:
- $argmax_{w,b} \frac{1}{\|w\|}$ such that $t_n(w^T(x_n) + b) \geq 1, \forall n \in (1,2,\ldots,N)$
- $\frac{1}{\|w\|}$ maximization $\equiv \frac{1}{2} \| w \|^2$ minimization (Ease of Computation)!
- min $L(w, b, \lambda) = \frac{1}{2} \| w \|^2 - \sum_{n=1}^{N} \lambda_n \left[t_n(w^T X_n + b) - 1\right], \lambda_n \geq 0, \forall n$



Machine Learning

Srihari

SVM Constrained Optimization

Optimize

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

subject to constraints

$$t_n(\boldsymbol{w}^t \phi(\boldsymbol{x}_n) + b) \geq 1, \quad n = 1, \ldots .N$$

- Can be cast as an unconstrained problem
- by introducing Lagrange undetermined multipliers with one multiplier $a_n \geq 0$ for each constraint
- The Lagrange function we wish to minimize is

2-3-5. KKT Condition :Prerequisite for this Sequential Lagrange Optimization

1) $\frac{\partial L}{\partial w} = 0: w = \sum_{n=1}^{N} \lambda_n t_n x_n$

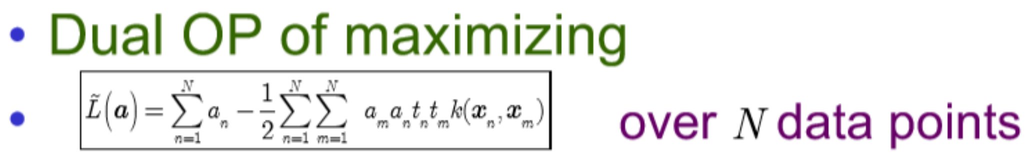
2) $\frac{\partial L}{\partial b} = 0: \sum_{n=1}^{N} \lambda_n t_n = 0$

3) $\forall n, \lambda_n \geq 0$

4) $\forall n, \lambda_n = 0$ or $t_n(w^T(x_n) + b) = 1$(Support Vectors)

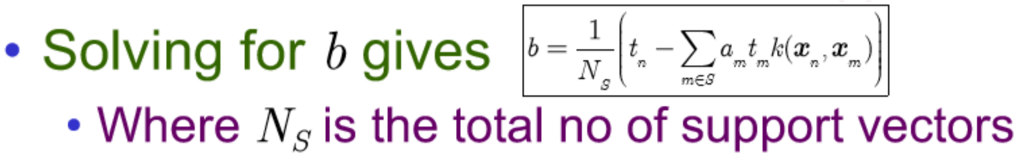
This task satisfies the KKT Condition. Changed into Dual Optimization Problem of Maximization (Too hard to handle for undergrads..)

2-3-6. Solve the equation



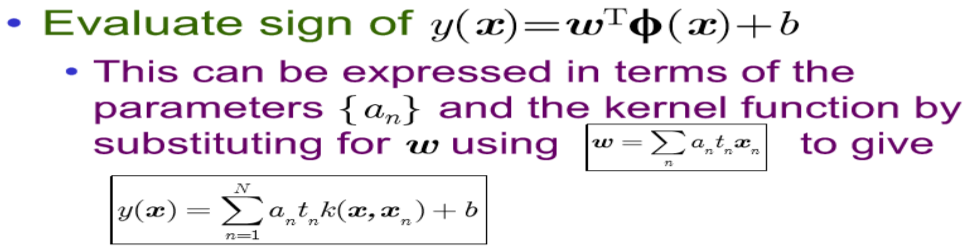
- Dual OP of maximizing
- $\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_m a_n t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ over N data points

Likewise, can get the solution for b.



- Solving for b gives $b = \frac{1}{N_S}\left(t_n - \sum_{m \in S} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)\right)$
  - Where N_S is the total no of support vectors

Put these Solutions into the formula so that the **Prediction** relies on the sign of $y(x)$!

