

The Support Vector Machine

4

B

CS 3244
Machine Learning



NUS | Computing

Forecast for Week 04B



Learning Outcomes for this week:

- Understand the Support Vector Machine Classifier as an optimal hyperplane;
- Understand how the optimization function is modified to allow errors (soft SVM).

Other important concepts:

- Noisy Targets
- Non-linear Mappings
- Kernels



Login to participate on Zoom



You'll need to annotate my screen today, so please join via Zoom or work with someone in the classroom to do the exercise.

(Don't worry, participation / attendance will be recorded by QR)

Projects Groupings (again! 🤯)



Apparently LumiNUS reports to us that some of your completed sub-team preference forms are incomplete although your interface shows it as complete.

We may need to redo the groups again! 😓 Very sorry

An aerial photograph of a city grid, likely New York City, with a complex network of streets. Several paths are highlighted in bright yellow and red, creating a visual representation of a classification or regression problem on a spatial dataset.

Using Regression for Classification?

CS3244 Machine Learning



Department of Computer Science
School of Computing

Linear regression for classification



Linear regression learns a real-valued function $y = f(\mathbf{x}) \in \mathbb{R}$

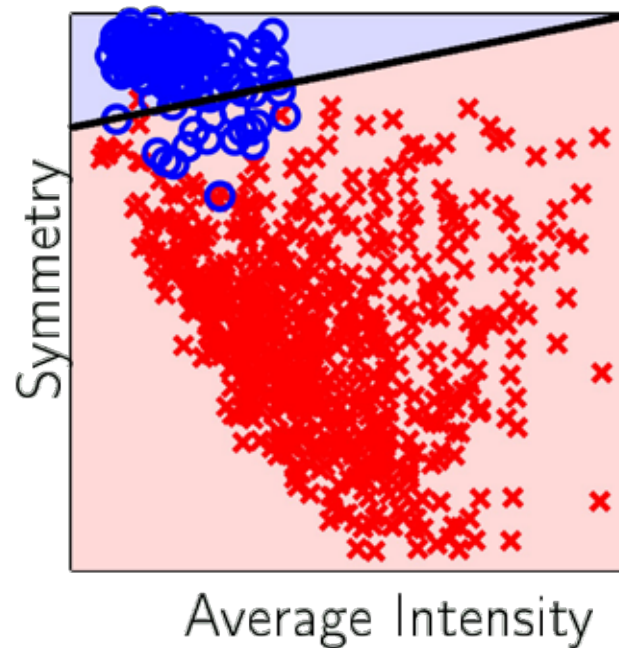
Binary valued functions are also real-valued! $\pm 1 \in \mathbb{R}$

Use linear regression to get θ where $\theta^\top \mathbf{x}^{(j)} \approx y^{(j)} = \pm 1$

In this case, $\text{sign}(\theta^\top \mathbf{x}^{(j)})$ is likely to agree with $y^{(j)} = \pm 1$


Good initial weights for classification

Why linear regression doesn't set good weights for classification



What's wrong
with this
picture?

Hint: think
squared error



Noisy Targets

CS3244 Machine Learning



Department of Computer Science
School of Computing

Noisy targets

The target function isn't always
a function $f: \mathcal{X} \rightarrow \mathcal{Y}$



Criterion	Value
Age	32 years
Gender	Male
Salary	40 K
Debt	26 K
...	...
Years in Job	1 year
Years at Current Residence	3 years

Consider two identical
customers for loan approval ...
could have two different
outcomes!

Why? How do
we characterize
these sources
of noise?

Your Turn: what do you think?

The target function isn't always
a function $f: \mathcal{X} \rightarrow \mathcal{Y}$



Criterion	Value
Age	32 years
Gender	Male
Salary	40 K
Debt	26 K
...	...
Years in Job	1 year
Years at Current Residence	3 years

Q1: Is misreporting salary a noisy target?

Q2: What other lecture featured noisy targets?

Target distribution

Instead of saying the target is a **function**, think of it as a **distribution**: $P(y|\mathbf{x})$

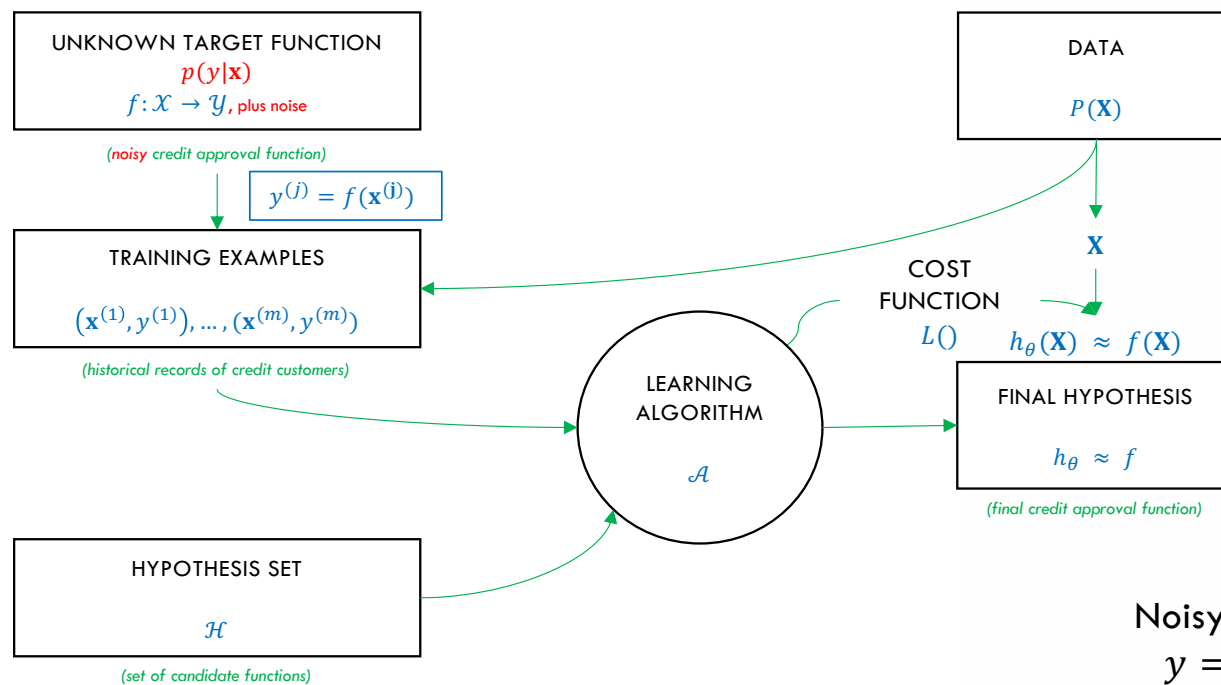
Our data (\mathbf{x}, y) is now generated by the joint distribution: $P(\mathbf{x})P(y|\mathbf{x})$

