Biostat 202: Opportunities and Challenges of "Big Data". Prediction Algorithms 2: Classification

John Kornak

Division of Biostatistics

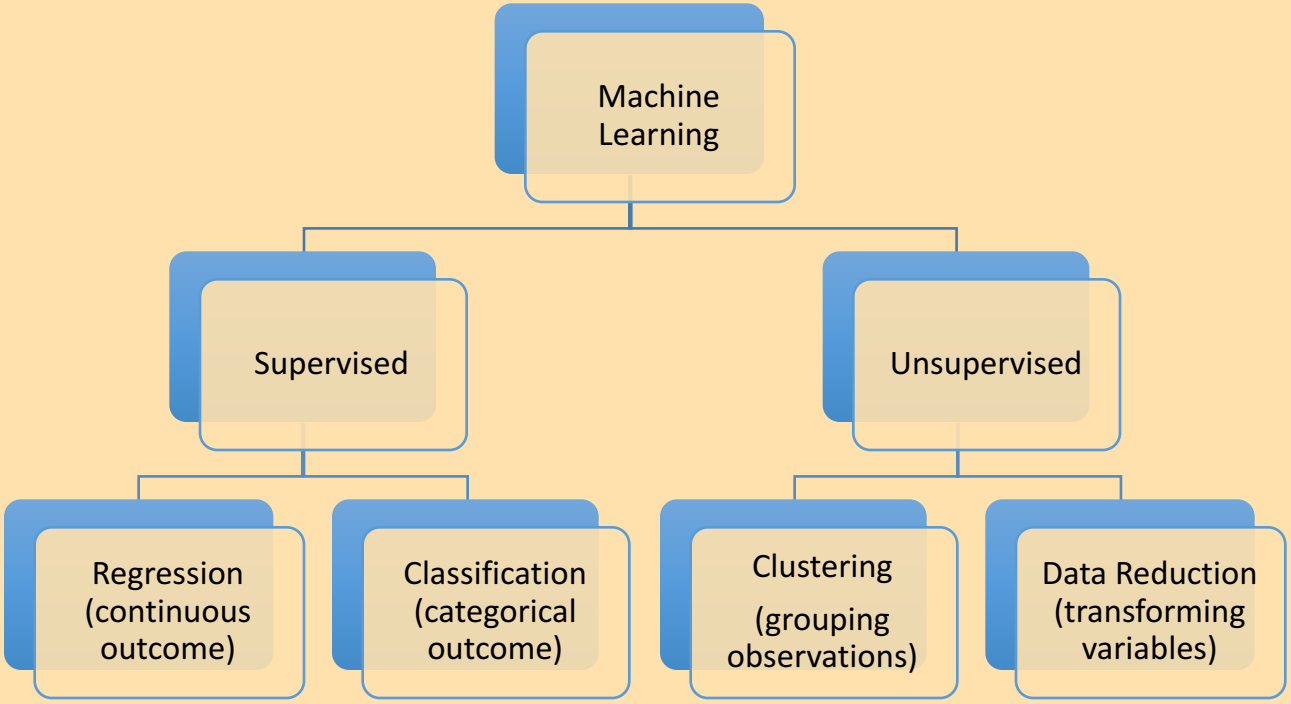
Department of Epidemiology and Biostatistics

08/18/2016

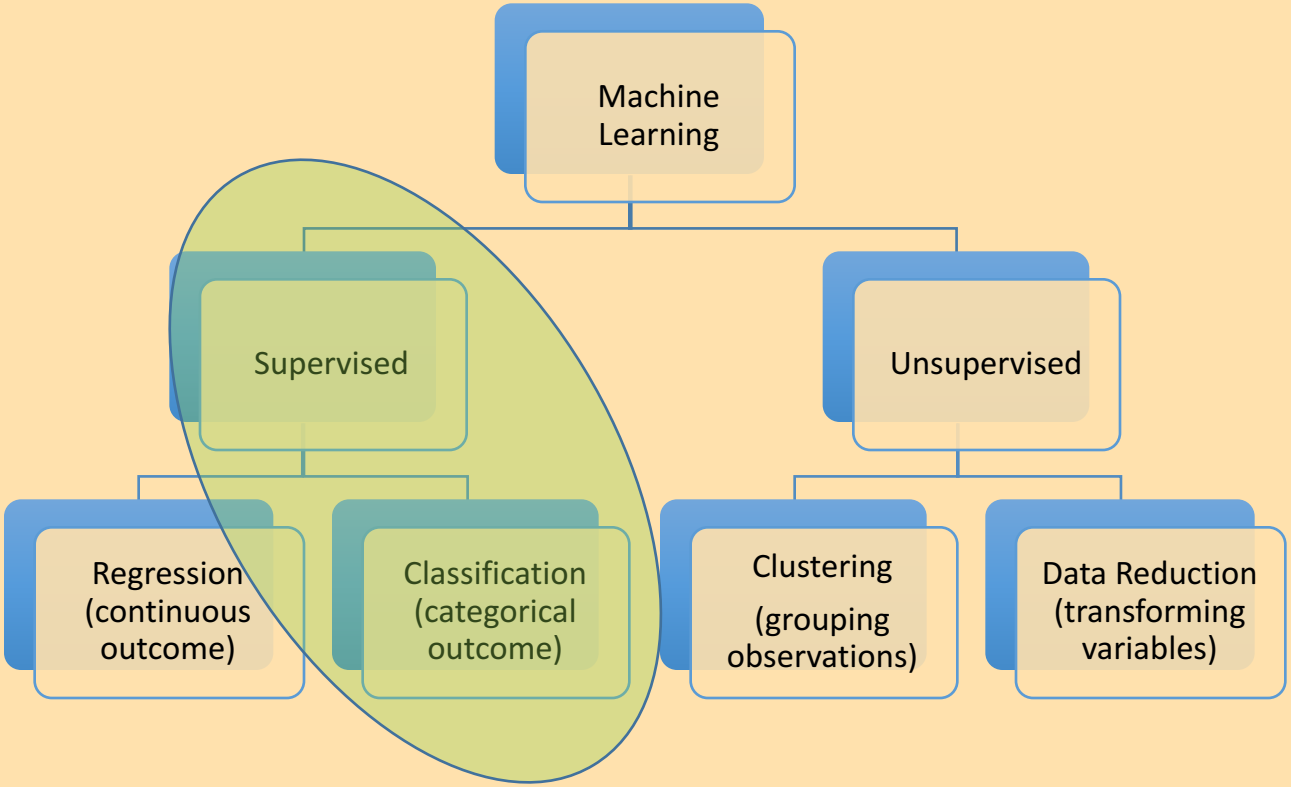
Overview

- Classification
- Model evaluation and training-test
- Classification methods
- Training-validation-test
- Cross validation
- More classification methods at a glance
- IBM SPSS Data Modeler examples left for todays lab

Supervised vs. unsupervised



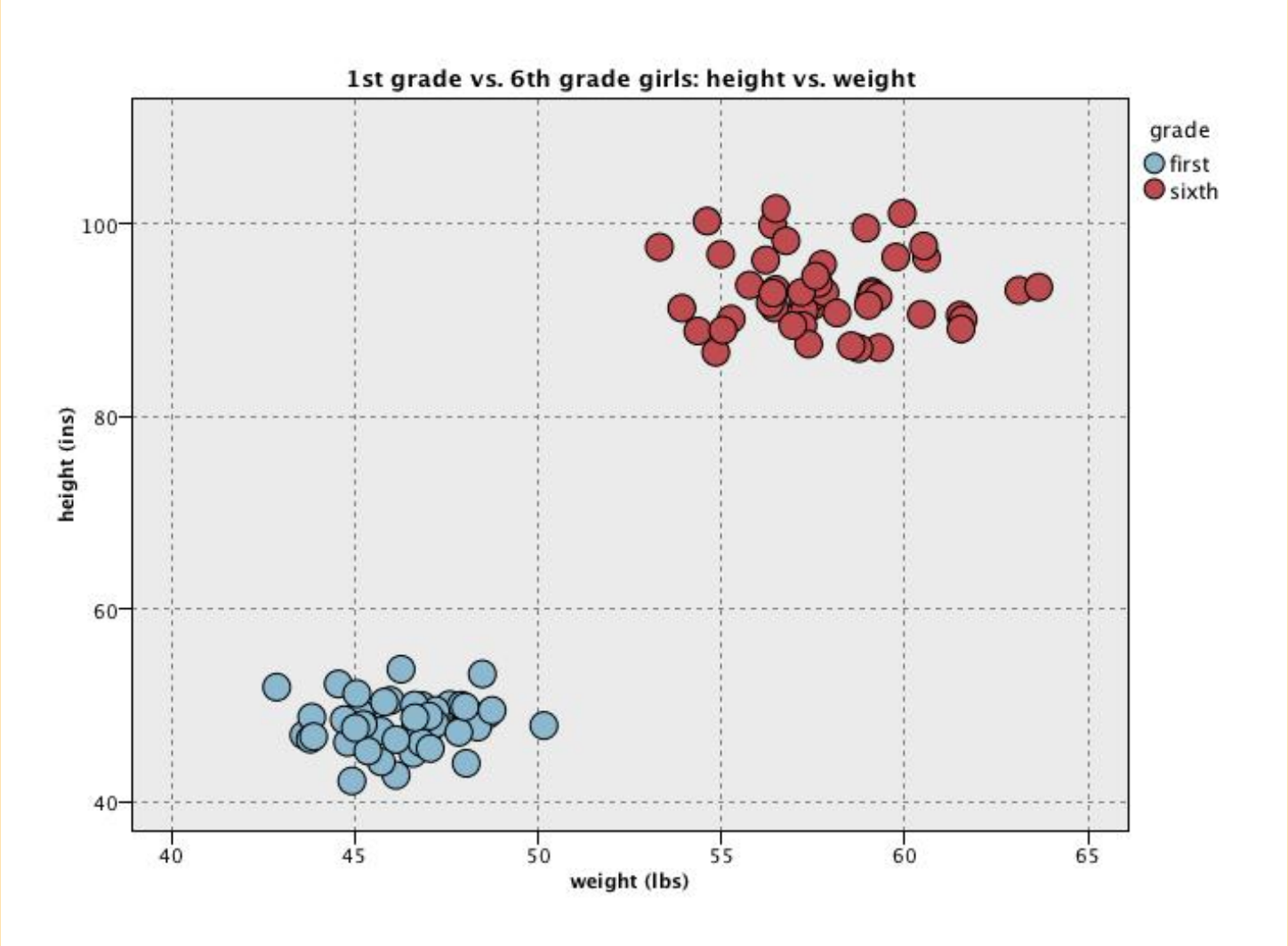
Supervised vs. unsupervised



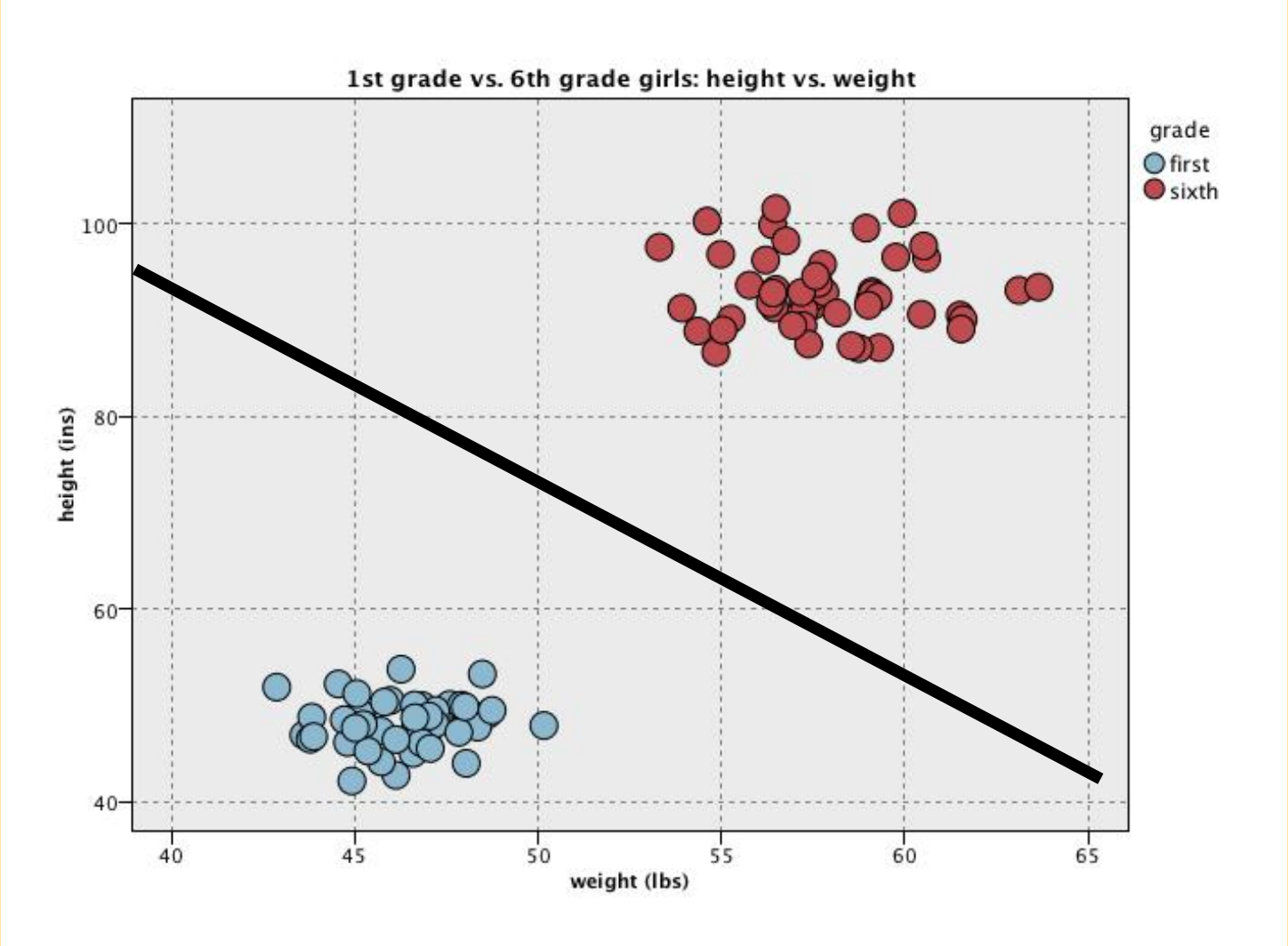
Classification

- Objective – given set of predictors and some training data, generate a rule (classifier) to predict which category a new instance/observation/subject belongs to
- Training data with predictors and outcomes used to develop classifier
- Classifier is applied to new data where the outcome (i.e. target group/class) is unknown
- Let's look at a toy example trying to classify 1st vs. 6th grade girls based on height and weight….