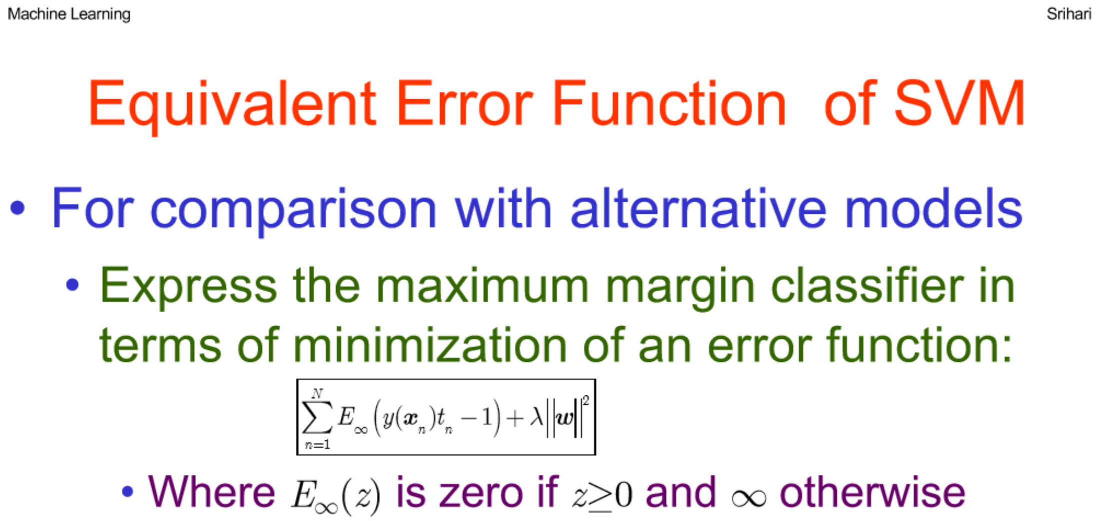


- Evaluate sign of $y(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}) + b$
  - This can be expressed in terms of the parameters $\{a_n\}$ and the kernel function by substituting for $\boldsymbol{w}$ using $\boxed{\boldsymbol{w} = \sum_n a_n t_n \boldsymbol{x}_n}$ to give

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

2-4. Error Function of Hard Margin Classifier

Supervised Machine Learning : The concept of Error has to be addressed!



Machine Learning

Srihari

Equivalent Error Function of SVM

- For comparison with alternative models
  - Express the maximum margin classifier in terms of minimization of an error function:

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

  - Where $E_{\infty}(z)$ is zero if $z \geq 0$ and ∞ otherwise

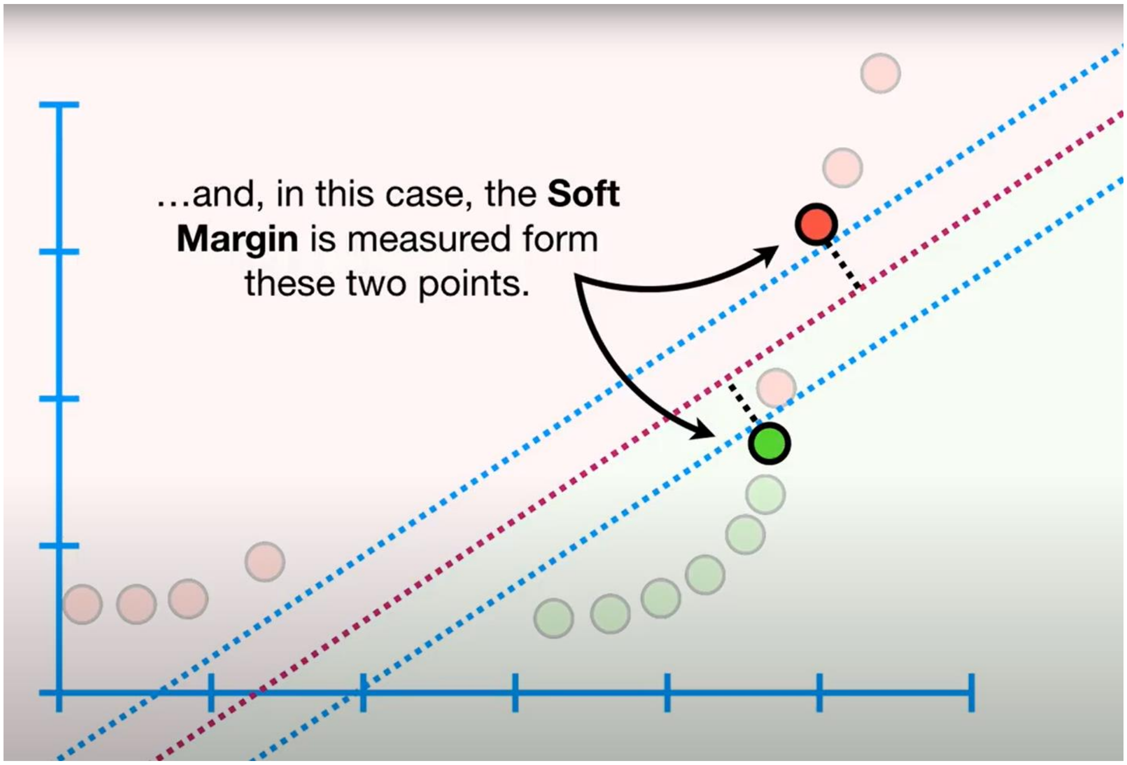
This Hard Margin SVM error is infinite in case of at least one Misclassification!

In pursuit of 0% $TRAIN$ error rate.

Compared to the Soft Margin SVM (addressed afterwards),
Small Bias & Big Variance.

3. Soft SVM without Expansion of Basis

Hard Margin is so susceptible to Outliers!


...and, in this case, the **Soft Margin** is measured form these two points.

3-1. Jargons

- **Soft Margin** : Distance between the Discriminant Function and the Frontier observation, Allowing Misclassification IN SAMPLE.