Noisy target = Deterministic target function $f(x) = \mathbb{E}(y|\mathbf{x}) + \text{Noise } (y - f(\mathbf{x}))$

A deterministic target is just a special case of the generalized (noisy) target:

$$P(y|\mathbf{x}) = 0, \text{ except where } y = f(\mathbf{x})$$

Learning diagram with noisy targets



Noisy Target:
 $y = f(\mathbf{x}) \rightarrow y \sim P(y|\mathbf{x})$



The Support Vector Machine

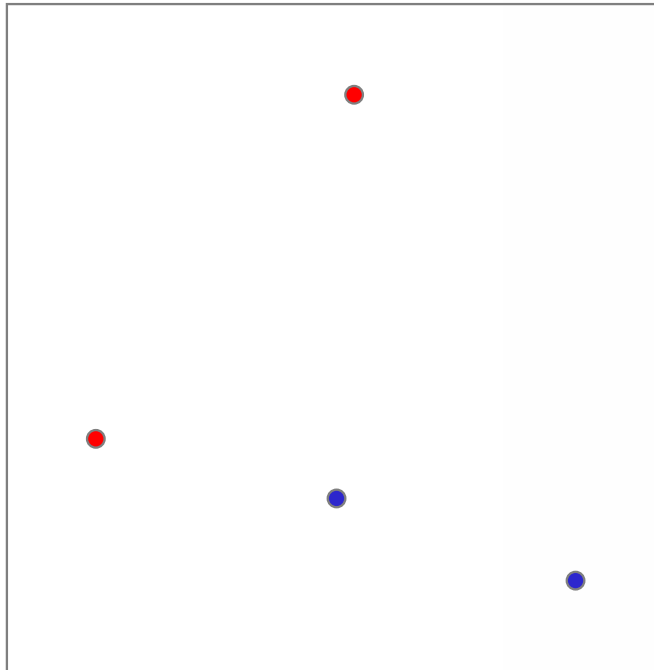
CS3244 Machine Learning



Department of Computer Science
School of Computing



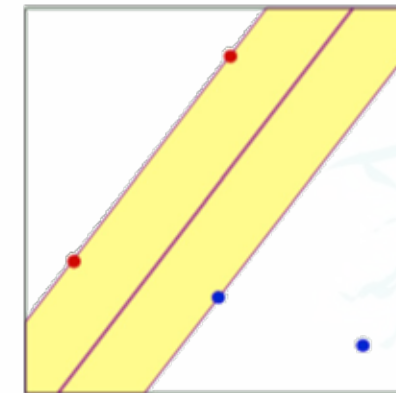
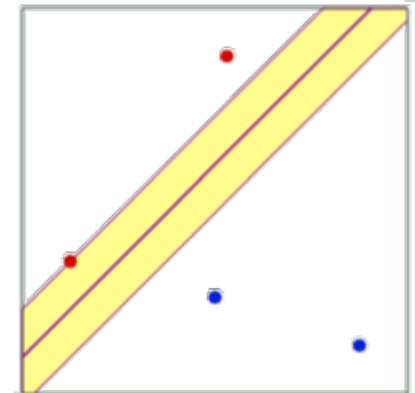
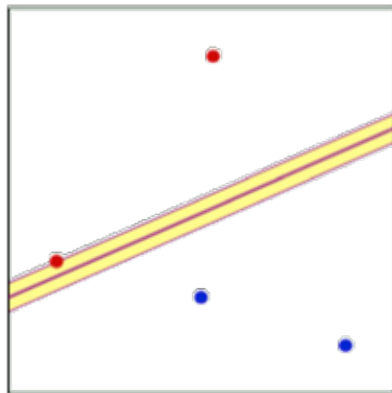
Land Transport Authority