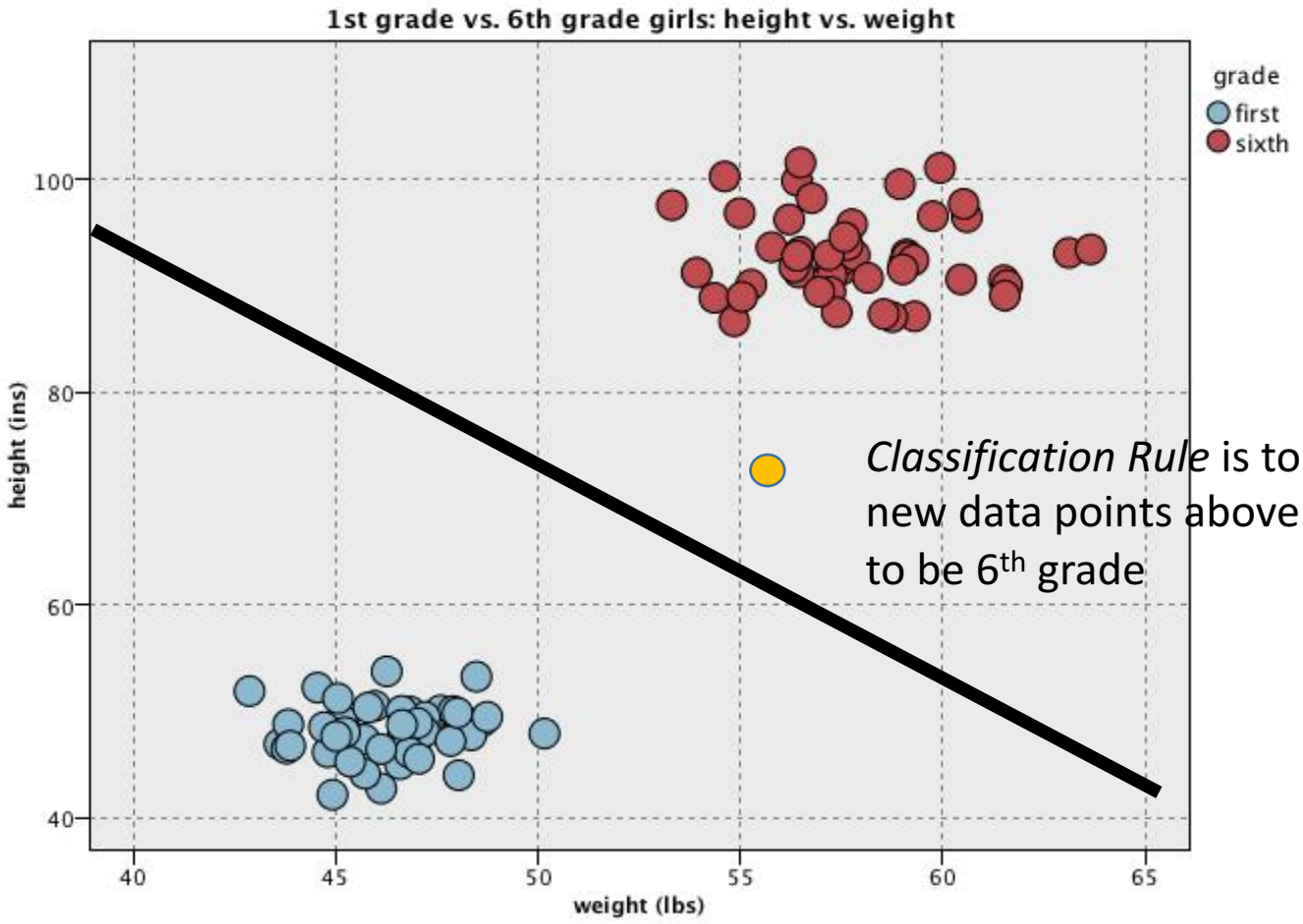
2D Classification example



2D Classification example

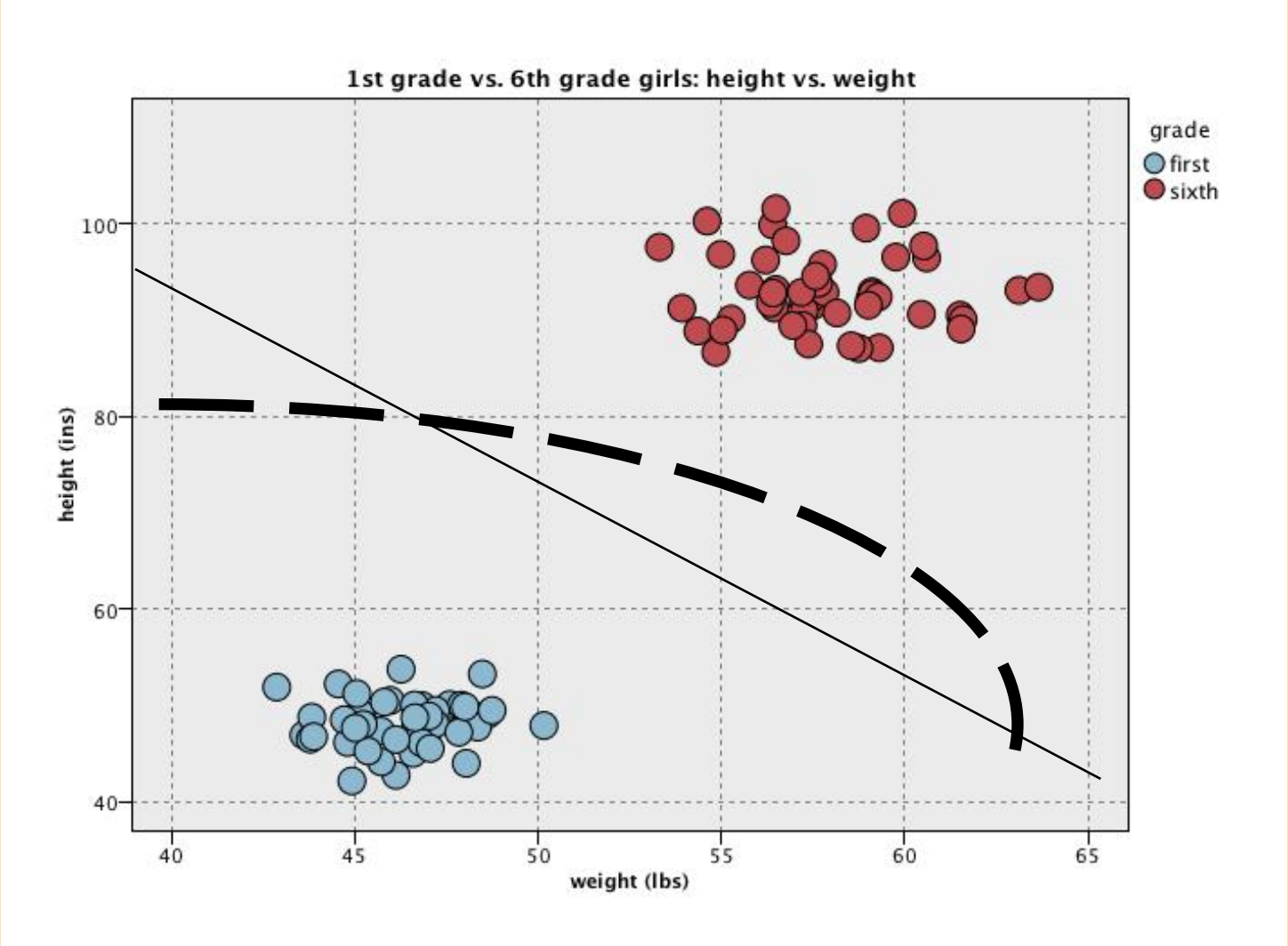


2D Classification example

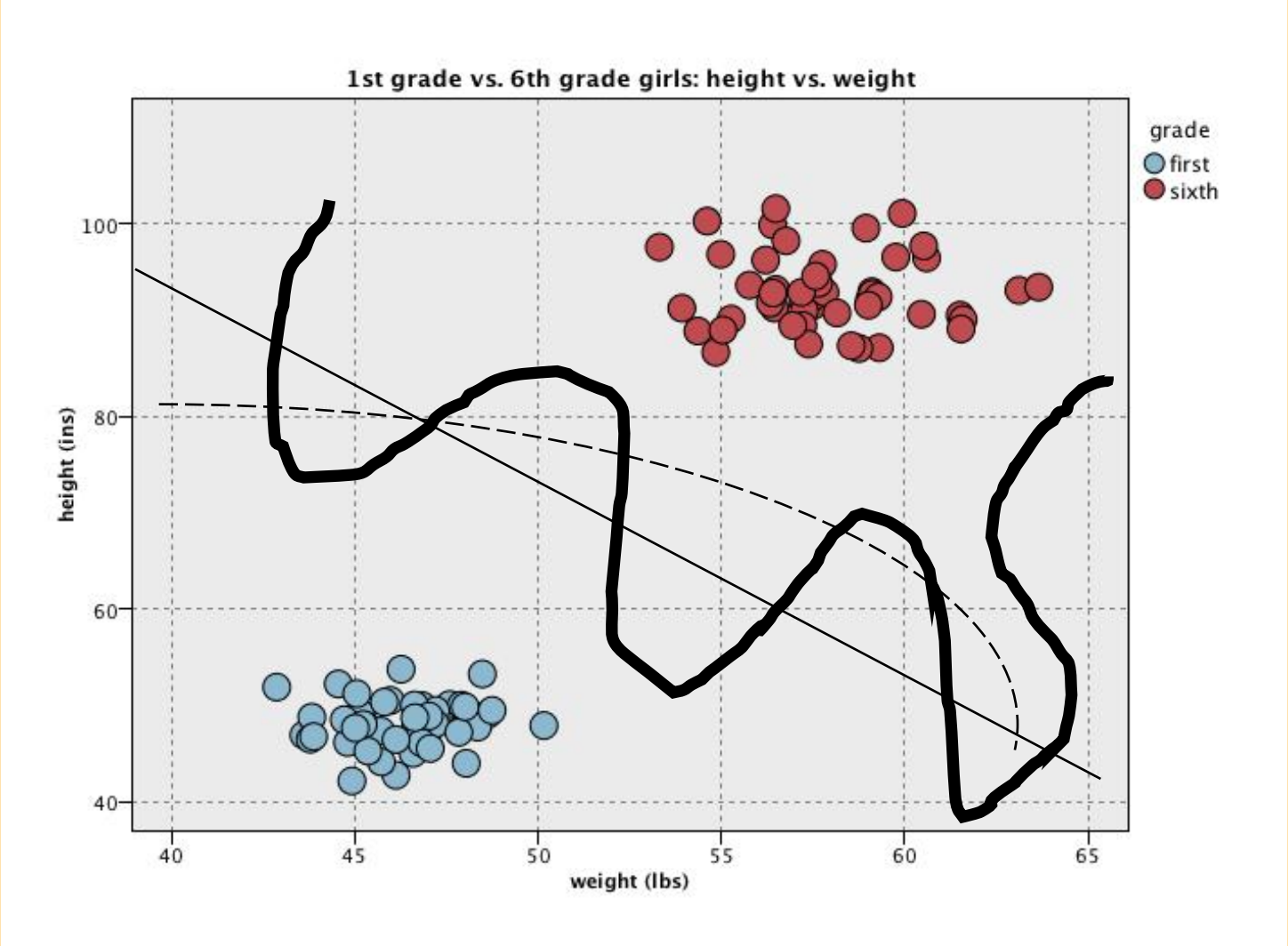


Classification Rule is to predict new data points above the line to be 6th grade

2D Classification example

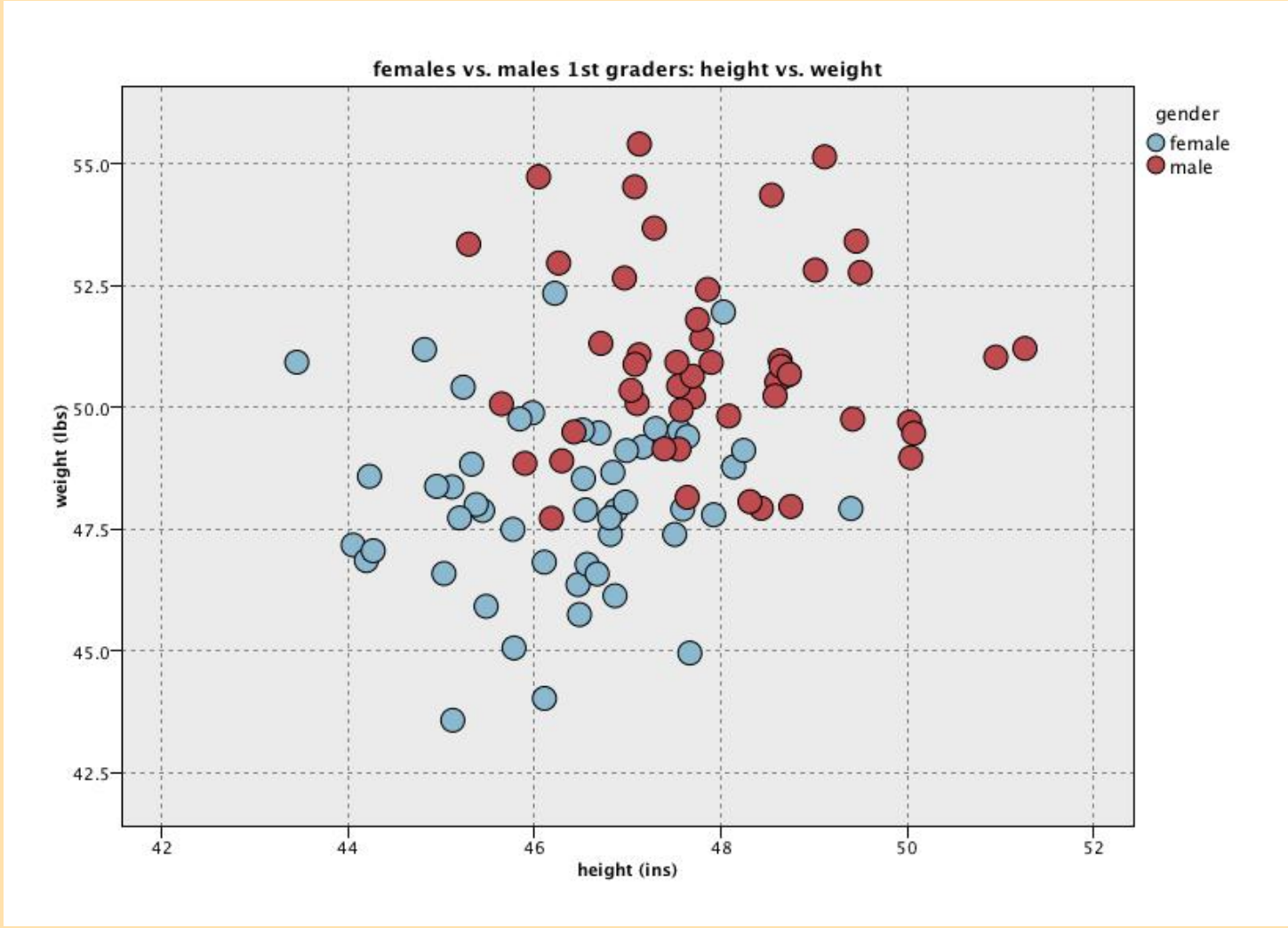


2D Classification example

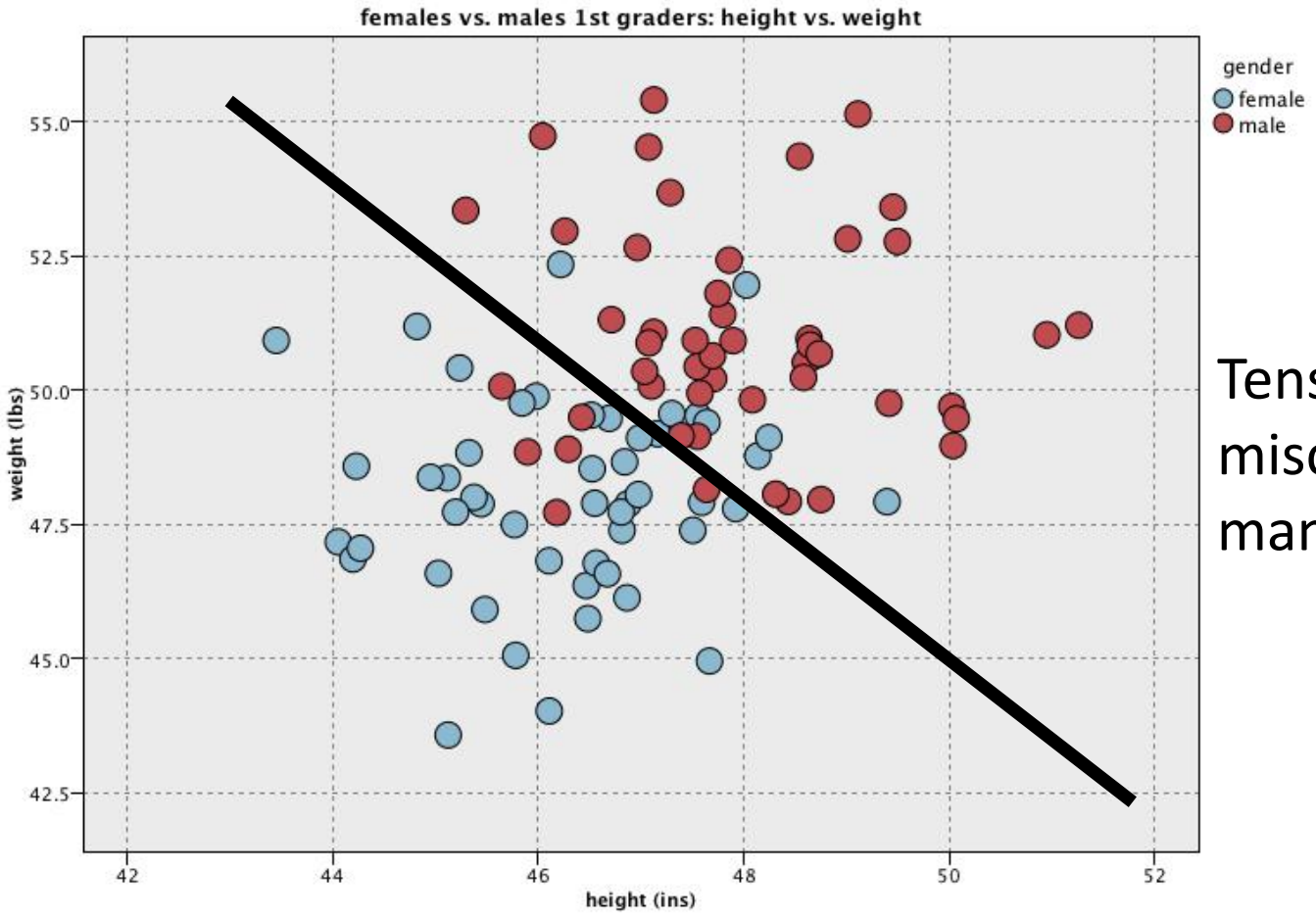


- But in real problems the separation is unlikely to be so neat
- Consider 1st grade boys vs. 1st grade girls