- The best Soft Margin is decided by **Validation**! (X an Optimal Problem)

Hard Margin vs Soft Margin (Bias Variance Tradeoff Seesaw!)

- **Hard Margin** : Bigger Variance, Smaller Bias

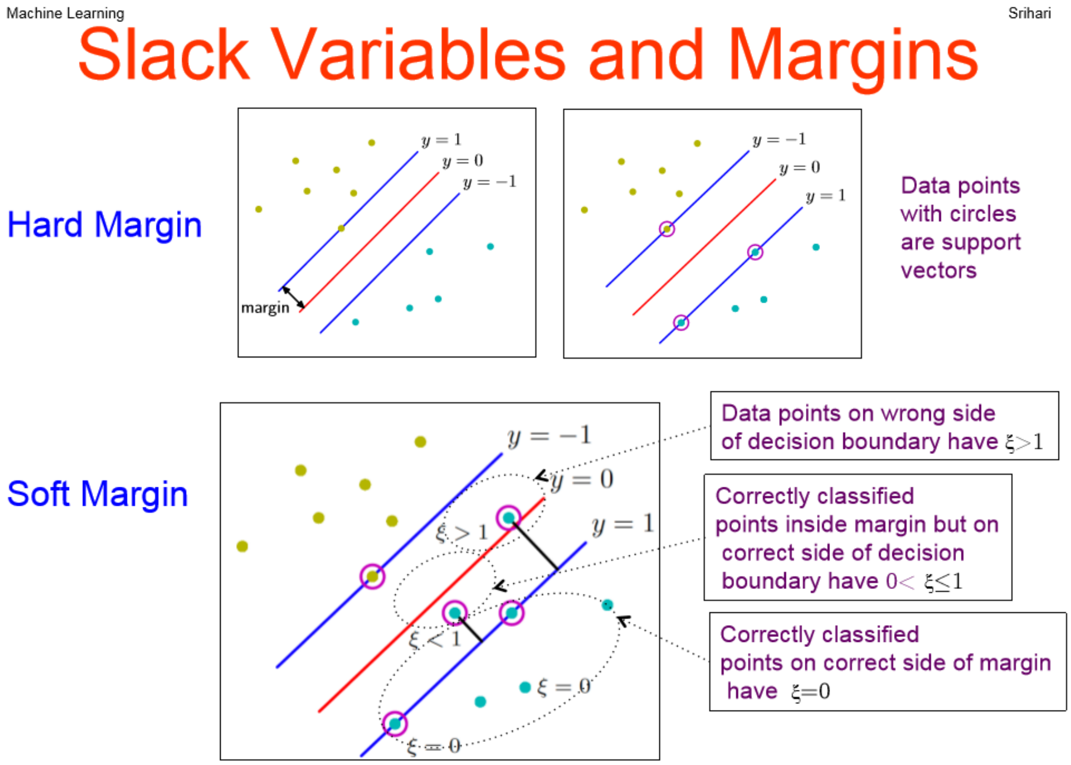
- **Soft Margin** : Bigger Bias, Smaller Variance. (Allow Bias but reduce Variance)

- **Soft Margin** is close to **Ridge, Lasso** although one is in the realm of classification and the other is in the realm of regressio! (Not only the idea but also the penalty term of the error function. addressed afterwards.)

3-2. Error Function

3-2-1. Our new Friend “Xi”

ξ_n (Slack Variable) : A variable that increases Proportionally with the distance of a point with the ‘correct’ Discriminant Function. (Linear Increase)



Machine Learning

Srihari

Slack Variables and Margins

Hard Margin

Data points with circles are support vectors

Soft Margin

Data points on wrong side of decision boundary have $\xi > 1$

Correctly classified points inside margin but on correct side of decision boundary have $0 < \xi \leq 1$

Correctly classified points on correct side of margin have $\xi = 0$

Question) What if we set ξ_n to increase quadratically with each point to the correct Discriminant Function?

- Understand Four Cases.

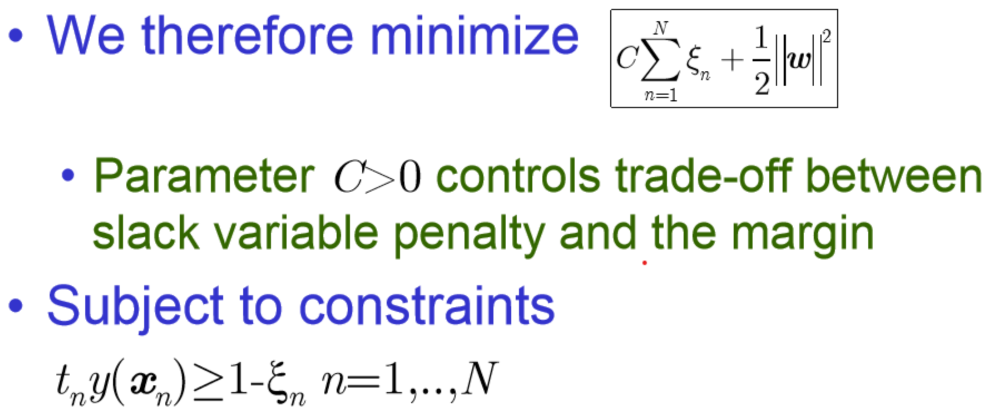
1) $\xi_n = 0$

2) $0 < \xi_n < 1, \xi = | t_n - y(x_n) |$

3) $\xi_n = 1,\ \xi = | t_n - y(x_n) |$

4) $\xi_n > 1, \xi = | t_n - y(x_n)$

3-2-2. Error Function



- We therefore minimize $\boxed{C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2}$
  - Parameter $C>0$ controls trade-off between slack variable penalty and the margin
- Subject to constraints

