Computer Vision: Fall 2022 — Lecture 6

Dr. Karthik Mohan

Univ. of Washington, Seattle

October 18, 2022

Check-In

1. Assignment 2

Check-In

1. Assignment 2
2. Calendly slots

Check-In

1. Assignment 2
2. Calendly slots
3. Other questions/thoughts?

Today

1. Summary on Total Variation (TV) methods for image smoothing
2. Supervised Learning
3. Binary Classification

References

1. Good Book for Machine Learning Concepts

Next Topic: Total Variation (TV) for Image Smoothing

Image Smoothing Motivation

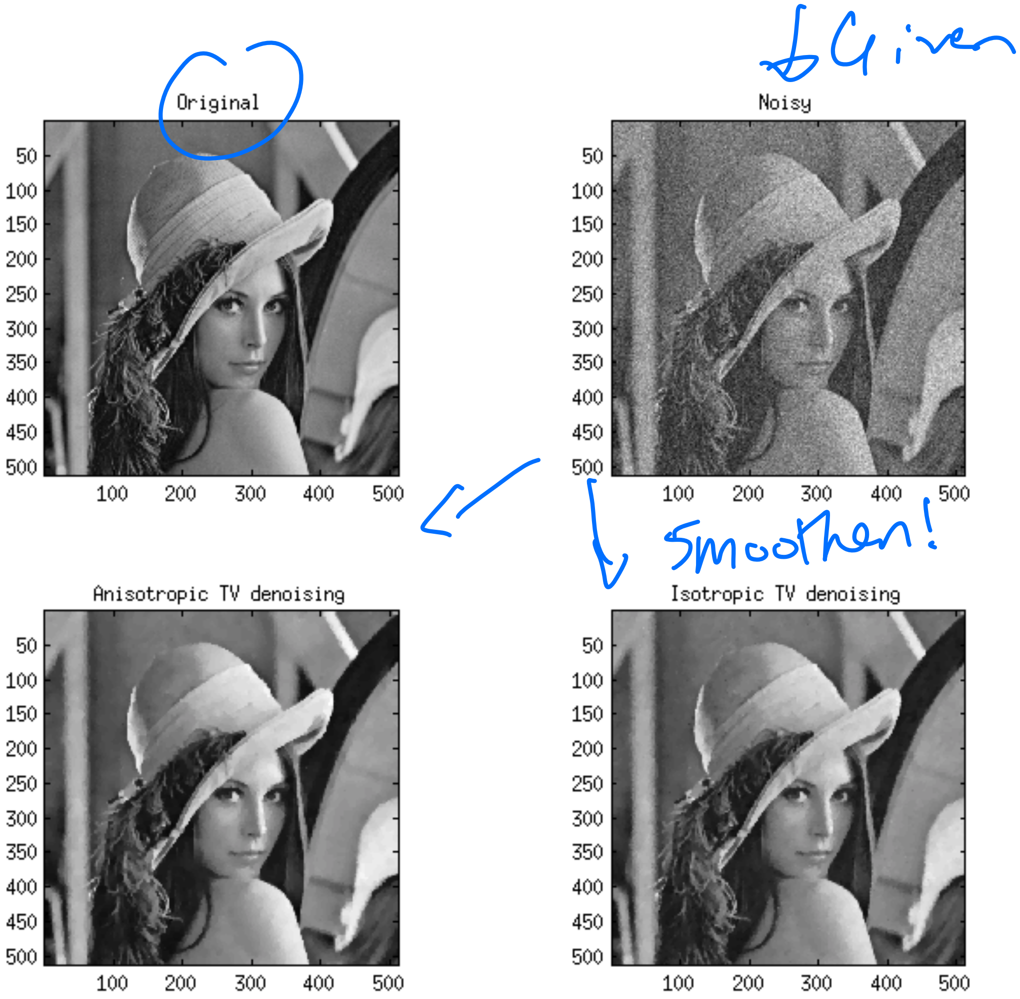
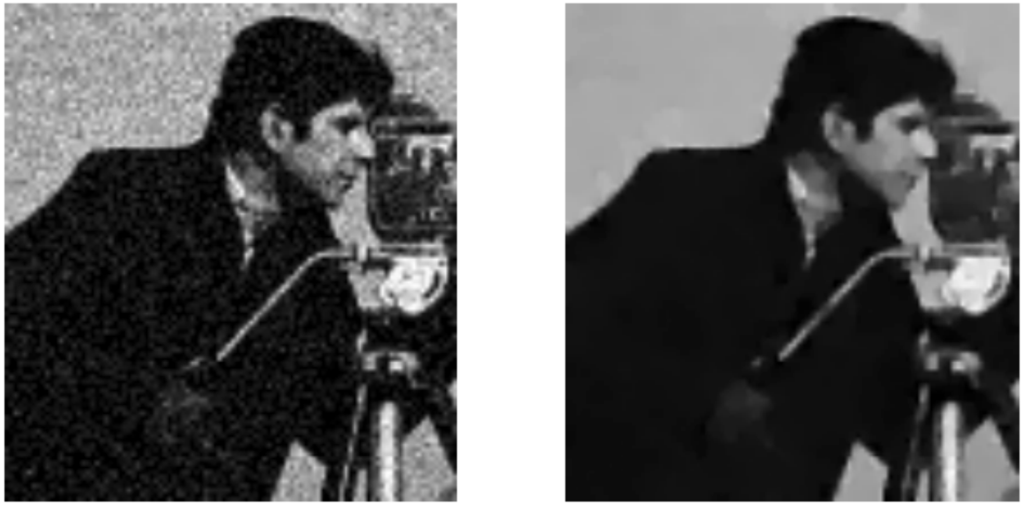
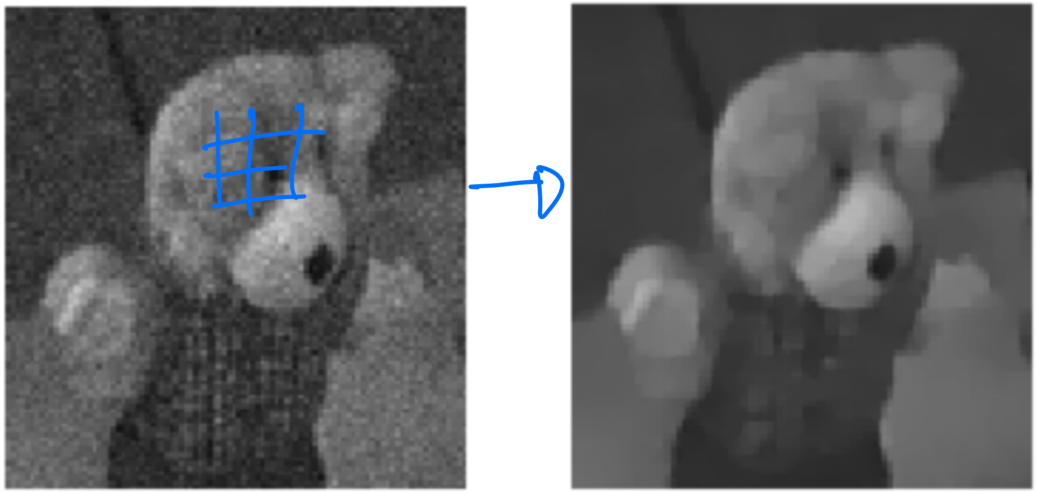


Image Smoothing Motivation

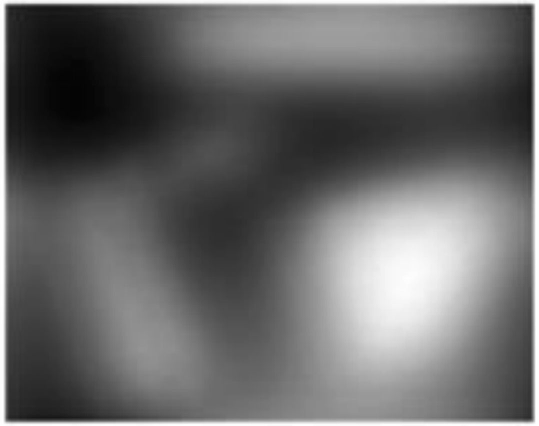




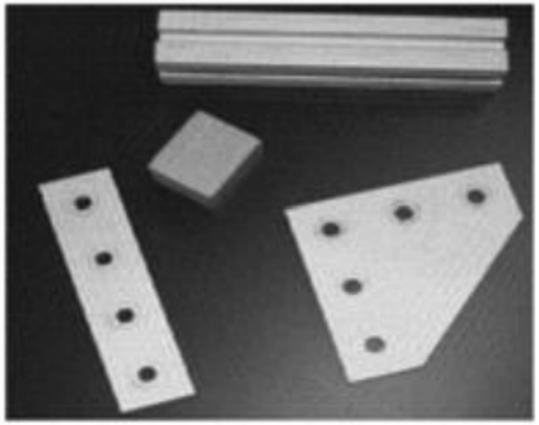
Google

{ Low Rank }
+ Sparse
Decomposition
for occlusions

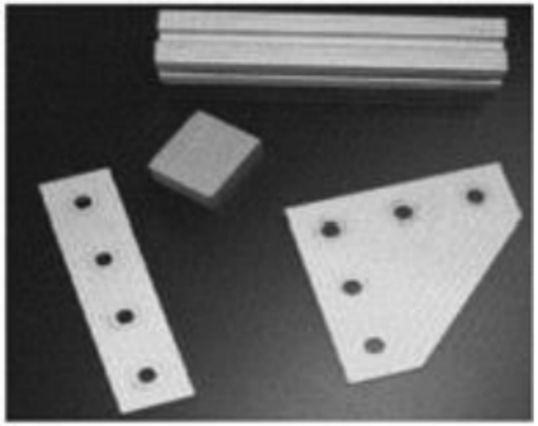
Image Smoothing Motivation



Blurred and Noisy image



Total Variation reconstruction



LASSO regularization reconstruction

- occlusion

Background for TV

Total Variation (TV)

Is based on the concept of Regularization and using ℓ_1 or ℓ_2 norms. We will look into this background next.