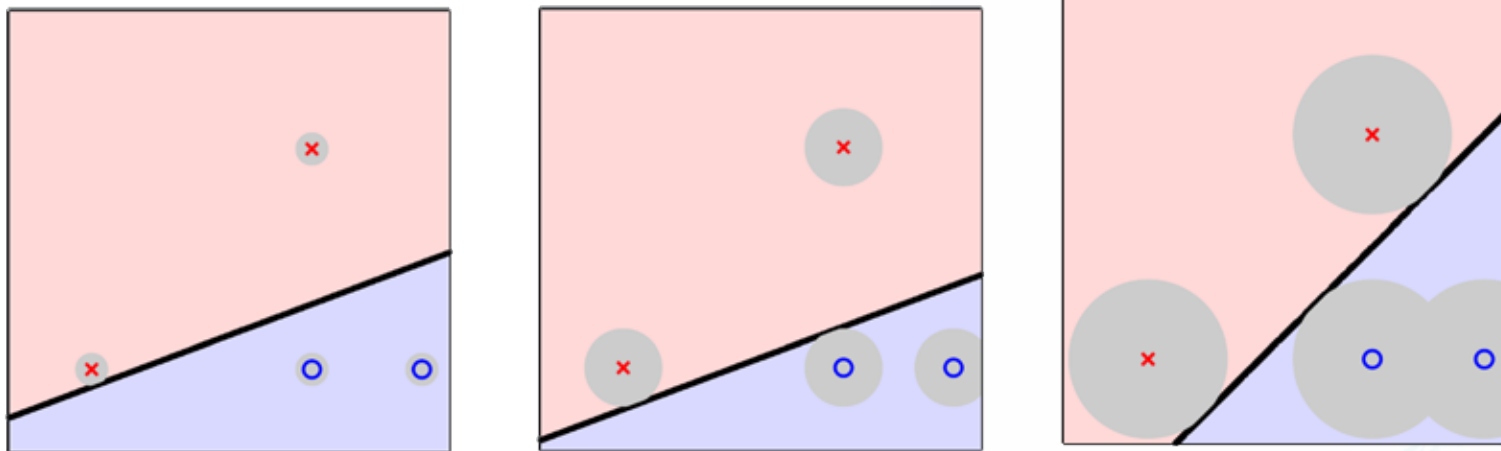
Better linear separation

Two questions:

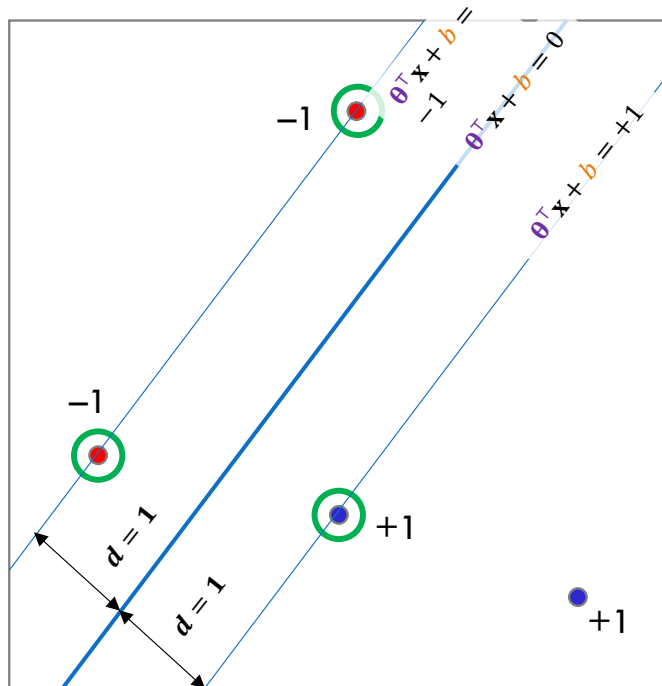
1. Why is a bigger margin better?
2. Which θ maximizes the margin?



Inherently handles noisy data



SVM: An analogizer



Hyperplane equation: $h(\mathbf{x}) = \theta^T \mathbf{x} + b$

+ve points **must** lie above the hyperplane; **-ve** points below.

θ dictates the orientation of the plane;
 b dictates the offset (bias).

Define distance d to the optimal plane as “1” (unit distance).

This sets up constrained quadratic optimization problem that identifies the unique $h(\mathbf{x})$.

Note: Only a subset of the dataset **determines** our unique $h(\mathbf{x})$.

These are the **support vectors**, the most difficult instances to classify.

⚠ **Dataset must be separable!** ⚠



Inductive Bias of the SVM



Let's answer together. Write on my slide!

What do you think the Inductive Bias of the SVM is?



Non-linear Mapping

CS3244 Machine Learning



Department of Computer Science
School of Computing

Feature Engineering

Raw input $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, \dots, x_{256})$

Too many (257) parameters!

Linear model: $(\theta_0, \theta_1, \theta_2, \dots, \theta_{256})$

Features: extract useful information, e.g.,

Intensity and symmetry: $\mathbf{x} = (x_0, x_1, x_2)$

Linear model: $(\theta_0, \theta_1, \theta_2)$

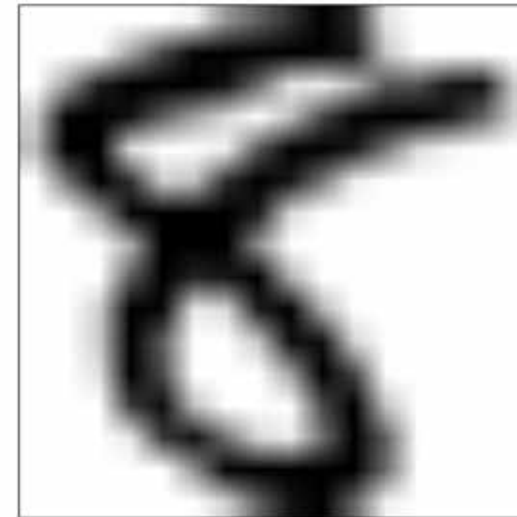
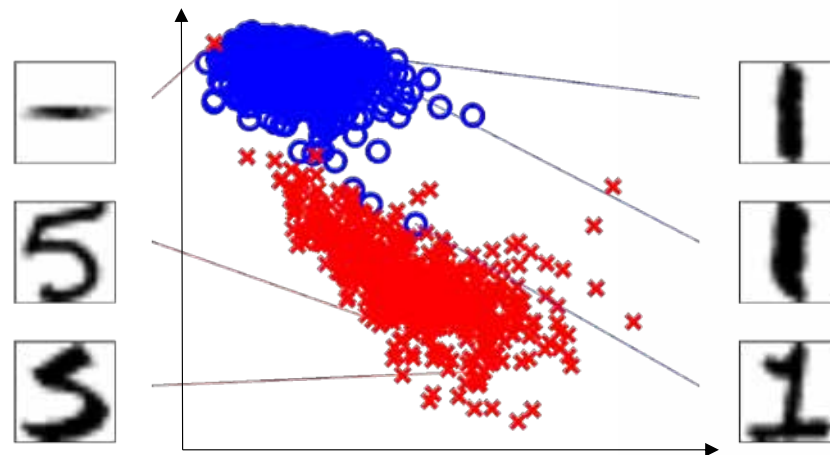


Illustration of features

$$\mathbf{x} = (x_0, x_1, x_2)$$

$x_1 = \text{intensity}$

$x_2 = \text{symmetry}$



Your Turn: which axes is which?

