Dall-E

Go to **tinyurl.com/cs441chat** to ask questions during lecture (CampusWire in-lecture chat room)

K-Nearest Neighbor

Applied Machine Learning
Derek Hoiem

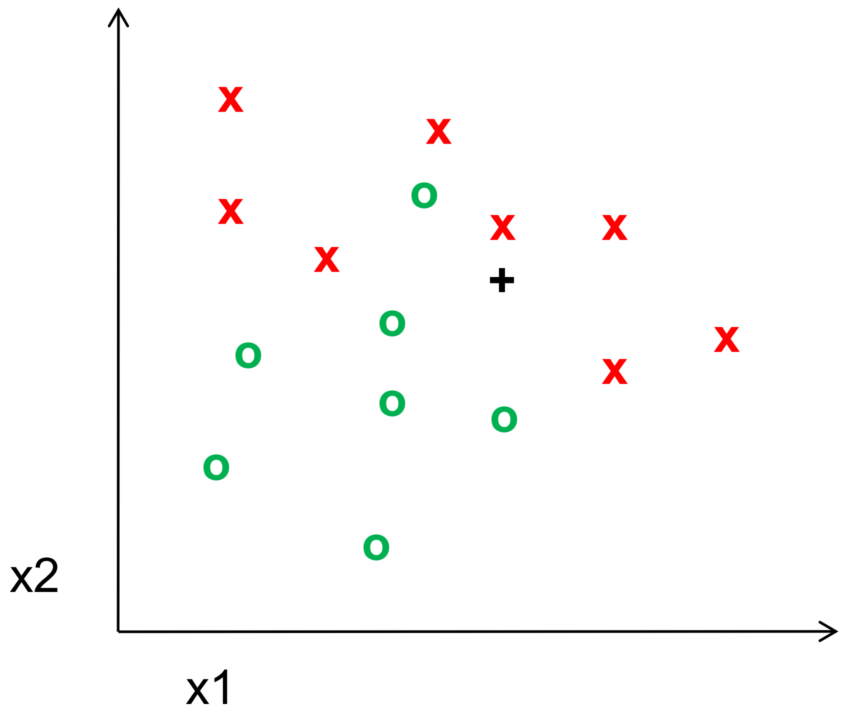
Last Class

- How to compute similarity
- Using similarity to retrieve and cluster

Today’s Lecture

- K-Nearest Neighbor Algorithm
- Example application
  - Deep Face
- Measuring and understanding error

What class do you think the '+' belongs to?



Key principle of machine learning

Given feature/target pairs $(X_1, y_1), \dots, (X_n, y_n)$:

if X_i is similar to X_j, then y_i is probably similar to y_j

With variations on how you define similarity and make predictions based on multiple similar examples, this principle underlies virtually all ML algorithms