Norms

ℓ_1 norm

The ℓ_1 norm of a vector is the sum of the absolute values of the elements in the vector!

$$\|x\|_1 = \sum_i |x_i|$$

Norms

ℓ_1 norm

The ℓ_1 norm of a vector is the sum of the absolute values of the elements in the vector!

$$\|x\|_1 = \sum_i |x_i|$$

ℓ_2 norm

$$\|x\|_2 = \sqrt{\sum_i |x_i|^2}$$

Norms

ℓ_1 norm

The ℓ_1 norm of a vector is the sum of the absolute values of the elements in the vector!

$$\|x\|_1 = \sum_i |x_i|$$

ℓ_2 norm

$$\|x\|_2 = \sqrt{\sum_i |x_i|^2}$$

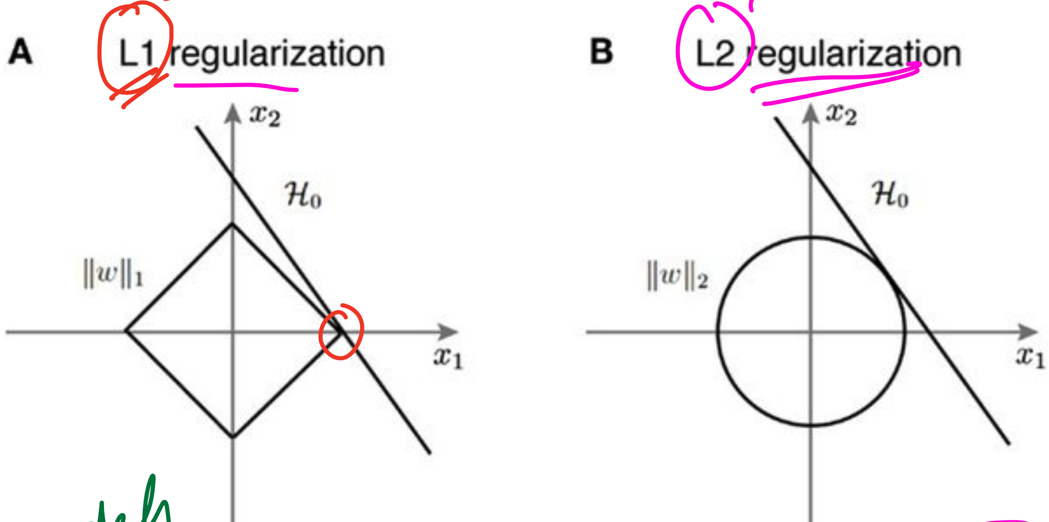
Notice!

$$\|x\|_2 \leq \|x\|_1$$

Norms and Norm Balls

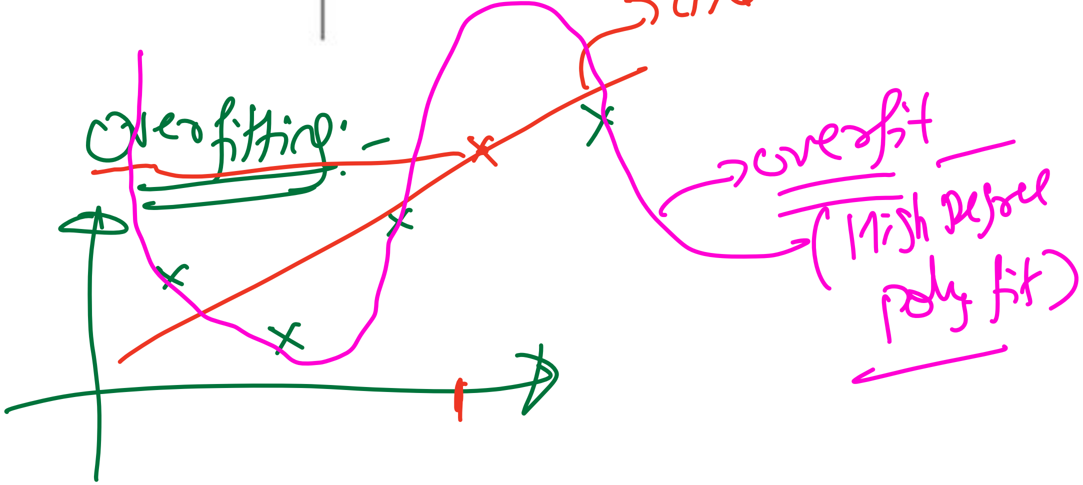
intuition → Sparsity

Prevent overfitting

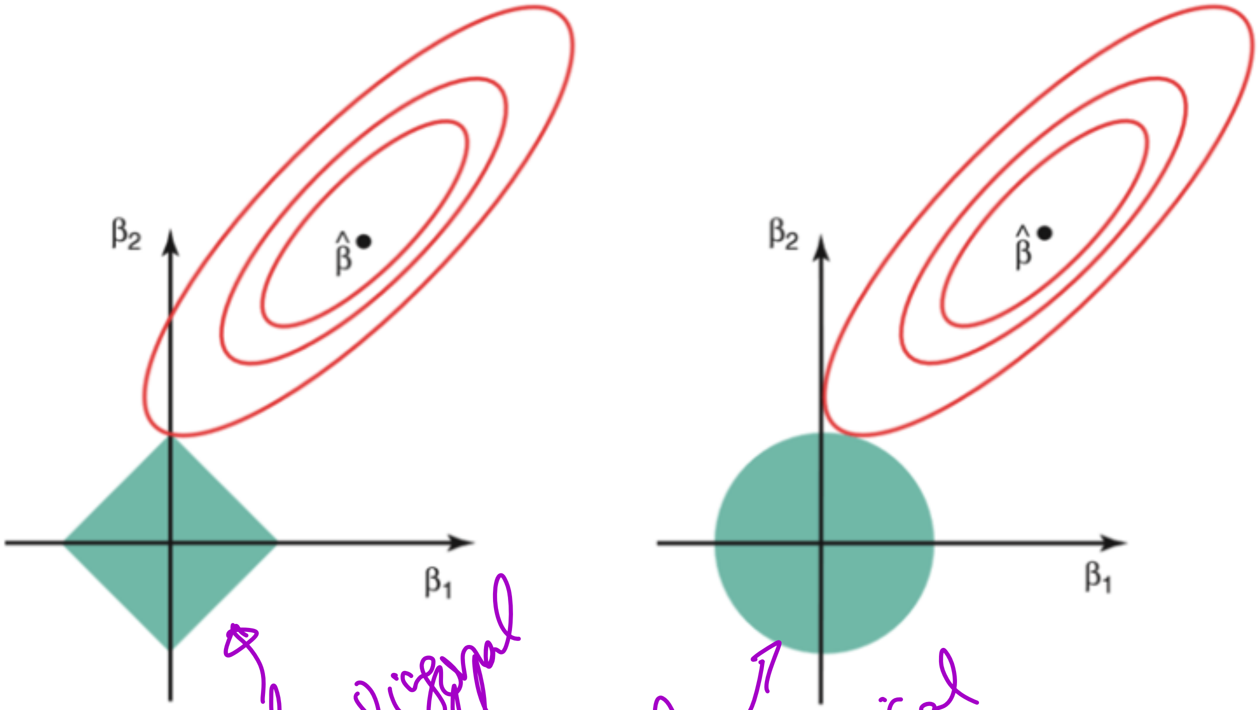


L_1 → Sparse models
→ Simpler models
→ Avoids overfitting

Overfitting: -



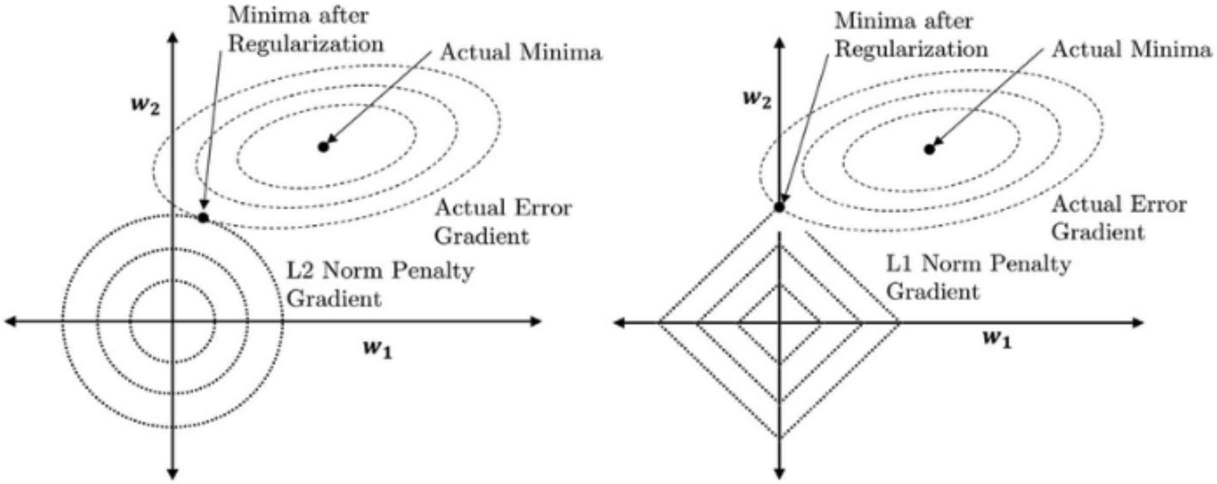
Norms and Norm Balls



l_1 - diagonal ball

l_2 Spherical

Norms and Norm Balls



ℓ_1 norm and sparsity

Sparsity as a regularizer

ℓ_1 norm for the reasons described in the previous slide is known to produce sparse solutions (i.e. a vector with a bunch of zeros).