$$t_n y(\boldsymbol{x}_n)\geq 1\text{-}\xi_n \quad n=1,..,N$$

3-2-3. Meaning of C (Pertinent C value decided by Validation as well.)

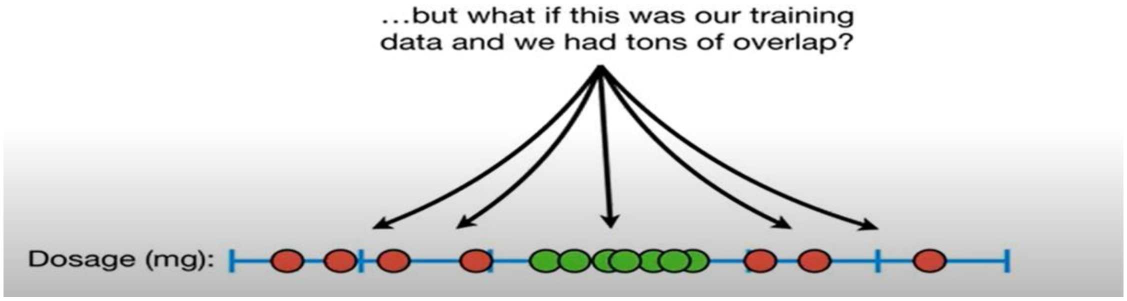
To minimize error…

Large C : $\sum_{n=1}^{N}\xi_n$ has to be small! : Takes care a lot about the penalty of Misclassification : **Close to the Hard Margin**

Small C : $\sum_{n=1}^{N}\xi_n$ can be large but margin has to be large! : Takes less care about the penalty of Misclassification : **Far from the Hard Margin**

Bias Variance Tradeoff Again!

4. Problems and Kernel Introduced


...but what if this was our training data and we had tons of overlap?
Dosage (mg):

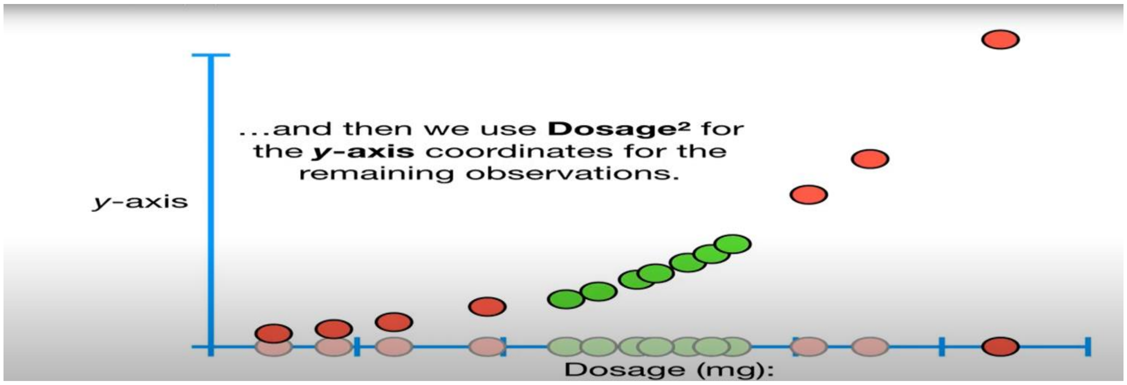
4-1. Problems. What can we do??

Choice 1) Allow about 50% of misclassification rate in sample.

Choice 2) “SVM cannot solve this. Give up!”

➔ Neither is right.

4-2. Solution : SVM can discriminate this using a STRAIGHT hyperplane (...) ?!



- **Solution : Find a Linearly Separable Hyperplane in the Feature space, dimensionally expanded from the input space !!**
- Note : When we report the clients, show the classification in the Original input space, not in the augmented Feature space! In this Picture, the discriminant is Curved.