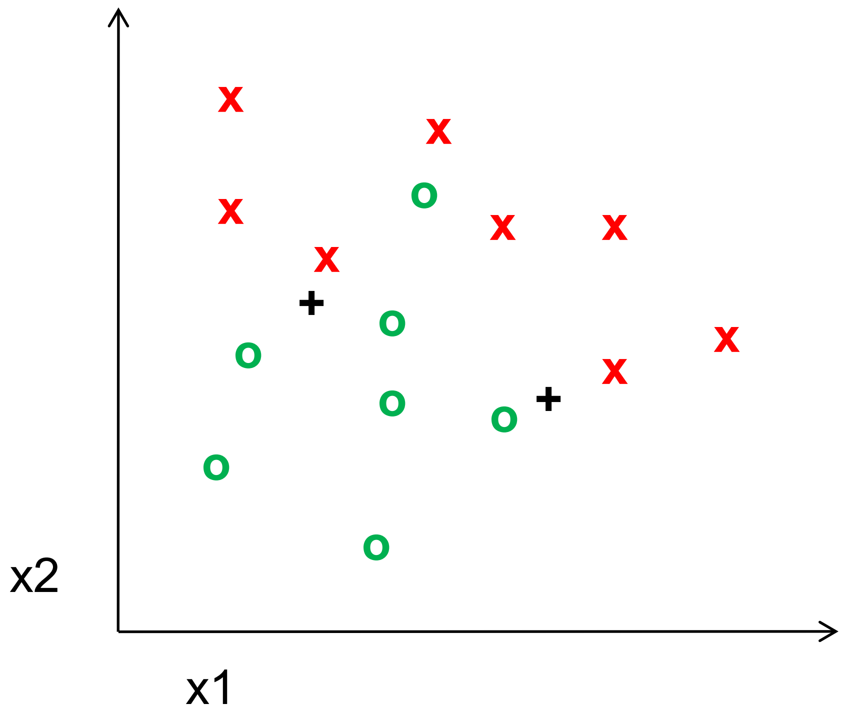
Nearest neighbor algorithm

For given test features, assign the label / target value of the most similar training features

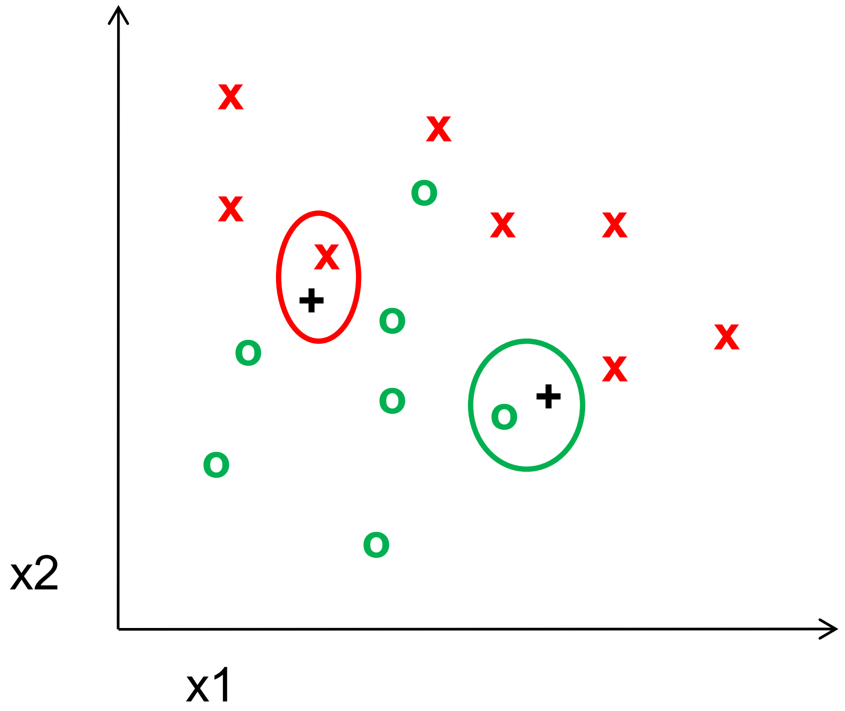
1. $i^* = \operatorname{argmin}_i \, distance(X_{train}[i], X_{test})$
2. $y_{test} = y_{train}[i^*]$

Distance function is up to designer. Simplest is L2 distance.