2D Classification example

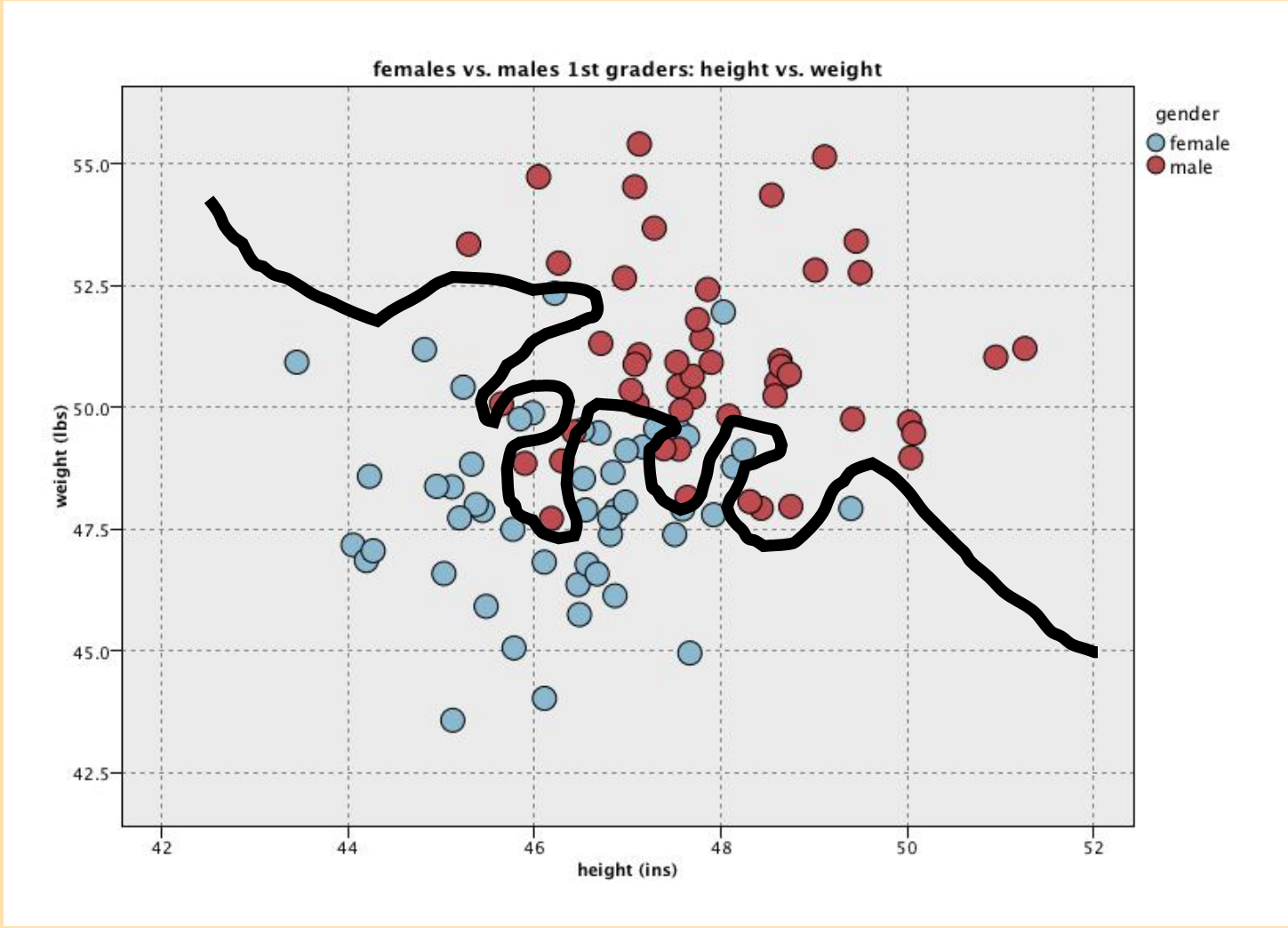


2D Classification example

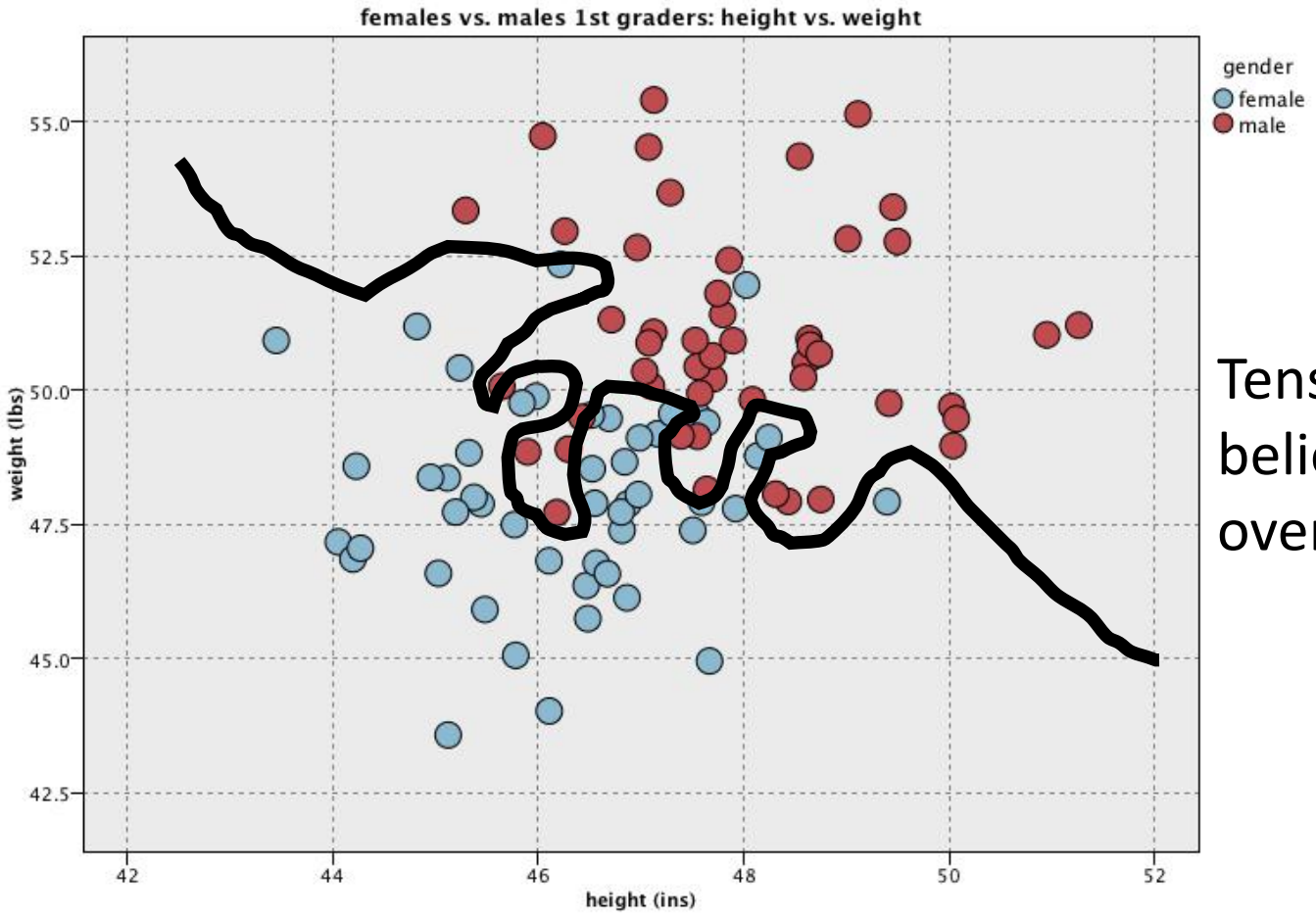


Tension: misclassifying many points

2D Classification example



2D Classification example



Tension: don't believe real - overfitting

The trade-off

- Want to find true predictive patterns in the data
- But, want to avoid overfitting spurious detail: i.e. don't want to make the model capture apparent patterns due to noise - we want results that are likely to be *generalizable* to future observations
- Solution – independent testing – reserve some data that has not been used in model fitting for the purpose of model evaluation and comparison

Evaluation

- Possible evaluation measures:
  - *Classification Accuracy* – or conversely, the *classification error rate* = 1 – classification accuracy
  - *Total cost/benefit* – different errors may have different costs
  - *Gini Index*, *cross-entropy* (we'll look at these later)
  - ….
- How reliable are the predicted results ?

Classifier error rate

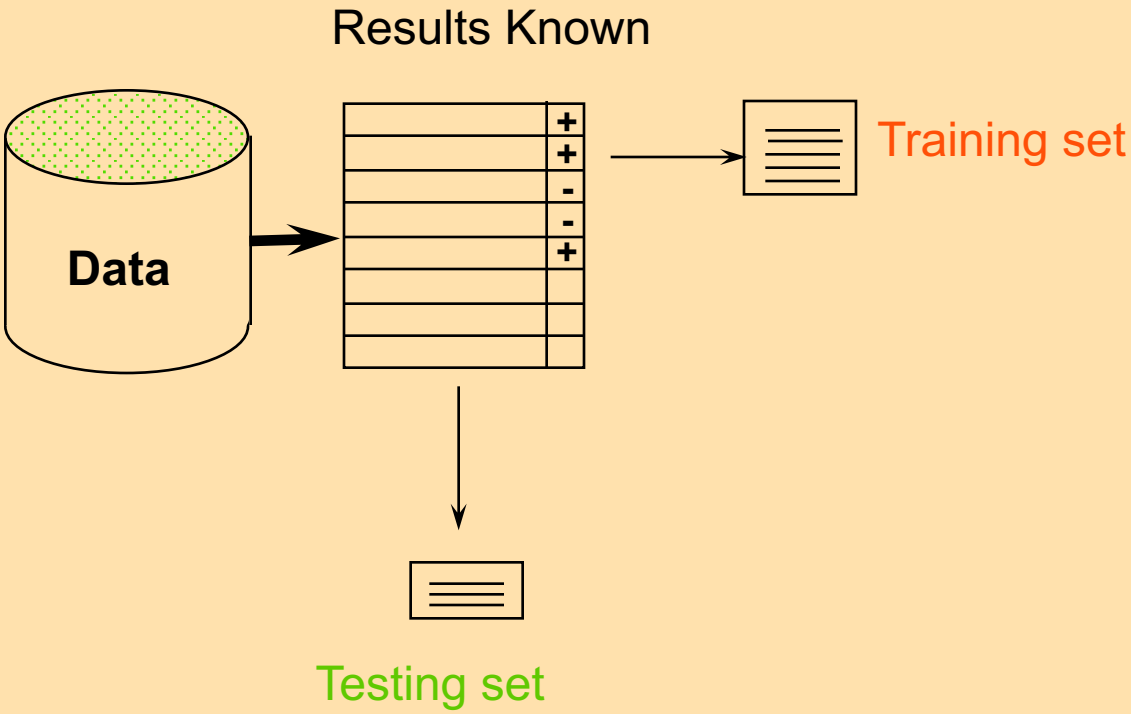
- Natural performance measure for classification problems: *error rate*
  - *Success*: instance's class is predicted correctly
  - *Error*: instance's class is predicted incorrectly
  - *Error rate*: proportion of errors made over the whole set of instances
- *Training set error rate:* is generally too optimistic!
  - you can find patterns even in random data

When you have lots of "instances"

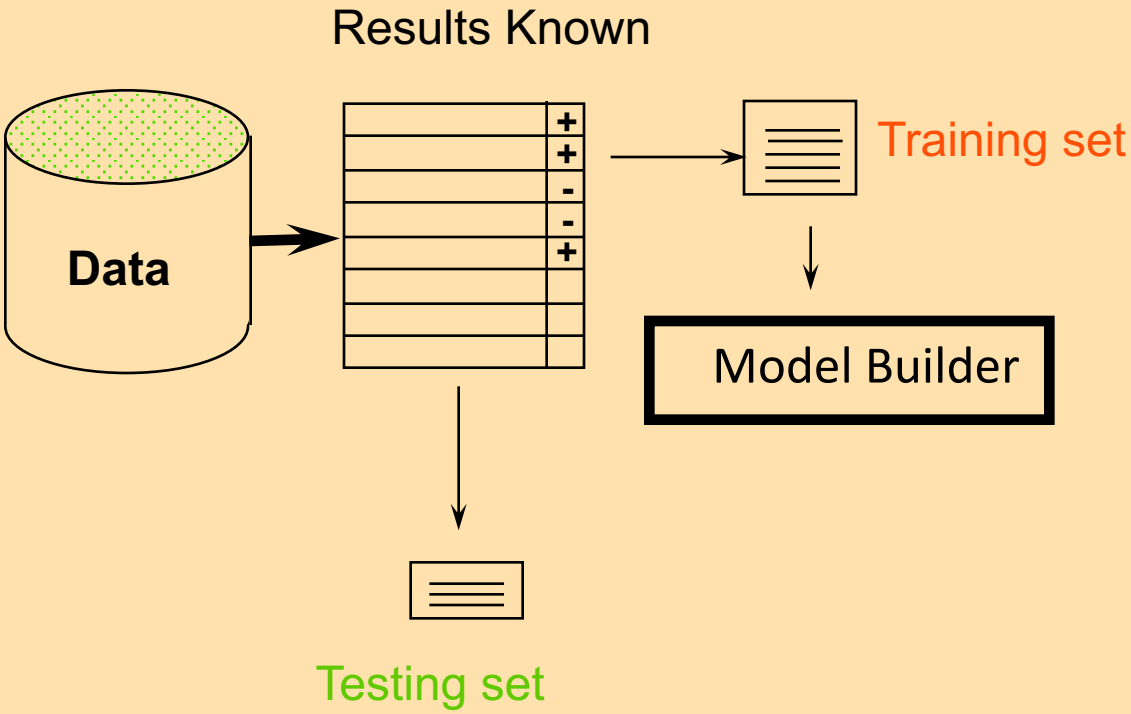
If many (>1000) instances are available, including >100 from each class

- A simple evaluation training/test evaluation will work
  - Randomly split data into training and test sets (often ~2/3 for train, 1/3 for test) – may want to *stratify* on outcome (have matching proportions of each class in each set)
- Build a classifier using the *training* set and evaluate it using the *test* set
- We will look at cross-validation later for when you do not have so many observations

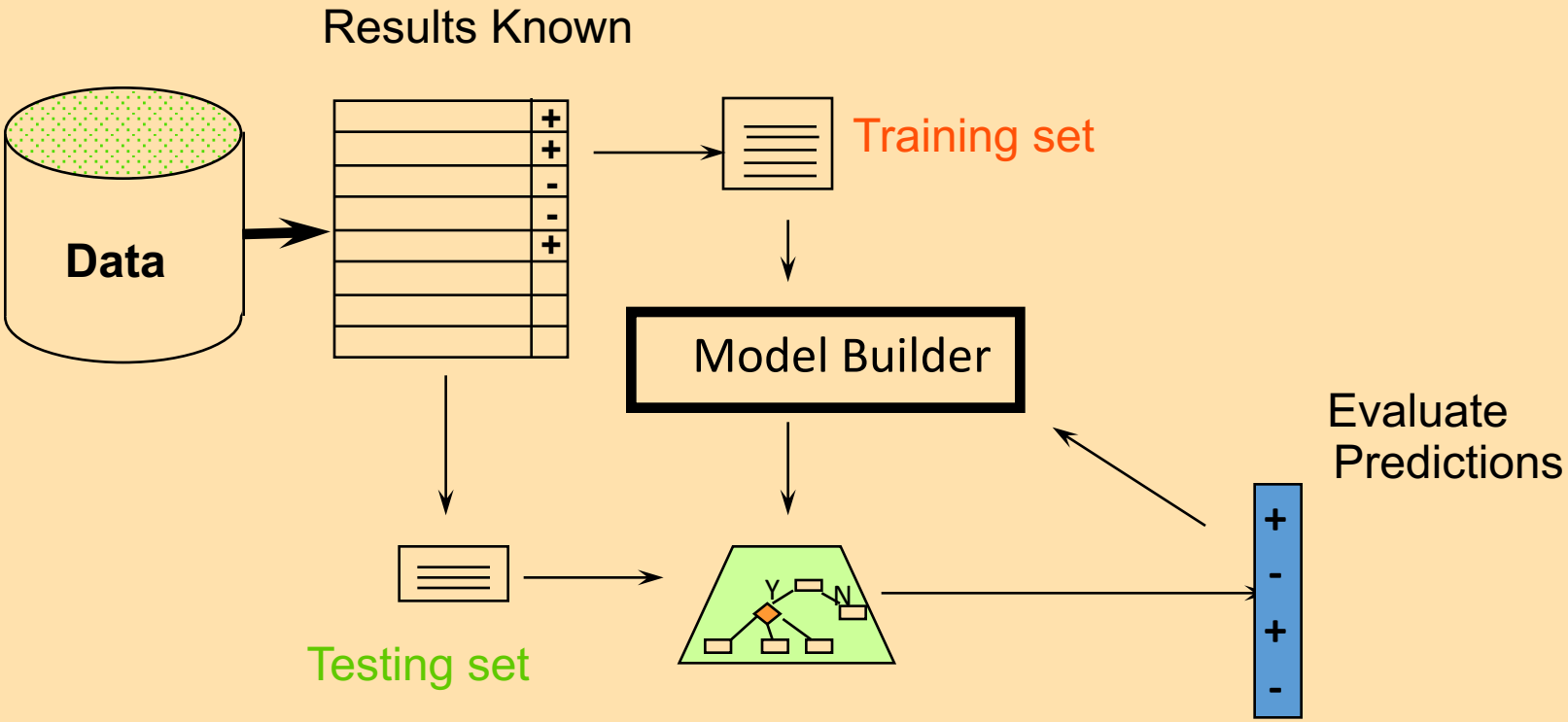
When you have lots of "instances"



When you have lots of "instances"



When you have lots of "instances"



Classification methods

- Logistic regression
- K-nearest neighbors
- Classification trees
- Support vector machines (SVMs)
- Linear Discriminant Analysis (LDA)

Logistic Regression

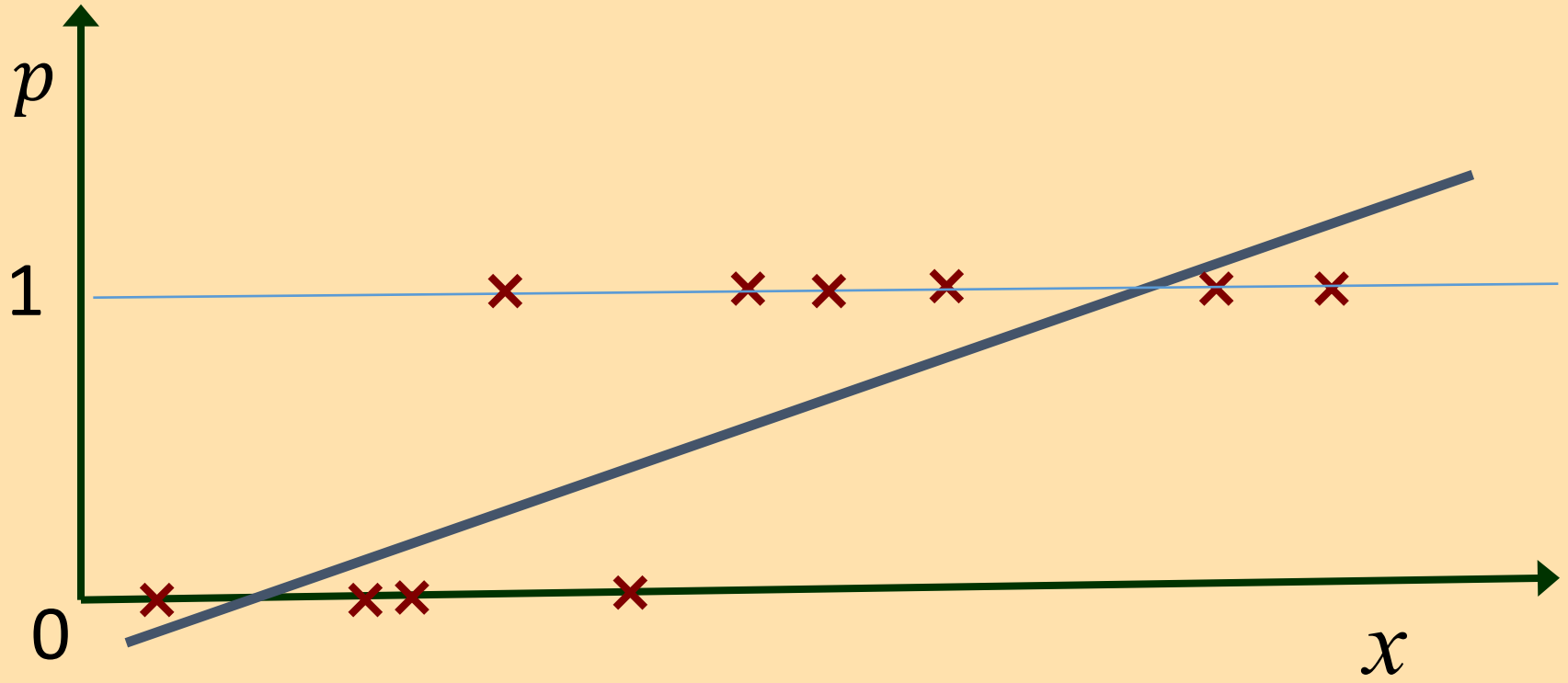
Binary/flag outcome regression modeling

- We want a model where the probability of being in a class depends on predictor(s)
- What about using linear regression?
- $p = ax + c$, where p is the probability of being in class 1 vs. 0, e.g. $p =$ probability boy (=1) vs. girl (=0)
- What might be the concern here?