Can, but not accurate.



probably not with a high degree of accuracy, so perhaps the more appropriate term is simply *estimate* rather than *model*. Of course these are dependent on how non-linear the data is : linear classifiers can give good estimates approximately linear data and vice versa



I think it is possible but the accuracy would be low



I believe we can model certain non-linearity with linear classifiers but in general, it is not possible and trying to model non-linearity with linear classifiers will result in poor classifiers with low accuracy.

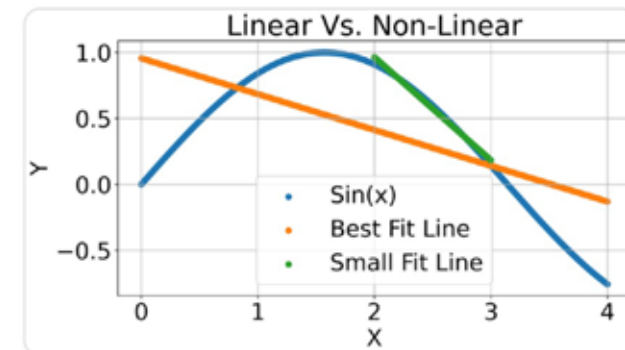


I believe it is not ideal even if it is possible as accuracy will be significantly affected.



In Teaching Experimental Physics, we often fit our experimental data to the known equation to check its validity. It can be either linear or non-linear. If we fit a linear equation to a non-linear model, it will only be valid for a small range of values.

image.png ▾



No, cannot.



I think we cannot model non-linearity with a linear classifier accurately. We can at most estimate. Linear classifiers, according to Robby's answer, do not model interactions between features. Mathematically, using a linear regression model would assume that the coefficients of any $(x_i)(x_j)$ term (where i is not equal to j) is always zero.



I don't think most non-linear problems can be modelled with a linear classifier since there is a reason why they are non-linear and thus, a lot of data will be lost if we actually model them with a linear classifier.

That being said, there can be some usages in modelling non-linearity with a linear classifier and sometimes it works and is accurate. The results varies in a case by case basis.



If a problem has many features, then I would wonder if using a linear model over non-linear would make a significant difference in the outcome. Ultimately, I think that a non-linear model would almost always be better in accuracy, but would suffer in perform with regards to time. So now, if we have many features involved, I think the strength of a non-linear model may not be as apparent, as linear models may be as accurate too



Yes, with some transformation



Yes, by scaling the non-linear features of the function to become linear. However, model may not be very accurate after scaling.



it may be possible if some transformations are made to the dataset



I believe we could model non-linearity with a linear classifier. We could easily find out whether the relationship is linear through scatter plot. Then we could use box-cox method or other method to do variable transformation. To convert the non-linear relationship to a linear relationship. For example if y has quadratic relationship with x , then transform x to x^2 will help a lot. (edited)



Yes, it is possible to model non-linearity with a linear classifier. For instance, kernels allow the linear support vector machines to incorporate non-linearity in some way. From what I read on this [website](#), kernels don't change the nature of the model but rather they use feature transformation to introduce some form of non-linearity into the model.

Of course, I think using a non-linear model would still be better. However, considering that non-linear models are harder to train, it is worth looking into how we can incorporate some non-linearity into a linear model so that we can sort of have the best of both worlds

Medium

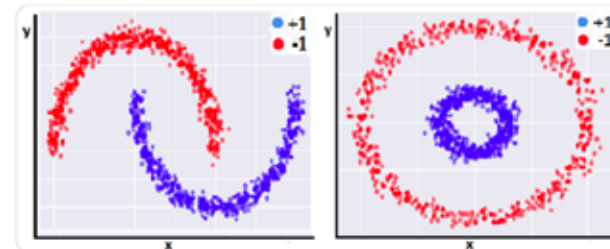
Using a Linear Model to deal with Nonlinear Dataset

Feature Transformation, Kernel-trick, SVM

Reading time

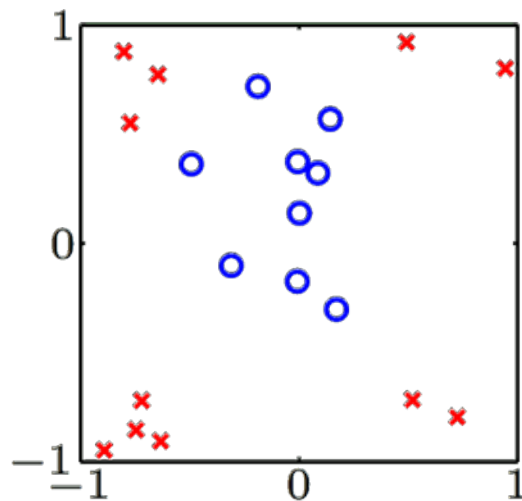
4 min read

Apr 5th, 2019 (186 kB)

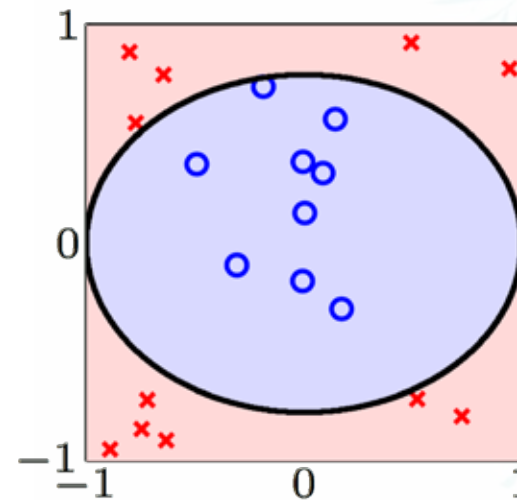


Linear models are limited

Data:



Hypothesis:



Another example



Credit line is affected by years in current residence x_i , but not in a linear way.

