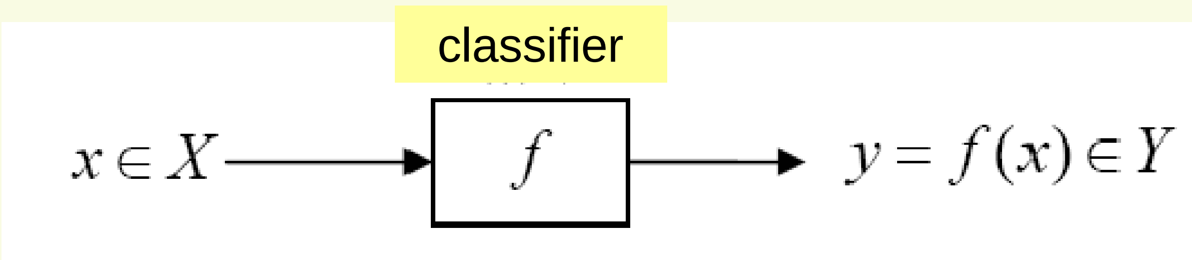
Intro to Classification

Definition of Classification

- A classifier is a function or an algorithm that maps every possible input (from a legal set of inputs) to a finite set of categories.
- X – input space, $\mathrm{x} \in \mathrm{X}$ sample from an input space.
- A typical input space is high-dimensional, for example $\mathrm{x} = \{x_1, ..., x_d\} \in R^d, \ \ d > 1$. We also call x a feature vector.
- Ω is a finite set of categories to which the input samples belong: $\Omega = \{1,2,\ldots,\mathrm{C}\}$.
- $w_i \in \Omega$ are called labels.