Binary/flag outcome regression modeling

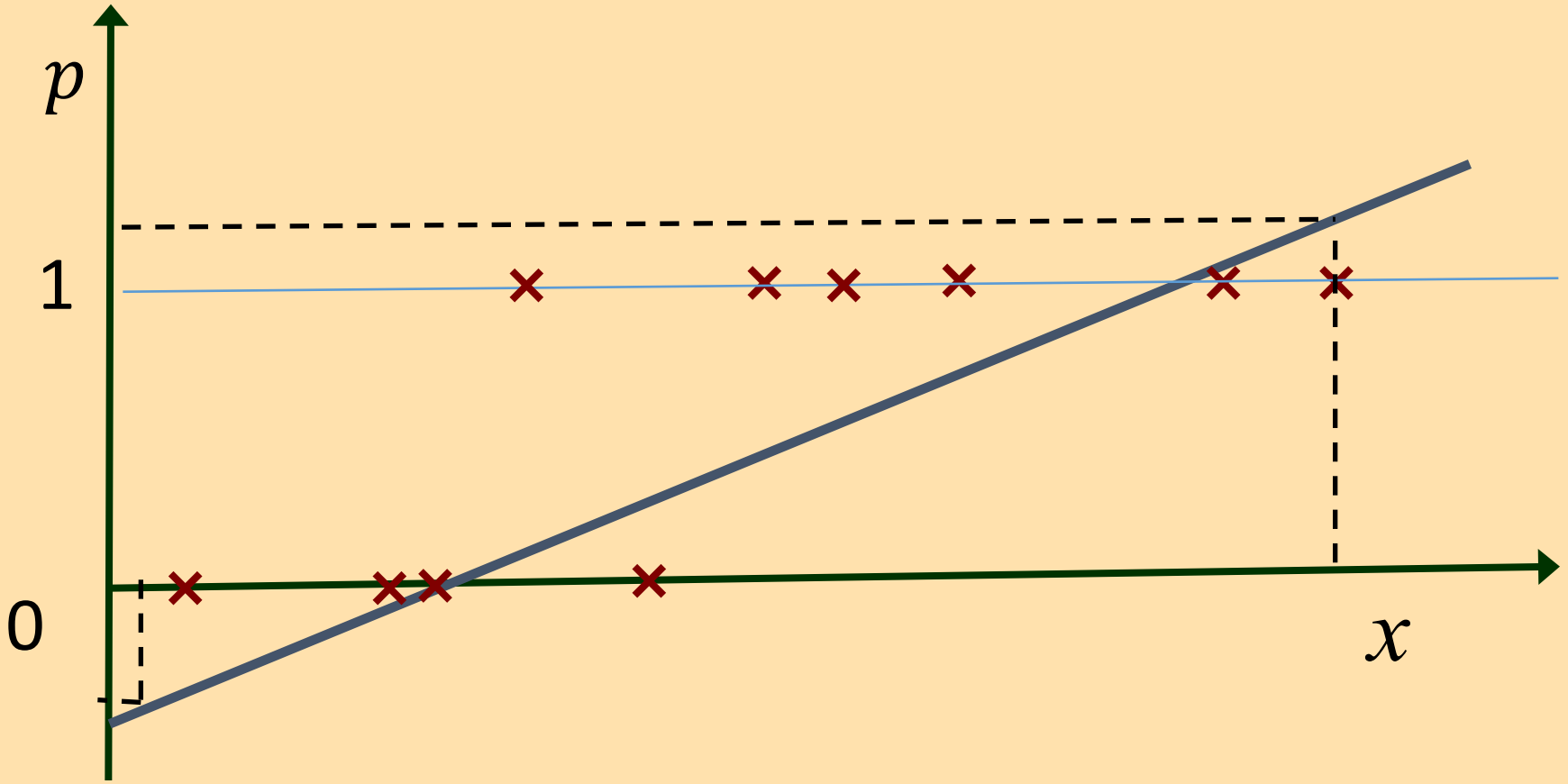
- We want a model where the probability of being in a class depends on predictor(s)
- What about using linear regression?
- $p = ax + c$, where p is the probability of being in class 1 vs. 0, e.g. $p =$ probability boy (=1) vs. girl (=0)
- What might be the concern here?
- Possible values for p are not restricted to be between 0 and 1 (as required for probabilities) – can be >1 or <0

Linear regression?



Linear regression

Problem predictions – outside range 0 to 1



Logistic regression

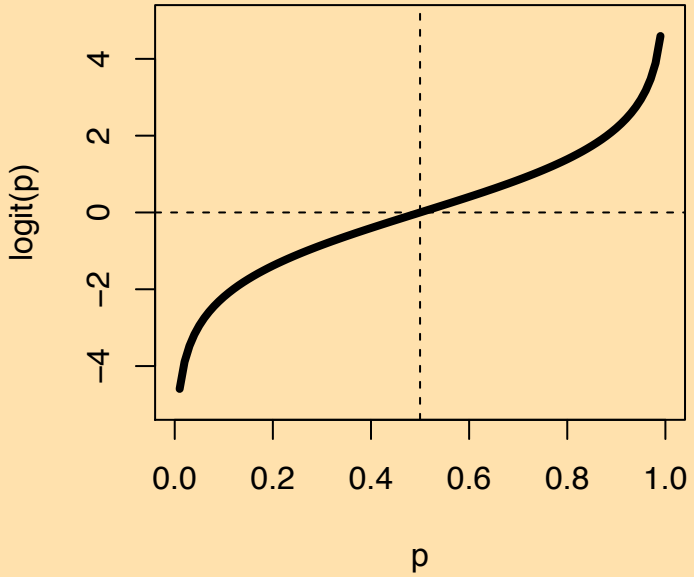
p must be between 0 and 1 (probability)

We can instead take the *odds* which is between 0 and infinity: $\frac{p}{1-p}$ but still not quite what we need…

to have values that are unrestricted we take the log of the odds (*logit function*) and get the logistic regression equation:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = ax + c$$

We can extend logistic regression to multiple predictors just as we did for linear regression.



Logistic regression for classification

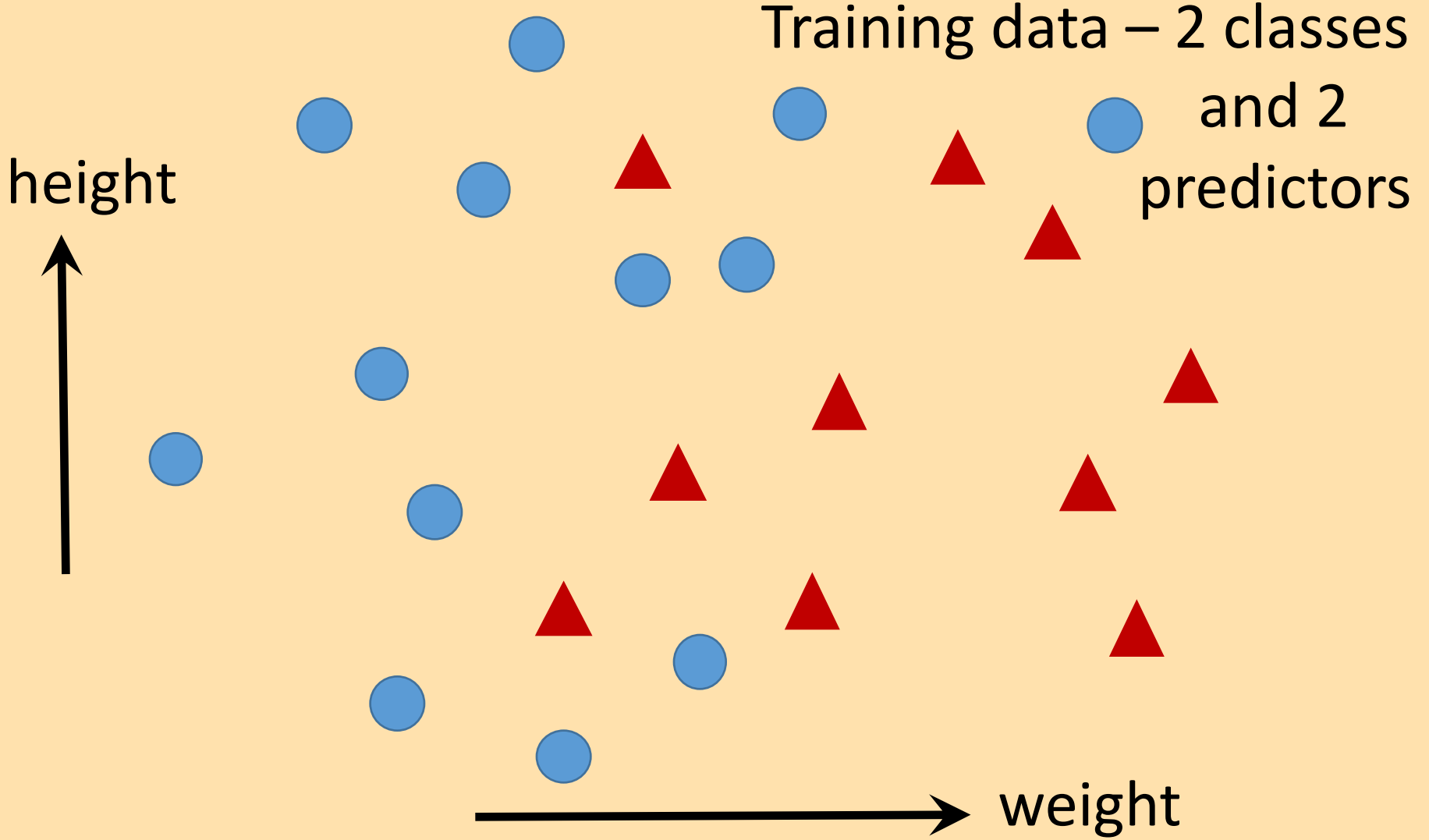
- When fitting logistic regression we get a model that relates predictors to a class probability
- To convert the probability to a class we need to threshold the probability
- E.g. if probability $p > 0.5$ then class as boy, otherwise if $p \leq 0.5$ class as girl

Logistic regression

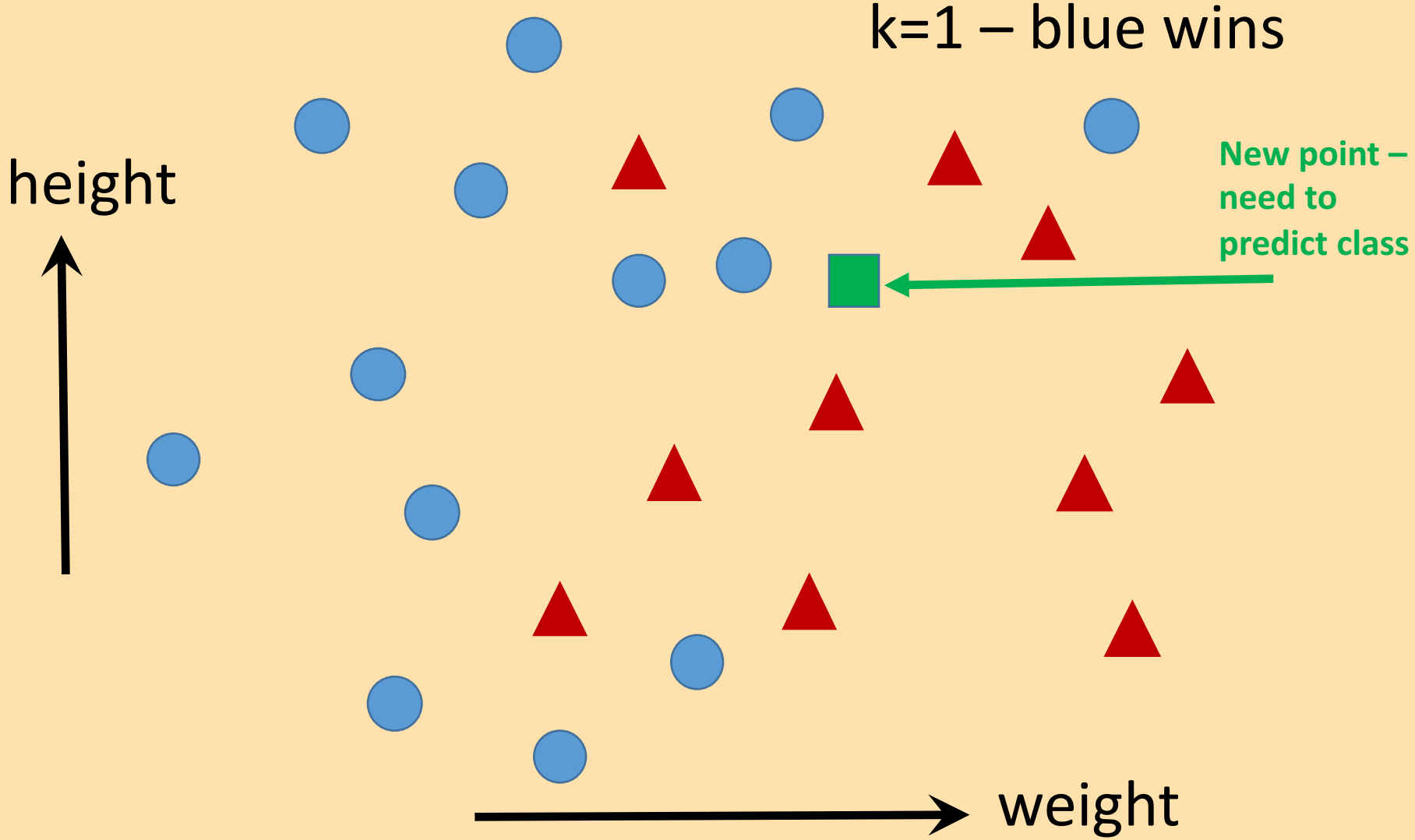
- A long history (relatively) – well known
- Much statistical theory
- Works well for many problems
- Can be extended to multiple categories
- Can use step-wise/subset selection techniques
- Can provide interpretable models