ℓ_1 norm and sparsity

Sparsity as a regularizer

ℓ_1 norm for the reasons described in the previous slide is known to produce sparse solutions (i.e. a vector with a bunch of zeros).

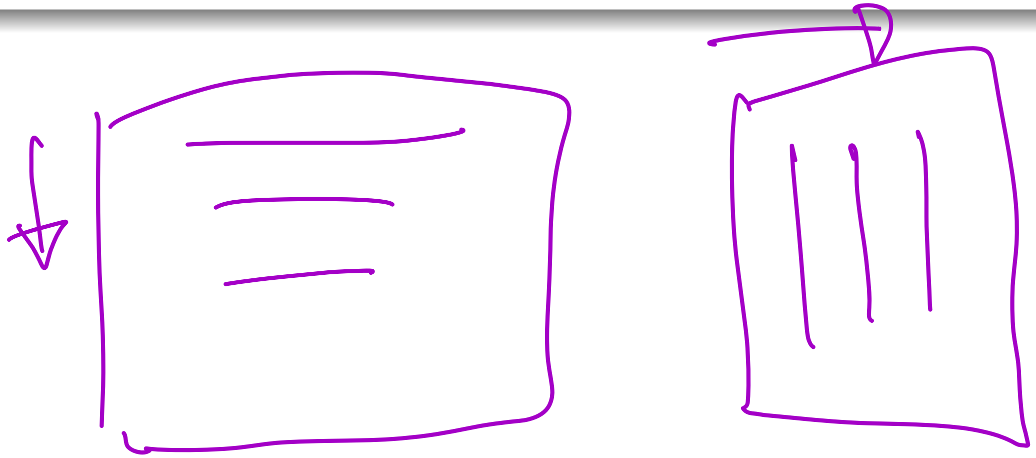
Sparsity for image denoising

This "sparsifying" property is what can be used to help denoise image. And that brings to TV!

Applying Sparsity to Image Smoothing

Horizontal Image Gradient Matrix, D_x

$$[D_x]_{i,j}(I) = I_{i+1,j} - I_{i,j}$$



Applying Sparsity to Image Smoothing

Horizontal Image Gradient Matrix, D_x

$$[D_x]_{i,j}(I) = I_{i+1,j} - I_{i,j}$$

Vertical Image Gradient Matrix, D_y

$$[D_y]_{i,j}(I) = I_{i,j+1} - I_{i,j}$$

Total Variation (TV) definitions

TV

Let A be a measurment matrix that measured an image and gave an output f. Given f and A, can you reconstruct the image u in such a way that it is denoised as well? To do this, we solve the following optimization problem!

$$\min_u \frac{1}{2}\|Au - f\|_2^2 + TV(u)$$

Total Variation (TV) definitions

Anisotropic TV

$$\min_{u} \frac{1}{2}\|Au - f\|_2^2 + TV_{aiso}(u)$$

where,

$$TV_{aiso}(u) = \|\underset{\sim\sim}{D_x(I)}\|_1 + \|\underset{\sim\sim}{D_y(I)}\|_1$$

Total Variation (TV) definitions

Anisotropic TV

$$\min_u \frac{1}{2}\|Au - f\|_2^2 + TV_{aiso}(u)$$

where,

$$TV_{aiso}(u) = \underline{\|D_x(I)\|_1 + \|D_y(I)\|_1}$$

Isotropic TV

$$\min_u \frac{1}{2}\|Au - f\|_2^2 + TV_{iso}(u)$$

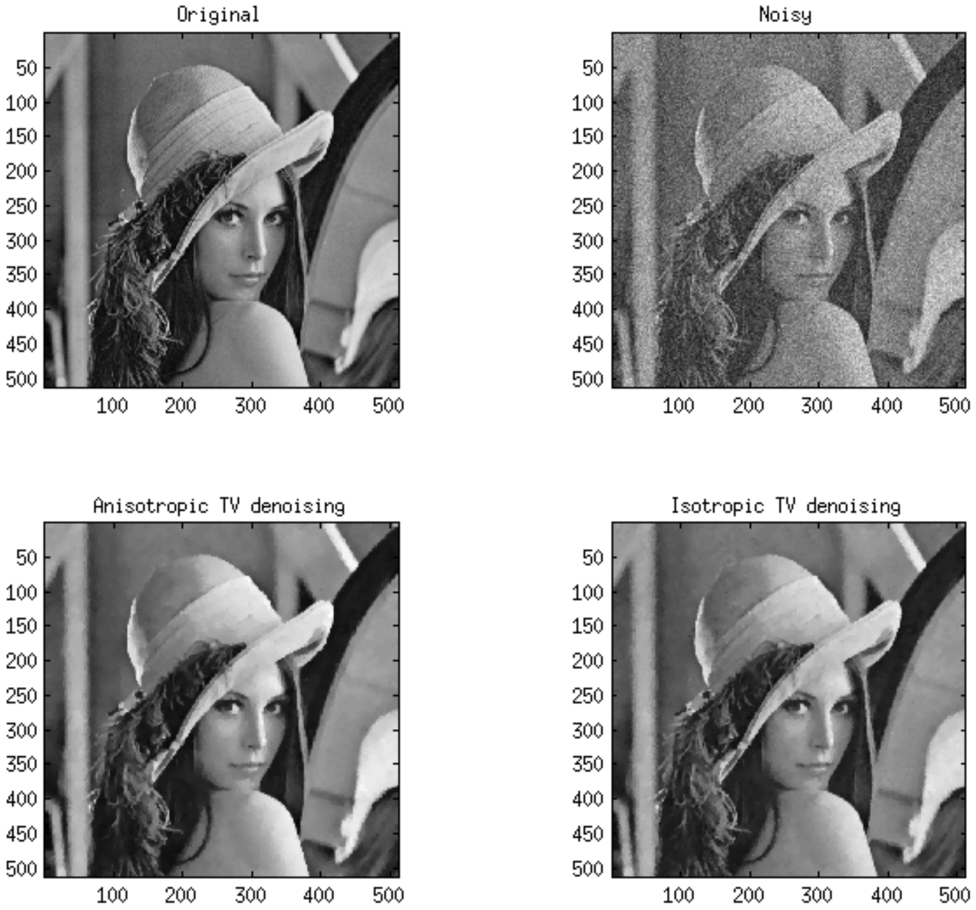
where,

$$TV_{\underset{iso}{\cancel{aiso}}}(u) = \|\underline{\sqrt{|D_x(I)|^2 + |D_y(I)|^2}}\|_1$$

Image Smoothing with TV

Using TV

Given the noisy image, using anisotropic and isotropic TV gives the results as below!



Original

Noisy

Anisotropic TV denoising

Isotropic TV denoising

ICE #0: Smoothing an Image

Smoothing an image

Consider an image, I given by the matrix:

$$I = \underbrace{\begin{bmatrix} \overset{\longrightarrow\phi}{1} & 2 & 1 \\ 2 & 3 & 2 \\ 3 & 4 & 3 \end{bmatrix}}$$

Consider two candidate smoothed versions of the I: I_1 and I_2. Let,

$$I_1 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 4 & 4 \end{bmatrix}, I_2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 3 \\ 3 & 3 & 4 \end{bmatrix}$$

Consider two metrics: $\|D_{\overset{y}{x}}(I)\|_1$ (or TV regularizer) and $\|\underbrace{I - \hat{I}}\|_2$ or denoising error. Which of the following statements are true:

ICE #0

original Image

Candidate 1

candidate 2

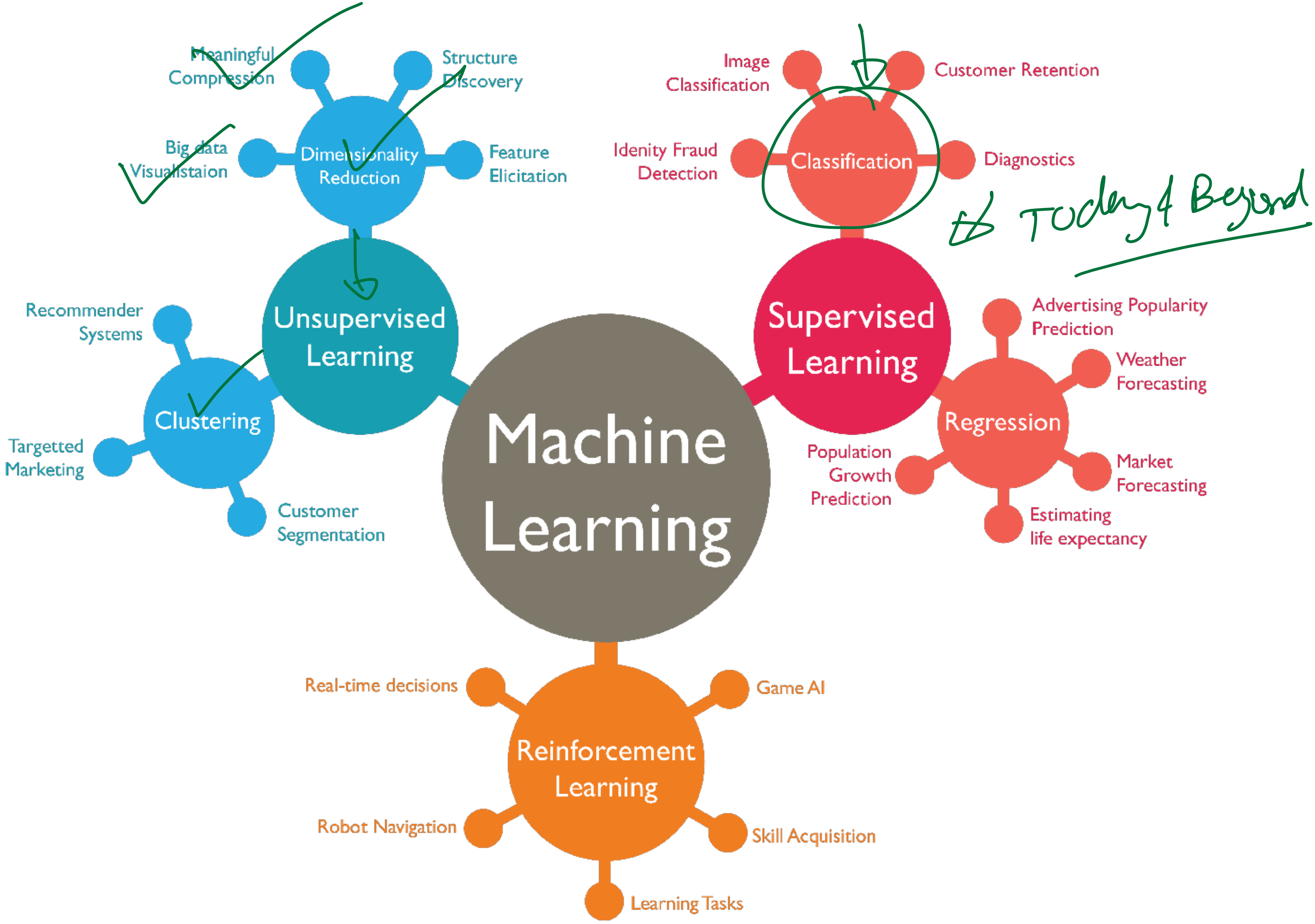
Smoothing an Image

$$I = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 3 & 4 & 3 \end{bmatrix}, I_1 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 3 & 4 & 4 \end{bmatrix}, I_2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 3 \\ 3 & 3 & 4 \end{bmatrix}$$

1. TV is better for I_1 but denoising error is better for I_2 $\rightarrow \|I_2 - I\|_2$
2. TV is better for I_2 but denoising error is better for I_1 $\rightarrow \|I_1 - I\|_2$
3. Both TV and denoising error are small for I_1 as compared to I_2
4. Both TV and denoising error are small for I_2 as compared to I_1

$TV_1 = \|D_y(I_1)\|_1$ $\quad$ $TV_2 = \|D_y(I_2)\|_1$

Next Topic: Supervised Learning and Classification!



Today & Beyond

Computer Vision Topics

1. Image Processing using convolutions ✓
2. Image De-noising ✓
3. Image Smoothing ✓
4. Image Clustering ✓
5. Image Classification
6. Object Detection
7. Semantic Segmentation
8. Instance Segmentation (maybe)
9. Image Embeddings
10. Image to Text
11. Image Captioning
12. Text to Image (high-level)

Flower Classification

IRIS!

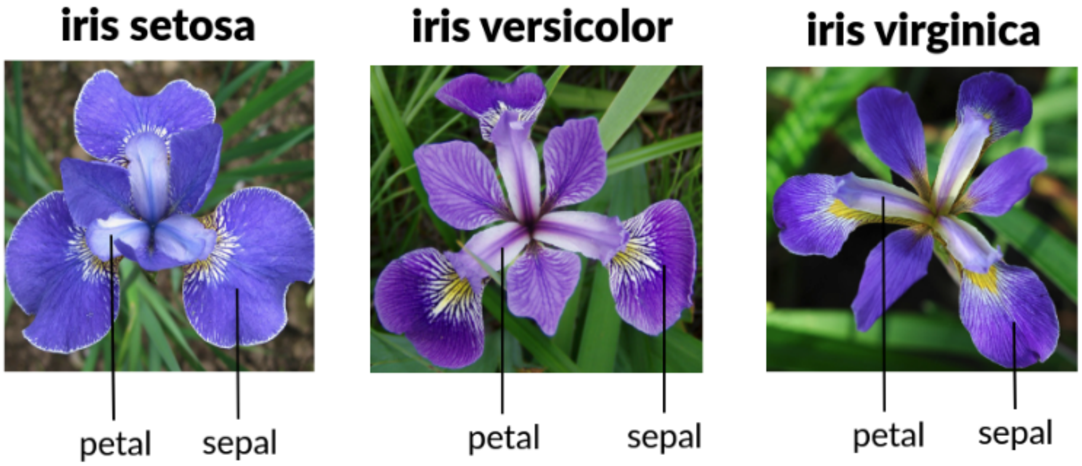
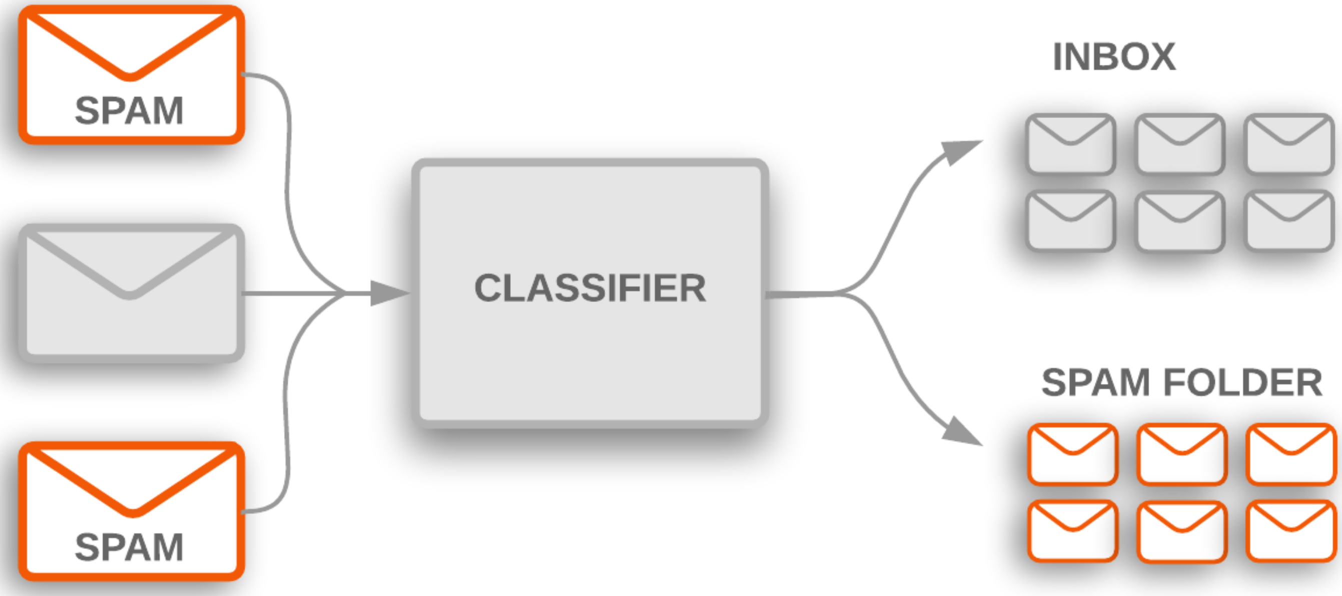


Image Classification → Meta-Data of the Images (1)

→ Actual Images as Data (2)