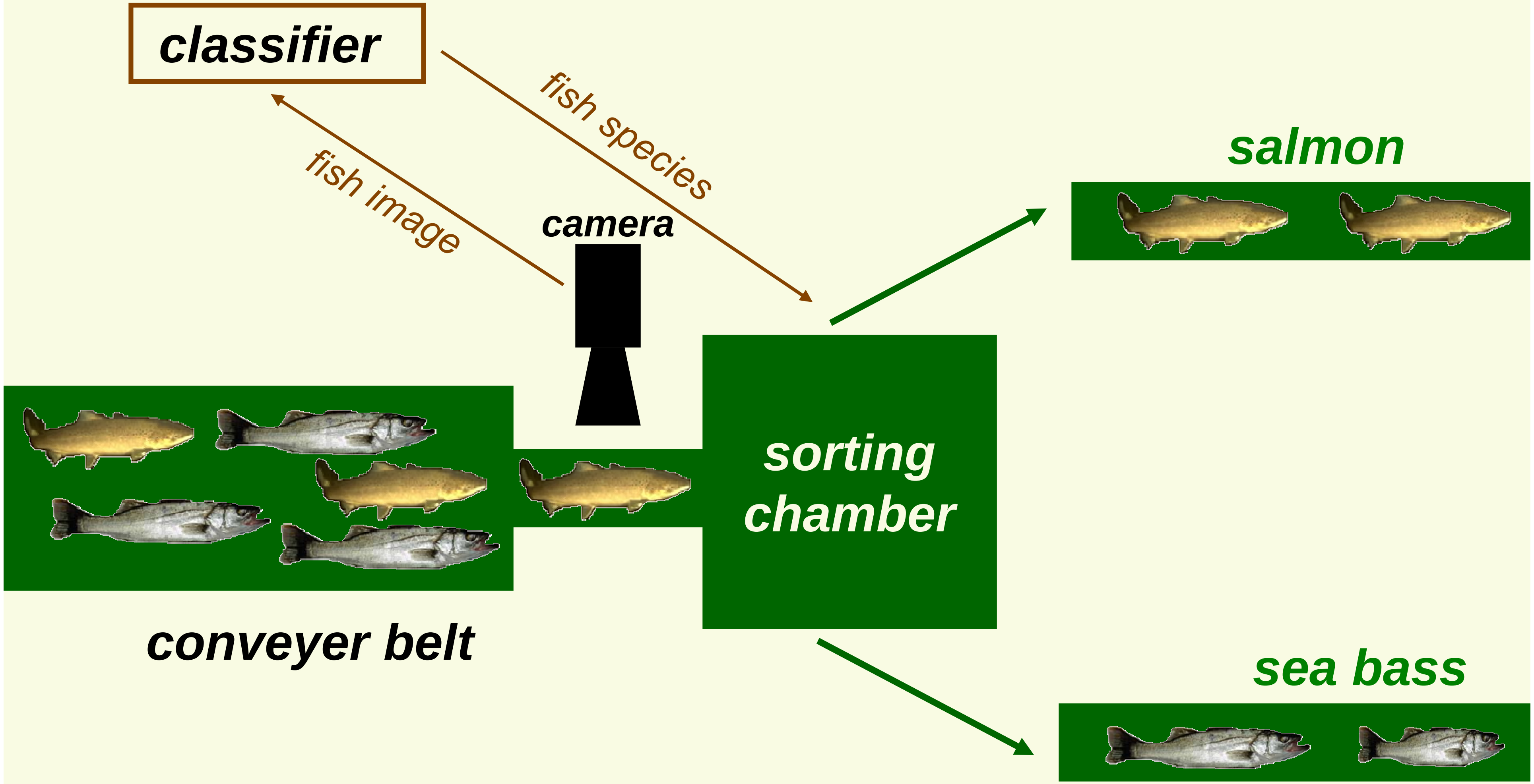
Definition of Classification

- Y is a finite set of decisions – the output set of the classifier.
- Usually $\mathrm{Y}=\Omega$, but it can also contain other decisions, such as “no decision”, “reject” (doesn’t belong to any category from Ω).
- A classifier is a function $f:\mathrm{X}\rightarrow\mathrm{Y}$



- Classification is also called Pattern Recognition.

Our Toy Application: fish sorting



How to design a PR system?

- Collect data and classify by hand

salmon *sea bass* *salmon* *salmon* *sea bass* *sea bass*

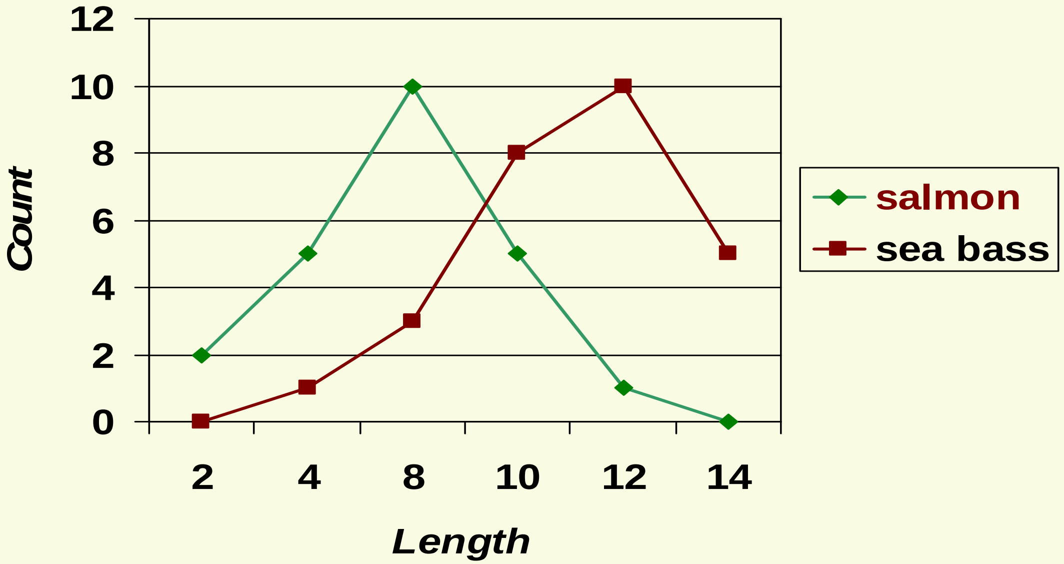
- Preprocess by segmenting fish from background