K-nearest neighbors classification

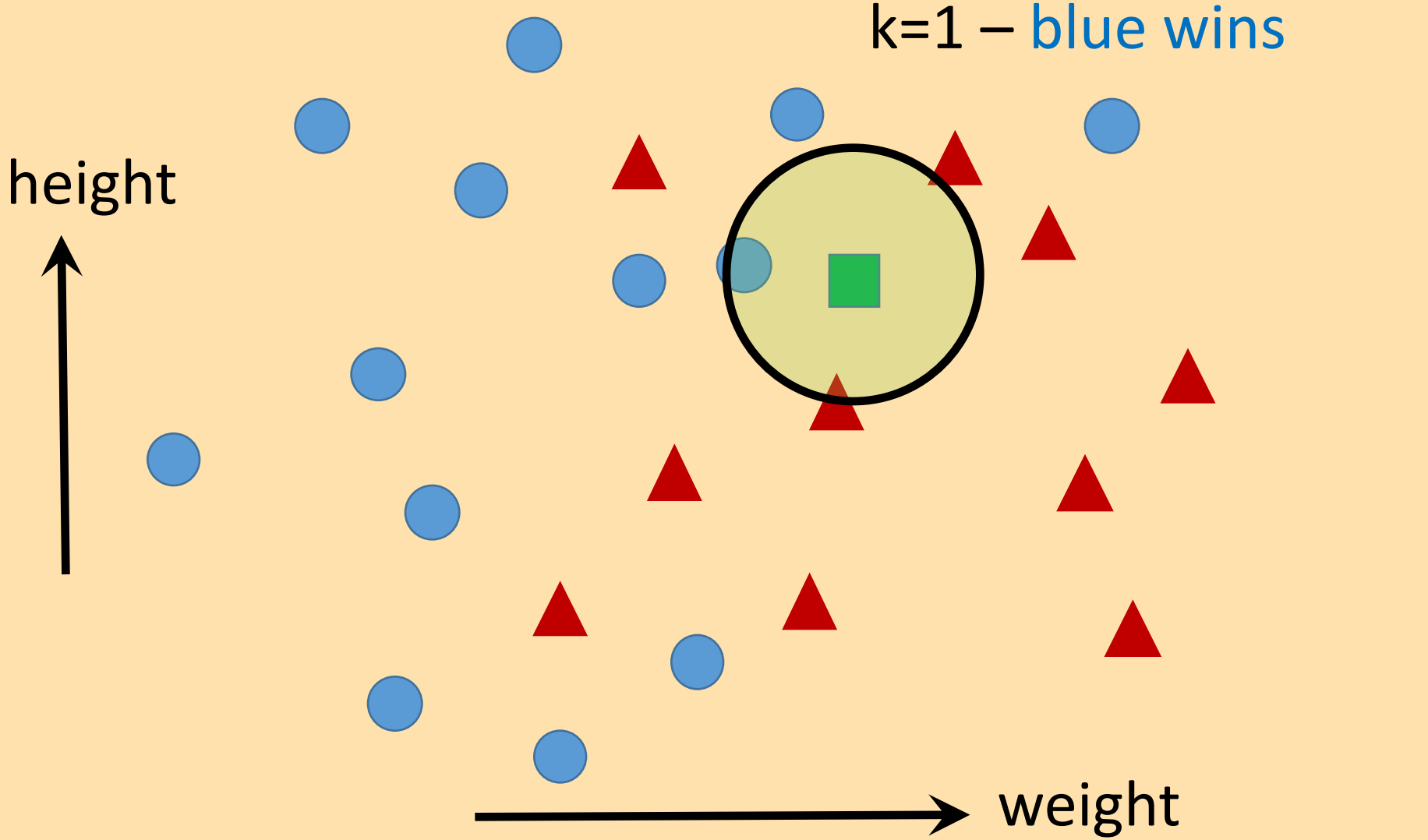
K-nearest neighbors classification



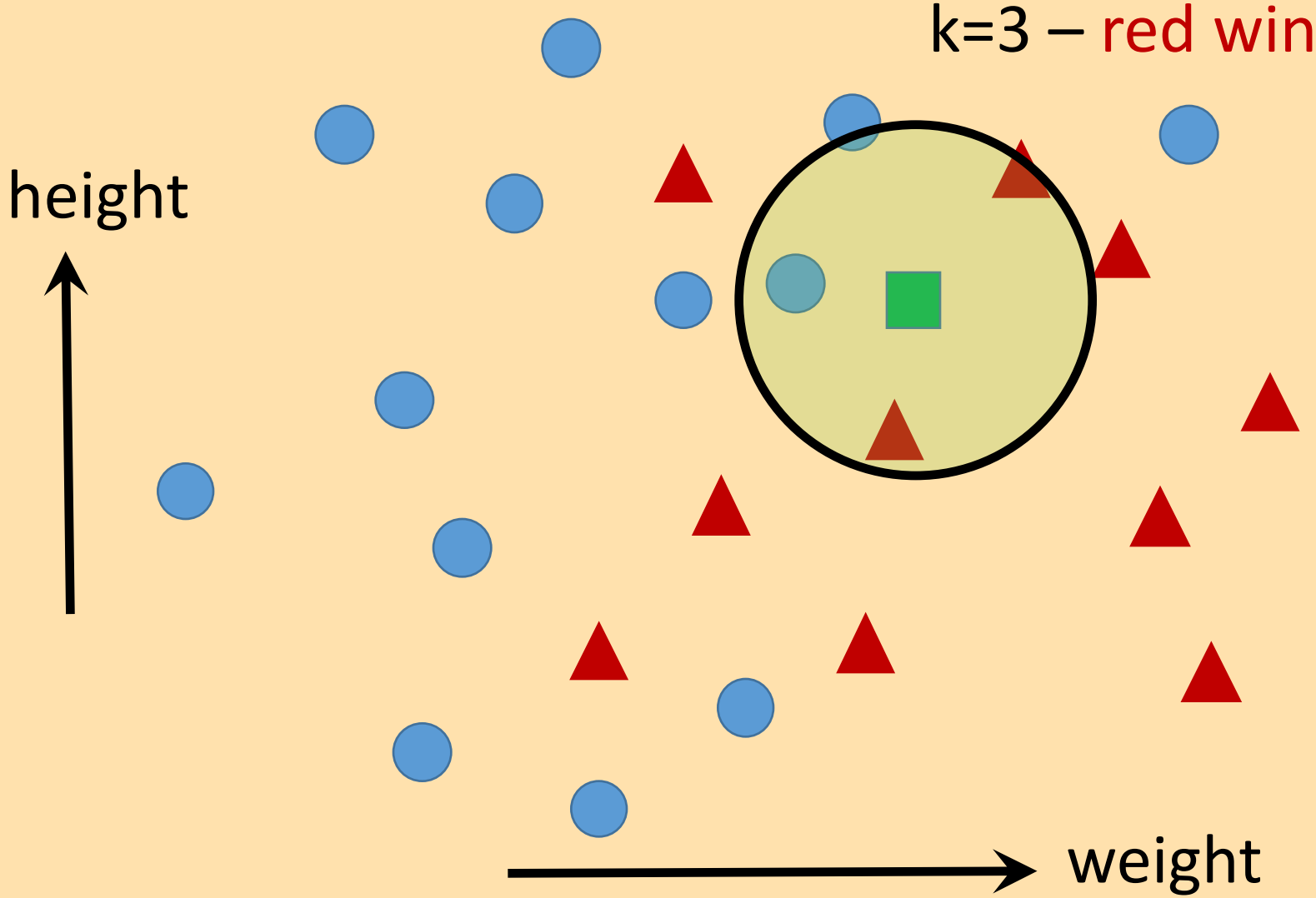
K-nearest neighbors classification



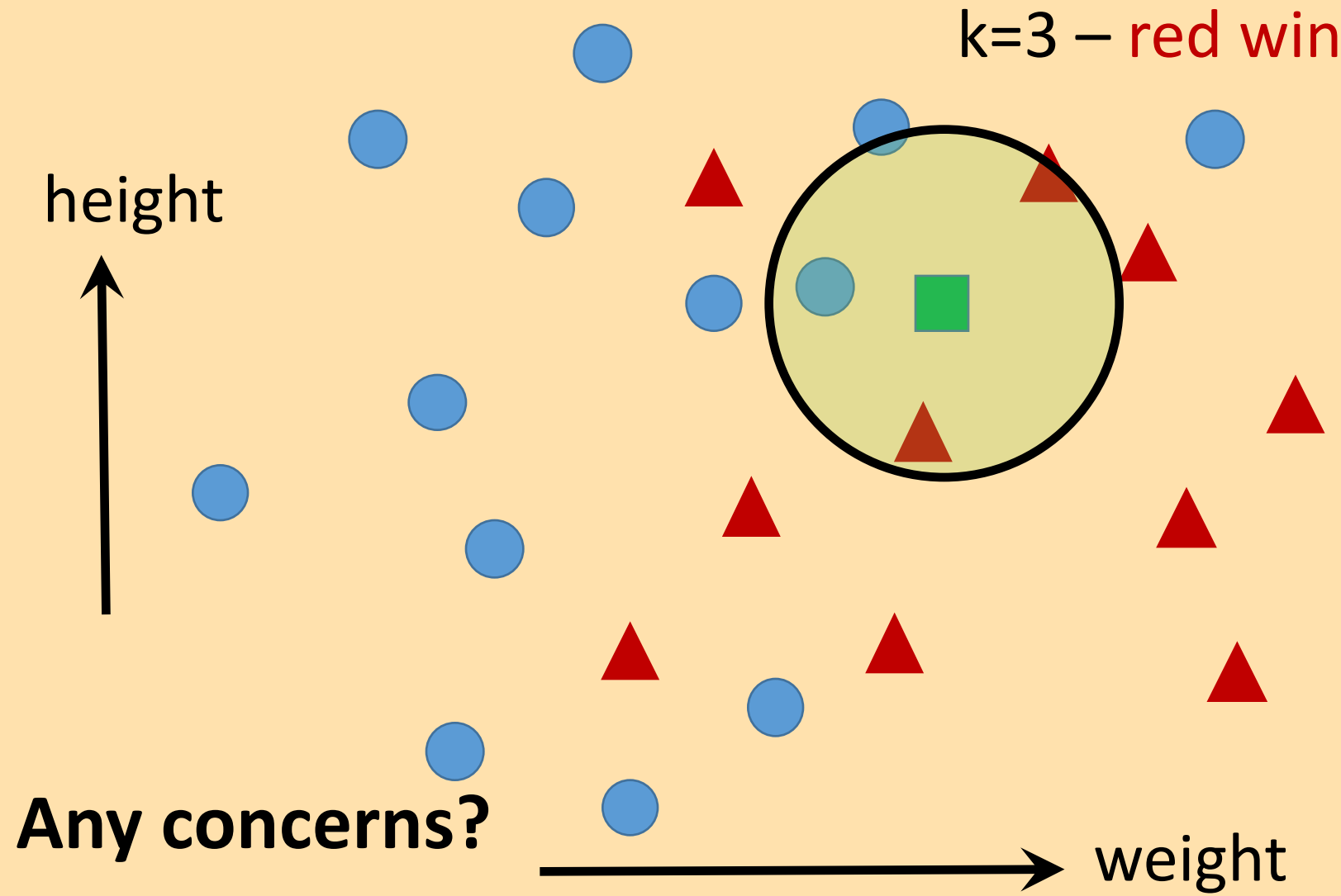
K-nearest neighbors classification



K-nearest neighbors classification



K-nearest neighbors classification



Any concerns?

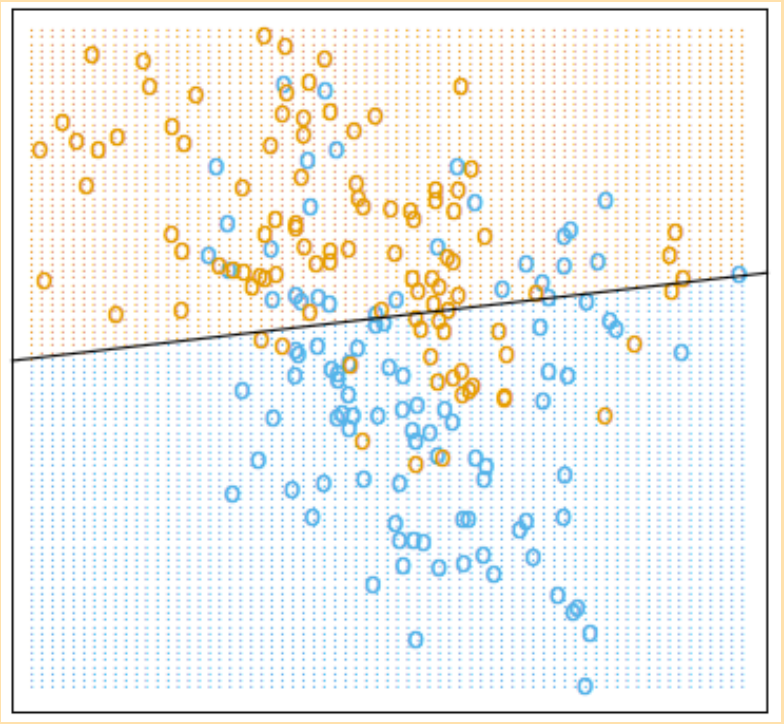
K-nearest neighbors classification

- Common to standardize predictors to have mean 0 and standard deviation of 1 (subtract mean and divide by standard deviation)
- Take majority vote among k-nearest neighbors
- Ties are broken at random (e.g. 1 red neighbor and 1 blue neighbor when k=2 – choose red or blue at random).
- Works for problems with more than 2 classes

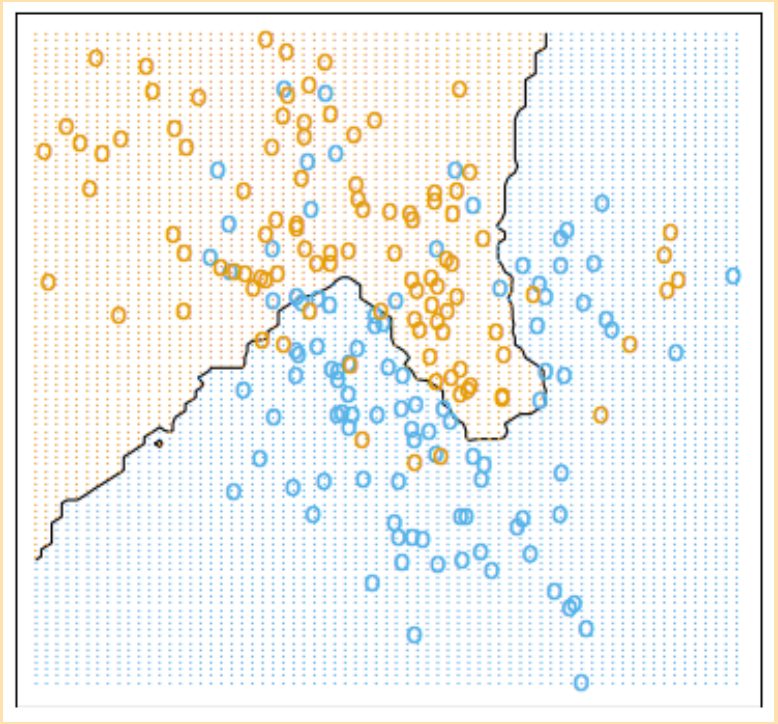
Choice of k

Example taken from Elements of Statistical Learning – Hastie et al. 2nd Ed. pp. 13-17

Linear decision boundary (LDA)