The value range $[1 < x_i < 5]$ is more significant.

Can we do that with linear models?

But linear *in what*?



Linear regression implements

$$\sum_{i=0}^n \theta_i x_i = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

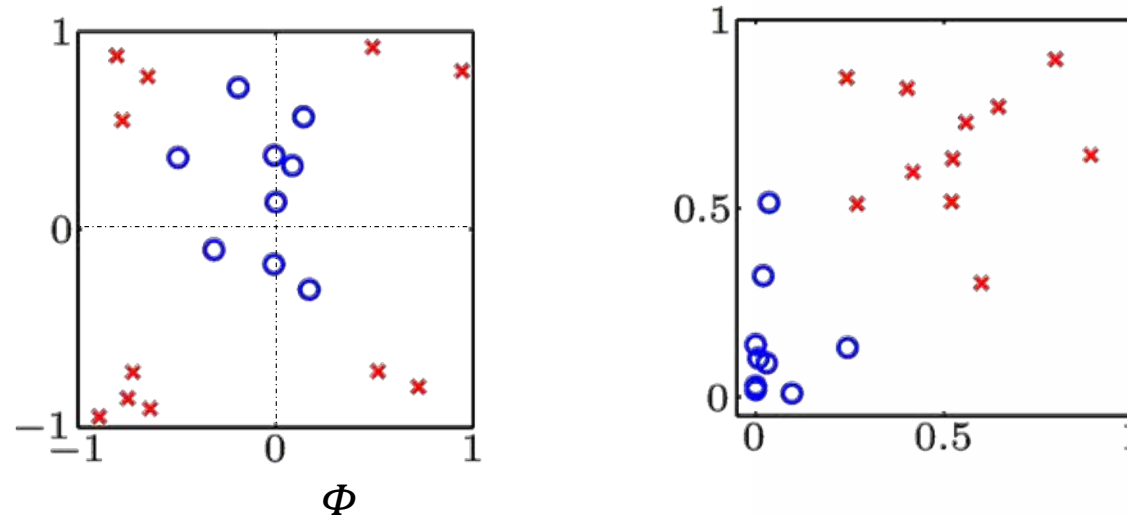
Linear classification implements

$$\text{sign}\left(\sum_{i=0}^n \theta_i x_i\right)$$

Algorithms work because of the **linearity of weights**, but it doesn't say anything about the observed data \mathbf{x} .

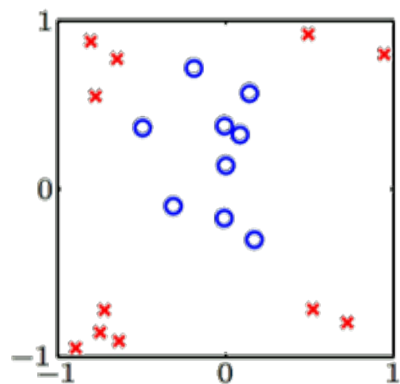
Transform the data nonlinearly

$$(x_1, x_2, \dots, x_n) \xrightarrow{\Phi} (x_1^2, x_2^2, \dots, x_n^2)$$



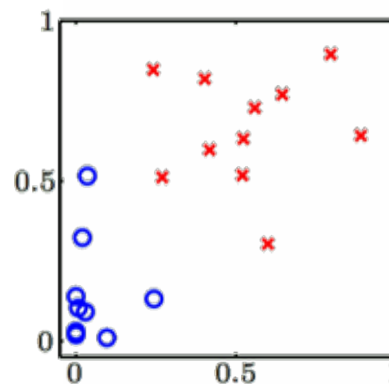
Any $\mathbf{x} \rightarrow \mathbf{z}$ preserves this linearity!

1. Original Data
 $\mathbf{x}^{(j)} \in \mathcal{X}$



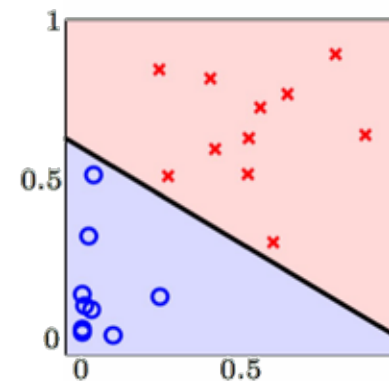
Φ

2. Transform the data
 $\mathbf{z}^{(j)} = \Phi(\mathbf{x}^{(j)}) \in \mathcal{Z}$



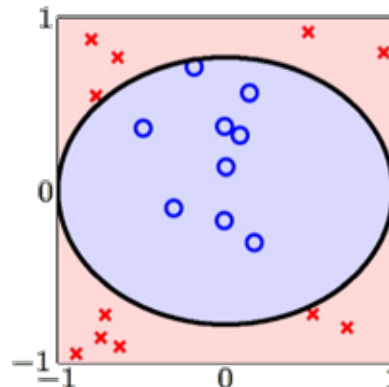
↓

3. Separate the data in
 \mathcal{Z} space
 $\tilde{h}(\mathbf{z}) = \text{sign}(\tilde{\boldsymbol{\theta}}^T \mathbf{z})$



Φ^{-1}

4. Classify in \mathcal{X} space
 $h_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) = \tilde{h}(\Phi(\mathbf{x}))$
 $= \text{sign}(\tilde{\boldsymbol{\theta}}^T \Phi(\mathbf{x}))$



What transforms to what

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \xrightarrow{\Phi} \mathbf{z} = (z_0, z_1, \dots, z_{\tilde{n}})$$

$$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)} \xrightarrow{\Phi} \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(m)}$$

$$y^{(1)}, y^{(2)}, \dots, y^{(m)} \xrightarrow{\Phi} y^{(1)}, y^{(2)}, \dots, y^{(m)}$$

$\boldsymbol{\theta}$?