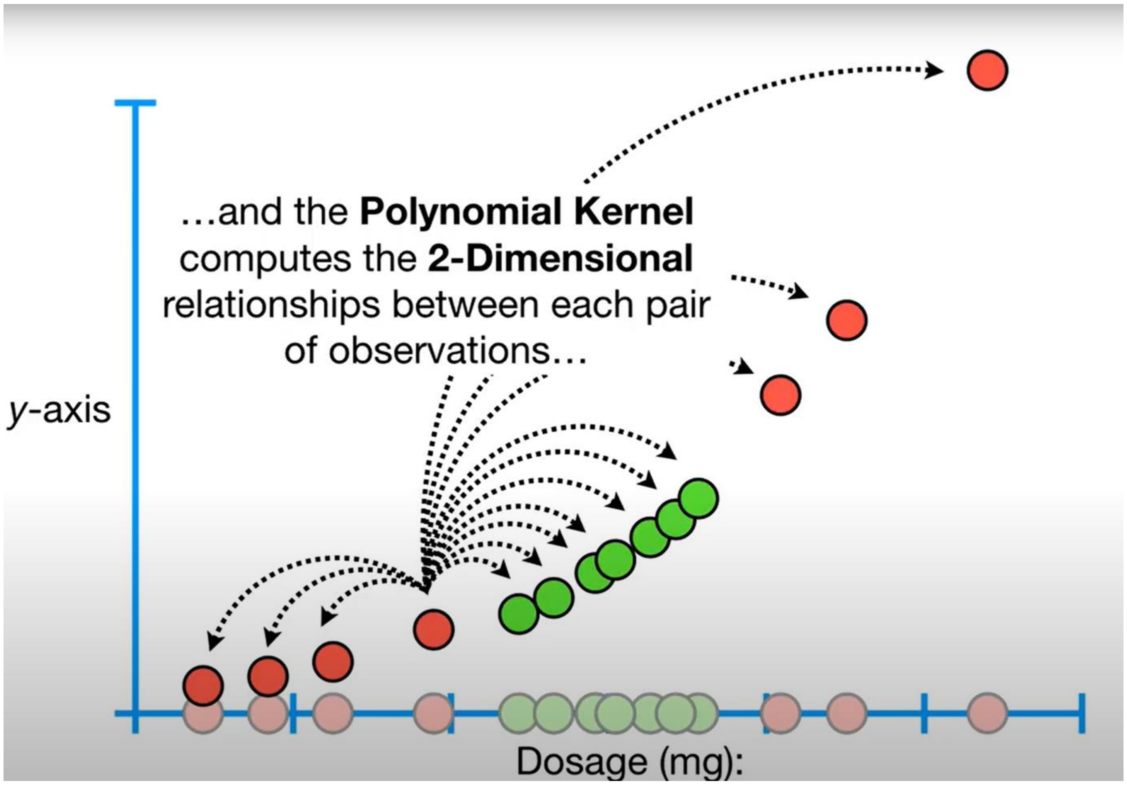
  ➔ **Kernel Function** that gets the augmented space and calculates the similarity needed.

- Thus to generalize, change all previous X’s to $\phi(X)$! Previously, I’ve addressed a special case of $\phi(X) = X$ for easy understanding of the concept of Maximal Margin.

4-3. A Kernel Function works in two steps

1st) Input Space →Feature Space (Feature expansion)

2nd) Calculate the similarity between all points : Gram Matrix is formed.



…and the **Polynomial Kernel** computes the **2-Dimensional** relationships between each pair of observations…

Q) : Why x^2? How about x^{e3}? How to decide 'The Best' Dimension?

A) Cross Validation… You know the answer…

5) What is Kernel Mathematically? (Similarity?)

Definition) $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ **,** $x_i, x_j \in R^n$

Definition) $\phi(X)$ **is a fixed nonlinear mapping (basis Function) from input space** R^n **to Inner product space** F **in** R^m

- Why talking about inner product? What's the relation of inner product with **Similarity?**

-**Cosine Similarity!** Think of the similarity between two points varying the degree of th angle two vectors form from 0 to 180!

- Disclaimer : Kernel Function Here is different from the Kernel Smoothing learned in Nonparametric Statistics & Time series, so be aware of that.

6) Kernel Trick : For Computational Ease

What if a overly complex discriminant function is needed? Computationally intensive following the definition of Kernel.

Thus, the computer doesn't transform the data into a high dimensional data and then inner product. It instead calculates the inner product in low dimensional space using the mathematically same function.

One example to show :

- $\psi(\vec{x}_i) = \phi([x_{i1}, x_{i2}]) = (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2)$

No Kernel Trick: **11times of multiplication** vs **Kernel Trick : 3 times of multiplication.**

Mercer's thm) $K(x_i, x_j)$ is a PSD matrix iff $K(x_i, x_j)$ can be expressed as $\phi(x_i)^T\phi(x_j)$

How to use the Mercer's thm : That Condition stasfied -> $\phi(X)$ **calculation not needed**

7) Examples of Kernel

7-1) Polynomial Kernel

$k(x_i, x_j) = (x_i^T x_j + C)^d$, C& d chosen by validation.