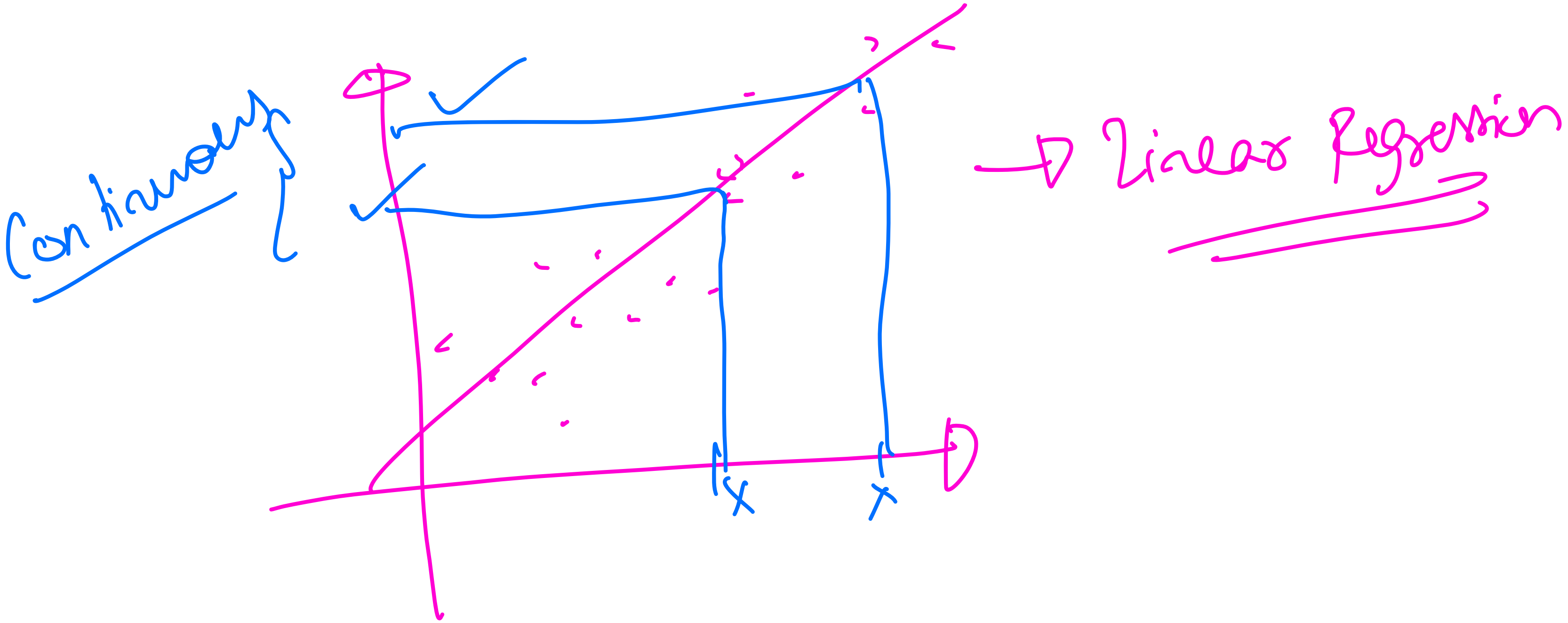
Classification in Machine Learning



Difference between Classification and Regression

Simple difference

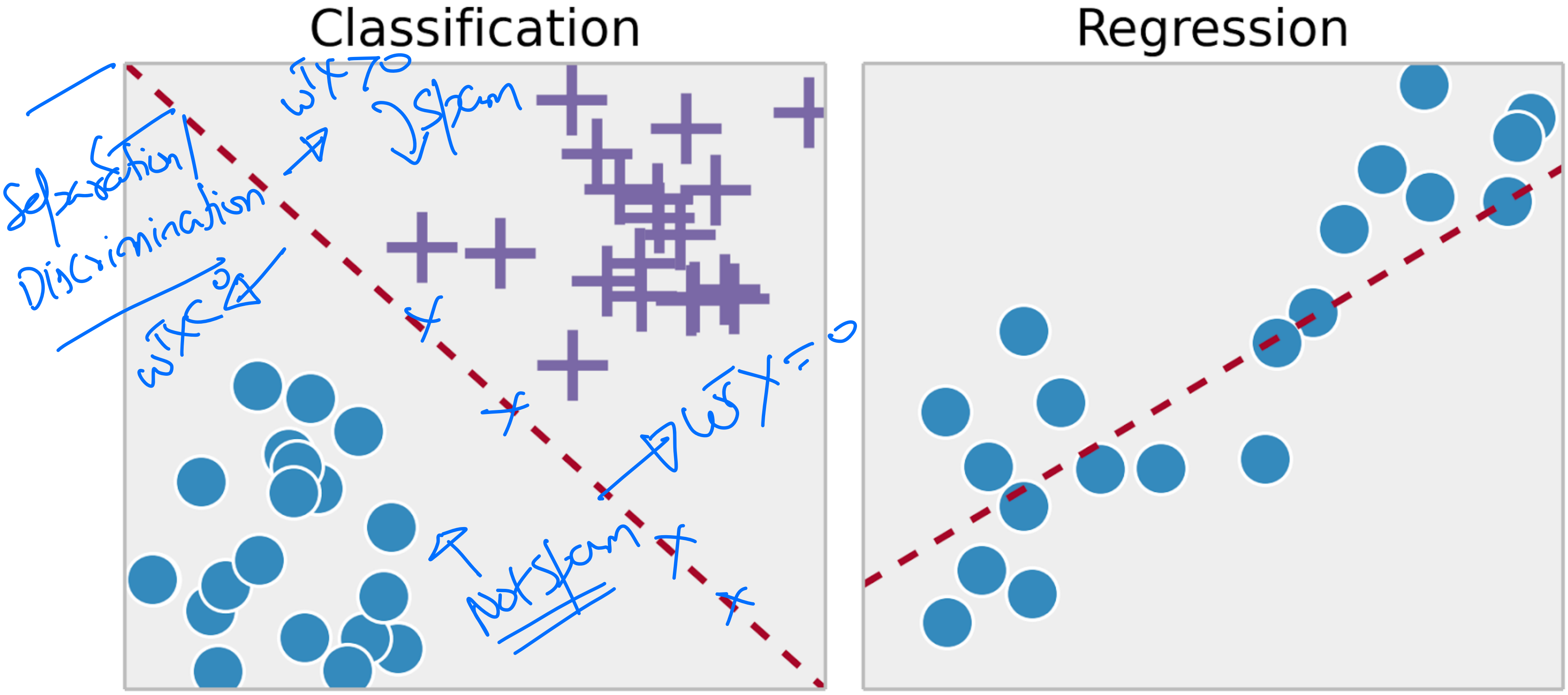
The target type in Regression is **numeric** whereas that in classification is **categorical**



Difference between Classification and Regression

Simple difference

The target type in Regression is **numeric** whereas that in classification is **categorical**



Types of Classification

Spam or Not Spam

Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

Iris1 Iris2 Iris3

Types of Classification

Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

Target is called Label

For binary classification, the convention is to label the target as positive or negative. Example: Positive for spam and negative for not-spam

Iain-1 → 0

Iain-2 → 1

Iain-3 → 2

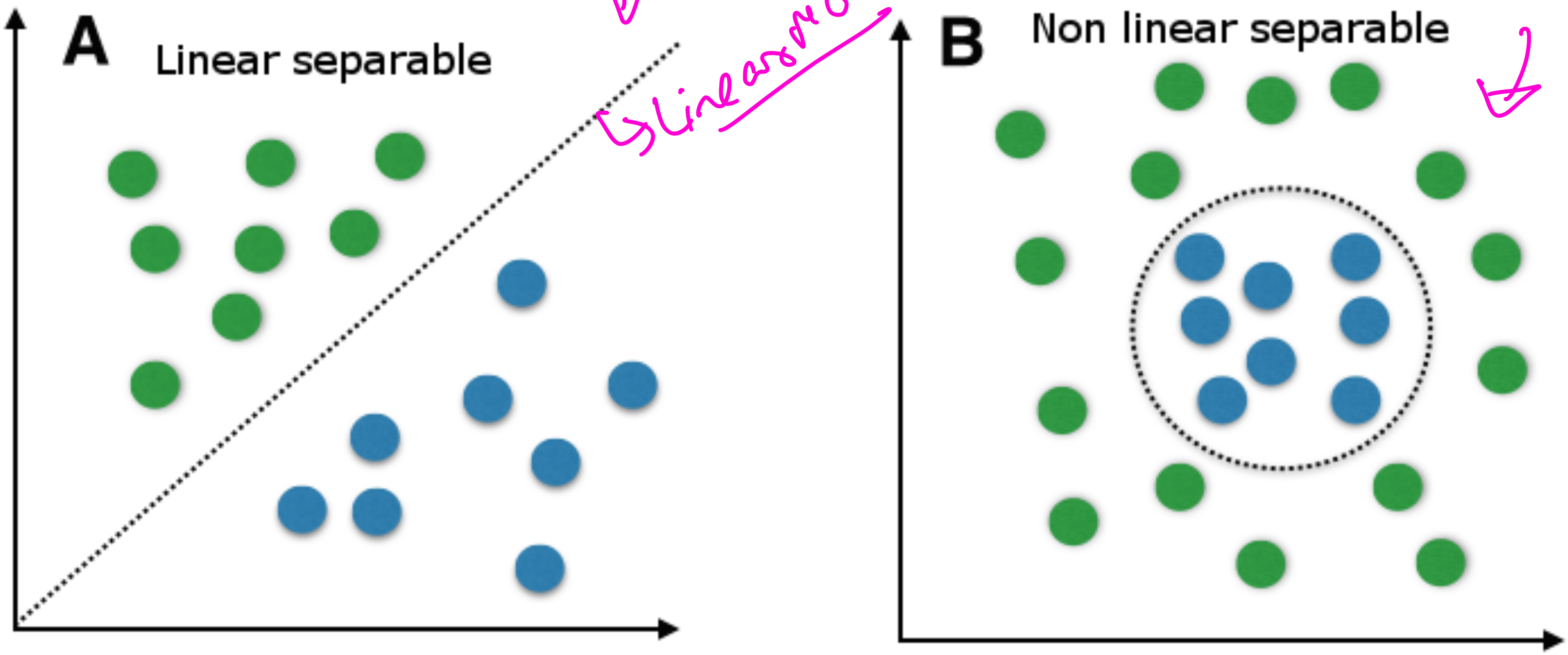
Spam Classification Example

Training Data

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

Linear Separability

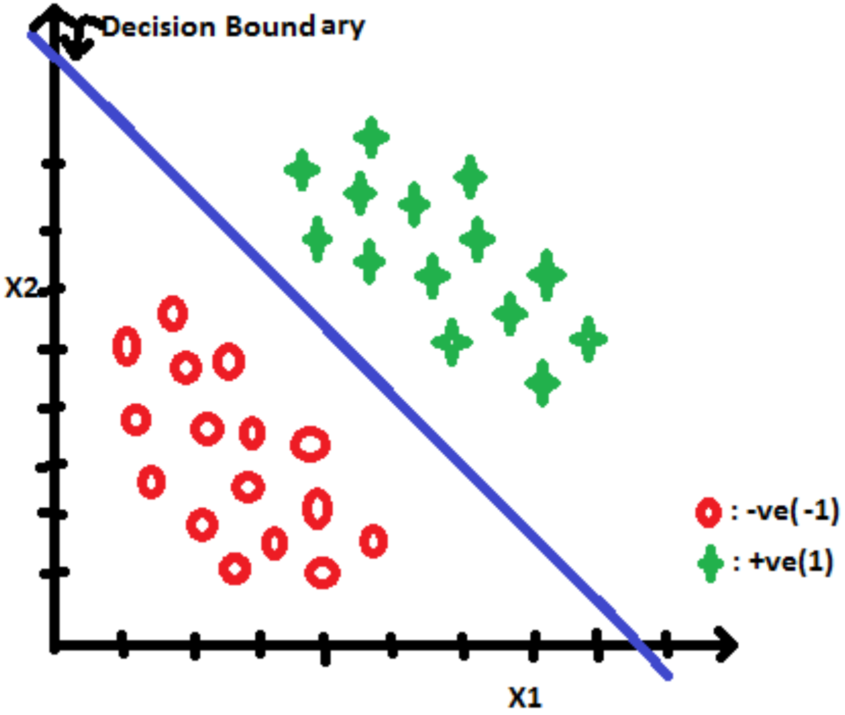
You can separate "C classes with a "linear model"