No weights in \mathcal{X}

$$\tilde{\boldsymbol{\theta}} = (\theta_1, \theta_2, \dots, \theta_{\tilde{n}})$$

$$\begin{aligned} h_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) &= \text{sign}(\tilde{\boldsymbol{\theta}}^\top \mathbf{z}) \\ &= \text{sign}(\tilde{\boldsymbol{\theta}}^\top \Phi(\mathbf{x})) \end{aligned}$$

“Support Vectors” in \mathcal{X} space

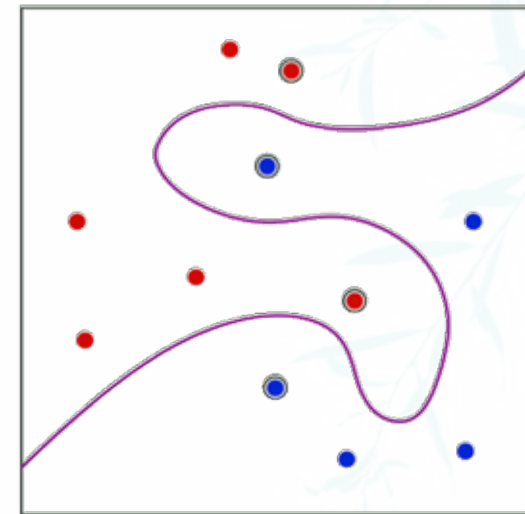


In \mathcal{X} space, we say that we have “pre-images” of support vectors.

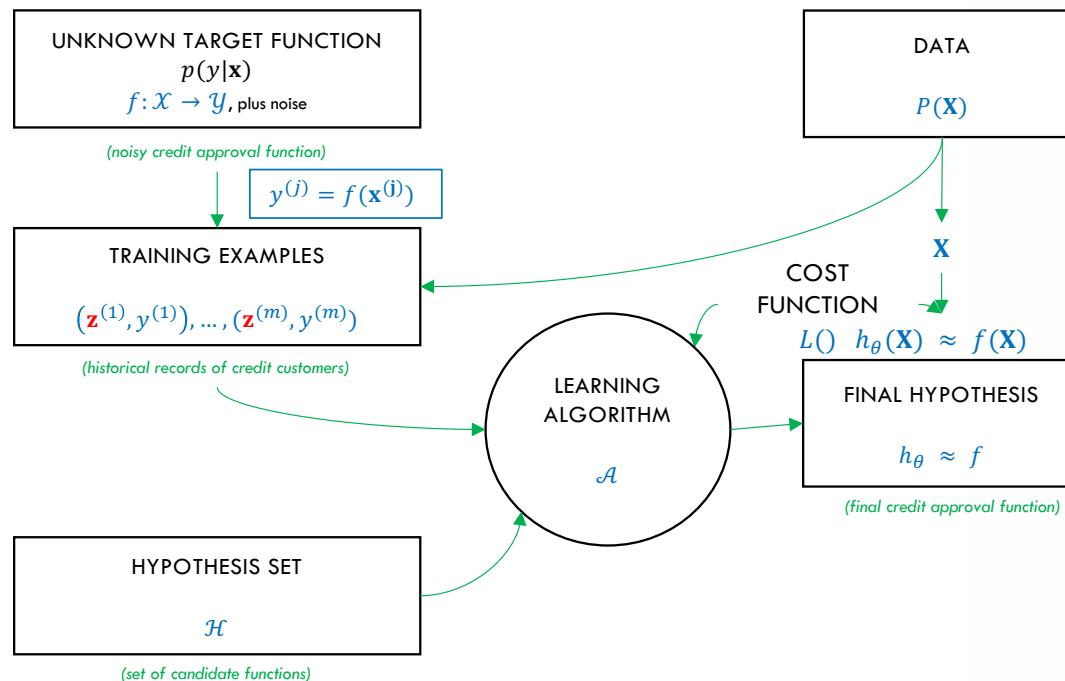
SVMs natively handle non-linear transforms through the use of **kernels**.

The margin is maintained in the \mathcal{Z} space.

Great generalization, since in the model, $\#$ of parameters \propto $\#$ of support vectors.



Final Learning Diagram



Nonlinear transformation

$\theta^T \mathbf{x}$ is linear in θ

Any $\mathbf{x} \rightarrow \mathbf{z}$ preserves this linearity.

