Lecture 9 – Classification and Regression

Bulat Ibragimov

bulat@di.ku.dk

Department of Computer Science
University of Copenhagen

Objectives

What is classification

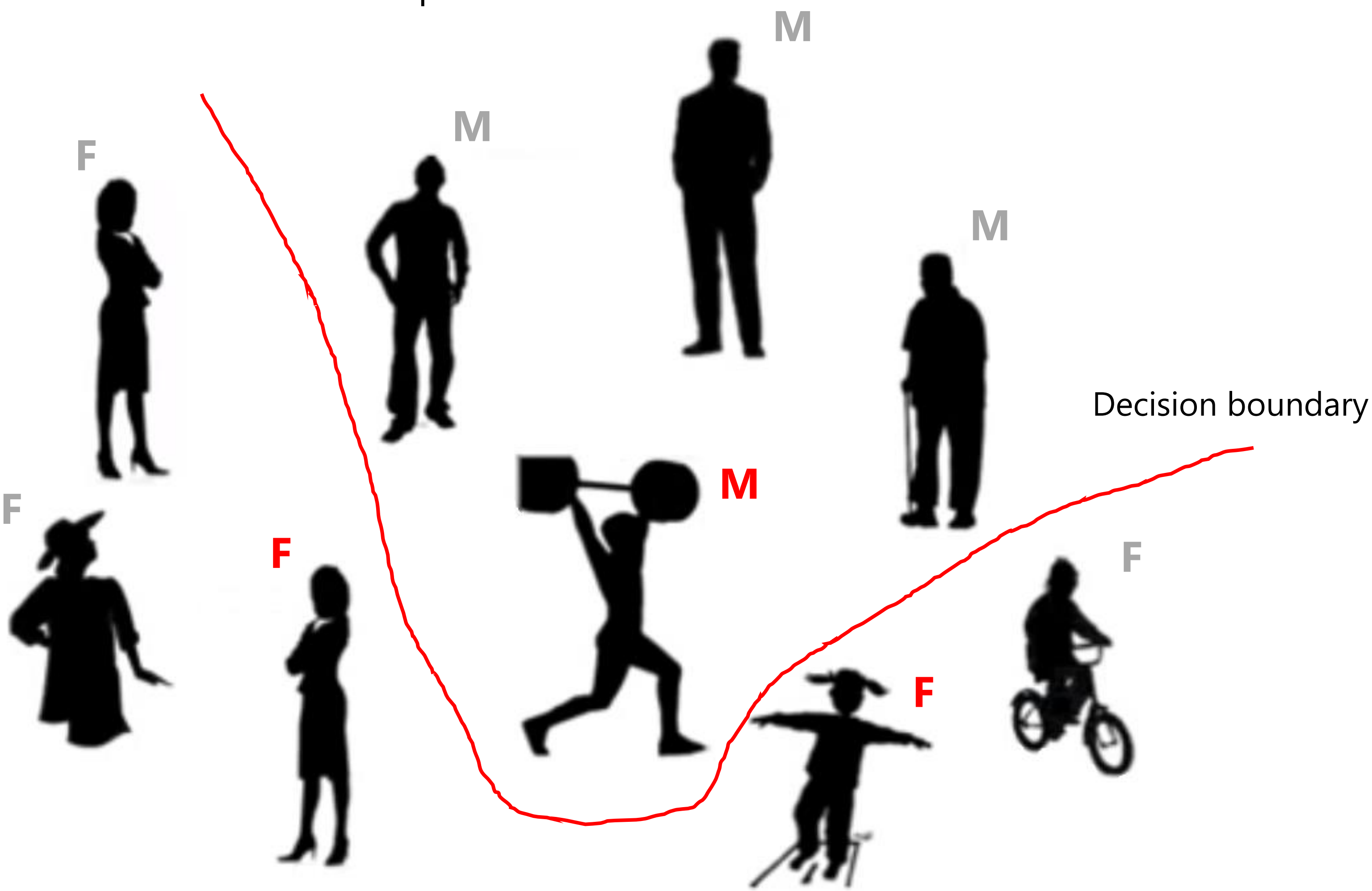
K-nearest neighbours

Support vector machines

Classification performance evaluation

Classification

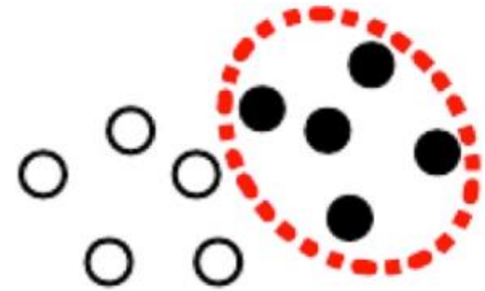
- Supervised
  - We have a database with samples and their labels



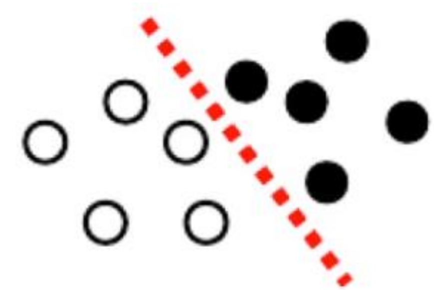
http://homepages.inf.ed.ac.uk/vlavrenk

Generative vs Discriminative models

- Generative:
  - Computes probabilistic model for each class
  - Can use unlabeled data



- Discriminative:
  - Focus on separation of classes
  - Cannot use unlabeled data



Training/Validation/Testing datasets

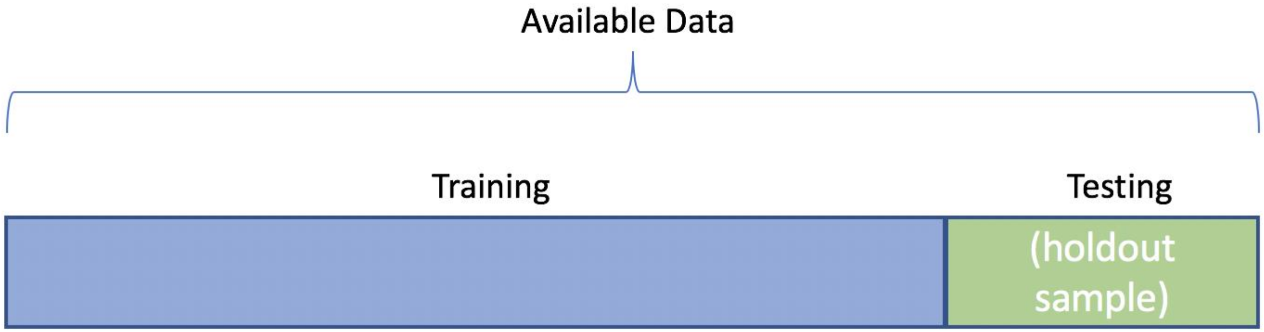
- Training dataset:
  - Model can use both training features and labels for changing its parameters.
- Validation dataset:
  - Needed to estimate how suitable is the selected model for solving the target problem.
- Testing dataset:
  - Can only be used when the model is completely finalized. Cannot be used to update anything about the model

Train-validation-test

Training Dataset: the part of the data used to optimize model f parameters.

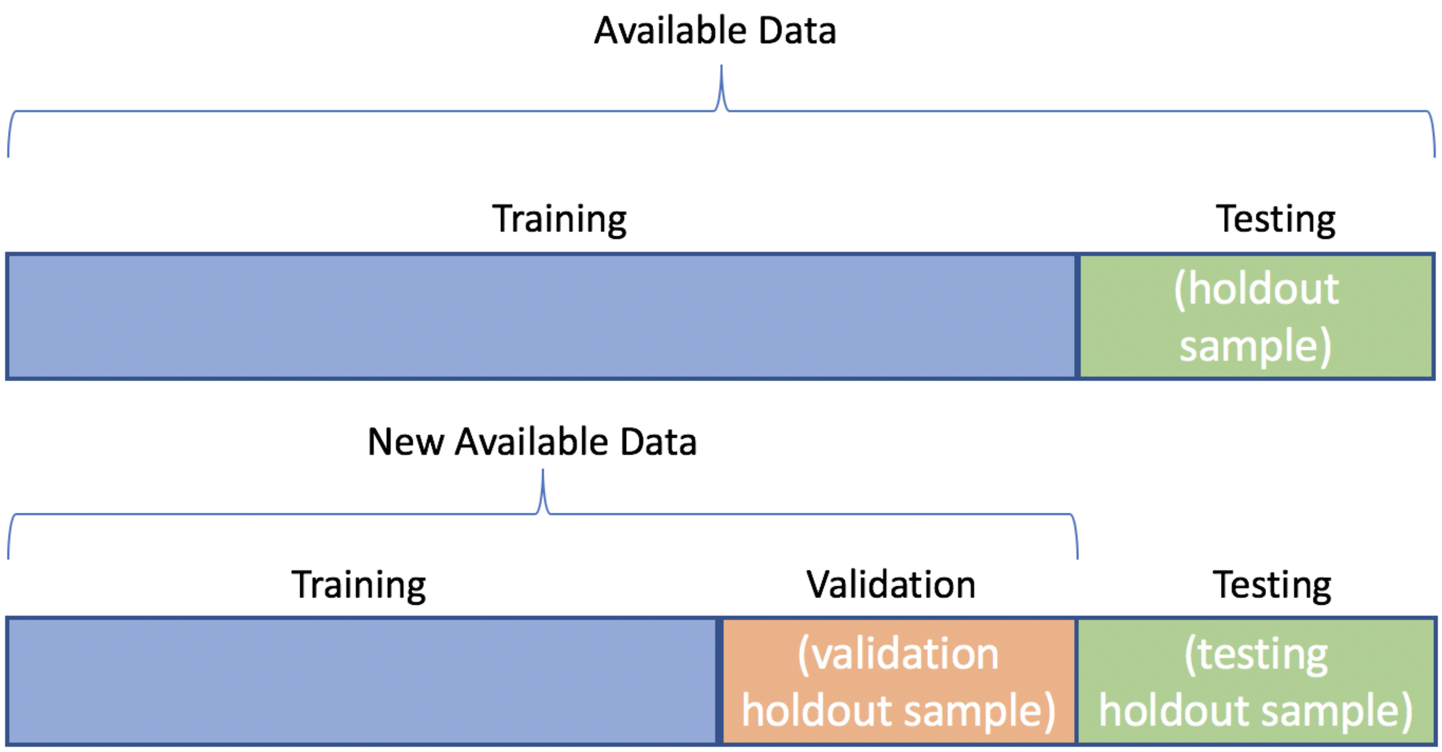
Testing Dataset: the part of the data used to evaluate how good the model f work.