- Extract possibly discriminating features
  - length, lightness,width,number of fins,etc.
- Classifier design
  - Choose model
  - Train classifier on part of collected data (training data)
- Test classifier on the rest of collected data (test data) i.e. the data not used for training
  - Should classify new data (new fish images) well

Classifier design

- Notice salmon tends to be shorter than sea bass
- Use *fish length* as the discriminating feature
- Count number of bass and salmon of each length

	2	4	8	10	12	14
bass	0	1	3	8	10	5
salmon	2	5	10	5	1	0



Fish length as discriminating feature

- Find the best length ***L*** threshold

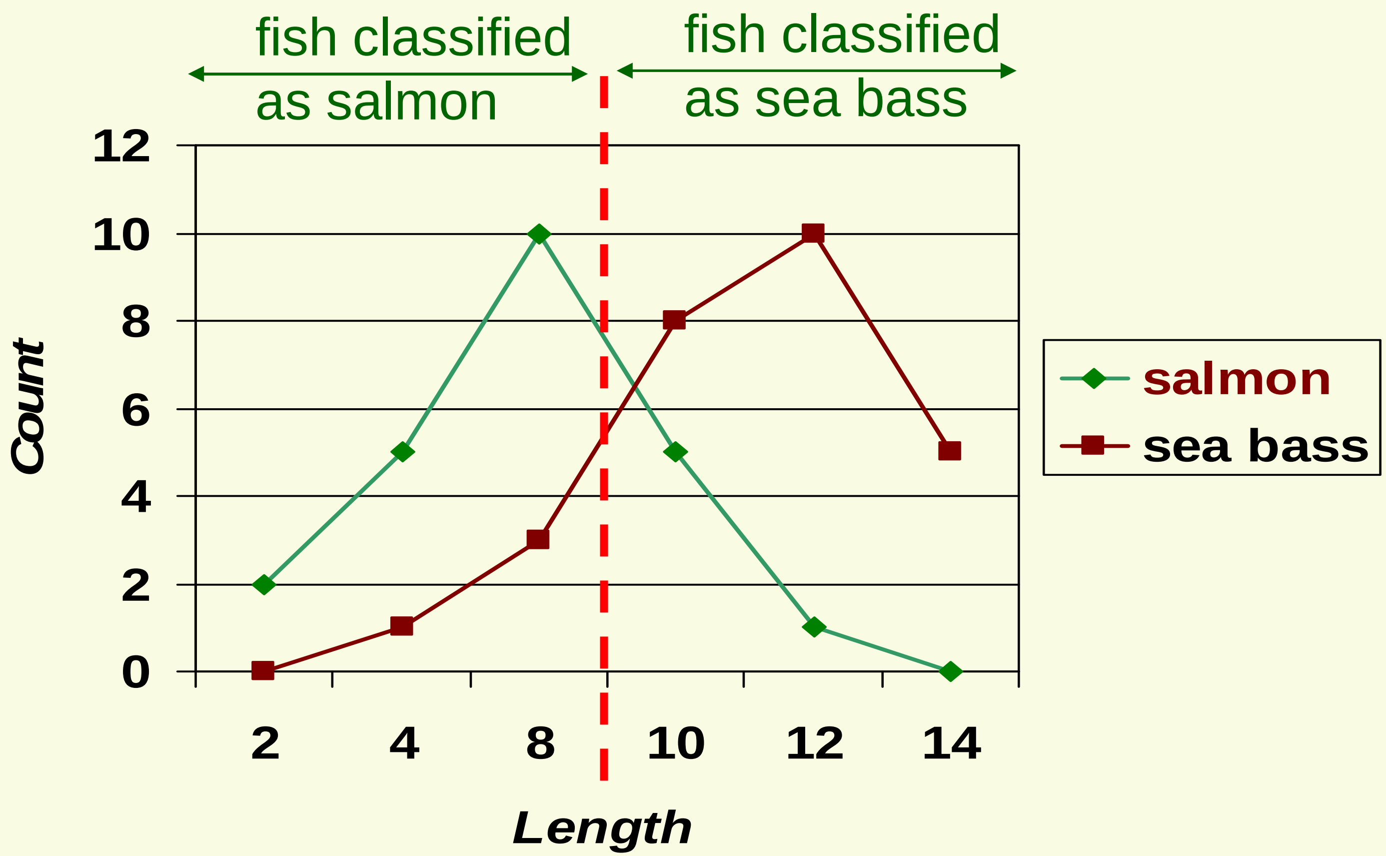
fish length < *L*	**fish length > *L***
⇓	⇓
classify as salmon	***classify as sea bass***