E.g., $(x_1, x_2) \xrightarrow{\Phi} (x_1^2, x_2^2)$



Soft Margin SVM

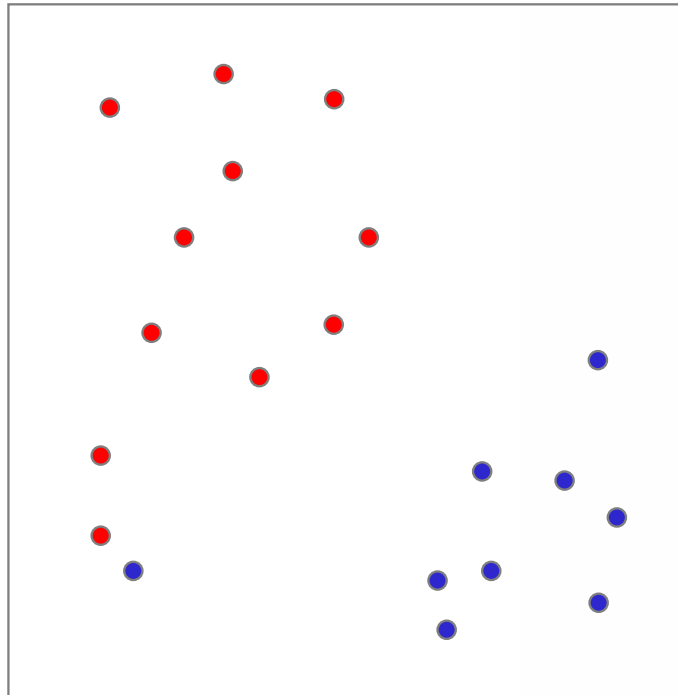
CS3244 Machine Learning



Department of Computer Science
School of Computing

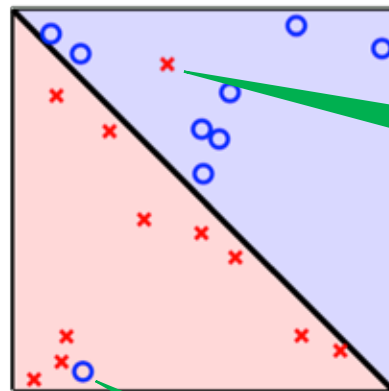


Land Transport Authority, Round 2



Two types of non-separable

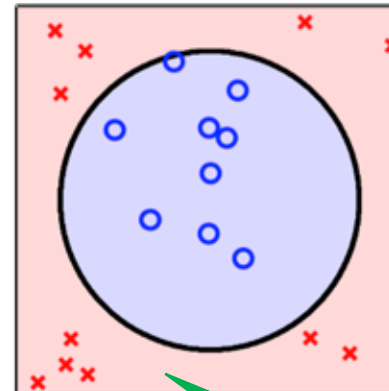
Slightly



Is this an outlier?

Is this an outlier?

Seriously



Ok, I give up.
Non-linear
transform me

Cost Function w Slack Variables

Margin violation: $y^{(*)}(\boldsymbol{\theta}^\top \mathbf{x}^{(*)} + b) \geq 1$ fails

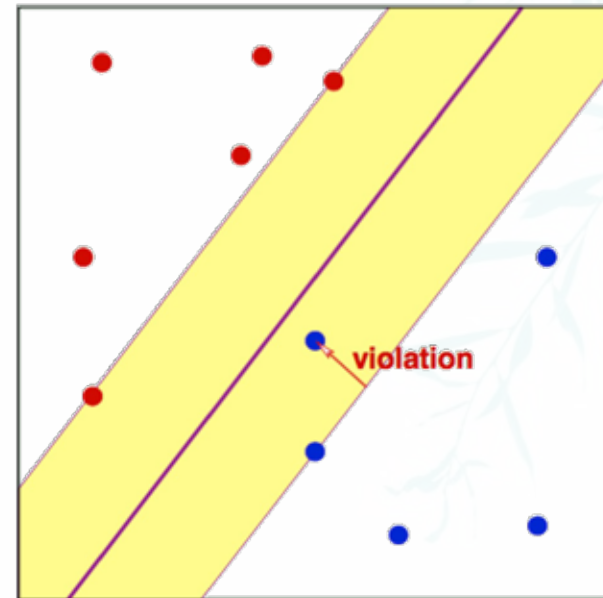
Quantify this:

$$y^{(*)}(\boldsymbol{\theta}^\top \mathbf{x}^{(*)} + b) \geq 1 - \xi^{(*)}$$

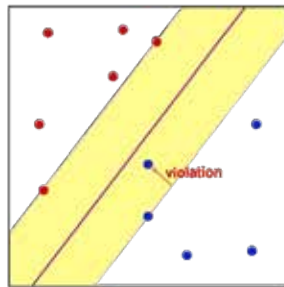
where $\xi^{(*)} \geq 0$

Slack variable: Soft error
on $(x^{(*)}, y^{(*)})$

Total violation: $\sum_{j=1}^m \xi^{(j)}$



Soft SVM's loss function: Hinge Loss



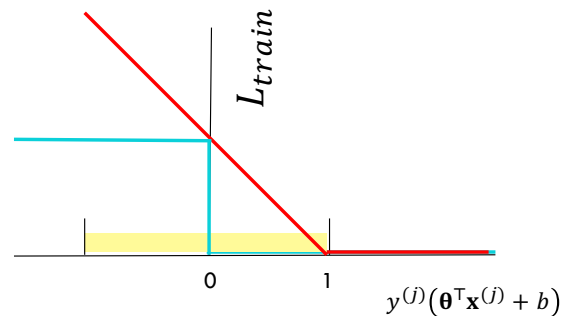
$$L_{train}(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{\theta}\|^2 + c \sum_{j=1}^m \max(0, 1 - y^{(j)}(\boldsymbol{\theta}^T \mathbf{x}^{(j)} + b))$$

width of highway

Penalty for per-instance obstacle

Soft margin SVM penalizes *misclassifications and correct classifications that fall inside the margin*.