If the model f has too many parameters it may perfectly capture the training dataset, but works poorly on the testing dataset. Imagine that you do not have access to the test dataset at all



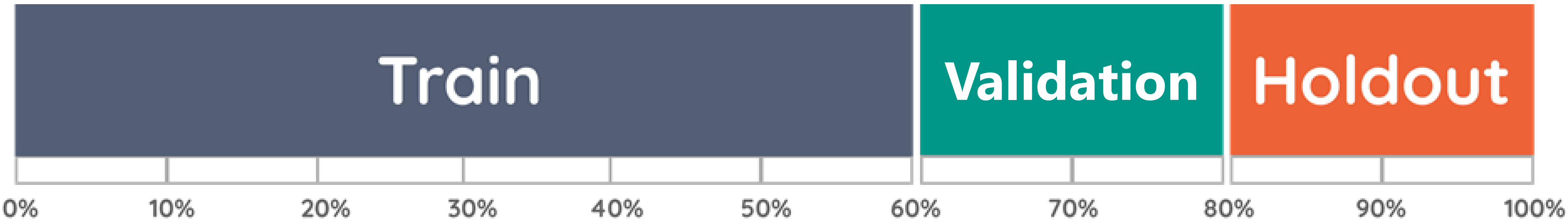
Train-validation-test: hold-out test

Training Dataset
Testing Dataset
Validation Dataset: the part of the data for estimating the performance of the model before final evaluation on the test dataset.

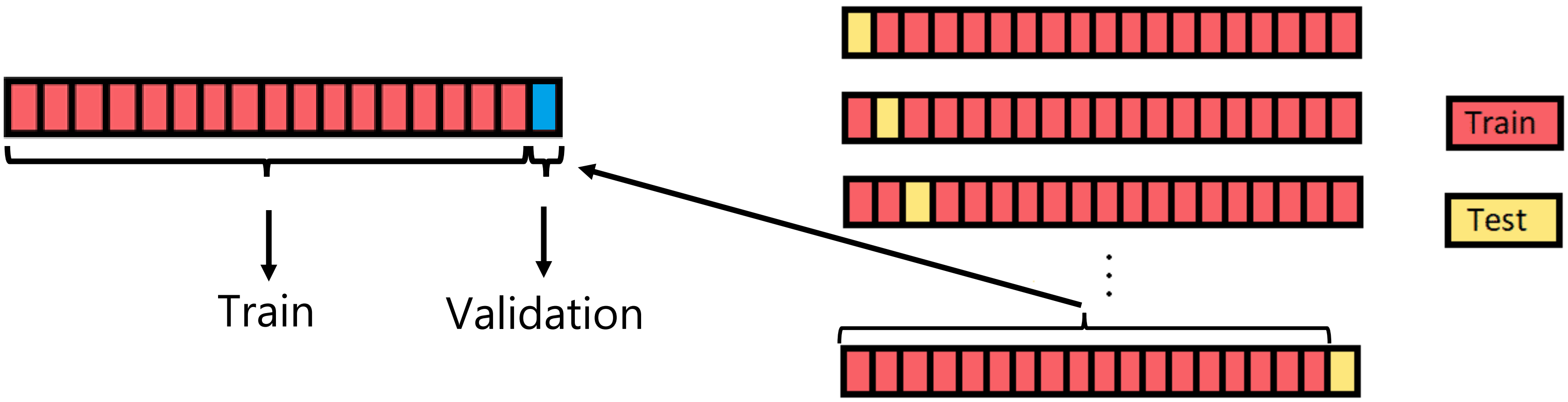


Hold-out test

The simplest strategy for model evaluation is to separate test data in advance from the complete database and only analyze it when the model development is finished



Cross-validation test



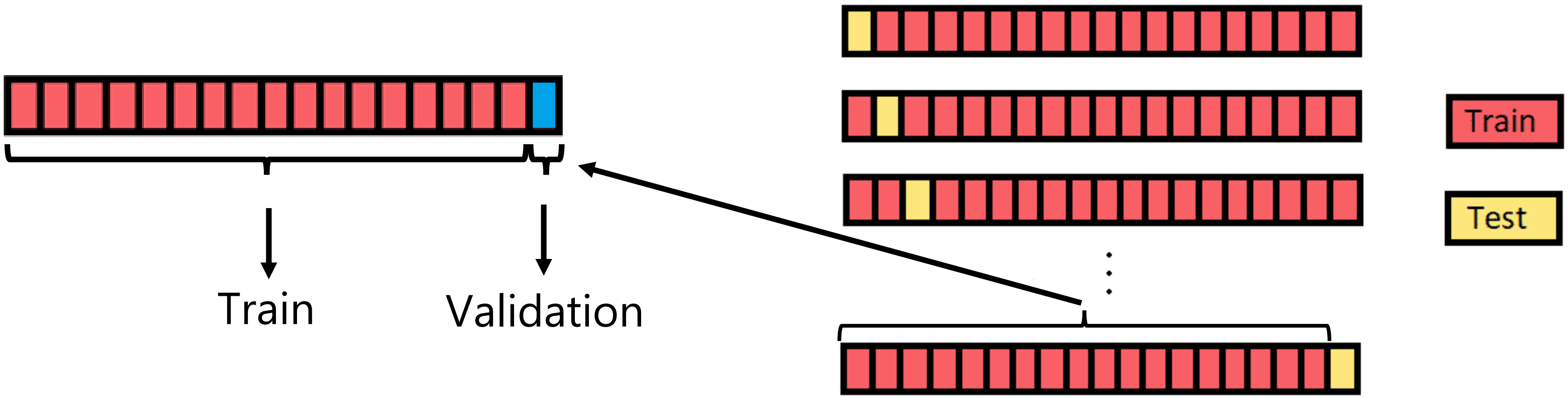
Cross-Validation

K-fold cross-validation splits the data into K (almost) equally-sized parts. We consider K "rounds":

1. Use $K-1$ parts for training the model. For instance, the optimal weight vector $\mathbf{w}$ is computed for linear regression.
2. Use the remaining part for validating the model by computing the induced loss on this part.

This yields K validation errors. Typically, the average of these values is considered.

Cross-validation test



Cross-Validation

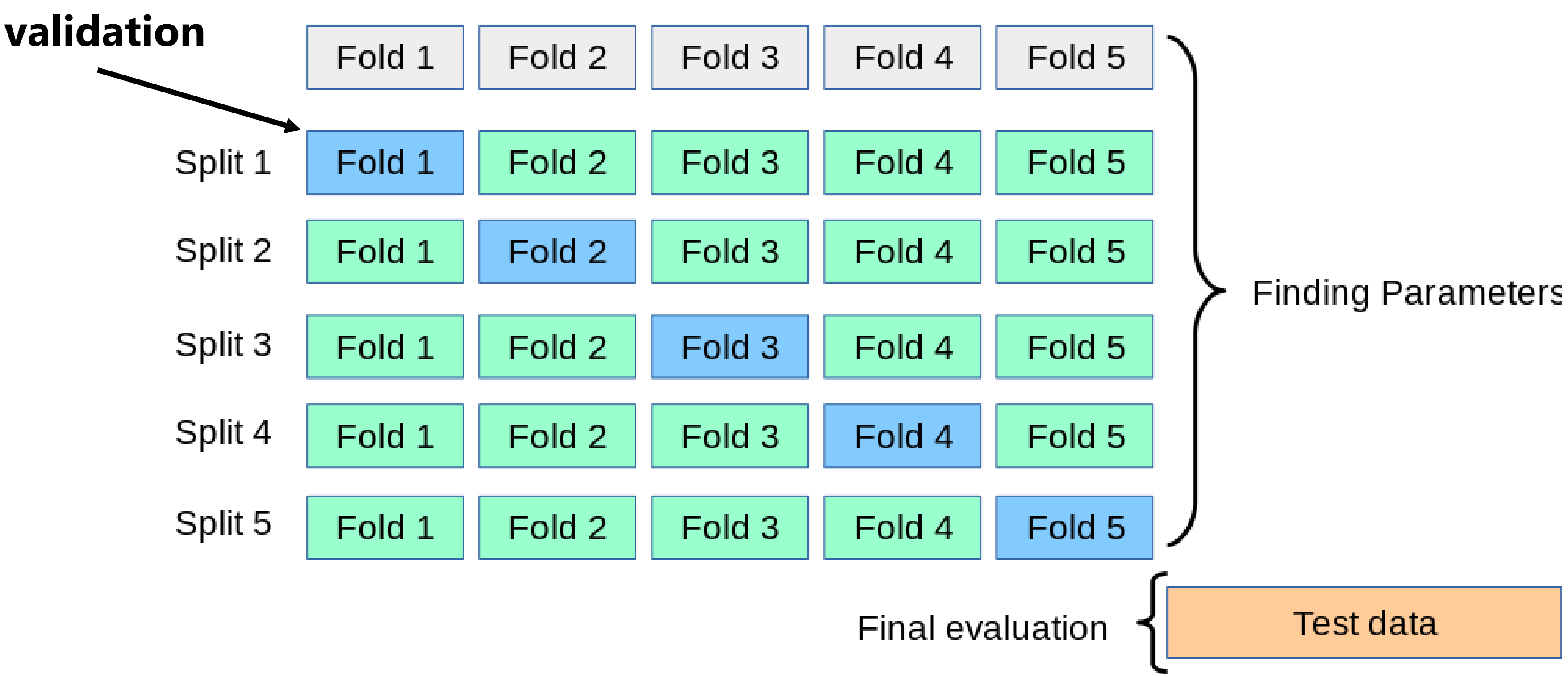
Typical values for K are $K = 2$, $K = 5$, $K = 10$, or $K = N$. The last case ($K = N$) is called Leave-One-Out Cross Validation (LOOCV). The average validation error for LOOCV is given by