- d is the degree for the basis expansion!
- For the meaning of C : 2 examples

ex1) $(x_i^T x_j + \frac{1}{2})^2$

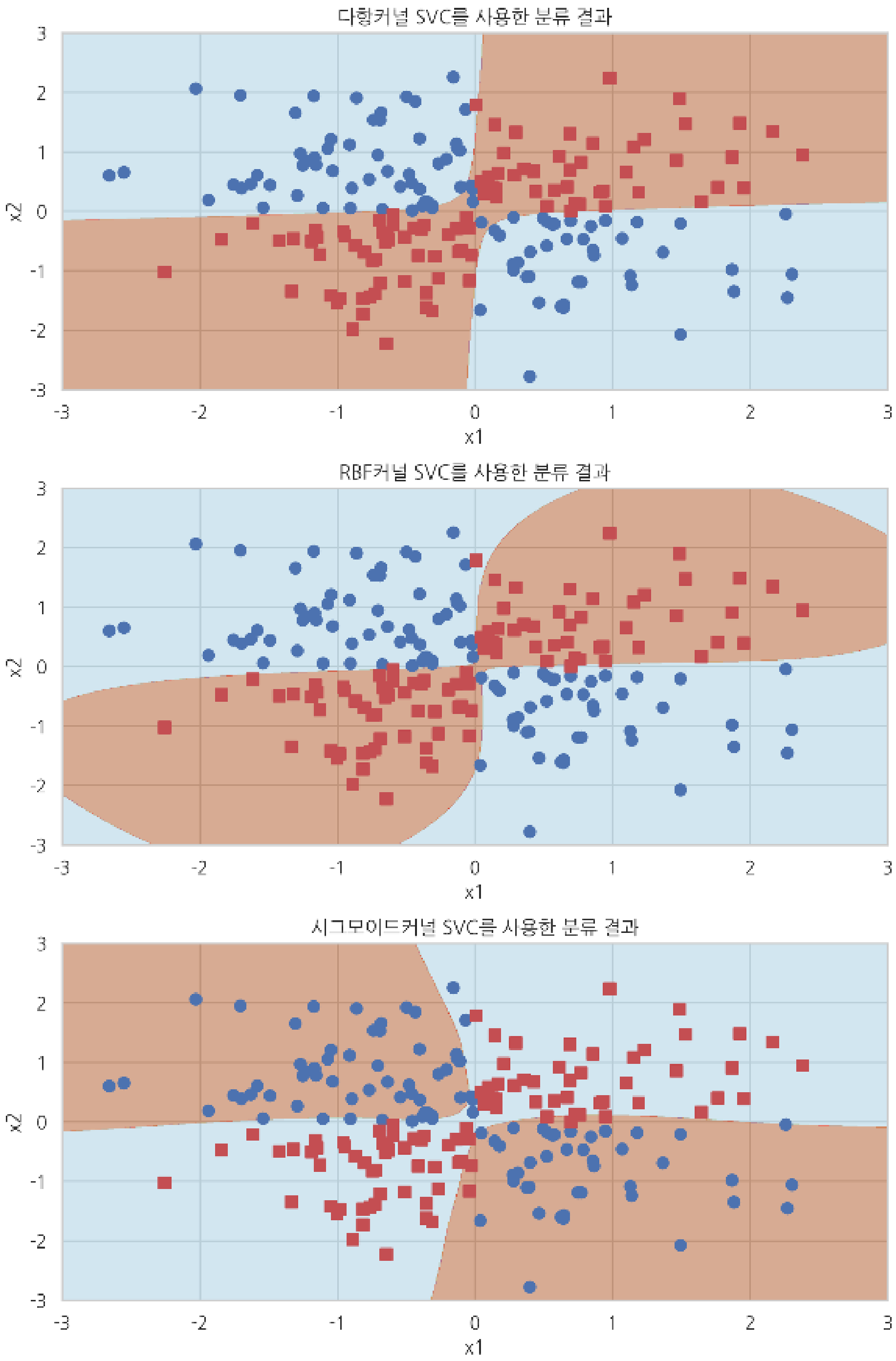
$= < x_i, x_i^2, \frac{1}{2} > < x_j, x_j^2, \frac{1}{2} >$

So called Z axis(perpendicular) is always 1/2 : No information ∴ One dimension Removed!

ex2) $(x_i^T x_j + 1)^2$

$= < \sqrt{2}x_i, x_i^2, 1 > < \sqrt{2}x_j, x_j^2, 1 >$

- Thus, When two classes are quite close that we need to expand a specific axis synthetically, large C this case helps! (Expanded and Contracted like a band)