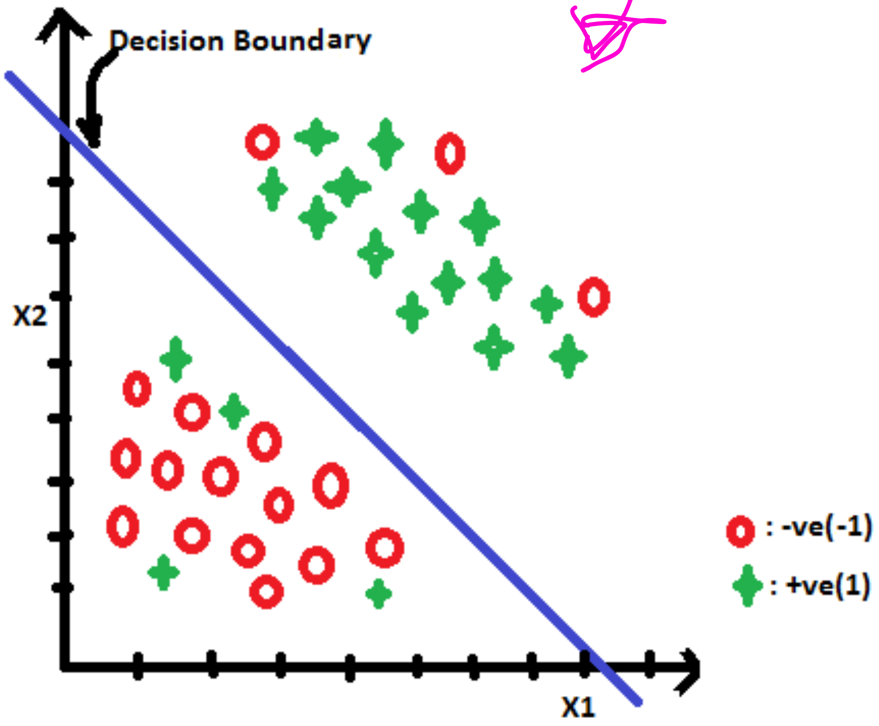
↳ linear model

Approximate Linear Separability

Real world classifiers



Perfect Linearly separable



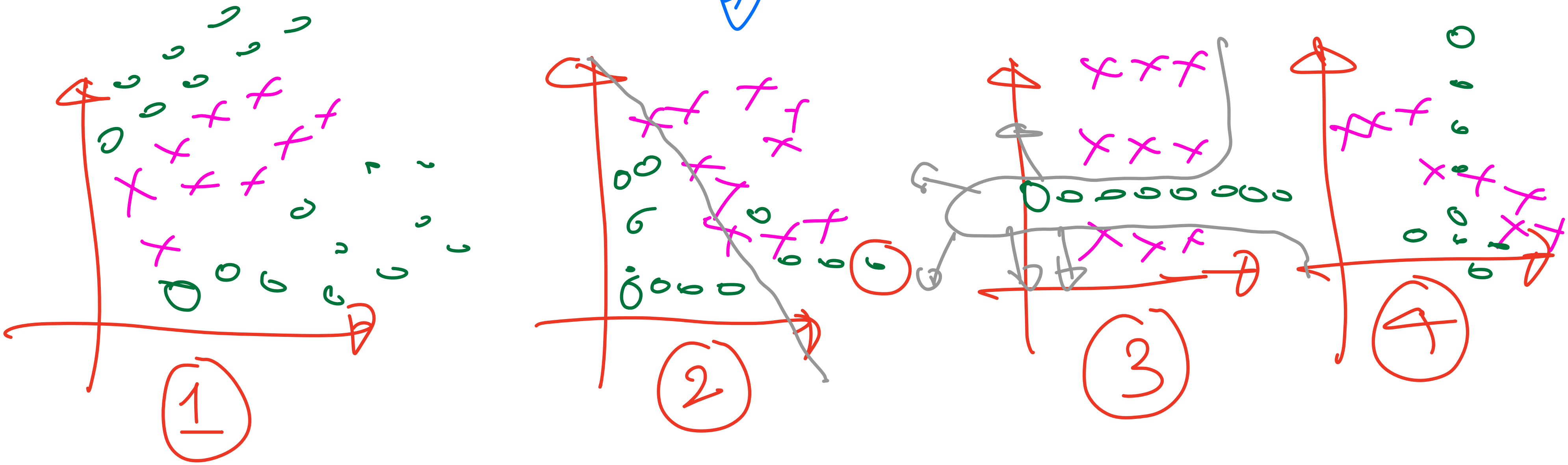
Almost Linearly Separable

ICE #1

("Non-linear model")

Good candidate for a ("Linear model")

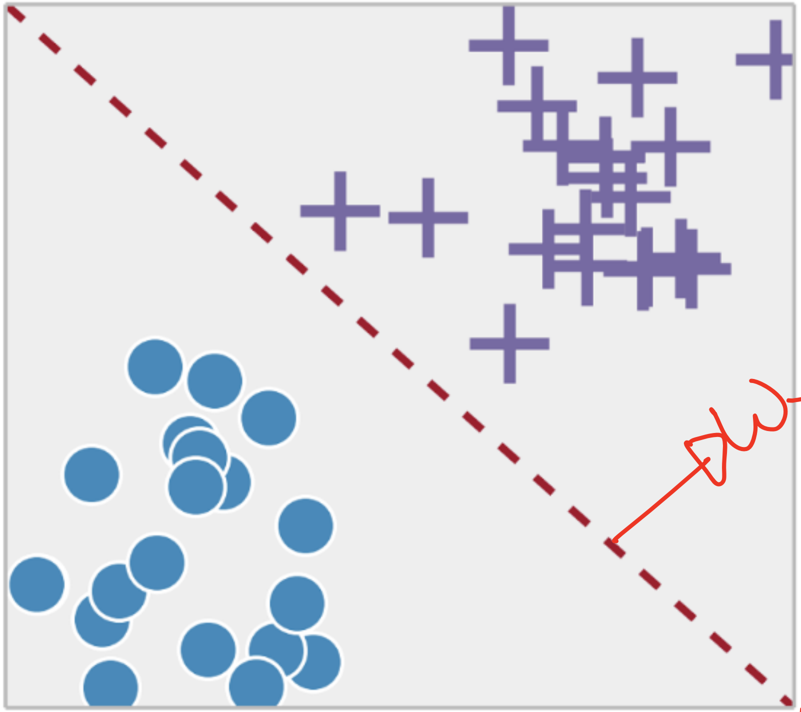
Which of the following data sets is the closest to being linearly separable?



Logistic Regression

Logit

Log(Probability)



w — Learn this!

LR fundamentals

Score

Label/Target

- Linear Model
- Want score $w^T x^i > 0$ for $y_i = +1$ and $w^T x_i < 0$ for $y_i = -1$!
- If linearly separable data, above is feasible. Else, minimize error in separability!!

Logistic Regression

Probability for a class

In LR, the score, $w^T x$ is converted to a probability through the sigmoid function. So we can talk about $P(\hat{y}^i = +1)$ or $P(\hat{y}^i = -1)$