$$\mathcal{L}^{CV} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - f(\mathbf{x}_n; \mathbf{w}_{-n})\right)^2$$

where $\mathbf{w}_{-n}$ is weight vector computed without the n-th training example.

Combination of cross-validation and hold out

Separate train and test datasets. Generate k-fold cross-validation-based models training on k splits of the train dataset, and the then evaluated on the test dataset.



Training/Validation dataset example

- Dataset:

	Student 1	Student 2	Student 3	Student 4	Student 5
Name	Thomas	Victor	Diana	Tiffany	Andrew
Hours of preparation	12	25	10	21	1
Grade [0-10]	5	9	4	10	1

Training set

Validation set

- Two models trained on training set:
  - Grade depends on student's name:
    name starts with [A-M] – grade < 5
    name starts with [N-Z] – grade ≥ 5
  - Grade depends on the length of preparation:
    # hours < 12 – grade < 5
    # hours ≥ 12 – grade ≥ 5
  - How to know which one is more reasonable

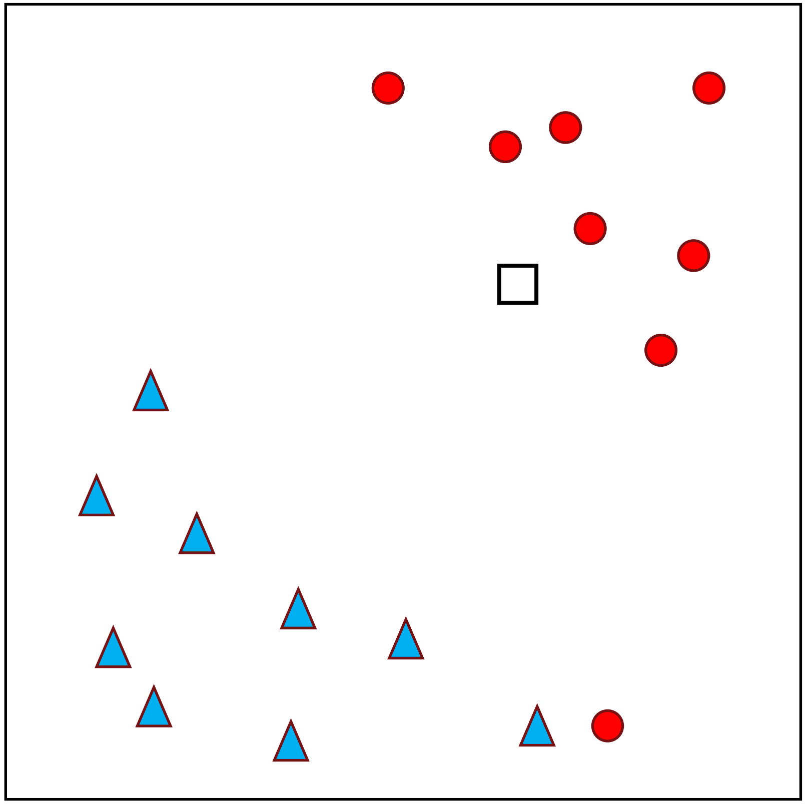
Accuracy on validation

50%

75%

k-nearest neighbor classifier

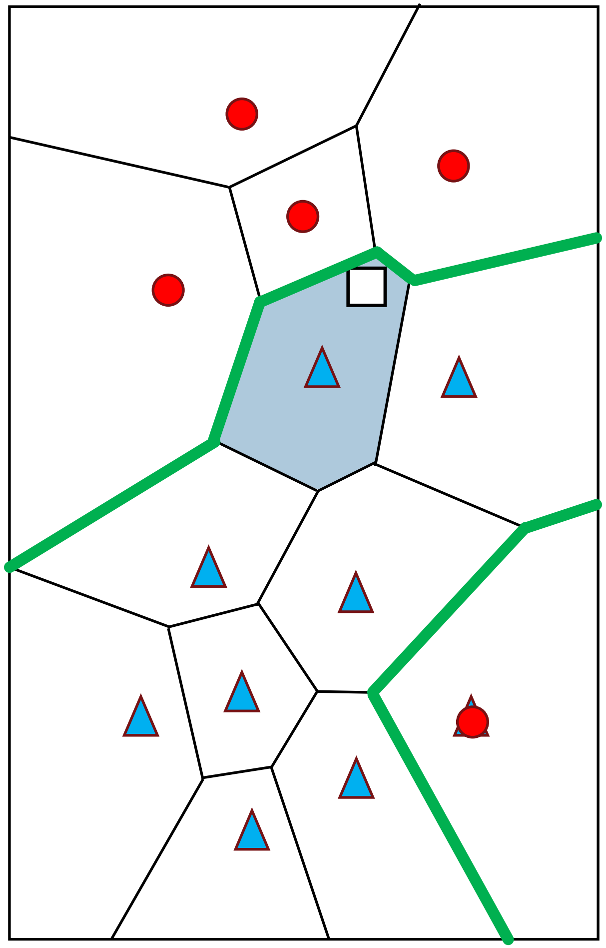
- How would you classify the white box:
  - Red or blue class?



- How did you do it:
  - Fit gaussian?
  - Coded a neural network?
- Nearby elements are red

1-nearest neighbor classifier

- If we formalize this intuition:
  - Find the most similar training example x’
  - Use its label y’ as the prediction
- Voronoi tessellation:
  - Separates space according to classes
  - Used to compute classification boundary
- Sensitive to outliers:
  - One point can change a very large regions
- Check more than 1 nearest neighbor



k-nearest neighbor (kNN) classifier

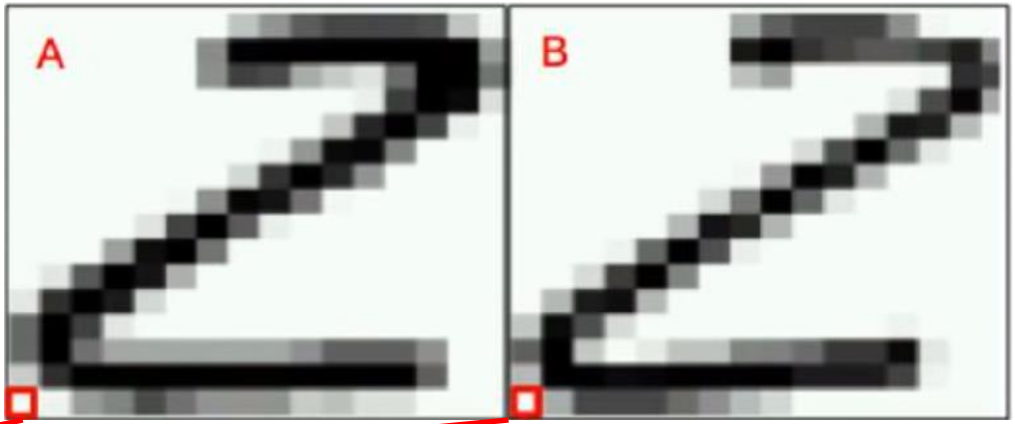
- Input:
  - Training examples $\{x_i, l_i\}$:
    - x_i - features of i-th examples
    - l_i - label of i-th examples

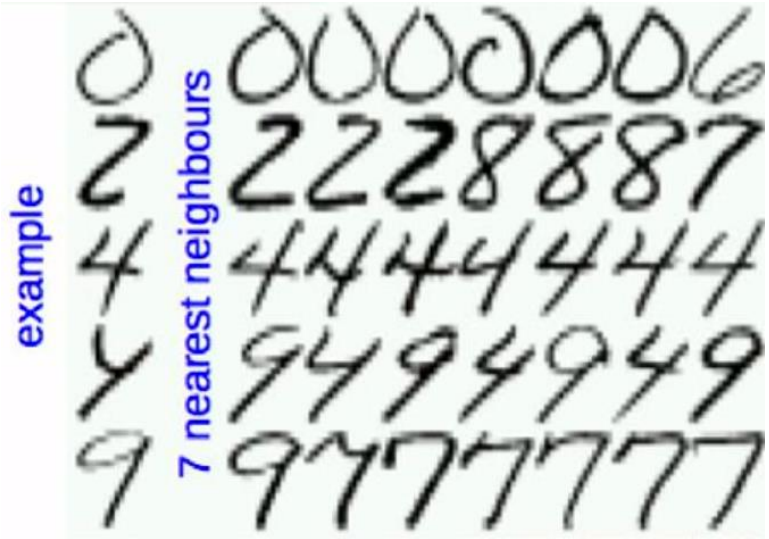
- Goal is to classify new example $\{z\}$