Three examples Discriminants using 3 types of kernel

1) Polynomial Kernel 2) RBF Kernel 3) Sigmoid Kernel

7-2) Radial Kernel

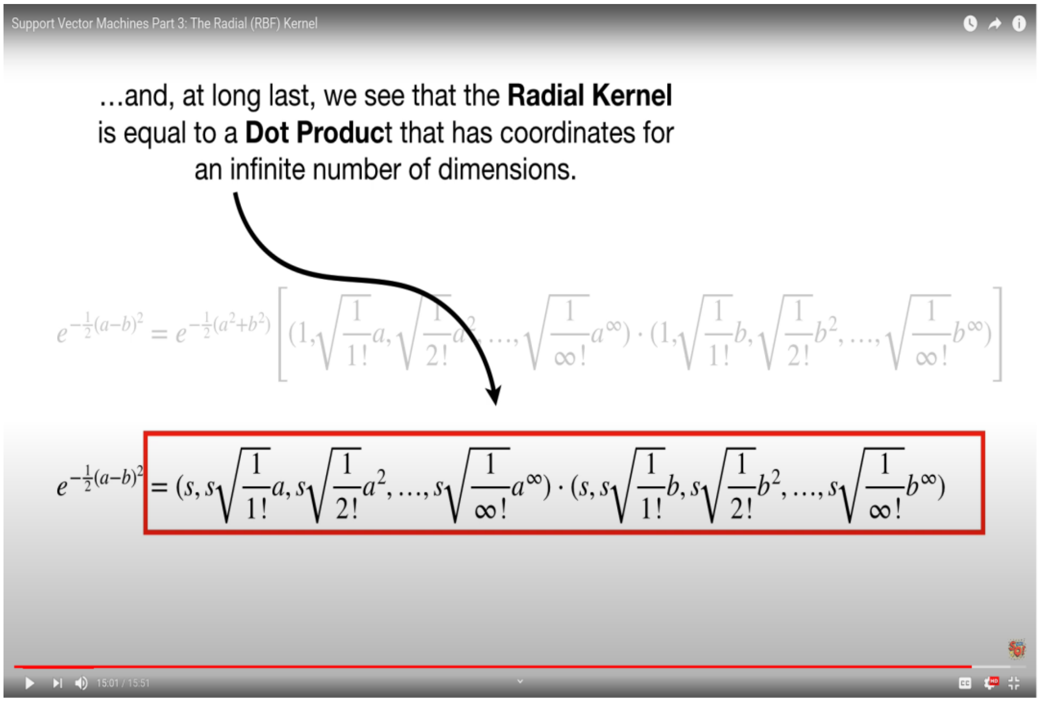
$k(x_i, x_j) = e^{-r(x_{ik}-x_{jm})^2}$, r chosen by validation..

Big r : Big Variance

Small r: Big Bias

- Importance : Can be expanded to an infinite dimensional space!

  Why important? No matter how Curvy Discriminant Function is needed, we can draw a discriminant hyperplane in infinite dimensional space.