- For example, at ***L*** = 5, misclassified:
  - 1 sea bass
  - 16 salmon

	2	4	8	10	12	14
bass	0	1	3	8	10	5
salmon	2	5	10	5	1	0

- Classification error (total error): $\frac{\mathbf{17}}{\mathbf{50}}$ = ***34%***

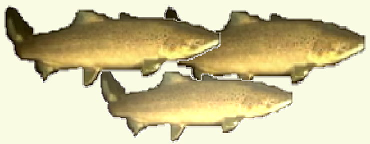
Fish Length as discriminating feature

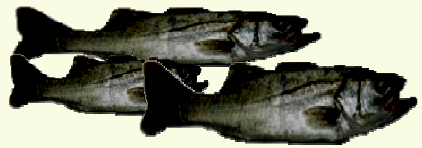


- After searching through all possible thresholds ***L***, the best ***L***= 9, and still 20% of fish is misclassified

Next Step

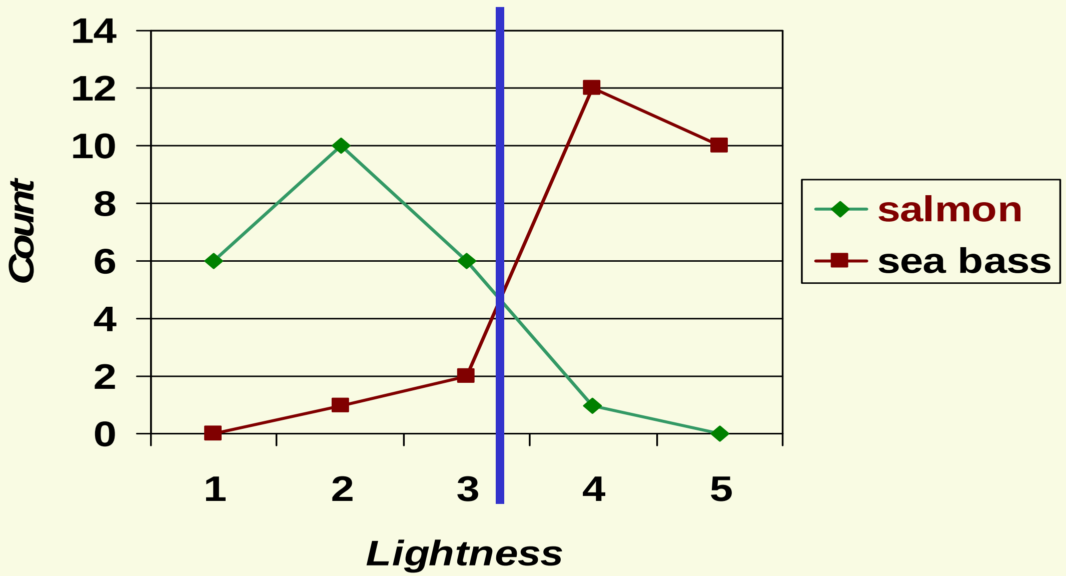
- Lesson learned:
  - Length is a poor feature alone!
- What to do?
  - Try another feature
  - Salmon tends to be lighter
  - Try average fish lightness