- Algorithm:
  - Compute distance $\mathrm{D}(z, x_i)$ to every training example x_i
  - Select k closest examples $x_{i1} \dots x_{ik}$ and their labels $l_{i1} \dots l_{ik}$
  - Output the most frequent class from $l_{i1} \dots l_{ik}$

Example: digit classification 7NN

- Database of 16x16 grayscale scale bitmaps of digits with known labels
- Recognize digit on new 16x16 grayscale bitmap
- Distance metric:
  - $D(A,B) = \sqrt{\sum_i \sum_j \left(A_{i,j} - B_{i,j}\right)^2}$





Predictions

- **0**; as majority voting
- **2**; parity voting, but 2s have higher scores
- **4**; unanimously voting
- **9**; as majority voting
- **7**; as majority voting

http://homepages.inf.ed.ac.uk/vlavrenk

kNN regression

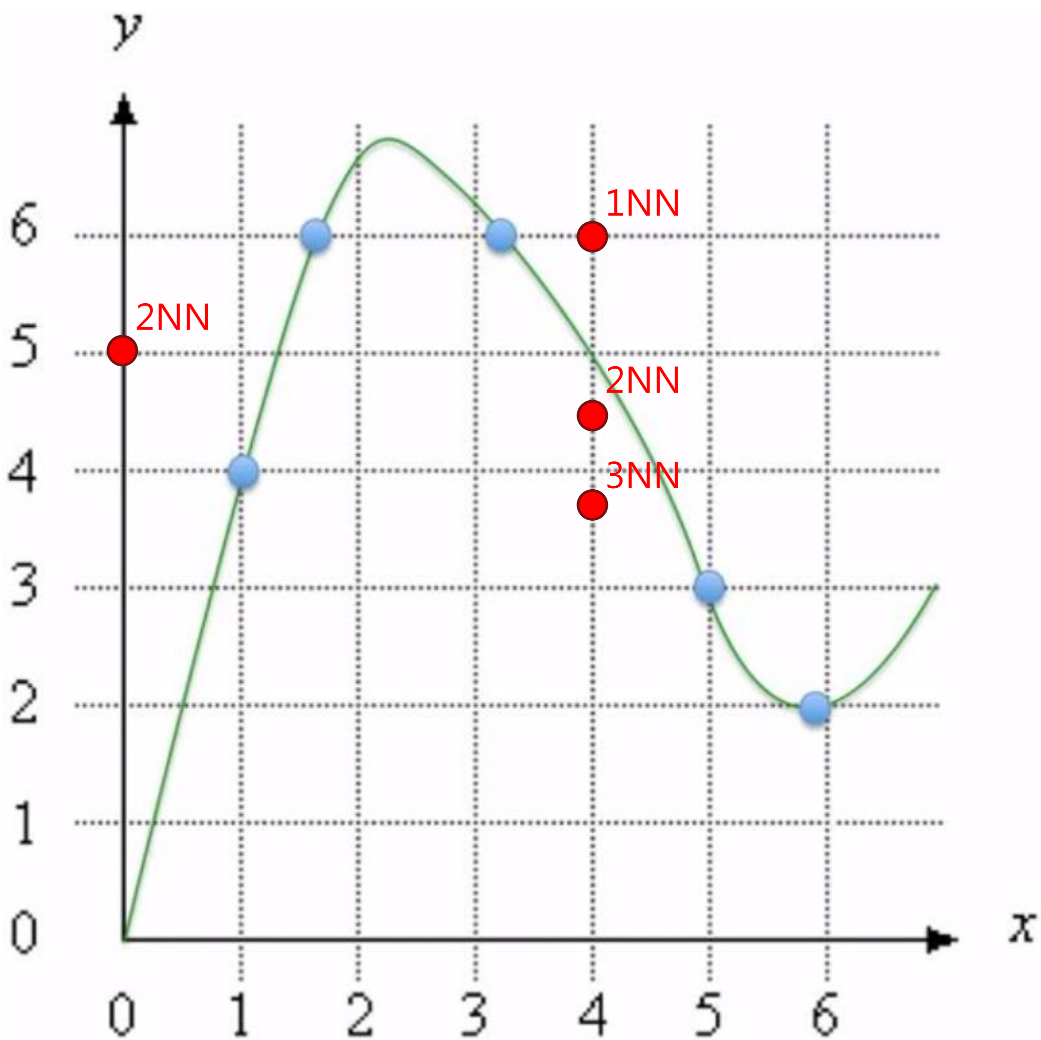
- Input:
  - Training examples $\{x_i, y_i\}$:
    - x_i - features of i-th examples
    - y_i - real value associated with examples (profit, exam result)

- Goal is to assign value to new example$\{z\}$

- Algorithm:
  - Compute distance $\mathrm{D}(z, x_i)$ to every training example x_i
  - Select k closest examples $x_{i1} \dots x_{ik}$and their values $y_{i1} \dots y_{ik}$
  - Output the mean of $y_{i1} \dots y_{ik}$

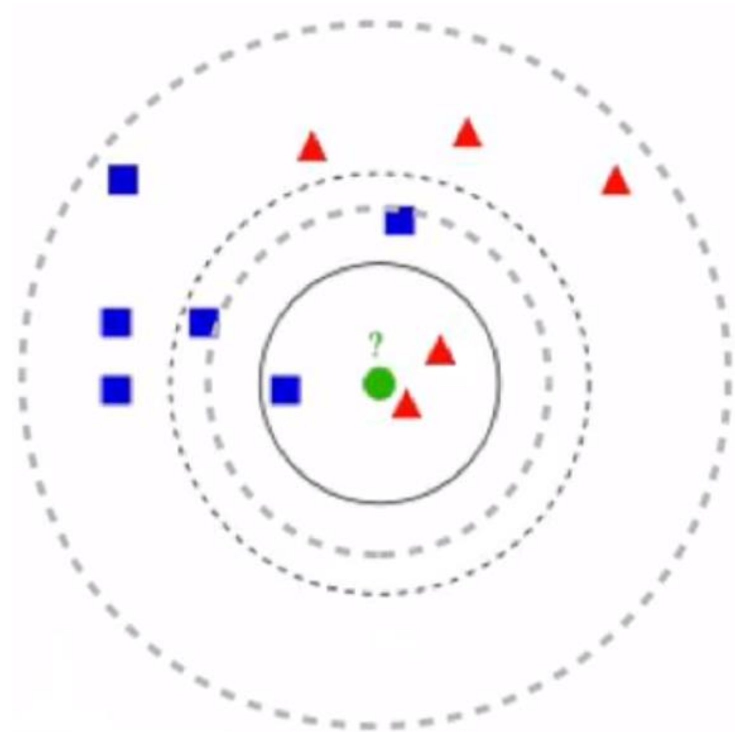
Example: kNN regression

- Let's predict values for x=4:
  - 1NN regression
  - 2NN regression
  - 3NN regression
  - etc.
- Results may significantly depend on the k selection:
  - What if we choose k = database size
- Which k is good for predicting for x=0?
- Extrapolation is less accurate than interpolation



kNN: how to select k?

- What will be the prediction for green point for:
  - 1NN classification
  - 2NN classification
  - 5NN classification



- Let's say you have a database of 10000 points in space with known red/blue labels. How to choose good k?

- Validation set!

http://homepages.inf.ed.ac.uk/vlavrenk

kNN: Distance measure

- Euclidian distance:

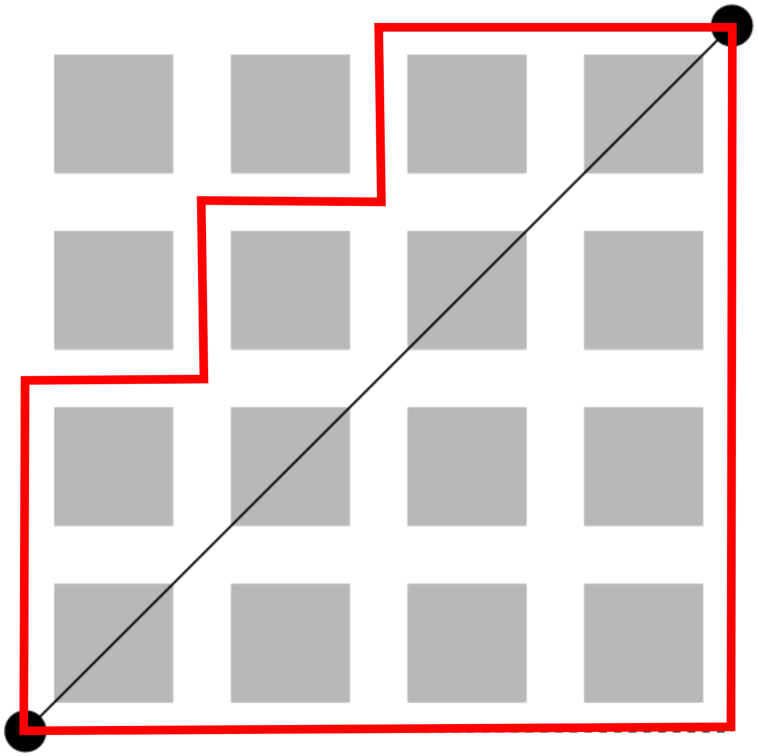
$$D(x, x') = \sqrt{\sum_d |x_d - x'_d|^2}$$