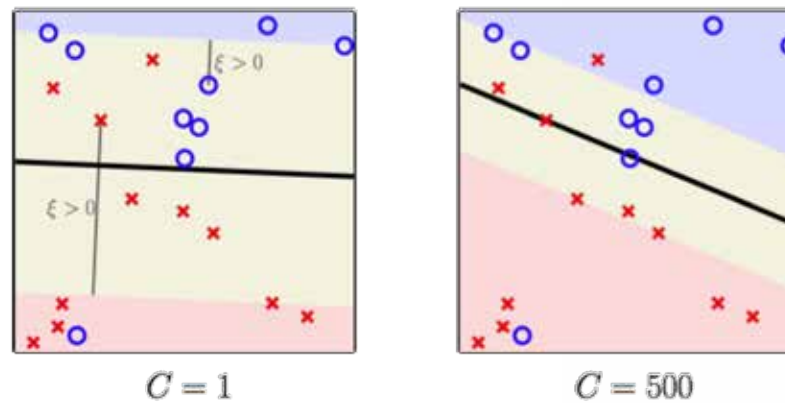
In hard margin SVM, by definition, there are no misclassifications.



green = 0-1 loss

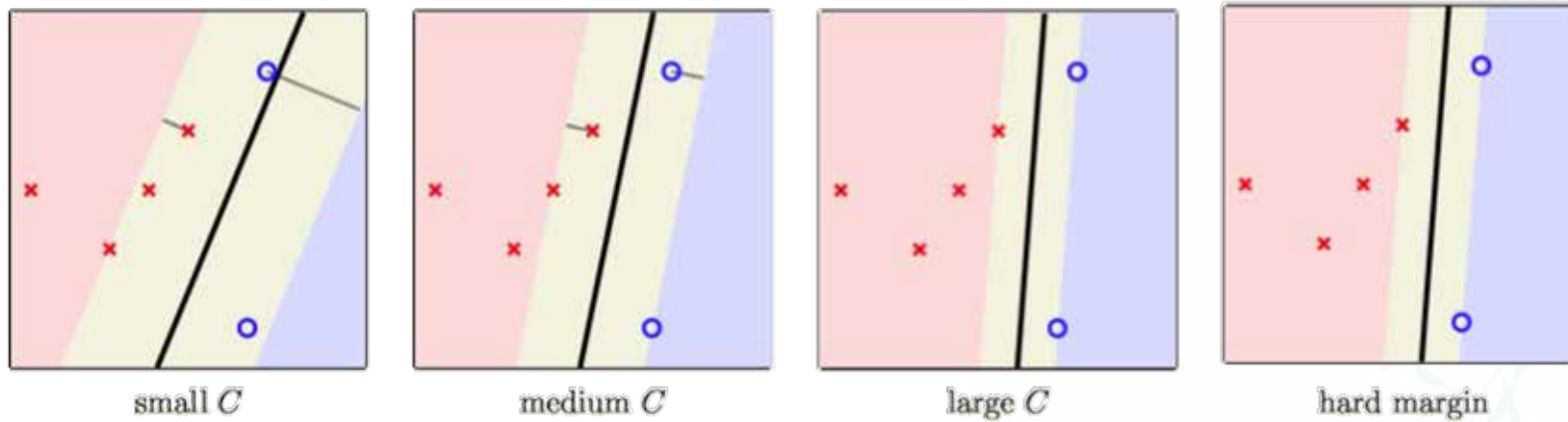
red = hinge loss

Soft Error C parameter



Soft error “badness”. Higher values indicate less tolerance.

Effect of Varying C





Wrapping up Week 04

CS3244 Machine Learning



Department of Computer Science
School of Computing

What did we learn this week?

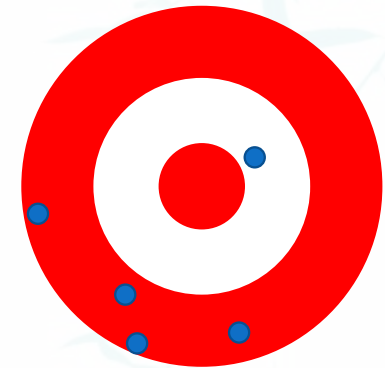
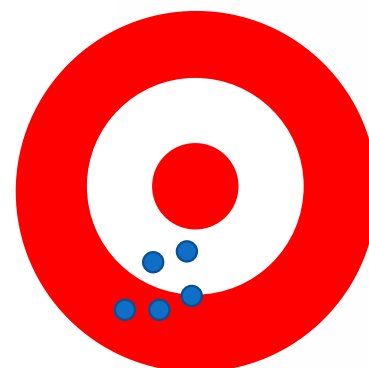
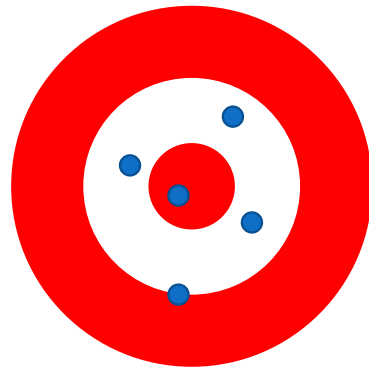
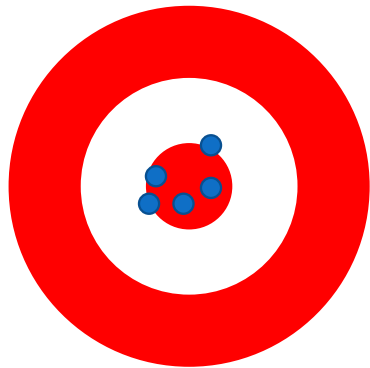


- Describe the basic idea of linear classification;
- Understand how both linear and logistic regression works;
- Understand the Support Vector Machine Classifier as an optimal hyperplane;
- Understand how the optimization function is modified to allow errors (soft SVM).

Other important concepts:

- Curse of Dimensionality
- Gradient Descent
- Noisy Targets
- Non-linear Mappings

Outlook for next week



Assigned Task (due before next Mon)



Read the post <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229> (4 mins)

Post a 1–2 sentence answer to the topic in your tutorial group: **#tg-xx**

Describe kNN or decision trees with respect to variance.

[Don't worry if you're not sure about the math,
we'll cover this again in Week 05.]