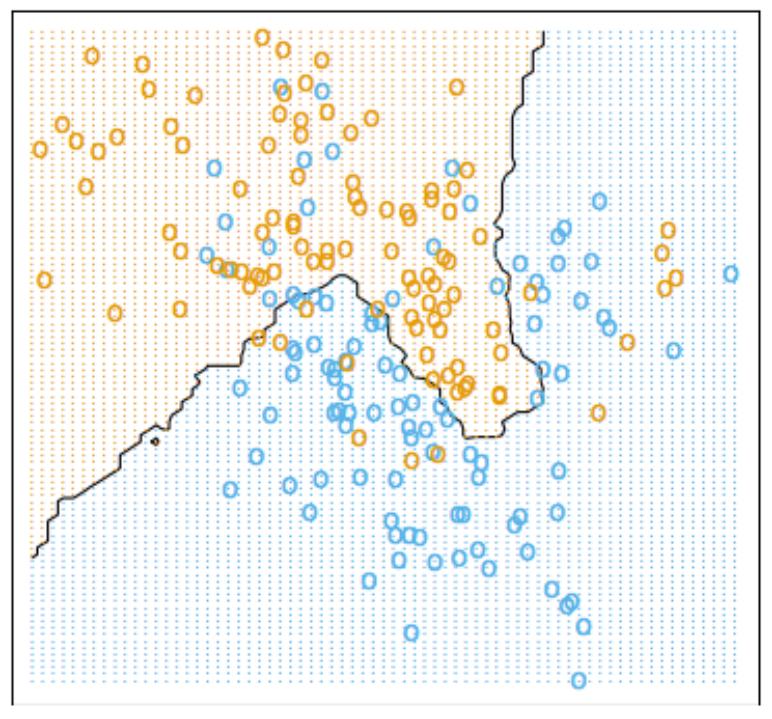


15 nearest-neighbor boundary

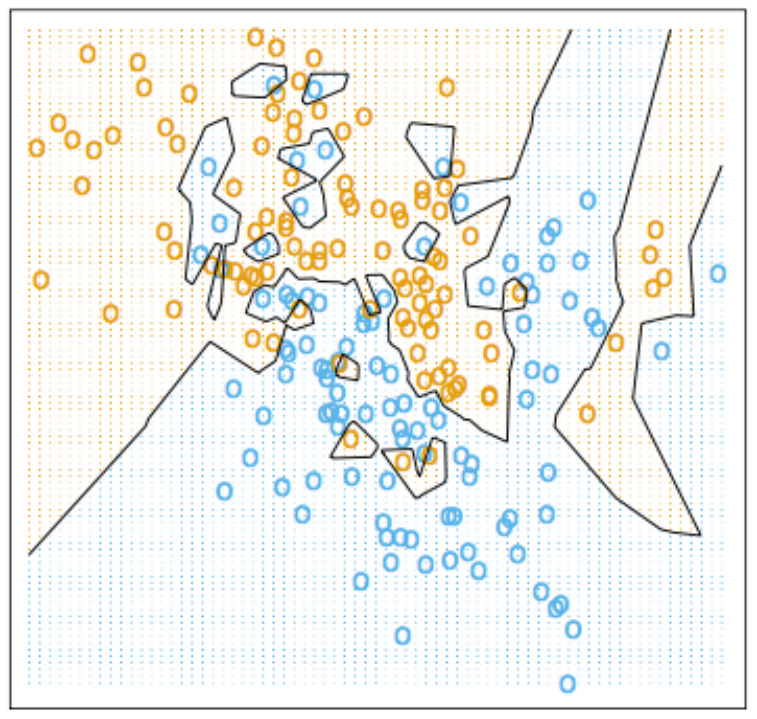


Choice of k

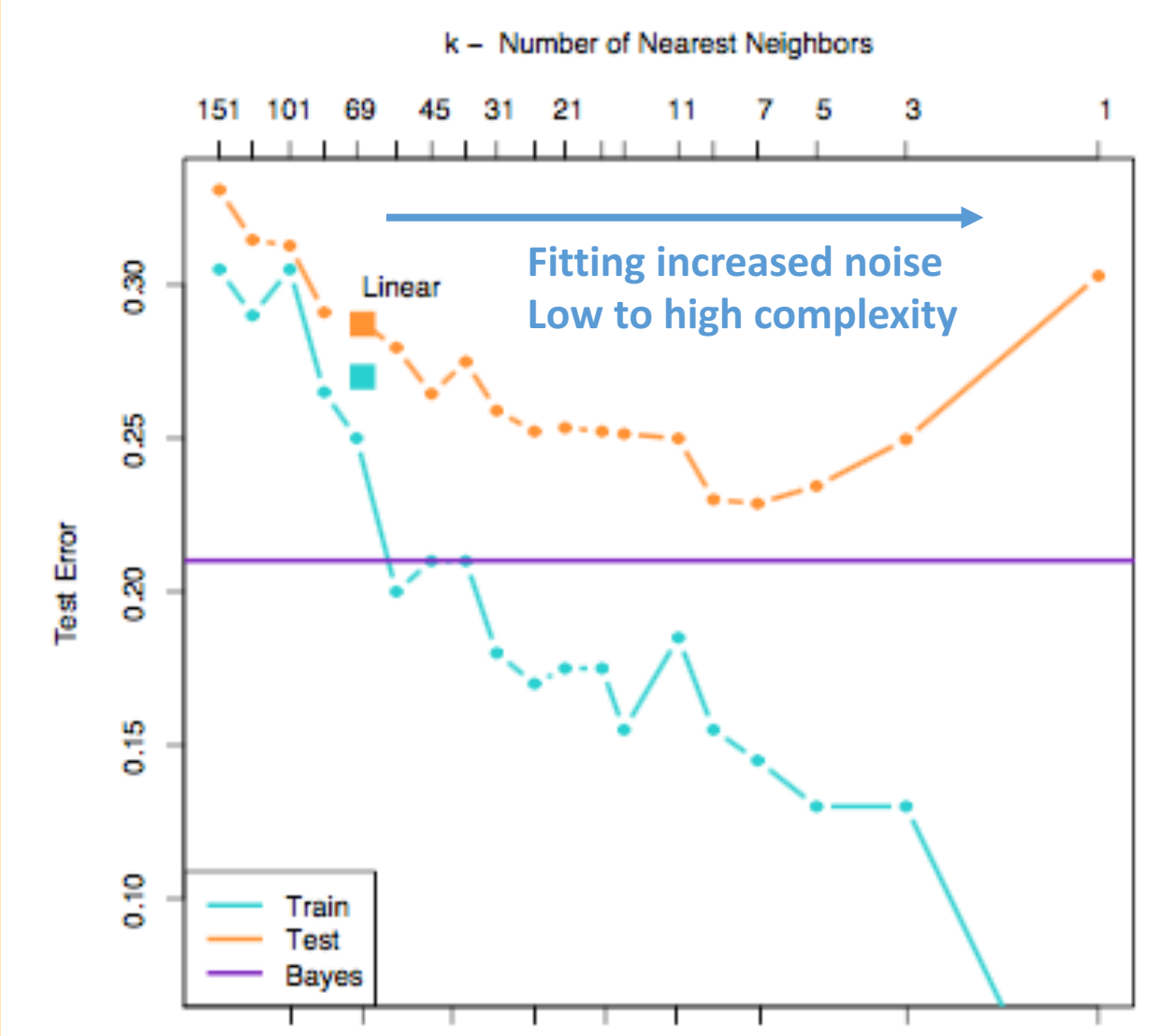
15 nearest-neighbor boundary



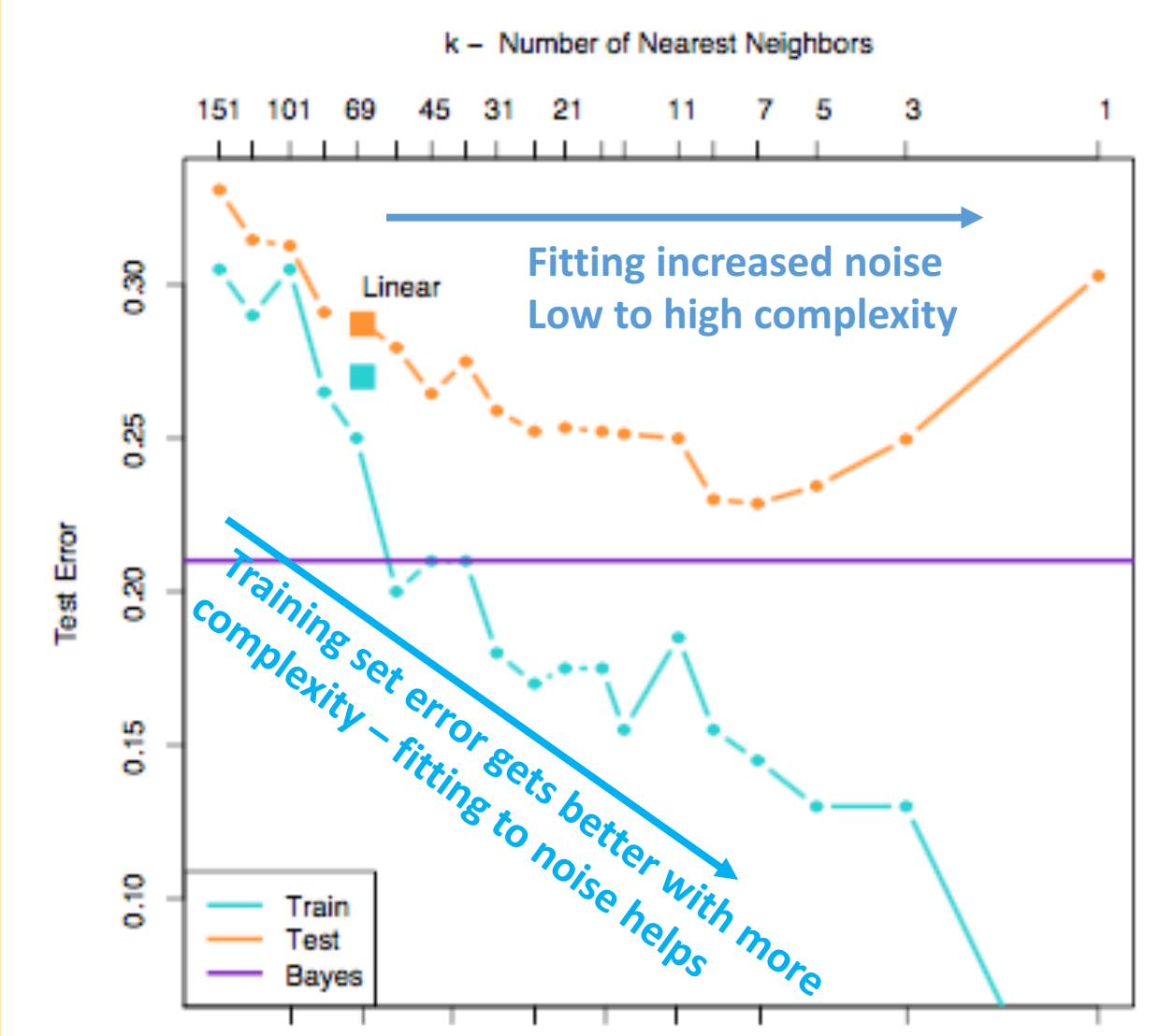
1-nearest neighbor boundary



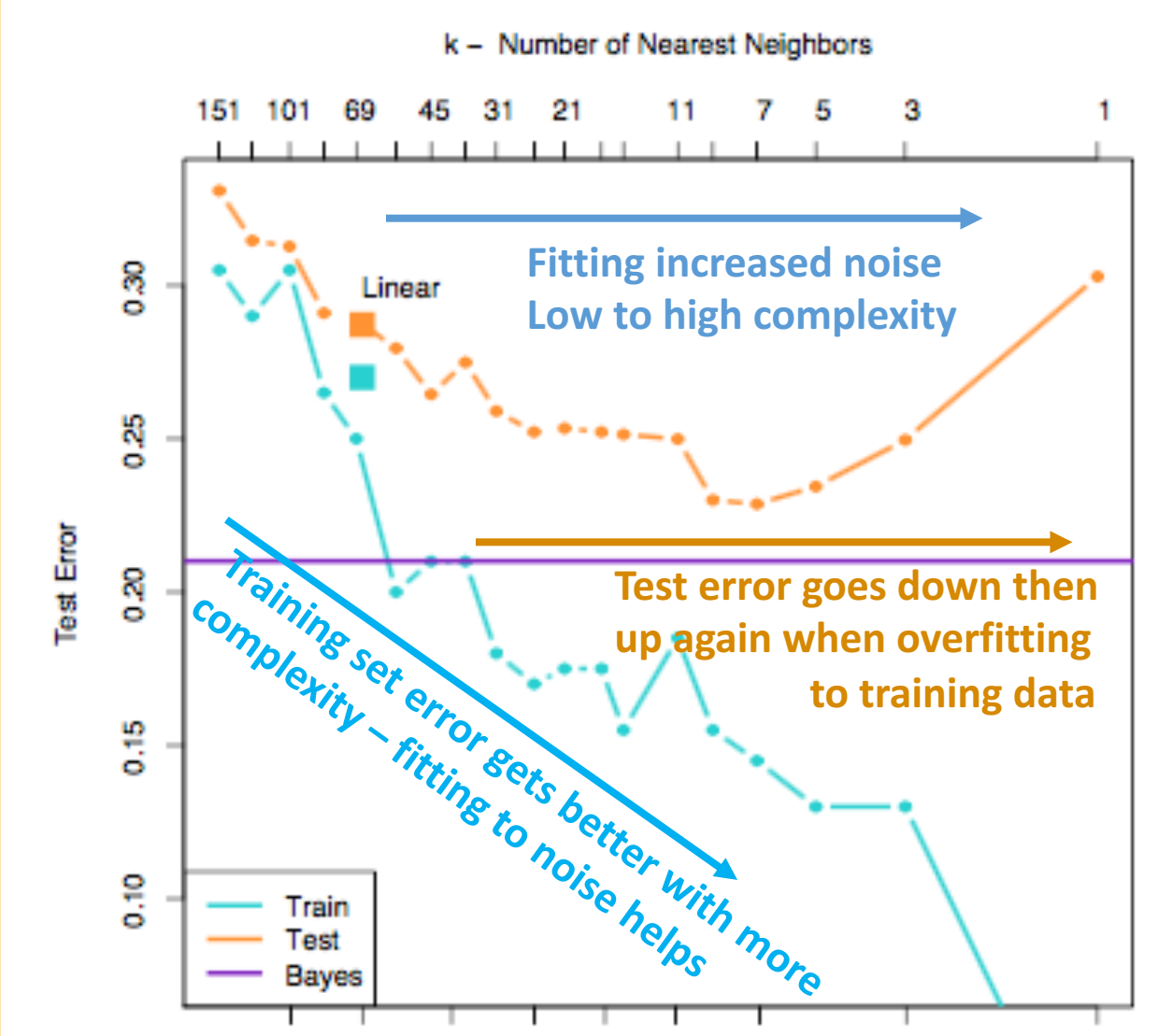
Choice of k


k – Number of Nearest Neighbors
151
101
69
45
31
21
11
7
5
3
1
Linear
Fitting increased noise
Low to high complexity
Test Error
0.30
0.25
0.20
0.15
0.10
Train
Test
Bayes

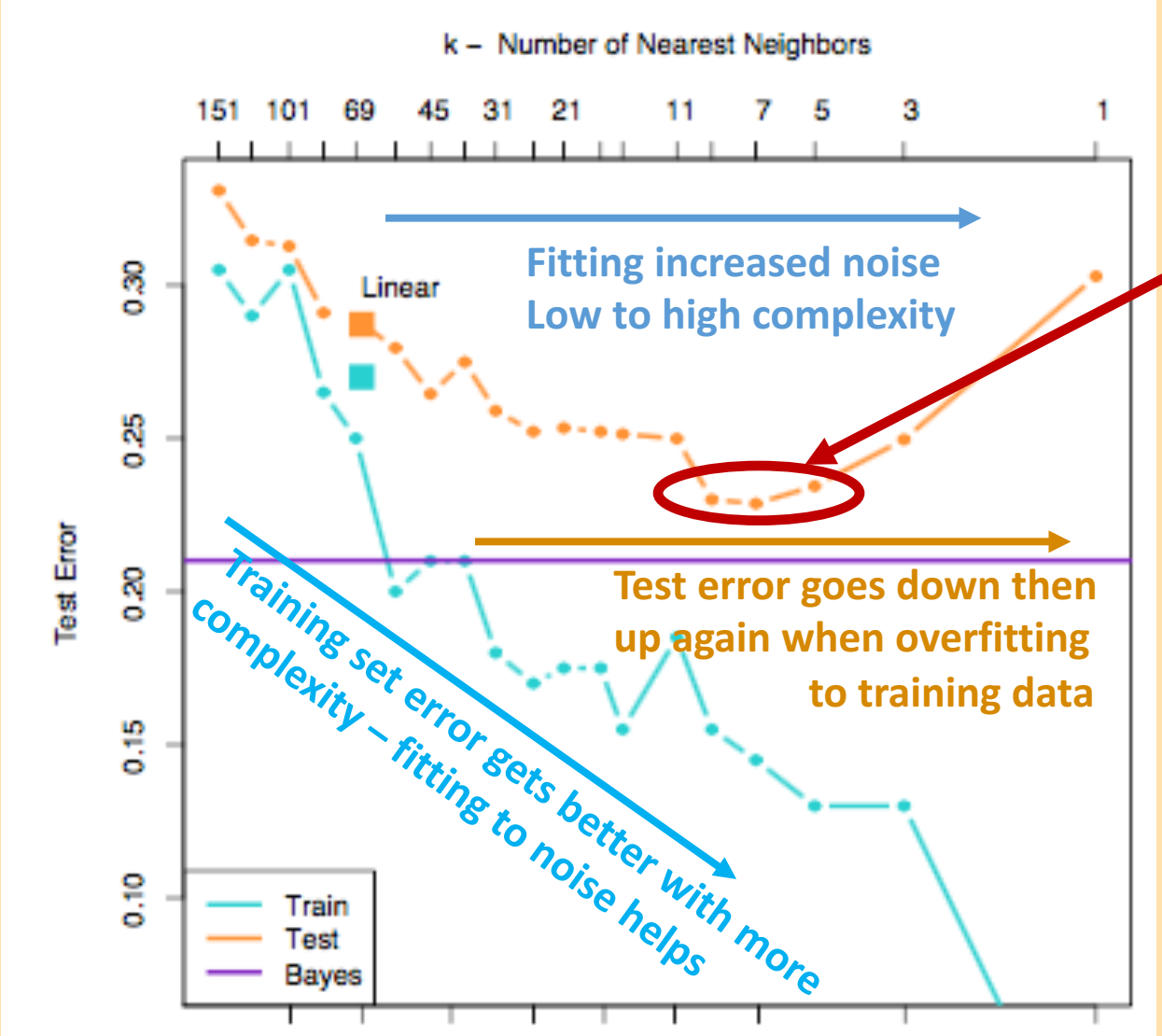
Choice of k


k – Number of Nearest Neighbors
151
101
69
45
31
21
11
7
5
3
1
Test Error
0.30
0.25
0.20
0.15
0.10
Linear
Fitting increased noise
Low to high complexity
Training set error gets better with more complexity – fitting to noise helps
Train
Test
Bayes

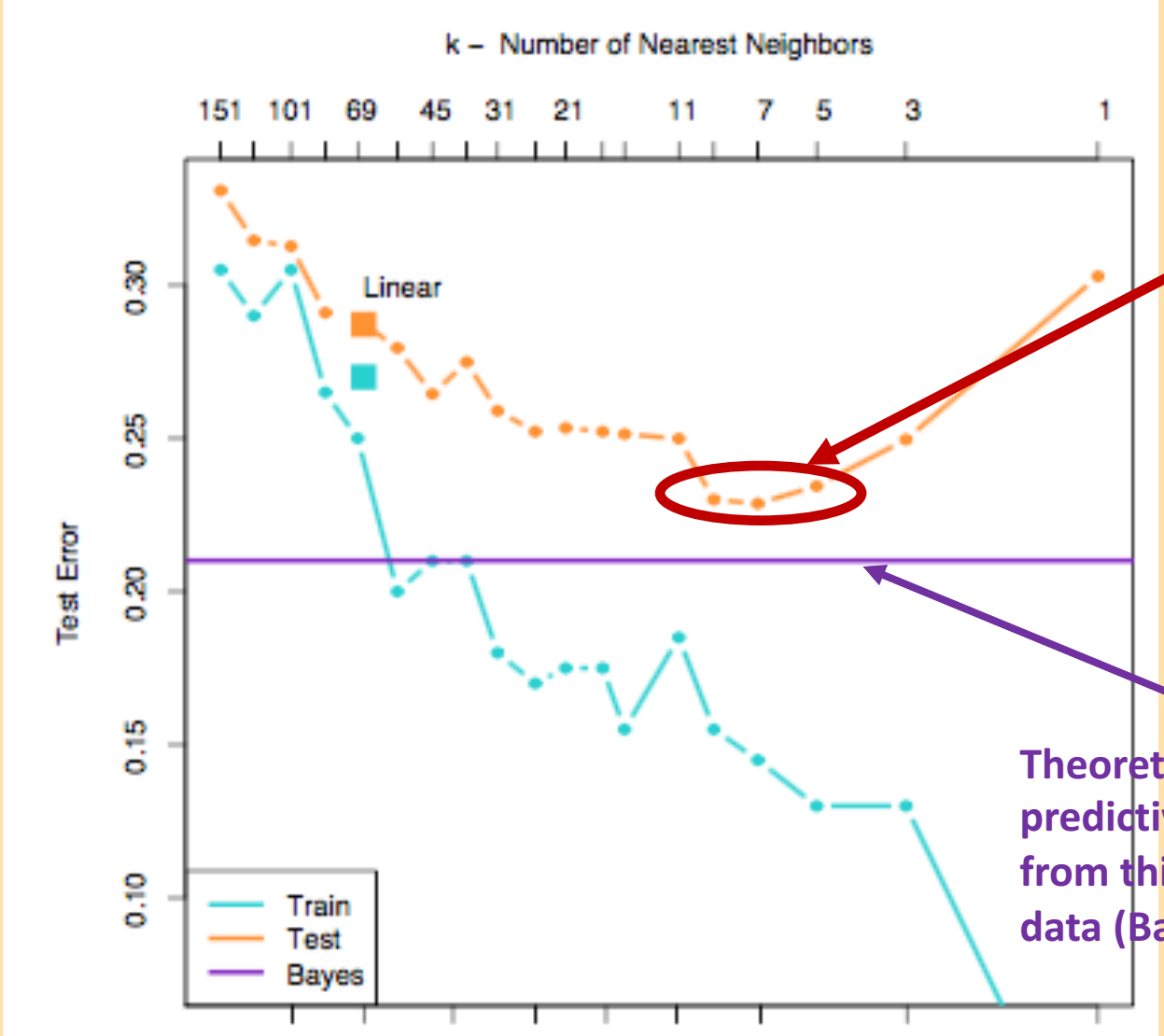
Choice of k


k – Number of Nearest Neighbors
151
101
69
45
31
21
11
7
5
3
1
Test Error
0.30
0.25
0.20
0.15
0.10
Linear
Fitting increased noise
Low to high complexity
Test error goes down then
up again when overfitting
to training data
Training set error gets better with more
complexity – fitting to noise helps
Train
Test
Bayes

Choice of k


k – Number of Nearest Neighbors
151 101 69 45 31 21 11 7 5 3 1
Test Error
0.10 0.15 0.20 0.25 0.30
Linear
Fitting increased noise
Low to high complexity
Best predictive models
Test error goes down then up again when overfitting to training data
Training set error gets better with more complexity – fitting to noise helps
Train
Test
Bayes

Choice of k

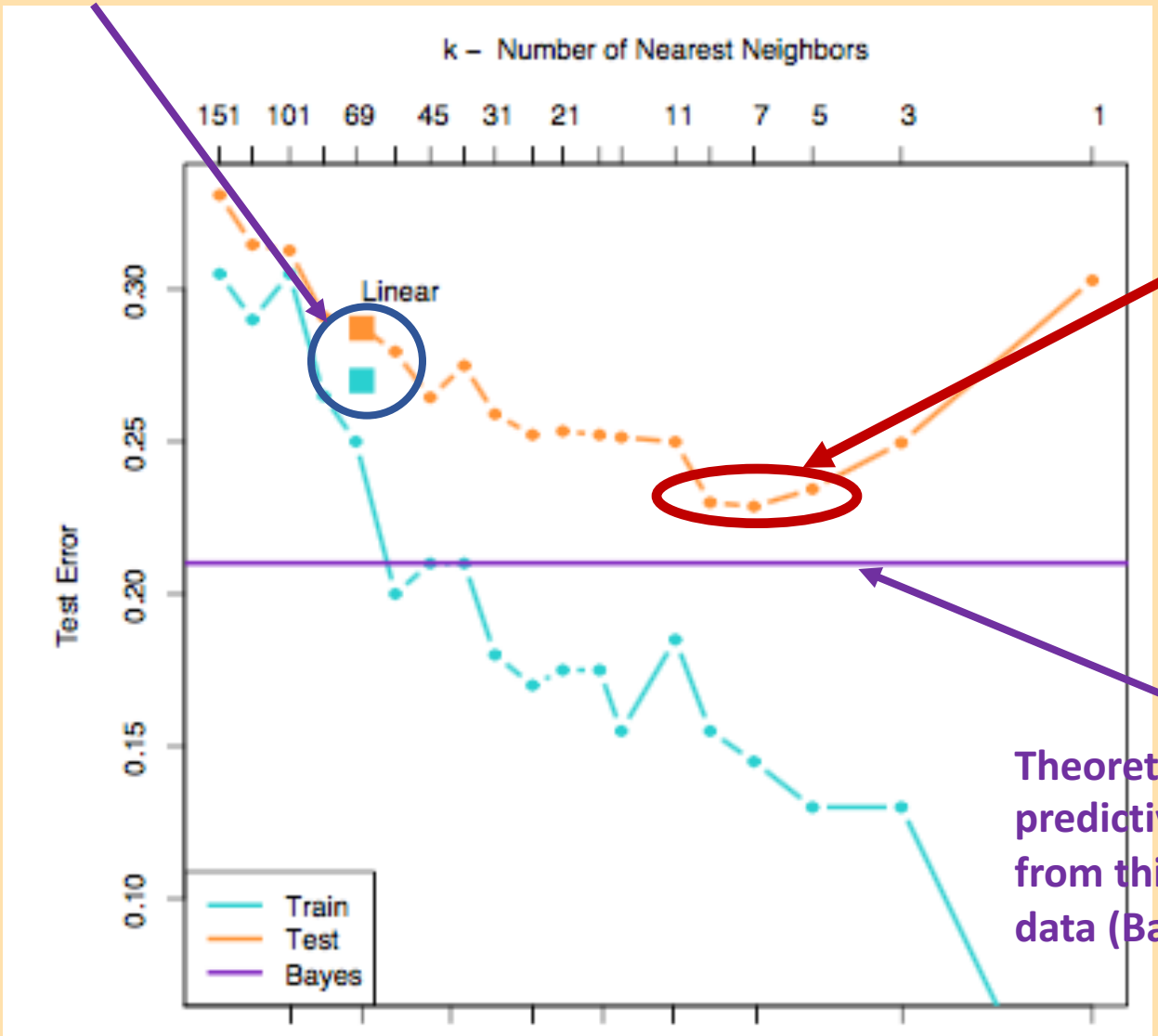


Best predictive models

Theoretically best true predictive error level from this simulated data (Bayes classifier)

Choice of k

Best linear separating line



Best predictive models

Theoretically best true predictive error level from this simulated data (Bayes classifier)