- Manhattan distance:

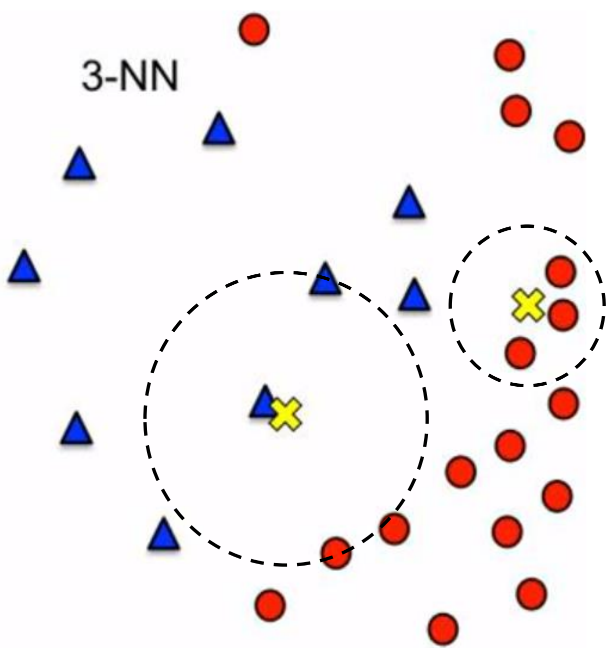
$$D(x, x') = \sum_d |x_d - x'_d|$$

- Logical distance (categorical attributes):

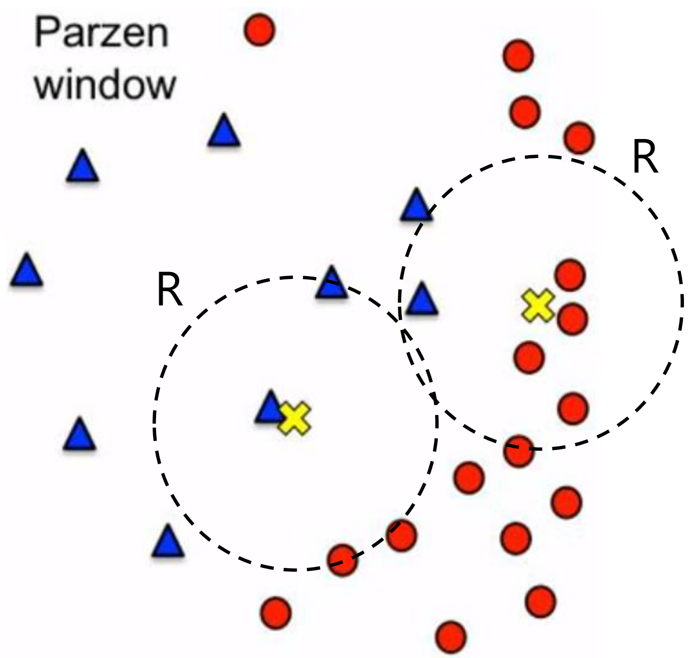
$$D(x, x') = \sum_d 1_{x_d \neq x'_d}$$



kNN and Parzen Windows



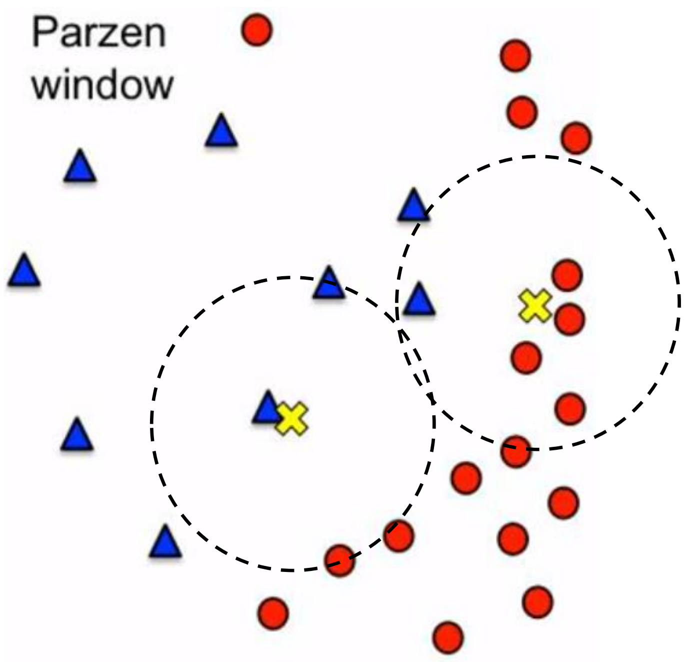
The size of the neighborhoods can be very different

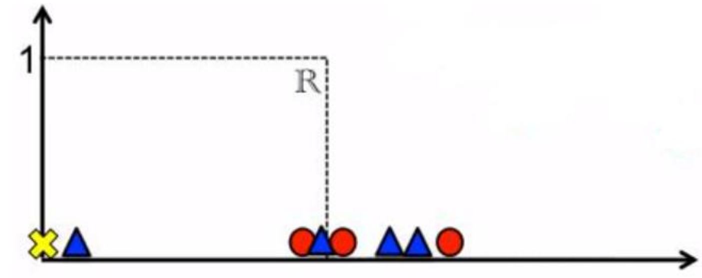


The size of the neighborhoods is the same

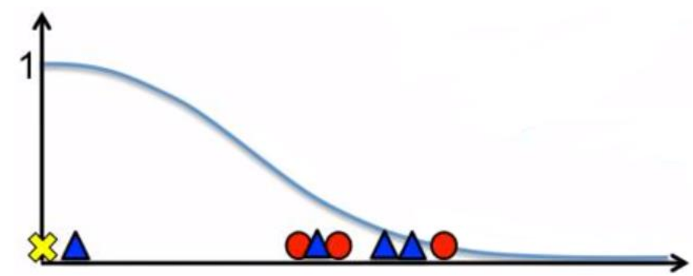
$$P(red|x) = \frac{\sum_i 1_{l_i=red} \cdot 1_{x_i \in R(x)}}{\sum_i 1_{x_i \in R(x)}}$$

kNN, Parzen Windows and Kernels





What is the problem with $1_{x_i \in R(x)}$?



Kernel-based predictor

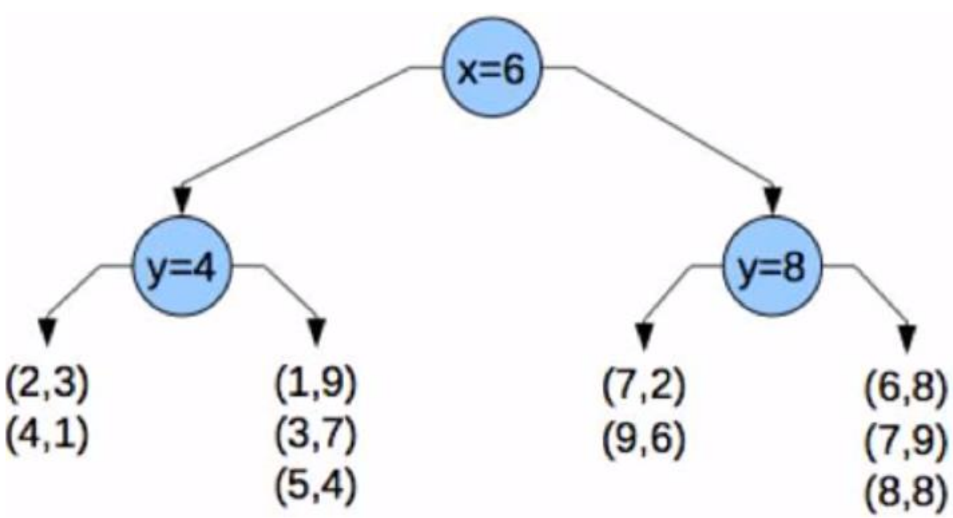
$$P(red|x) = \frac{\sum_i 1_{l_i=red} \cdot 1_{x_i \in R(x)}}{\sum_i 1_{x_i \in R(x)}}$$

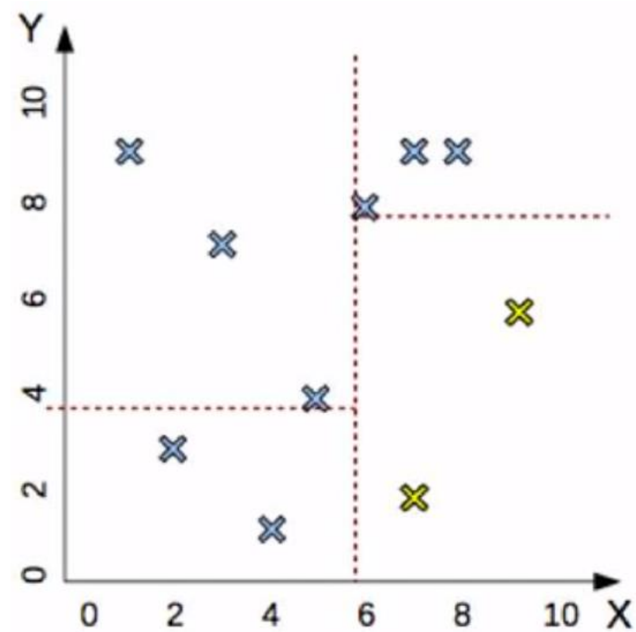
$$P(red|x) = \frac{\sum_i 1_{l_i=red} \cdot K(x_i, x)}{\sum_i K(x_i, x)}$$

kNN problem: speed!