Sigmoid Function

$\downarrow$

$$\frac{1}{1+e^{-z}}$$

$-\infty \le w^T x_i \le +\infty$

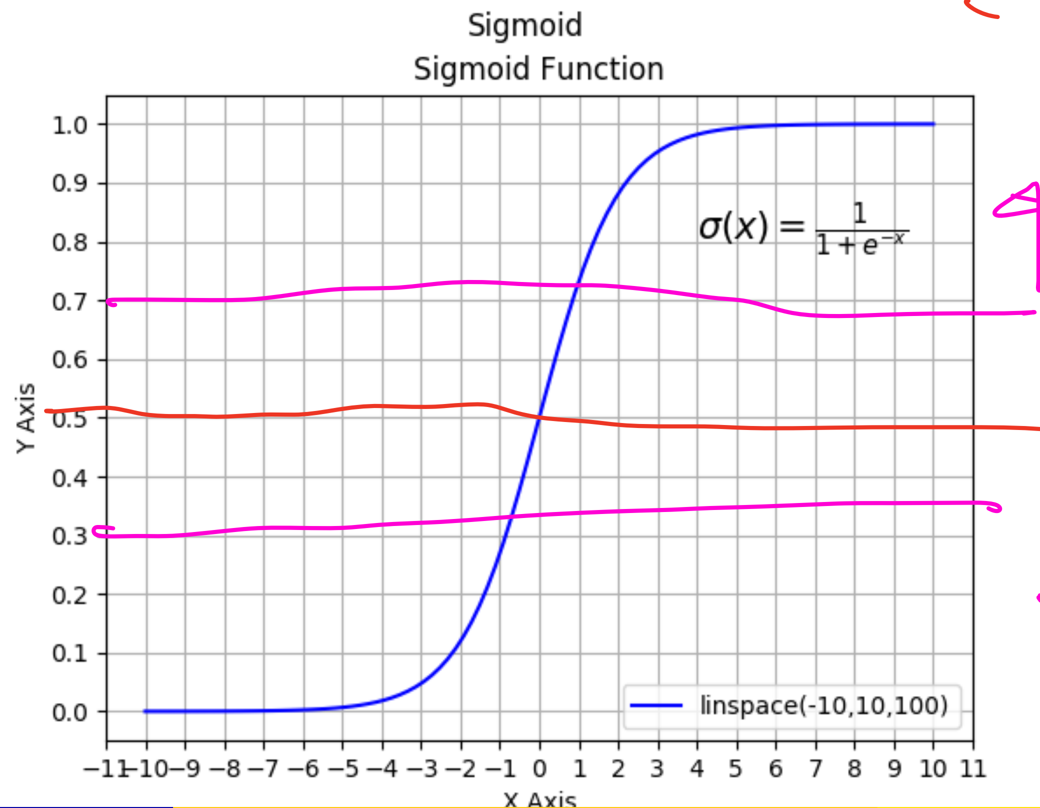
$\hookrightarrow$ Data i

$\downarrow$

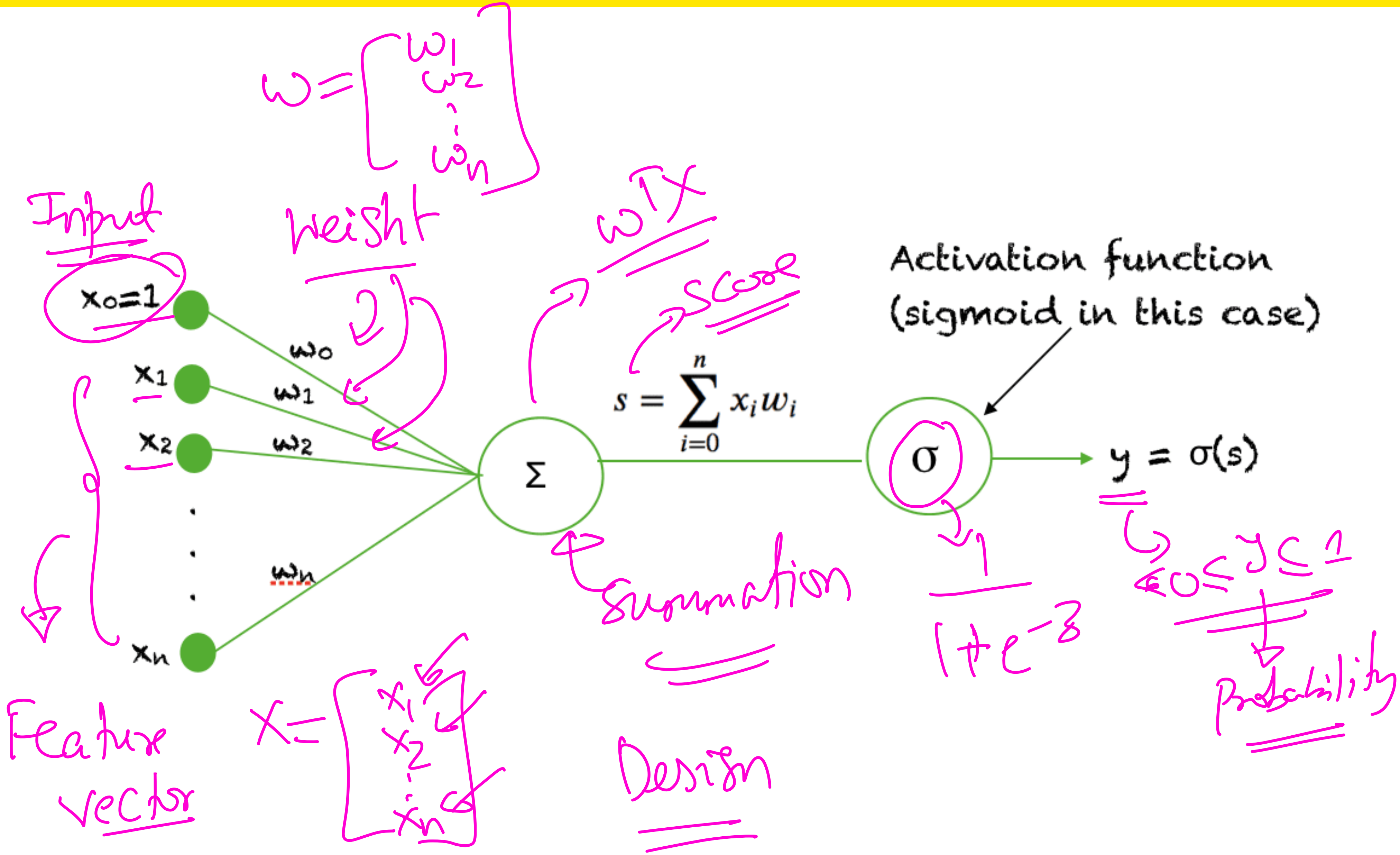
Find +ve

@ Threshold

$\downarrow$ -ve



LR represented Graphically



$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

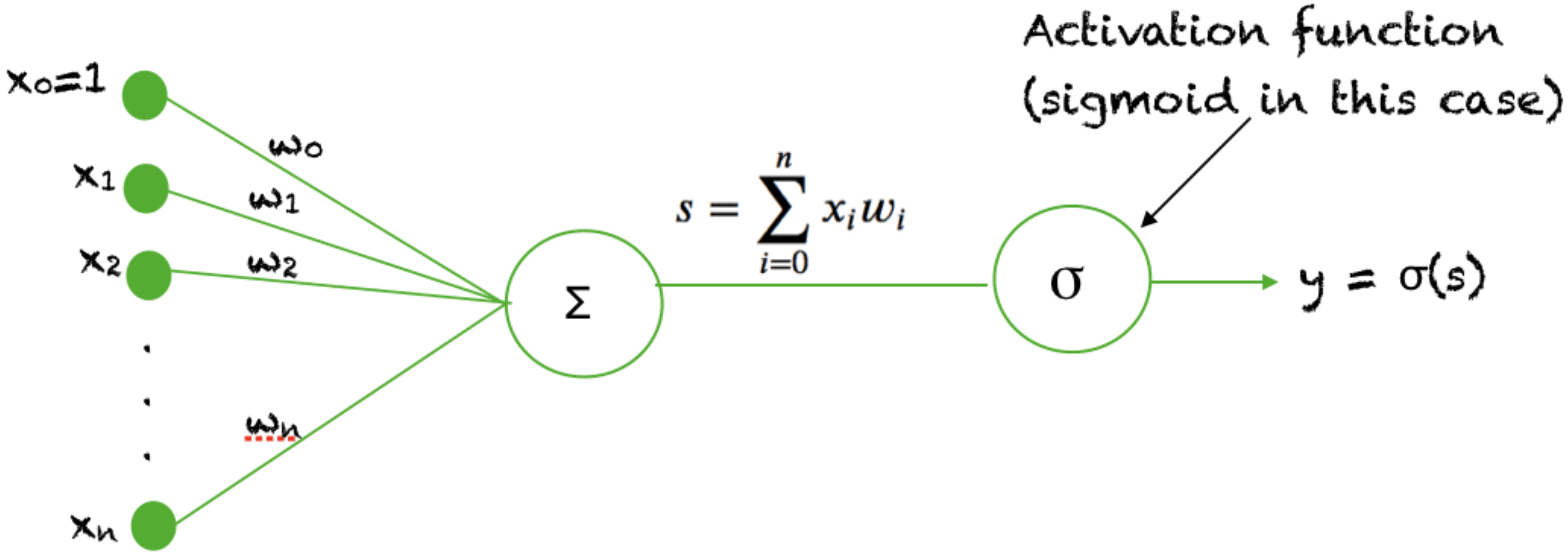
$$s = \sum_{i=0}^{n} x_i w_i$$

Activation function (sigmoid in this case)

$\frac{1}{1+e^{-s}}$

LR vs Neural Networks/Deep Learning

LR ⟷ 1 layer Neural Network



Logistic Regression

(-1 or 1)

y_i → truth (0 or 1)

LR Prediction

$$\boxed{\hat{y}_i} = \frac{1}{1 + e^{-\hat{w}^T x^i}} \longrightarrow \text{Probability} \quad 0 \leq \hat{y}_i \leq 1$$

LR Loss function - a.k.a what function do you optimize to learn a classifier?