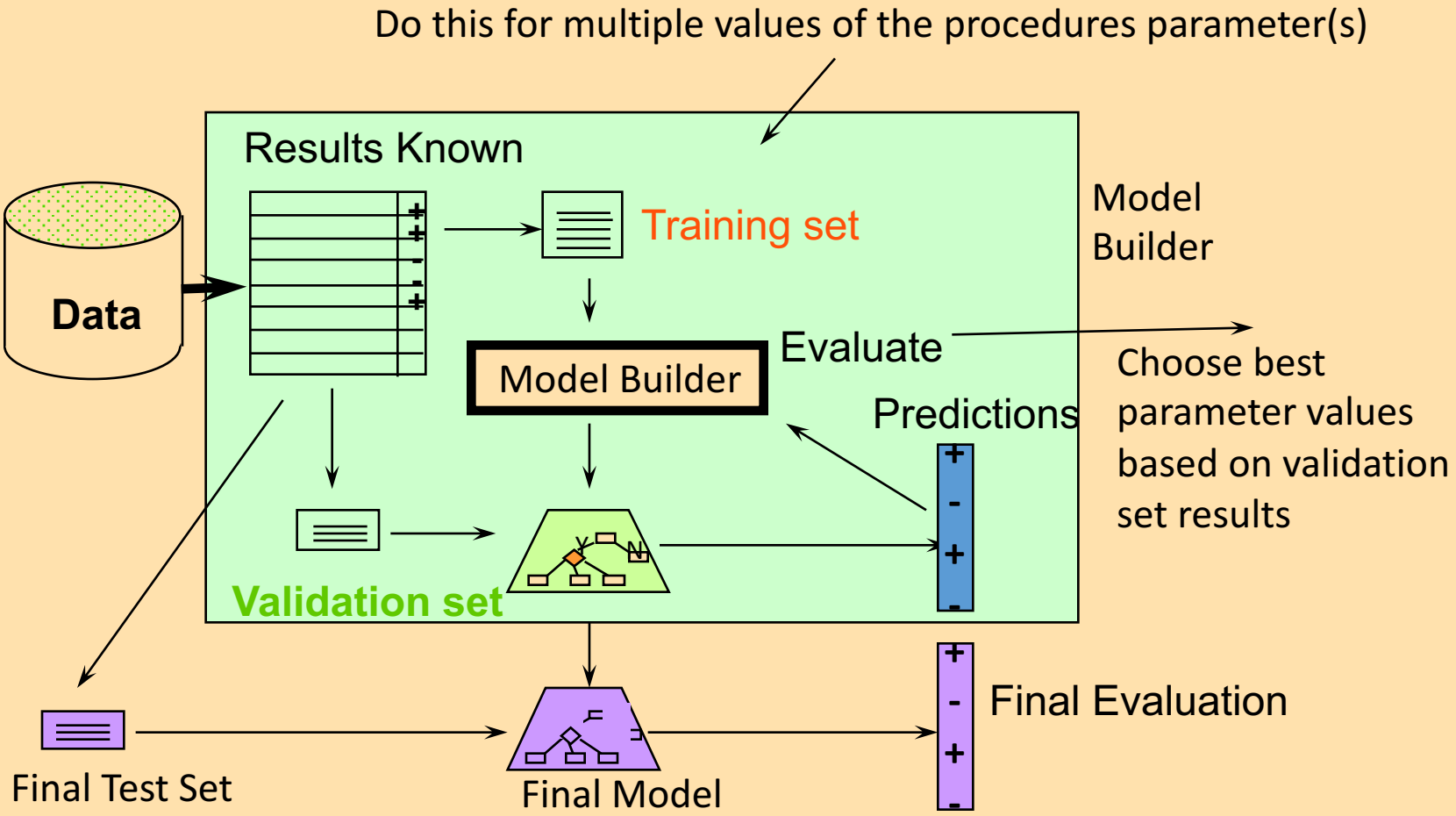
Nearest neighbors classifier

- Simple technique
- No modeling
- Has been applied to many machine learning problems successfully

Parameter estimation

- Many classifiers (and regression approaches) have inbuilt "tuning parameters" (like choosing k for nearest neighbors)
- Some learning schemes operate in two stages:
  - Stage 1: builds the basic structure
  - Stage 2: optimizes parameter settings
- The test data can't be used for parameter tuning!
- Proper procedure uses three sets: **training data, validation data, and test data**
  - Validation data is used to optimize parameters
- Once evaluation is complete, *all the data* can be used to build the final classifier

Train-validate-test



Warning: IBM Modeler goes against convention here:

IBM = training/test/validate
convention = training/validate/test

We will use convention in the slides but be careful to switch test and validate definitions when working in Modeler

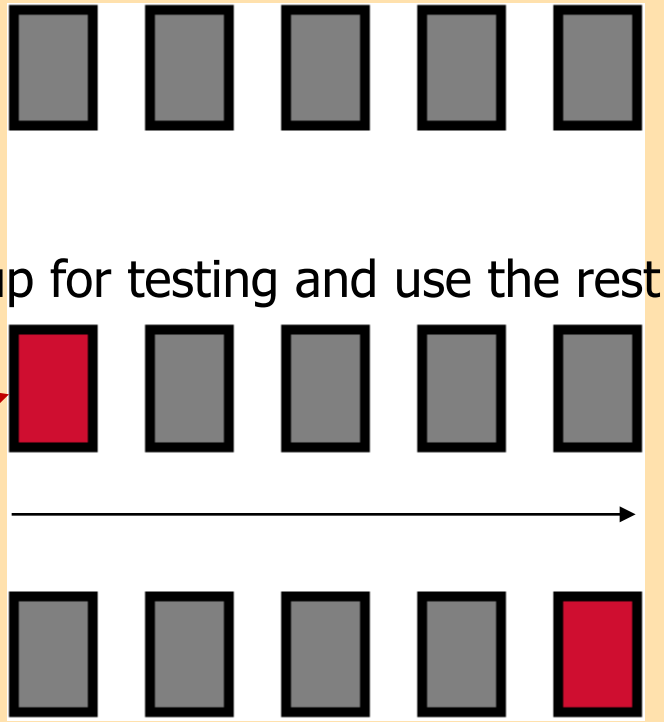
But what if you don’t have enough data to split 2, let alone 3 ways?

Cross-validation

- *Cross-validation* avoids overlapping test sets
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset in turn is used for validation and the remainder for training
- This is called *k-fold cross-validation* (10-fold is common choice, also 5-fold)
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

5-fold cross-validation

Break up data into groups of the same size



Hold aside one group for testing and use the rest to build model

Test set

Repeat

Another Warning:

IBM text book (Wendler and Gröttrup) wrongly considers cross-validation to be a simple *train-test-validate* analysis (no rotation of dataset)

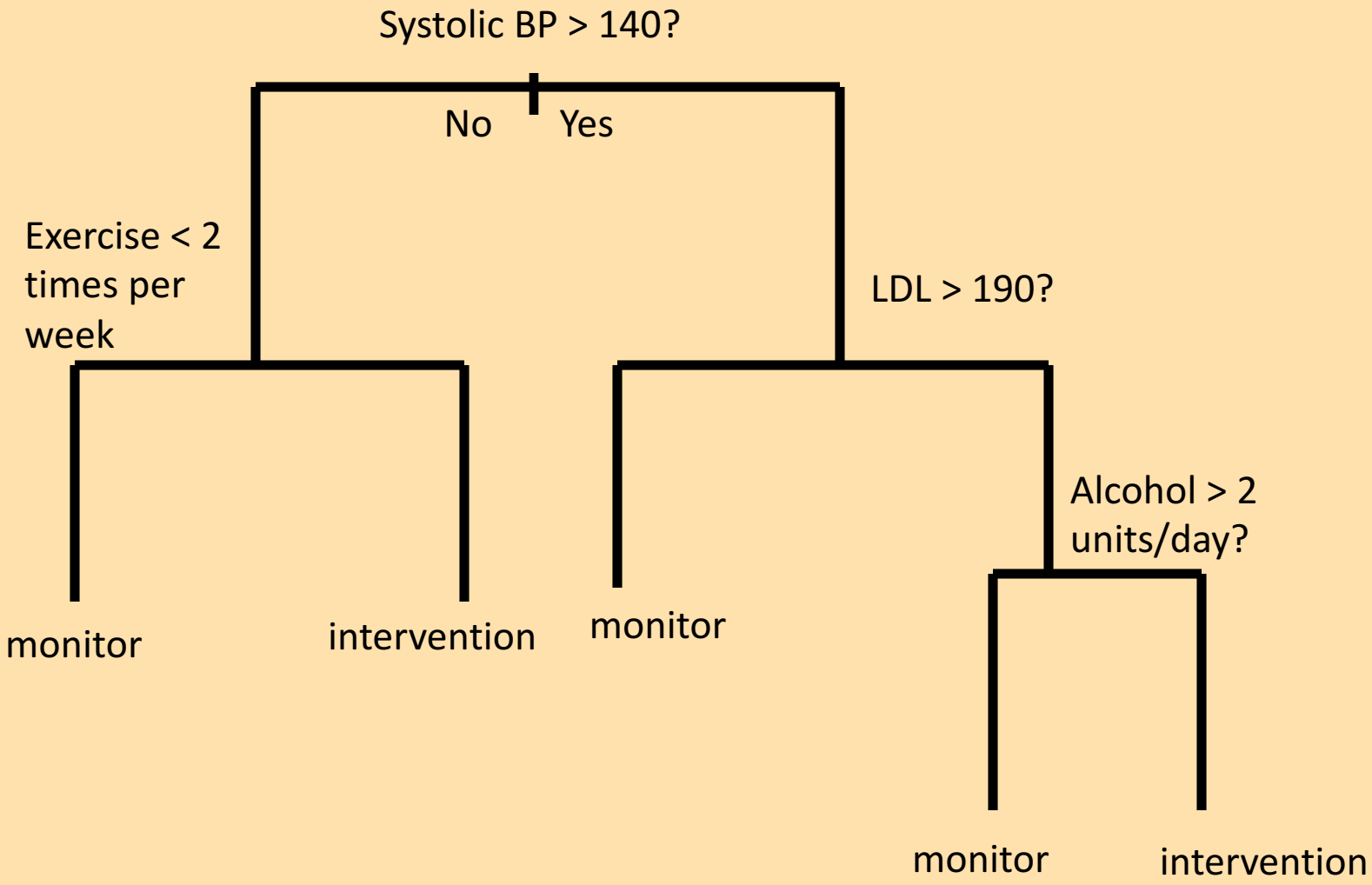
This flies in the face of the definition for cross-validation!

Recursive Partitioning (Decision Trees)

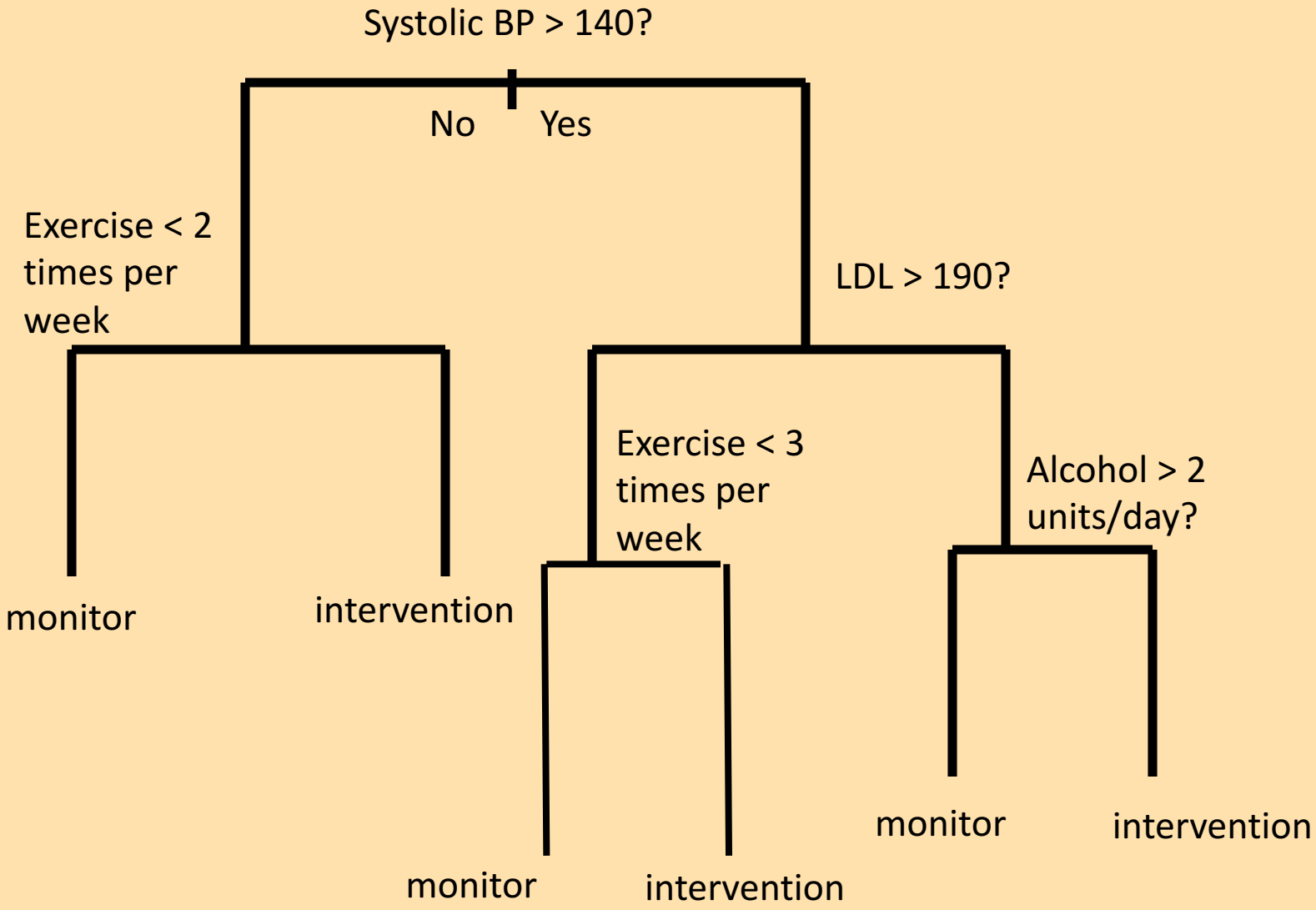
Recursive Partitioning

- Classification based on a series of binary decision cut-points e.g.,
  - Is systolic blood pressure greater that 140?
  - If yes, is LDL Cholesterol greater than 190?
  - If no, exercise > 2 times per week
- At the end of the series of questions a *decision node* is reached (note that the series of questions may not be the same for every subject – it depends which branch of the tree you go down)

Risk of heart problem



Risk of heart problem



Growing Trees

- Greedy algorithmic approach
- Start with first best split on a single variable (based on *Gini Index*)
  - The *Gini Index* is related to classification error, but penalizes errors more as the proportion of observations in a node is further from 0.5 – favors *pure nodes*
  - A *pure node* is a decision node where all of the observed data is of a single class
- Then for each branch look for the best split on a single variable for the observations within that branch
- Keep going untill each branch is pure or has less than e.g. 5 observations

Pruning Trees

- Full tree after growing is too complex
- Goal is to find the sub-tree (contained in the fully grown tree) that minimizes a cost-complexity criterion.
- The cost-complexity criterion is a trade-off between the classification error rate and the complexity (size) of the tree
- The best balance (weighting) for the trade-off is determined via cross-validation to minimize the classification error rate

Recursive Partitioning

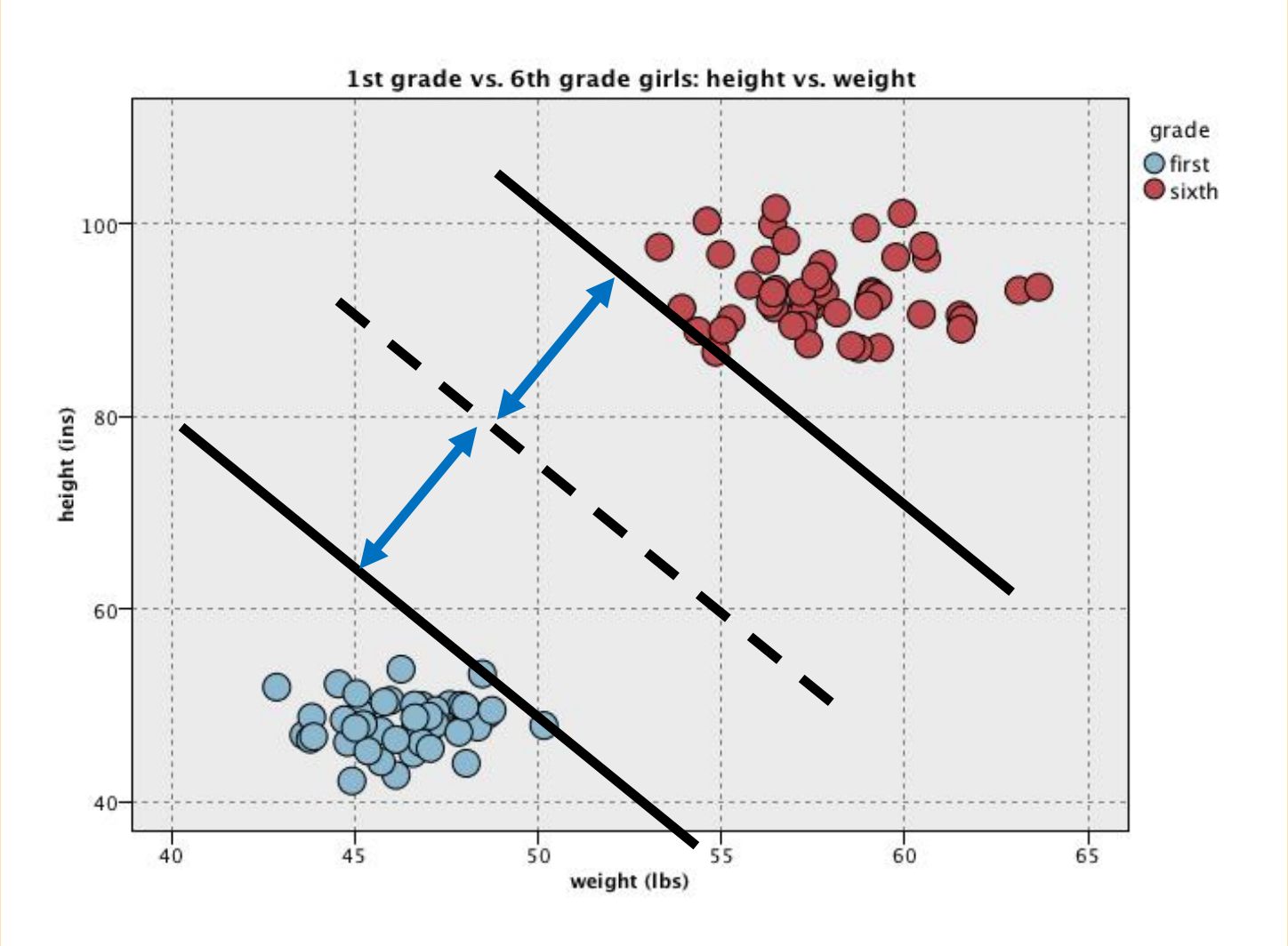
- Recursive partitioning methods are a very popular classification approach and come in many flavors: Classification and Regression Trees (CART), C5.0, Random Forests…
- Partitioning on cut-points approach is very popular in medicine because mimics decision approach of clinicians