- There are 60000 examples digit recognition database:
  - It will be extremely slow to measure distance to all of them

- Many acceleration techniques. For example, KD search tree:
  - Training samples: {(1, 9), (2, 3), (4, 1), (3, 7), (5, 4), (6, 8), (7, 2), (8, 8), (7, 9), (9, 6)}
  - Pick random dimension, find median, split
  - For a test sample (7, 4), follow the tree and search the specific subregion

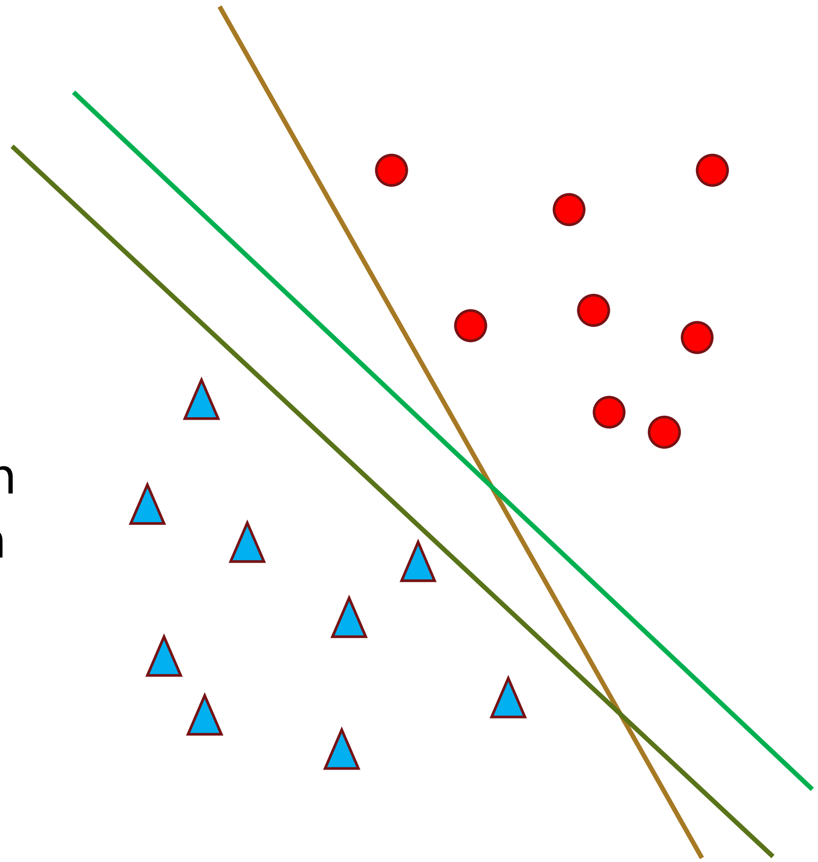




http://homepages.inf.ed.ac.uk/vlavrenk

Support vector machines

- Let's say we want to best separate red and blue points
- Many solutions are possible, but are they equally good?
- Somehow this solution looks better than other solutions. How did you make such intuitive conclusion?



Support vector machines

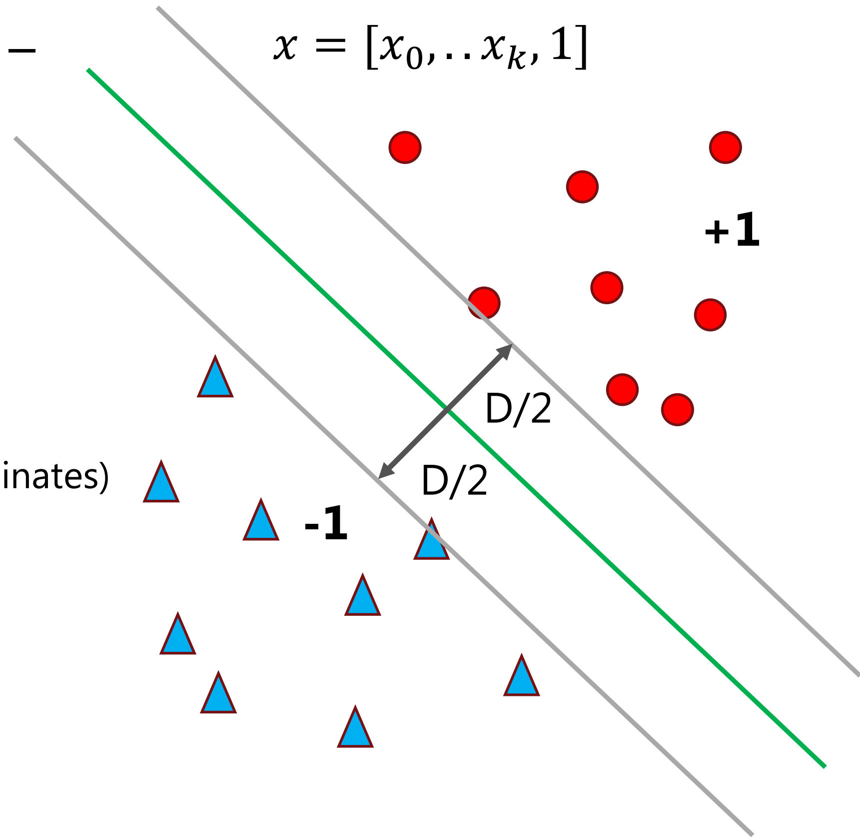
- Let’s say the points above the line have positive labels (+1), while points below – negative (-1)

- A prediction line can be defined as:

$$y = w^T x$$

label

Features (point coordinates)



Support vector machines

- The prediction is uncertain at the green borderline:

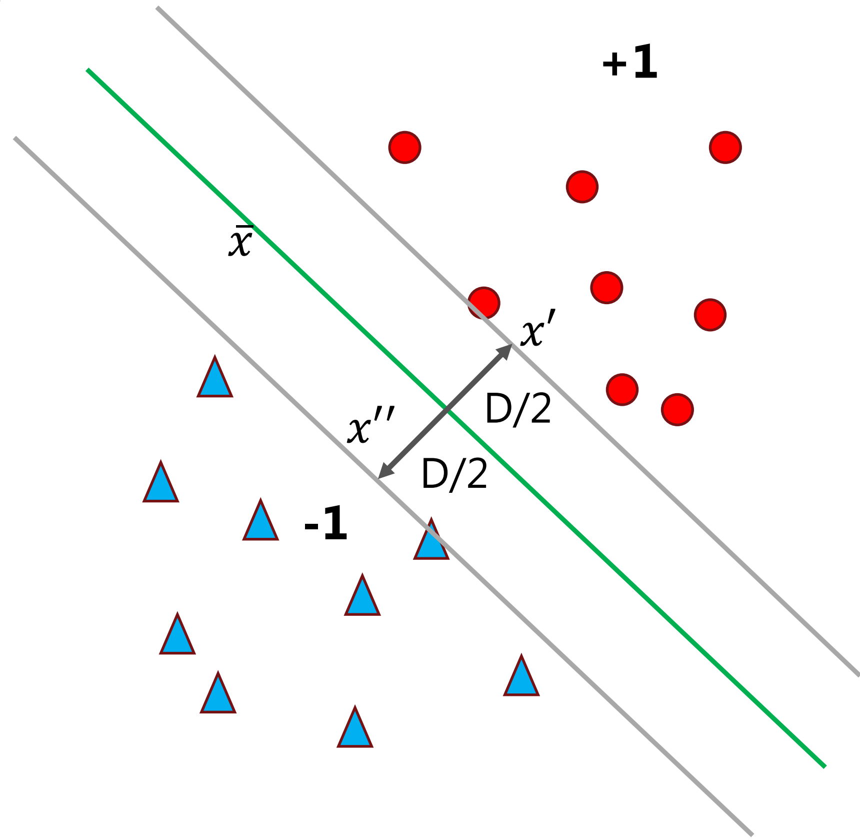
$$w^T \bar{x} = 0$$

- What is the equations for points above the upper gray line:

$$w^T x' \geq 1$$

- The equations for points below the lower gray line:

$$w^T x'' \leq -1$$



Support vector machines

- Hinge loss function:

$$h(x, y, w^T x) = \begin{cases} 0 & \text{if } y \cdot w^T x \geq 1 \\ 1 - y \cdot w^T x & \text{else} \end{cases}$$

- L2 Regularization (*why do we need it?*):

$$l = \|w\|^2$$

- We want to optimize:

$$\min_w \left(\lambda \|w\|^2 + h(x, y, w^T x) \right)$$

Support vector machines

- Partial derivatives of $\min_{w}\left(\lambda\|w\|^2 + h(x, y, w^T x\ )\right)$:

$$\frac{\delta}{\delta w_k}\lambda\|w\|^2 = 2\lambda w_k \qquad \frac{\delta}{\delta w_k}h(\cdot) = \begin{cases} 0 & \text{if } y \cdot w^T x\ \geq 1 \\ -y_i x_{ik} & \text{else} \end{cases}$$