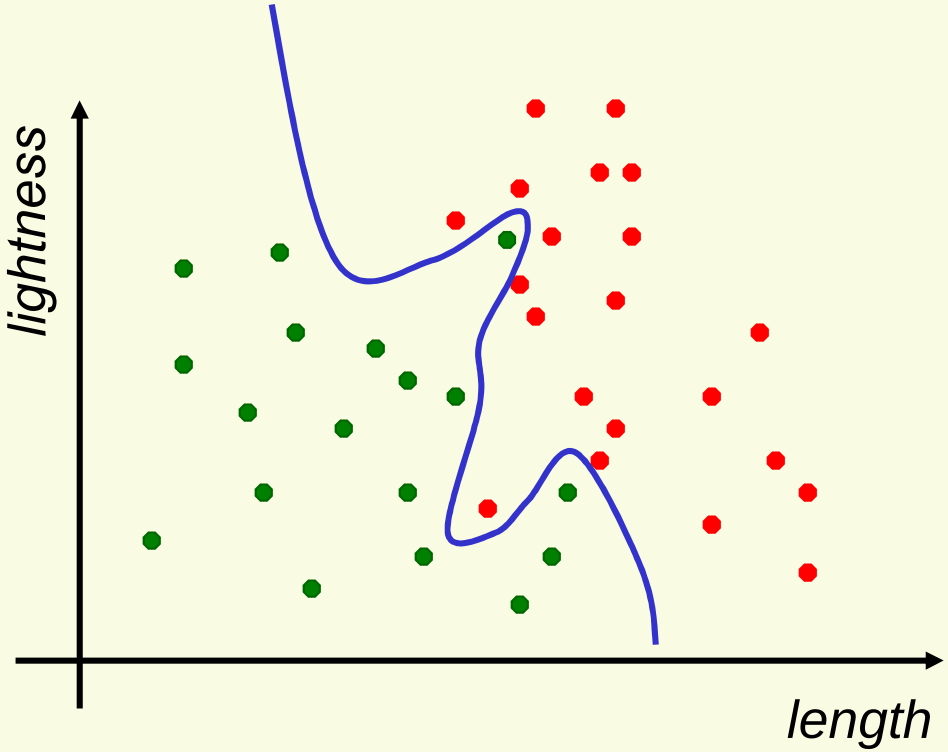
Fish lightness as discriminating feature

	1	2	3	4	5
bass	0	1	2	10	12
salmon	6	10	6	1	0



- Now fish are well separated at lightness threshold of 3.5 with classification error of 8%