LR Loss is based on the **cross-entropy** function

Entropy Function

Entropy

What is it a measure of? → uncertainty

↑Entropy ↑uncertainty

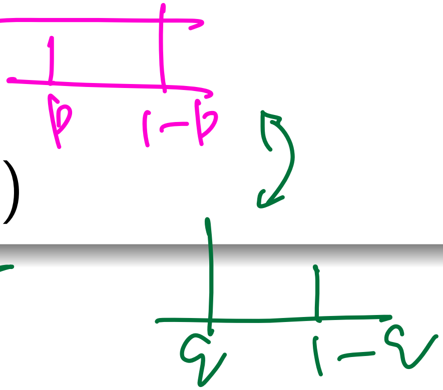
$$\underline{H(\mathbf{p}) = \sum_i -p_i \log(p_i)} \qquad \geq 0$$

where $\mathbf{p} \in \mathbb{R}^K$ is a probability distribution over K objects (e.g. K classes)

Binary Cross Entropy

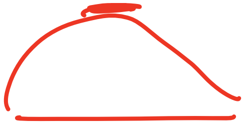
Binary Cross Entropy is a measure of distance between two binary probability distributions!

$$H(p, q) = \underbrace{-p\log(q) - (1-p)\log(1-q)}$$



ICE #13

$$H(p) = -\sum_i p_i \log p_i$$



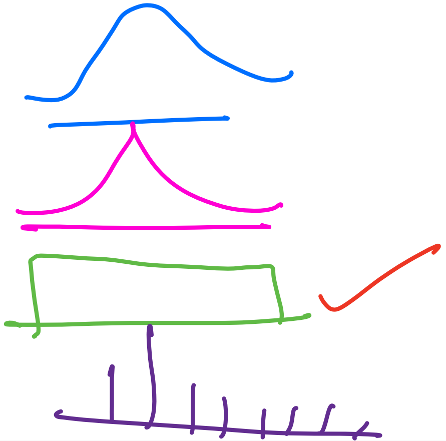
$$\nabla_p H(p) = 0$$

$$-p_i 1 + \log p_i = 0$$

Max Entropy

Which of the following distributions have the maximum entropy among all probability distributions?

1. Gaussian distribution
2. Laplace distribution
3. Uniform distribution ✓
4. Binomial distribution



Binary Cross Entropy Loss Function

Truth

LR Loss

Assume that $y_i = 0$ or $y_i = 1$ (i.e. the negative class has a label 0). Then the binary cross-entropy loss applies to LR:

$$\min_w \sum_i y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

Prediction

$0 \le \hat{y}_i \le 1$

$$\hat{y}_i = \frac{1}{1+e^{-w^T x_i}}$$

Binary Cross Entropy Loss Function

LR Loss

Assume that $y_i = 0$ or $y_i = 1$ (i.e. the negative class has a label 0). Then the binary cross-entropy loss applies to LR:

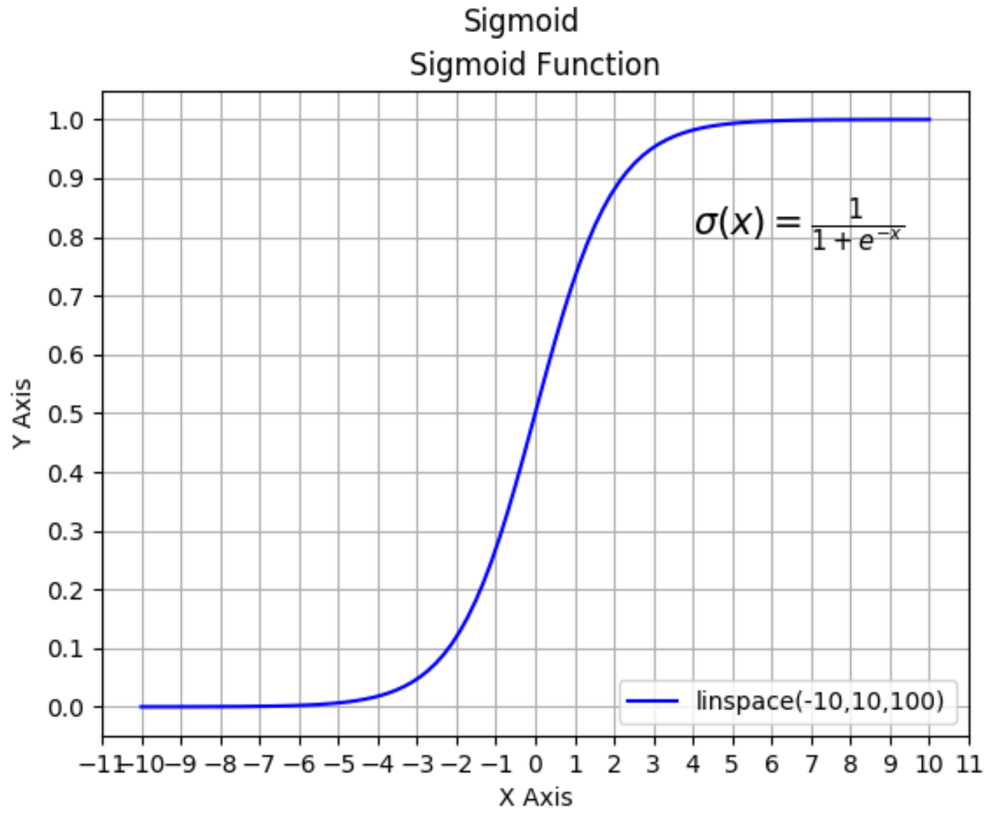
$$\min_w y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Minimizing the LR loss

What distribution minimizes the LR loss function?

Score to a Probability

Sigmoid Function



Logistic Regression Loss Function

LR Loss

Assume that $y_i = 0$ or $y_i = 1$ (i.e. the negative class has a label 0). Then the binary cross-entropy loss applies to LR:

$$\min_w \sum_{i=1}^{N} -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)$$

Logistic Regression Loss Function

LR Loss

Assume that $y_i = 0$ or $y_i = 1$ (i.e. the negative class has a label 0). Then the binary cross-entropy loss applies to LR:

$$\min_w \sum_{i=1}^{N} -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)$$

Let's understand the loss better?

Logistic Regression || Probabilistically Speaking

Probability of a class

$$P(\hat{y}_i = 1) = \frac{1}{1 + e^{-\hat{w}^T x^i}}$$

Logistic Regression || Probabilistically Speaking

Probability of a class

$$P(\hat{y}_i = 1) = \frac{1}{1 + e^{-\hat{w}^T x^i}}$$

Predictions vs True Labels

We want $\hat{y}_i = 1$ when $y_i = 1$ and $\hat{y}_i = 0$ when $y_i = 0$ - For as many data points as possible, isn't it? This means the classsifier is making good predictions!

Logistic Regression || Probabilistically Speaking

Probability of a class

$$P(\hat{y}_i = 1) = \frac{1}{1 + e^{-\hat{w}^T x^i}}$$

Predictions vs True Labels

We want $\hat{y}_i = 1$ when $y_i = 1$ and $\hat{y}_i = 0$ when $y_i = 0$ - For as many data points as possible, isn't it? This means the classsifier is making good predictions!

Predictions vs True Labels

What's the joint probability that $\hat{y}_i = 1$ when $y_i = 1$ and $\hat{y}_i = 0$ when $y_i = 0$?

Logistic Regerssion — Connection to Maximum Likelihood

LR as Maximum Likelihood Estimate

Discussed in depth in the "Advacned Intro to ML" course next quarter!

ICE #2 (2 mins)

Handling the math

Let $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$. What's the expression for $\log(1-\hat{y}_i)$? (Working out the math on paper is recommended here!)

a) $\frac{e^{-\hat{w}^T x^i}}{1+e^{-\hat{w}^T x^i}}$

b) $\log\left(\frac{e^{-\hat{w}^T x^i}}{1+e^{-\hat{w}^T x^i}}\right)$

c) $\log\left(\frac{1}{1+e^{\hat{w}^T x^i}}\right)$

d) $\log\left(\frac{1}{1+e^{-\hat{w}^T x^i}}\right)$

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!
6. Logistic Regression uses the log-loss or cross-entropy loss whereas Linear Regression uses the quadratic loss

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!
6. Logistic Regression uses the log-loss or cross-entropy loss whereas Linear Regression uses the quadratic loss
7. Logistic Regression loss can be derived as a MLE - So its well grounded in statistics.

Evaluating Classifiers!

ICE #3

Let's say you own an email server and want to provide a service to your email customers to help sort their emails into spam vs not-spam. So you go ahead and build a spam classifier on a training data set. Your data set has 100 spam emails and 900 non-spam emails. You notice that your classifier has 90% accuracy on the training data set and also your validation data set. Should you be happy with your classifier?

a) Yes

b) No

c) Maybe!

d) Something's fishy!

Evaluating classifiers

Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

Evaluating classifiers

Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

Better metric than accuracy

Consider the **confusion matrix** for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Better metric than accuracy

Accurcay is how many data points the classifier got right divided by the total data points. What's accuracy here?

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Accuracy, Precision, Recall and F1-score

	Predicted Positive	Predicted Negatives
Positives (P)	TP	FN
Negatives (N)	FP	TN

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Accuracy, Precision, Recall and F1-score

Precision (Pr) = TP/(TP + FP)
Recall (R) = TP/(TP + FN) = TP/P
F1-score = $\frac{2 \times Pr \times R}{Pr + R}$
Accuracy (Acc) = $(TP + TN)/(P + N)$

ICE #4

More Confusion!

Let's say we computed a **Confusion Matrix** for another Spam Classifier on a different data set and we obtained:

	Predicted Positive	Predicted Negatives
Positives (P)	50	50
Negatives (R)	100	400

Metrics!

Accuracy, **Pr**, **R** and **F1** are as follows:

a) $75\%, 0.2, 0.5, 0.285$

b) $80\%, 0.3, 0.4, 0.285$

c) $80\%, 0.5, 0.3, 0.1875$

d) $75\%, 0.3, 0.5, 0.1875$

Summary

1. Total Variation for Image denoising and smoothing
2. Supervised Learning and Binary Classification
3. Logistic Regression
4. Metrics for measuring goodness of a classifier
5. Confusion Matrix