- Update of w:

for mis-classified samples:

$$w = w - \eta(-y_i x_{ik} + 2\lambda w_k)$$

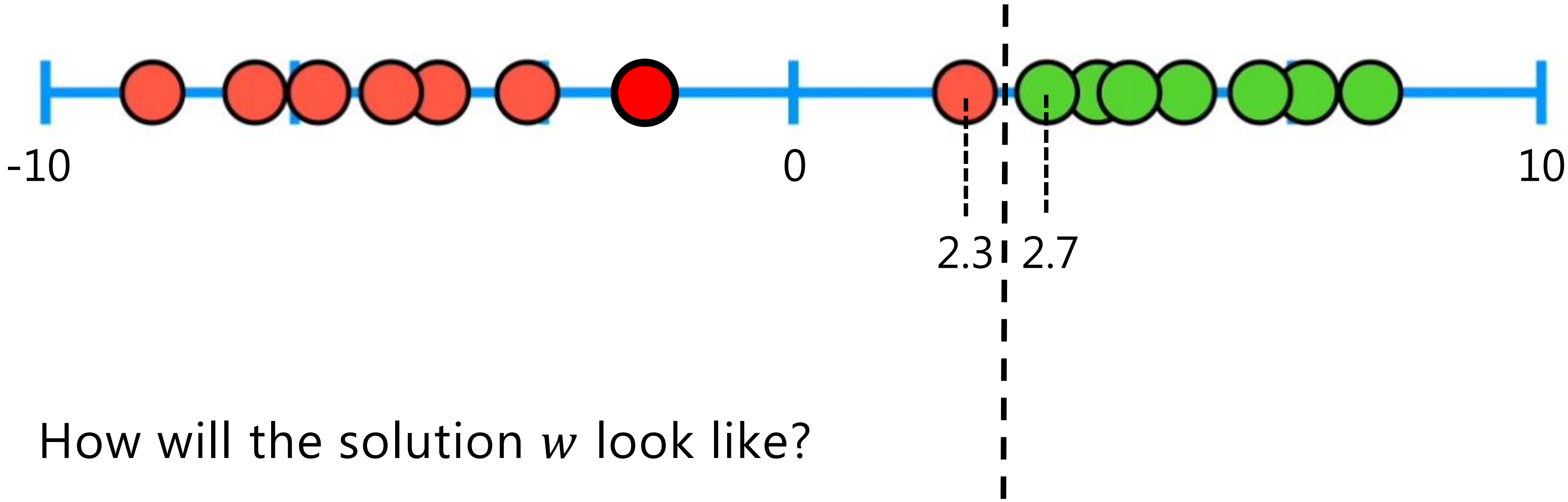
for correctly classified samples:

$$w = w - \eta(2\lambda w_k)$$

Support vector machines: regularization

- Separation with very low $\lambda = 1e - 10$:

$$\min_w \left(\lambda \|w\|^2 + h(x, y, w^T x\,)\right)$$



- How will the solution w look like?

$$\boxed{w = 5x \ - 12.5}$$

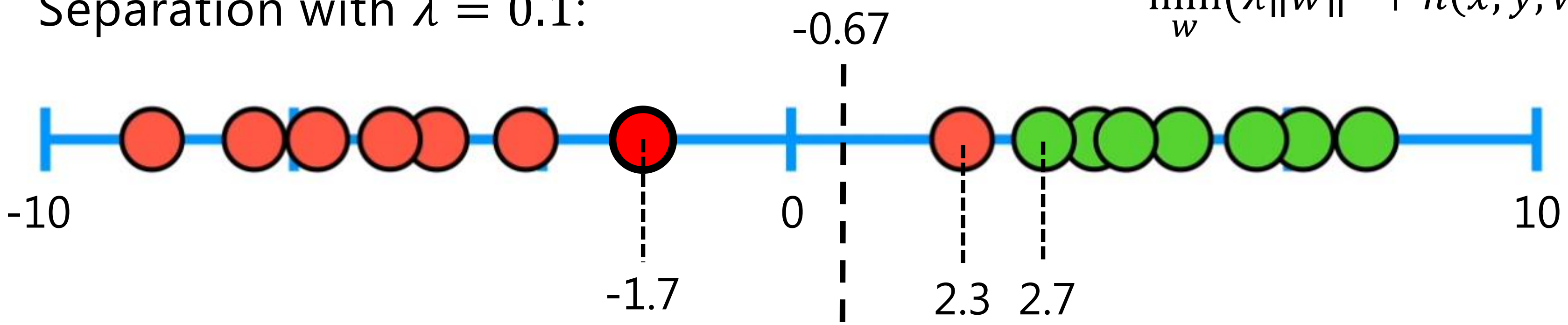
- Could we get something like?
  - $w \ = 10x - 25$

- Is this a good solution?

Support vector machines: regularization

$$\min_{w}\left(\lambda\|w\|^2 + h(x, y, w^T x\ )\right)$$

- Separation with $\lambda = 0.1$:



- The solution w will look like:

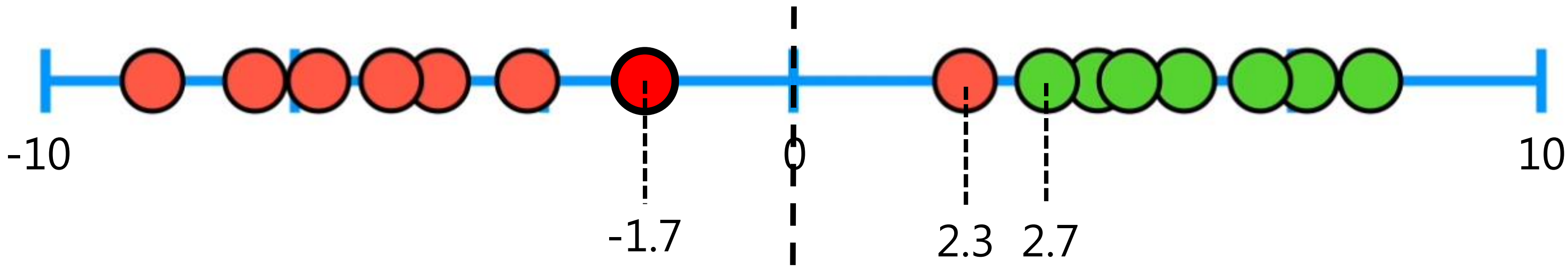
$$\boxed{w = 0.43x\ - 0.28}$$

- This separation does not classify samples perfectly, but seems to be more reliable

Support vector machines: regularization

- Separation with $\lambda = 10$:

$$\min_{w}\left(\lambda\|w\|^2 + h(x, y, w^T x\,)\right)$$



- The solution w will look like:

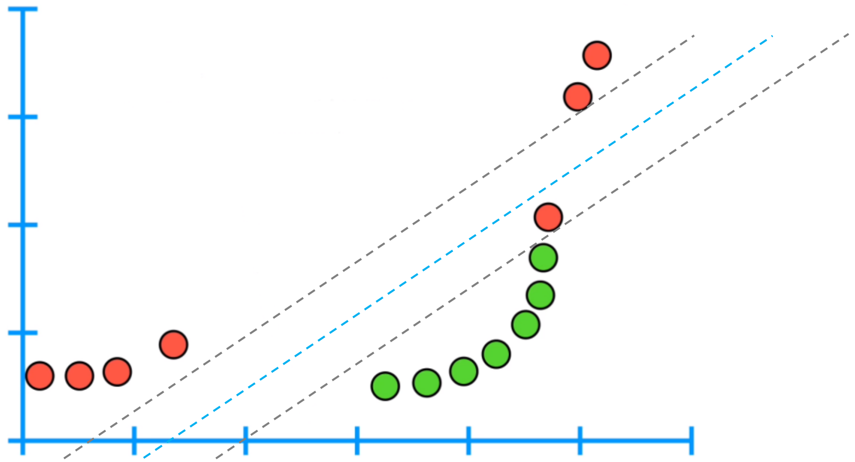
$$\boxed{w = 0.065x}$$

- The solution w comes closer to 0, with growth of λ:

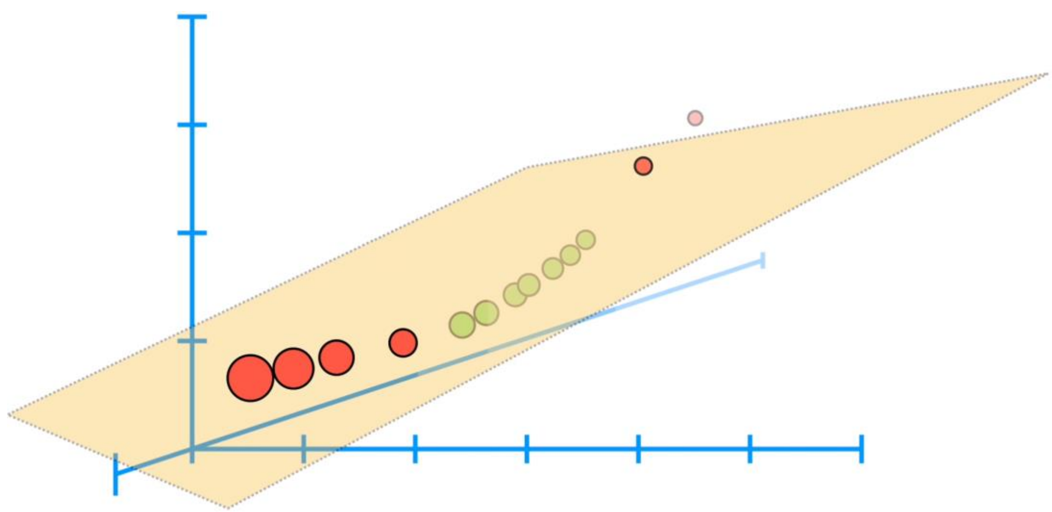
λ	w
1e-10	[5, -12.5]
0.1	[0.43, -0.28]
10	[0.065, 0]