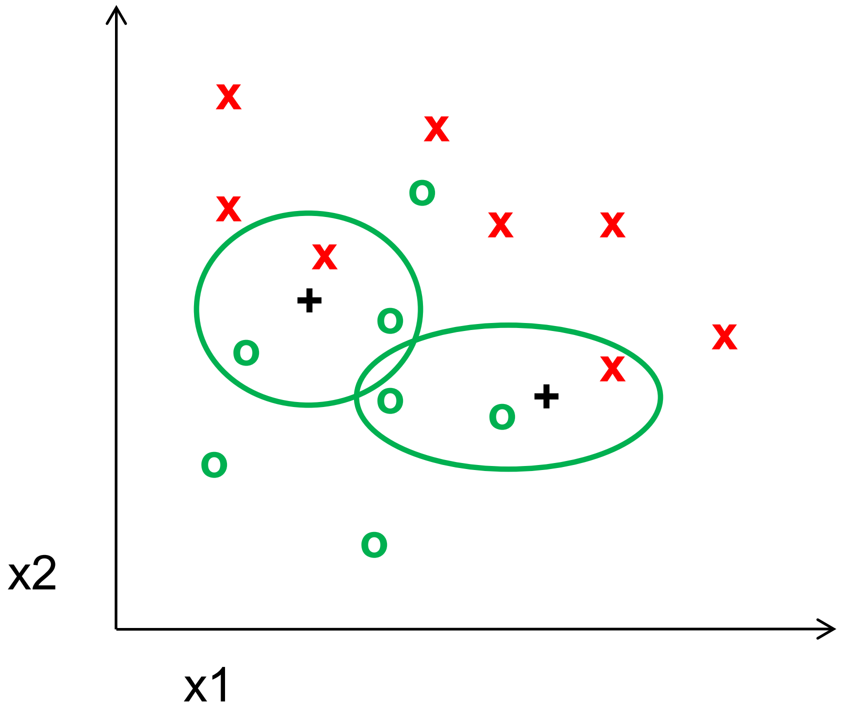
K-nearest neighbor: predict based on K closest training samples