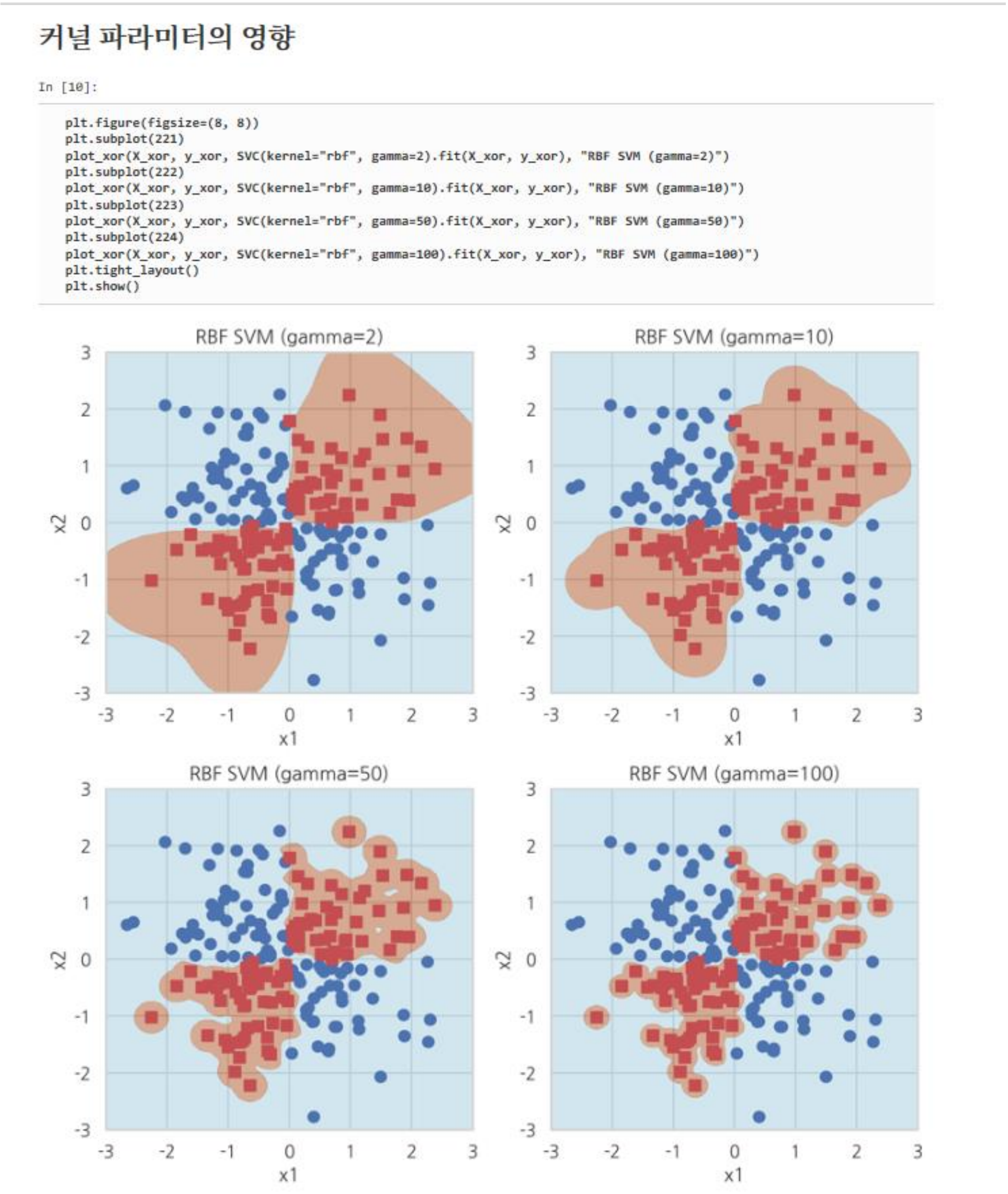


Support Vector Machines Part 3: The Radial (RBF) Kernel

…and, at long last, we see that the **Radial Kernel** is equal to a **Dot Product** that has coordinates for an infinite number of dimensions.

$$e^{-\frac{1}{2}(a-b)^2} = e^{-\frac{1}{2}(a^2+b^2)}\left[(1, \sqrt{\frac{1}{1!}}a, \sqrt{\frac{1}{2!}}a^2, \ldots, \sqrt{\frac{1}{\infty!}}a^{\infty}) \cdot (1, \sqrt{\frac{1}{1!}}b, \sqrt{\frac{1}{2!}}b^2, \ldots, \sqrt{\frac{1}{\infty!}}b^{\infty})\right]$$

$$e^{-\frac{1}{2}(a-b)^2} \boxed{= (s, s\sqrt{\frac{1}{1!}}a, s\sqrt{\frac{1}{2!}}a^2, \ldots, s\sqrt{\frac{1}{\infty!}}a^{\infty}) \cdot (s, s\sqrt{\frac{1}{1!}}b, s\sqrt{\frac{1}{2!}}b^2, \ldots, s\sqrt{\frac{1}{\infty!}}b^{\infty})}$$



커널 파라미터의 영향

In [10]:

```
plt.figure(figsize=(8, 8))
plt.subplot(221)
plot_xor(X_xor, y_xor, SVC(kernel="rbf", gamma=2).fit(X_xor, y_xor), "RBF SVM (gamma=2)")
plt.subplot(222)
plot_xor(X_xor, y_xor, SVC(kernel="rbf", gamma=10).fit(X_xor, y_xor), "RBF SVM (gamma=10)")
plt.subplot(223)
plot_xor(X_xor, y_xor, SVC(kernel="rbf", gamma=50).fit(X_xor, y_xor), "RBF SVM (gamma=50)")
plt.subplot(224)
plot_xor(X_xor, y_xor, SVC(kernel="rbf", gamma=100).fit(X_xor, y_xor), "RBF SVM (gamma=100)")
plt.tight_layout()
plt.show()
```

and Gaussian, Sigmoid Kernel, etc.

8. Why called “Sparse Kernel Method?”

To understand this, have a look at **Memory Based Model**, that contains SVM.

Linear Model vs Memory Based Model

- **Linear Model** : train data → $y(x, w)$ model → predict the test data
- Train data itself is never used for Test.
- However, **Memory Based Model** uses Train data directly to predict the Test Data.
- **Support Vector Machine is a memory based model! but Sparse!**
- Why sparse? **Only the frontier Support Vectors used for Test Data Prediction!**

9. Citations / image used

P2 : https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15_097S12_lec12.pdf MIT Opencourseware, p8

PP 3,4,7,8 : https://cedar.buffalo.edu/~srihari/CSE574/Chap7/7.1-SVMs.pdf Introduction to Machine Learning Course (Instructor : Sargur Srihari, University at Buffalo) pp 10,15,20,24,27,25

P9 : https://www.youtube.com/watch?v=efR1C6CvhmE Statquest with Josh Starmer -Youtube

PP 10, 11 : https://cedar.buffalo.edu/~srihari/CSE574/Chap7/7.2-SVM-Overlap.pdf Introduction to Machine Learning Course (Instructor : Sargur Srihari, University at Buffalo) pp 8,10

PP 12, 13 : https://www.youtube.com/watch?v=efR1C6CvhmE Statquest with Josh Starmer -Youtube

PP 16,18 : https://www.youtube.com/watch?v=Toet3EiSFcM Statquest with Josh Starmer -Youtube

PP 17,19 : https://datascienceschool.net/view-notebook/69278a5de79449019ad1f51a614ef87c/ Datascienceschool.net (Korean site)