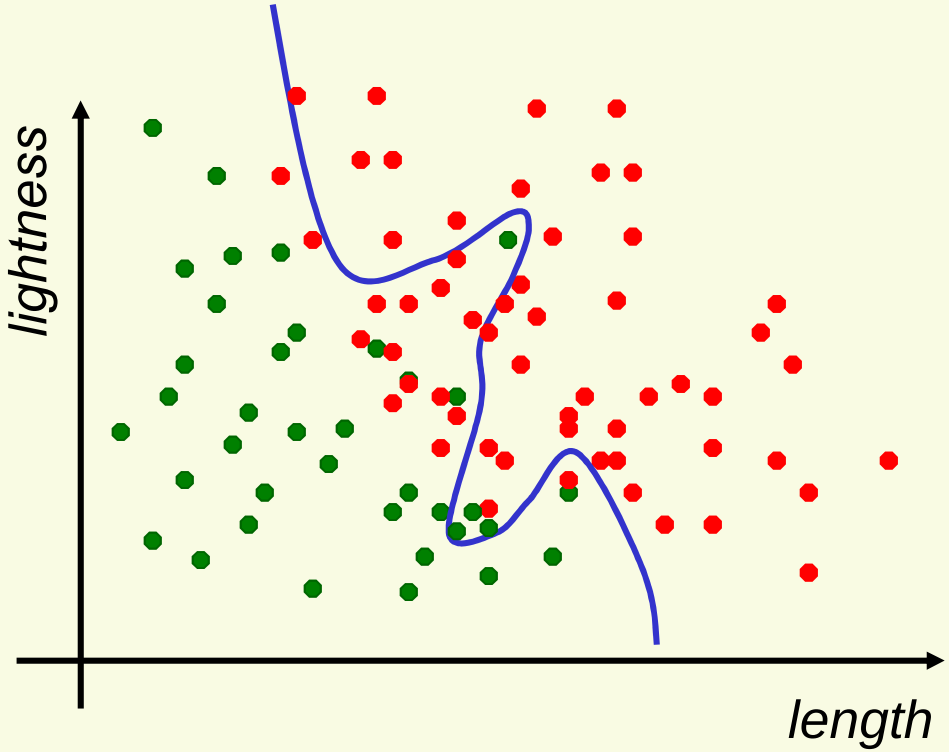
Better decision boundary



- Ideal decision boundary, 0% classification error

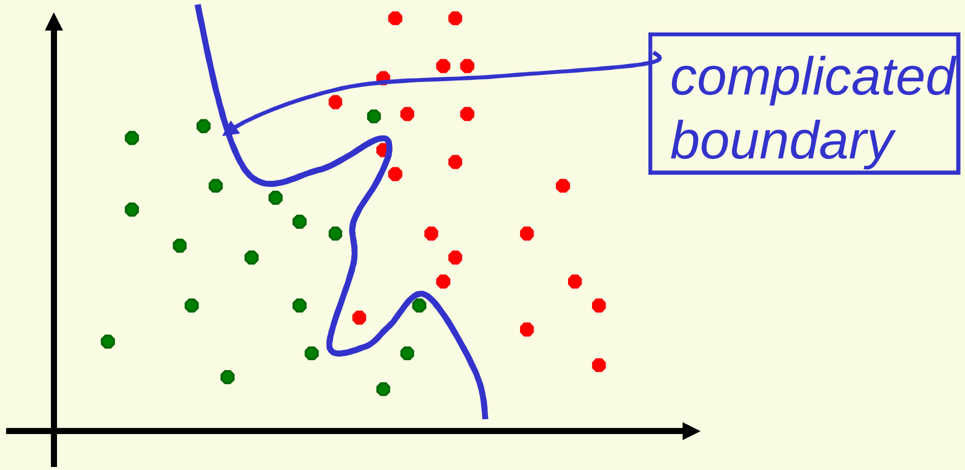
Test Classifier on New Data

- Classifier should perform well on new data
- Test "ideal" classifier on new data: 25% error



What Went Wrong?