Support vector machines

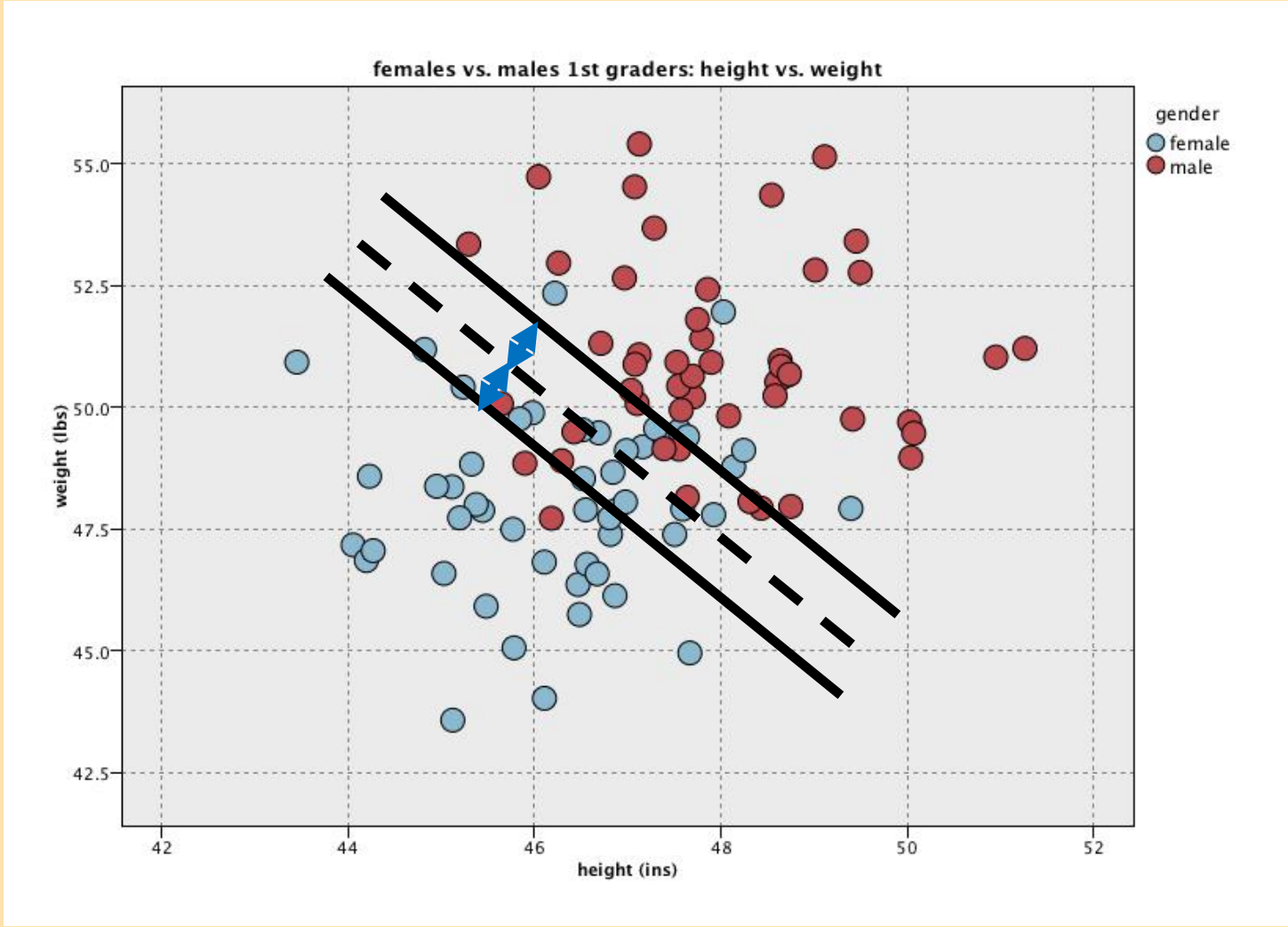
Support vector machines

- Idea is to generate a (non-linear) hyper-plane that optimally separates points from different classes
- Developed in computer science community in 1990s

Maximizing margins

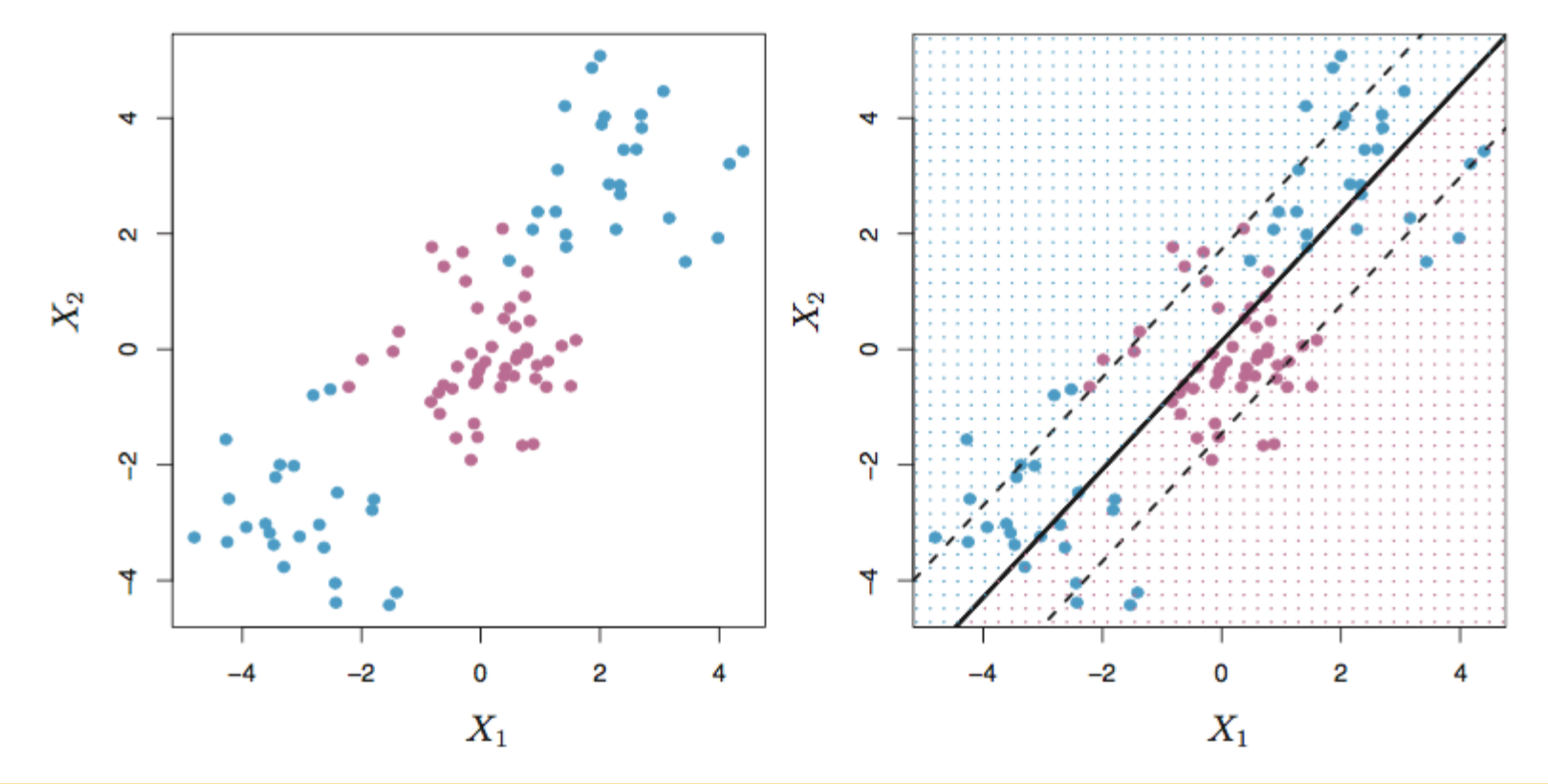


But usually we have overlapping data…

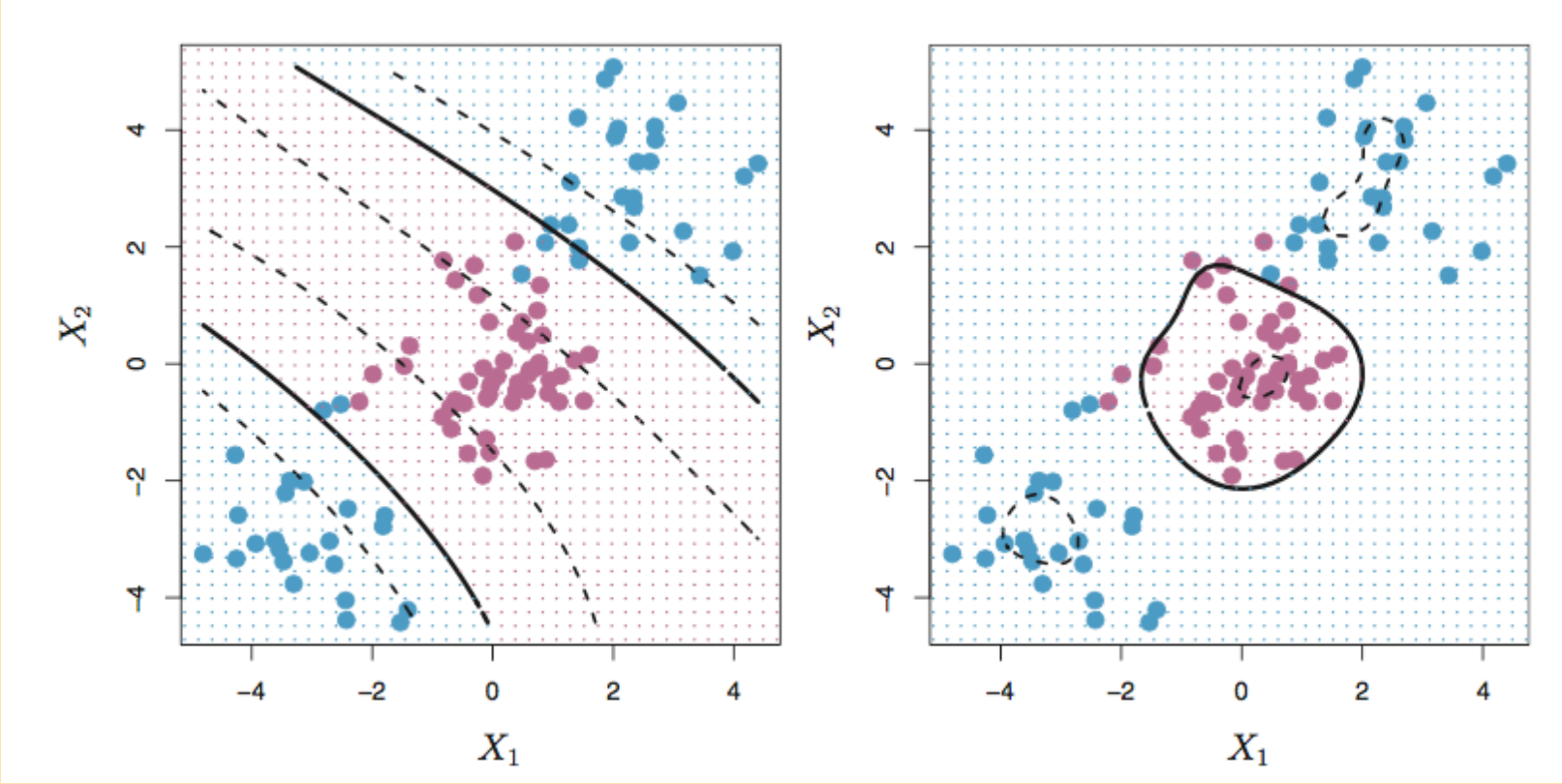


But linearity is limited

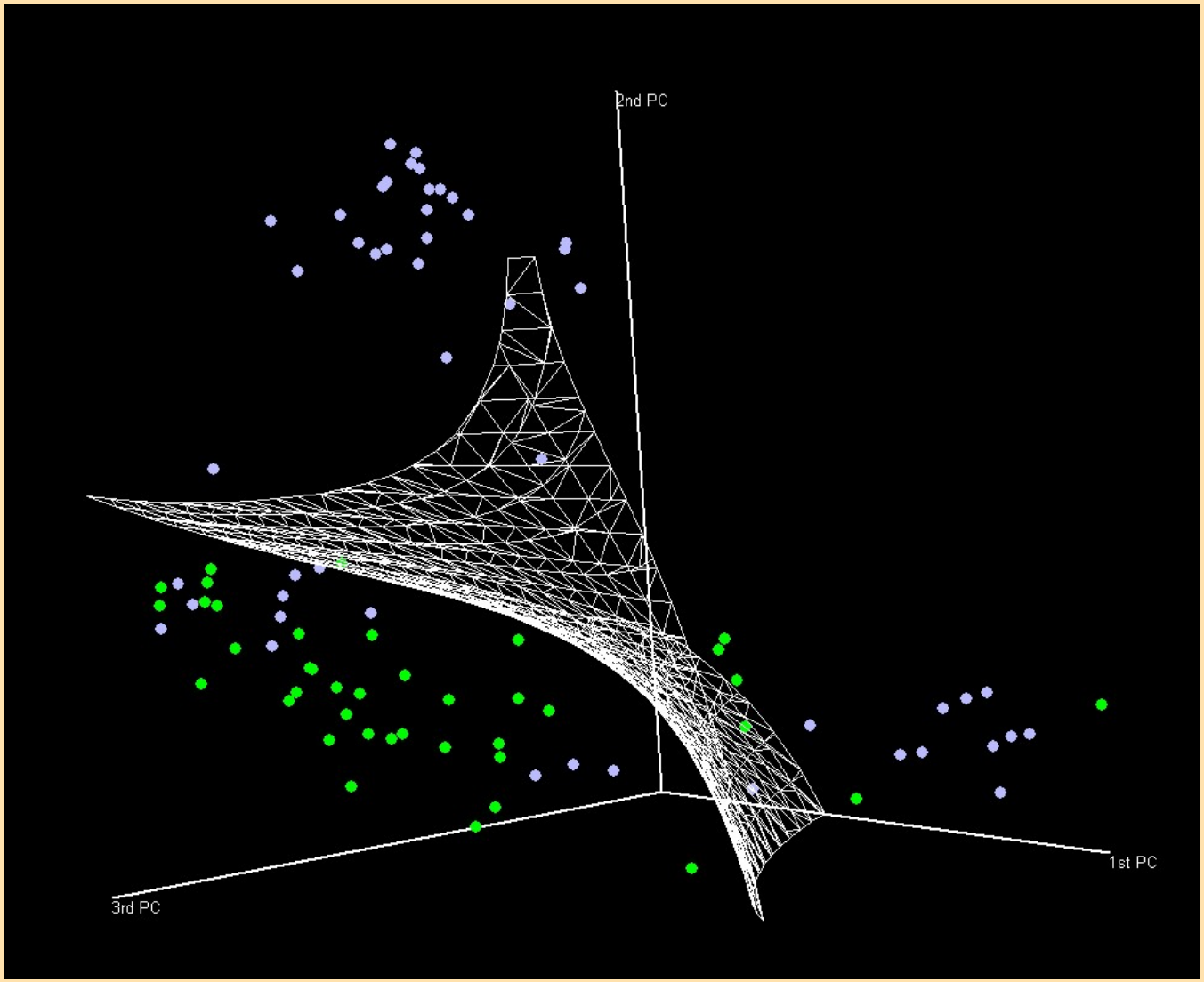
Example from ISLR



Non-linear decision boundary



3 predictor - non-linear decision boundary



Support Vector Machine

- Considered one of the best “out of the box” classifiers
- Has gained massive popularity
- Solved a lot of tough problems

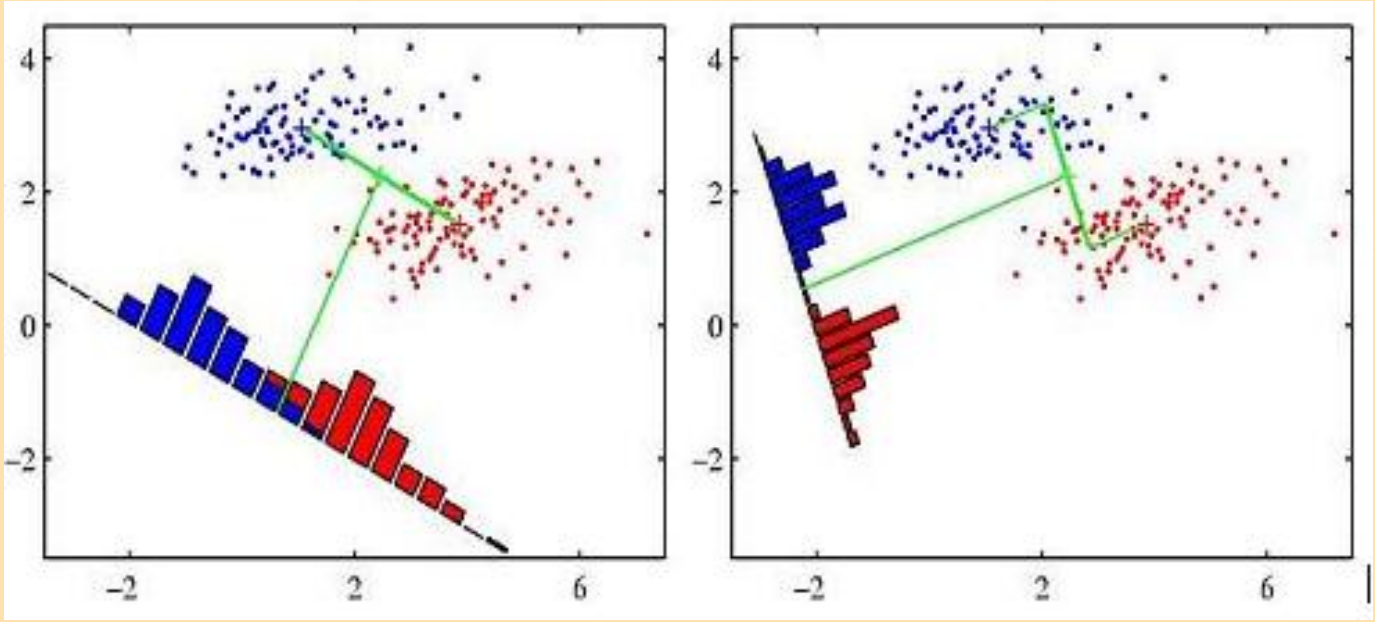
Discriminant Analysis

Linear discriminant analysis (LDA)

- Consider 2 or more classes with correct class known for each subject.
- Objective: find a linear combination

  $a_1 x_1 + a_2 x_2 + \ldots + a_p x_p$ of the variables that maximizes the ratio of the between-class to within-class variance – this is the first linear discriminant function.
- The 2nd linear discriminant finds the best linear combination *orthogonal (perpendicular)* to the first to maximize the same ratio etc.

Discriminant analysis



- Gaussian (normal) probability densities: linear discriminant analysis (LDA) based on assuming within-class covariances are equal
- Provides the probability of belonging to each class, not just a classification.

Discriminant analysis

- Long standing method (goes back to Ronald Fisher – see lab)
- We are more interested in summary patterns of what is important in determining classification
- Can extend to quadratic discriminant analysis (QDA) and more complex non-linear forms

Monday's lecture – clustering and data reduction

- Clustering
  - K-means clustering
  - Kohonen
- Data reduction
  - Principal components analysis (PCA)
- Neural Networks (Neural Nets)
- Bayesian Networks (Bayes Nets) – causal interpretation

Note: HW #4 will be up some time tomorrow
– due 11.55pm on Thursday August 25

Hosted by Institute for Computational Health Sciences
Director, Atul Butte, MD, PhD

Thursday, September 1st
9:00-10:30
Mission Bay—Genentech Hall, Byers Auditorium
Simulcast at Parnassus—Library, CL 221-222

Predicting Adolescent Suicide Risk & Preventing Suicidal Behavior: Integrating Precision Phenotyping & Systems Biology for Personalized Interventions

John Pestian, PhD, MBA

Director, Computational Medicine Center
Professor of Pediatrics, Psychiatry, and Biomedical Informatics
Cincinnati Children's Hospital Medical Center, University of Cincinnati