Support vector machines: dimensionality

- 2D data:

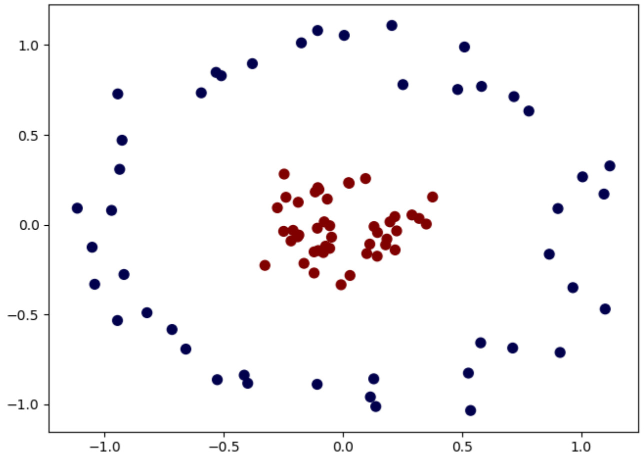


- 3D data:



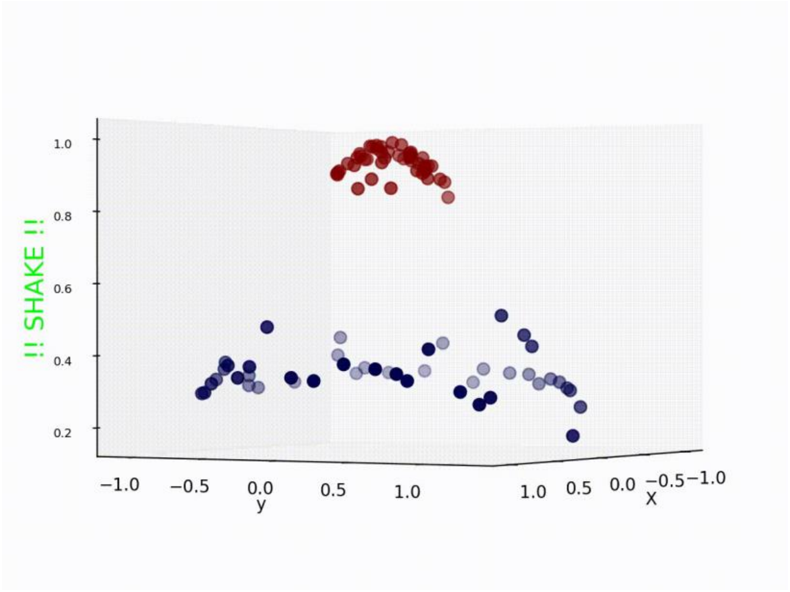
Support vector machines: kernel

- Can we classify these samples with support vectors:



- Let's transform data:

$$z = e^{-\|x\|^2}$$

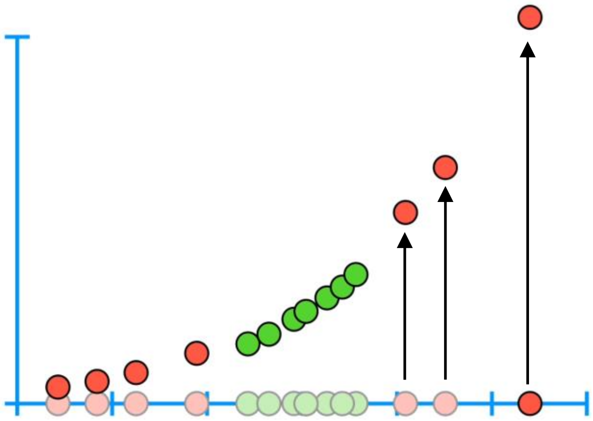


https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d

Support vector machines: kernels

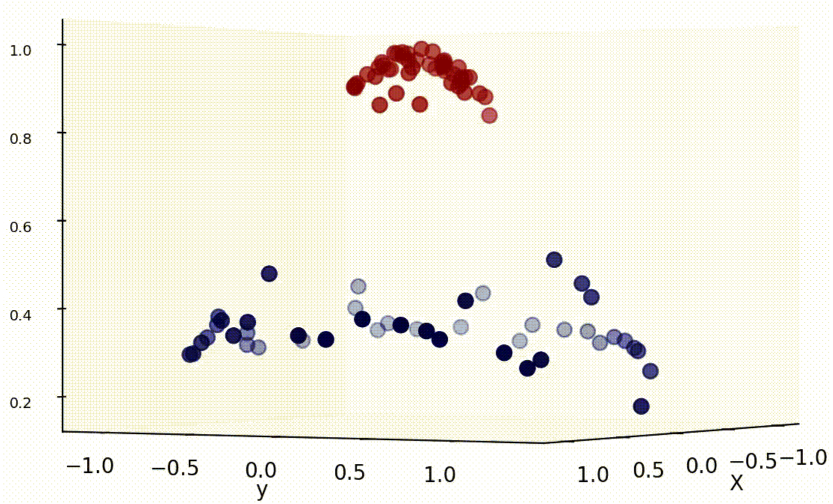
- Polynomial kernel:

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



- Radial basis function kernel:

$$K(x_i, x_j) = e^{-\gamma (x_i - x_j)^2}$$



- Hinge loss:

$$h(x, y, w^T x) = h(x, y, K(w, x))$$

Classification performance evaluation: 2 classes

- Accuracy:

$$A = \frac{\sum_i \text{sign}(y_i f(x_i))}{|y|}$$

correctly classified testing examples

testing examples

- What are the problems of such metric:
  - Class imbalance problem:
    - 0.9 healthy subject, 0.1 diseased
    - Naïve classification will result in 0.9 accuracy

  - Let's say $f(x) \in [-1,1]$, $f(x) = 0.01$ will be as good as $f(x) = 0.99$ for $y = 1$

Performance evaluation: sensitivity/specificity

- Let's normalize all labels y and $f(x)$ to $[0, 1]$
- True positive (TP) - # of cases, where $y_i = 1$ and $f(x_i) \geq 0$
- True negative (TN) - # of cases, where $y_i = -1$ and $f(x_i) < 0$
- False positive (FP) - # of cases, where $y_i = -1$ and $f(x_i) \geq 0$
- False negative (FN) - # of cases, where $y_i = 1$ and $f(x_i) < 0$

- Sensitivity:

$$\frac{TP}{TP + FN}$$

- Specificity:

$$\frac{TN}{TN + FP}$$

Performance evaluation: ROC curve

- Example of testing mosquito killing spray:

Among survivals, how many survived receiving the current dose?

Among dead, how many died because of the current dose?

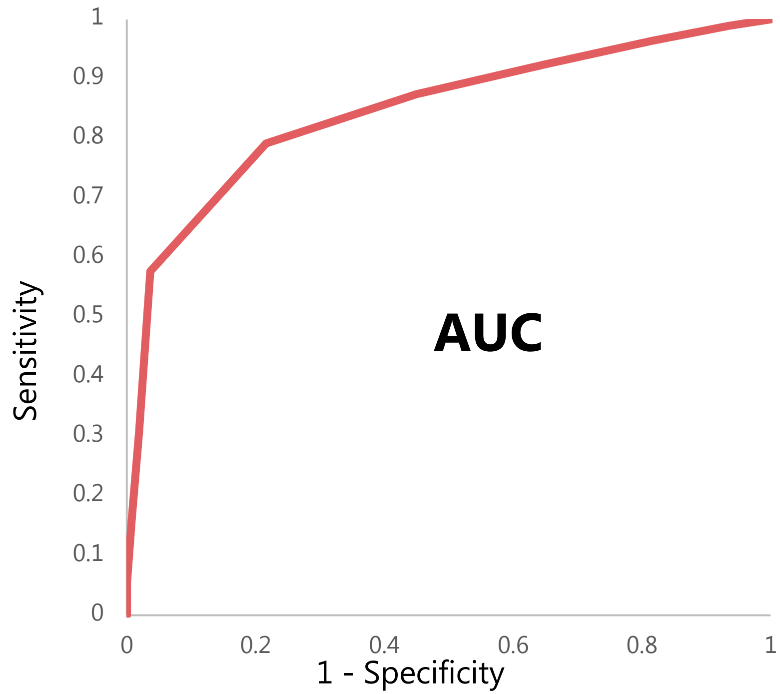
	A	B	C	D	E	F	G
3	ROC Table						
4							
5		Observed		Cumulative			
6	*Dosage*	*Lives*	*Dies*	*Lives*	*Dies*	*FPR*	*TPR*
7				0	0	1	1
8	less than 2.00	34	3	34	3	0.935484	0.989247
9	2.00 - 3.99	63	7	97	10	0.815939	0.964158
10	4.00 - 5.99	88	11	185	21	0.648956	0.924731
11	6.00 - 7.99	105	14	290	35	0.449715	0.874552
12	8.00 - 9.99	123	23	413	58	0.216319	0.792115
13	10.00 - 11.99	95	60	508	118	0.036053	0.577061
14	12.00 - 13.99	9	75	517	193	0.018975	0.308244
15	14.00 - 15.99	6	41	523	234	0.00759	0.16129
16	16.00 -17.99	4	30	527	264	0	0.053763
17	18.00 or more	0	15	527	279	0	0
18		527	279				

$$1-\frac{35}{279}$$

$$1-\frac{290}{527}$$

Performance evaluation: ROC curve

- Receiving operator curve:



- Closer AUC to 1 the better

Questions?