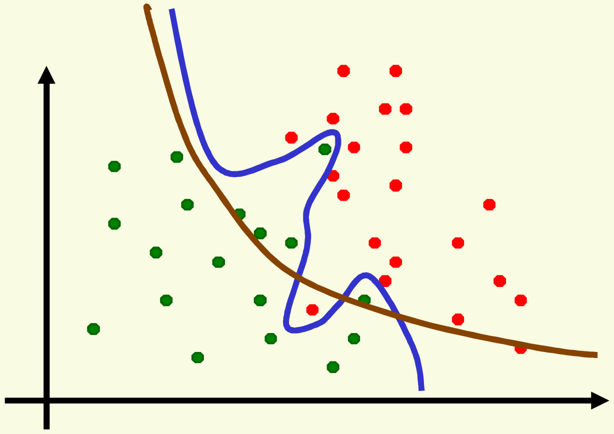
- Poor ***generalization***



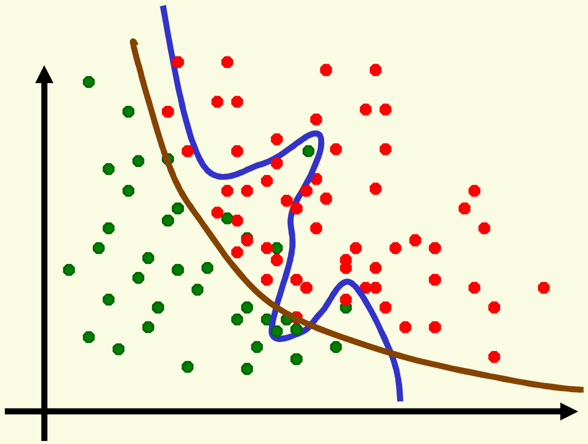
- Complicated boundaries do not generalize well to the new data, they are too “tuned” to the particular training data, rather than some true model which will separate salmon from sea bass well.
  - This is called overfitting the data

Generalization

training data



testing data



- Simpler decision boundary does not perform ideally on the training data but generalizes better on new data
- Favor simpler classifiers

Classification Overview

a lot is known ”easier”

↓

little is known “harder”

Bayesian Decision theory

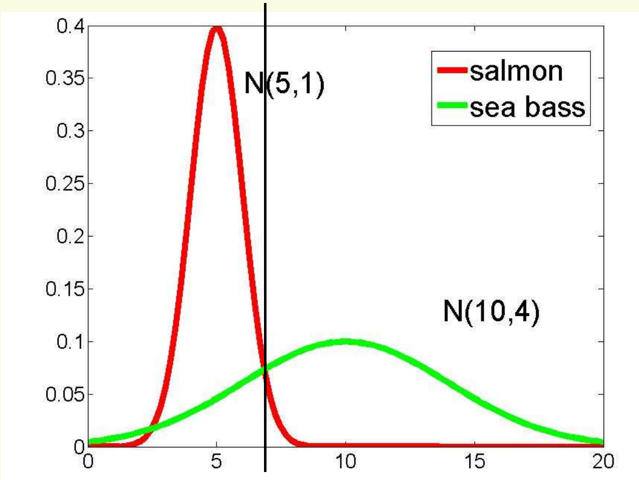
- Known probability distribution of the categories
  - never happens in real world
- Do not need training data
- Can design optimal classifier

a lot is known ”easier”

little is known “harder”

Example

respected fish expert says that salmon’s length has distribution ***N***(5,1) and sea bass’s length has distribution ***N***(10,4)



ML and Bayesian parameter estimation

- Shape of probability distribution is known
  - Happens sometimes
- Labeled training data

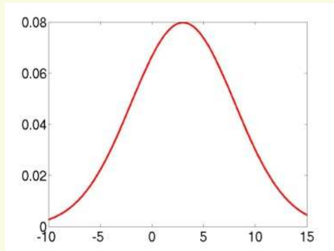
salmon bass salmon salmon

- Need to estimate parameters of probability distribution from the training data

a lot is known "easier"

Example

respected fish expert says salmon's length has distribution $N(\mu_1, \sigma_1^2)$ and sea bass's length has distribution $N(\mu_2, \sigma_2^2)$



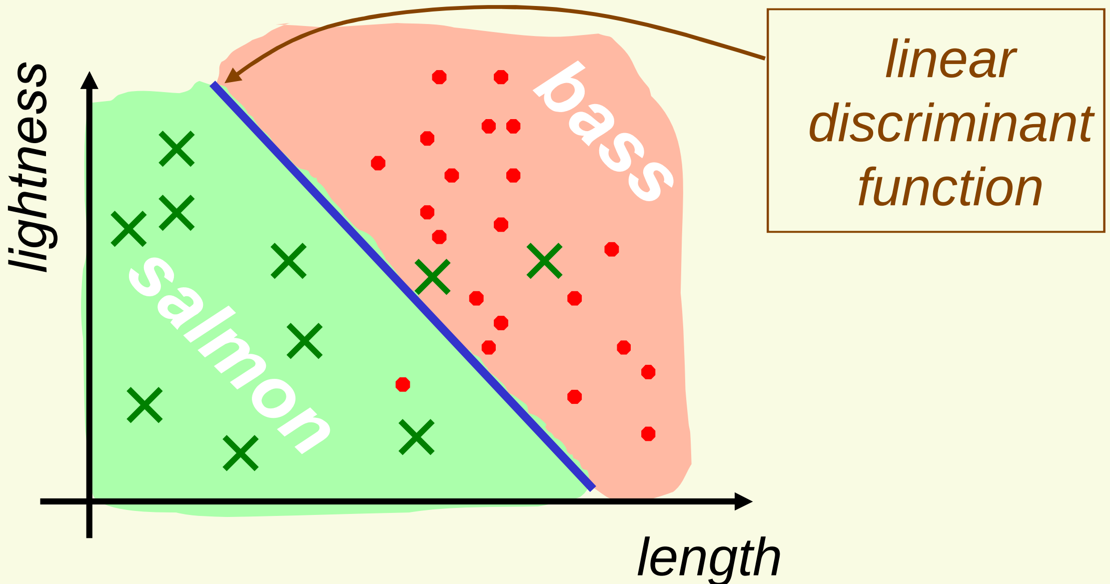
- Need to estimate parameters $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2$
- Then can use the methods from the Bayesian decision theory

little is known "harder"

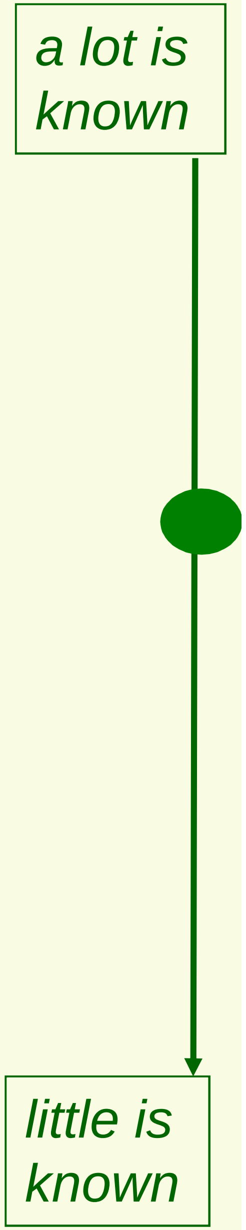
Linear discriminant functions and Neural Nets

- No probability distribution (no shape or parameters are known)
- Labeled data


- The shape of discriminant functions is known



- Need to estimate parameters of the discriminant function (parameters of the line in case of linear discriminant)



Non-Parametric Methods

- Neither probability distribution nor discriminant function is known
  - Happens quite often
- All we have is labeled data

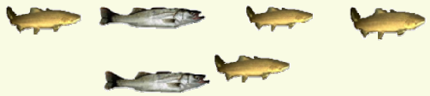
- Estimate the probability distribution from the labeled data

a lot is known "easier"

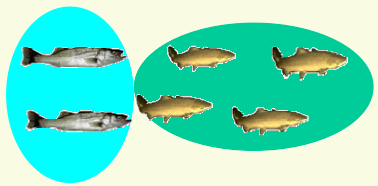
little is known "harder"

Unsupervised Learning and Clustering

- Data is *not labeled*
  - Happens quite often



1. Estimate the probability distribution from the *unlabeled* data
2. Cluster the data



a lot is known ”easier”

little is known “harder”

Classification Summary

1. Bayesian Decision theory (rare case)
   - Know probability distribution of the categories
   - Do not even need training data
   - Can design optimal classifier
2. ML and Bayesian parameter estimation
   - Need to estimate Parameters of probability dist.
   - Need training data
3. Linear discriminant functions and Neural Nets
   - The shape of discriminant functions is known
   - Need to estimate parameters of discriminant functions
4. Non-Parametric Methods
   - No probability distribution, labeled data
5. Unsupervised Learning and Clustering
   - No probability distribution and unlabeled data

a lot is known

little is known