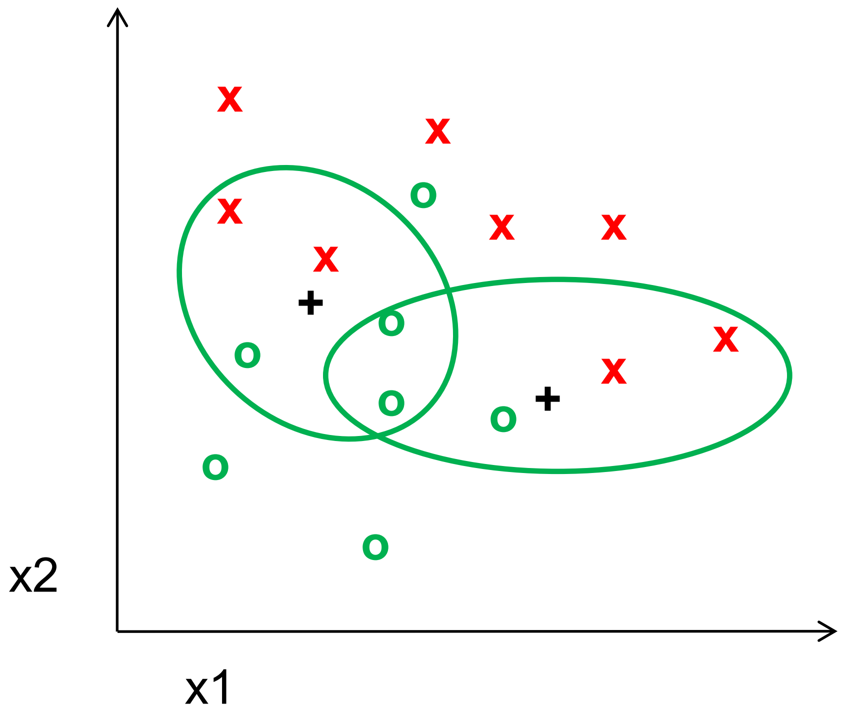
1-nearest neighbor



3-nearest neighbor



5-nearest neighbor



KNN Distance Function

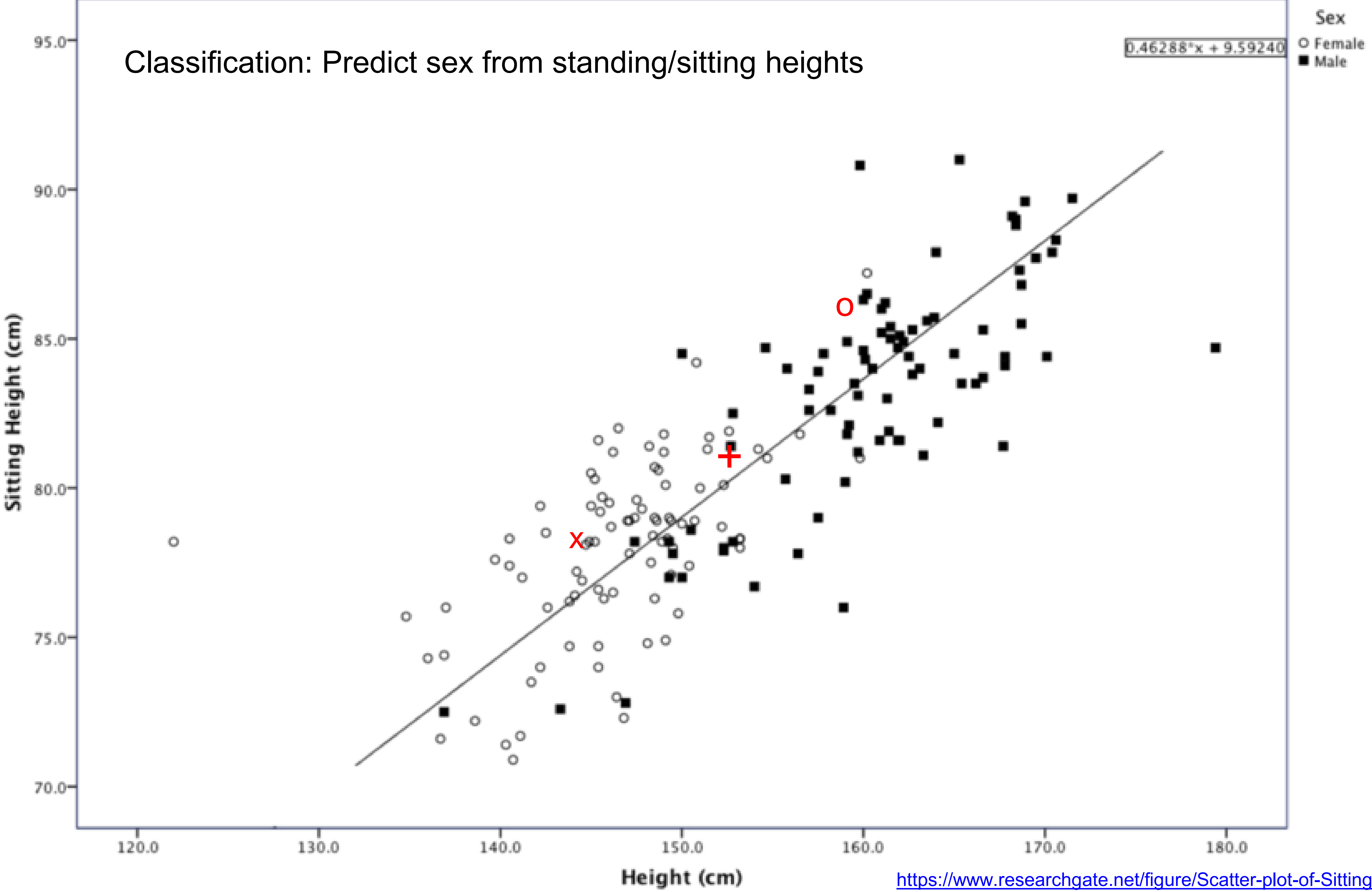
- Euclidean or L2 norm: $\|\boldsymbol{x}-\boldsymbol{t}\|_2 = \sqrt{\sum_k (x_k - t_k)^2}$
  - Assumes all dimensions are equally scaled
  - Dominated by biggest differences
- Citi Block or L1 norm: $\|\boldsymbol{x}-\boldsymbol{t}\|_1 = \sum_k |x_k - t_k|$
  - Assumes all dimensions are equally scaled
  - Less sensitive to very large differences along one dimension
- Mahalanobis distance: $d_M(\boldsymbol{x},\boldsymbol{t}) = \sqrt{(\boldsymbol{x}-\boldsymbol{t})^T \sum^{-1} (\boldsymbol{x}-\boldsymbol{t})}$
  - Normalized by inverse feature covariance matrix: “whitening”
  - When diagonal covariance is assumed, this is equivalent to scaling each dimension by $1/\sigma_k$

$\boldsymbol{x}$ and $\boldsymbol{t}$ are training and test sample feature vectors

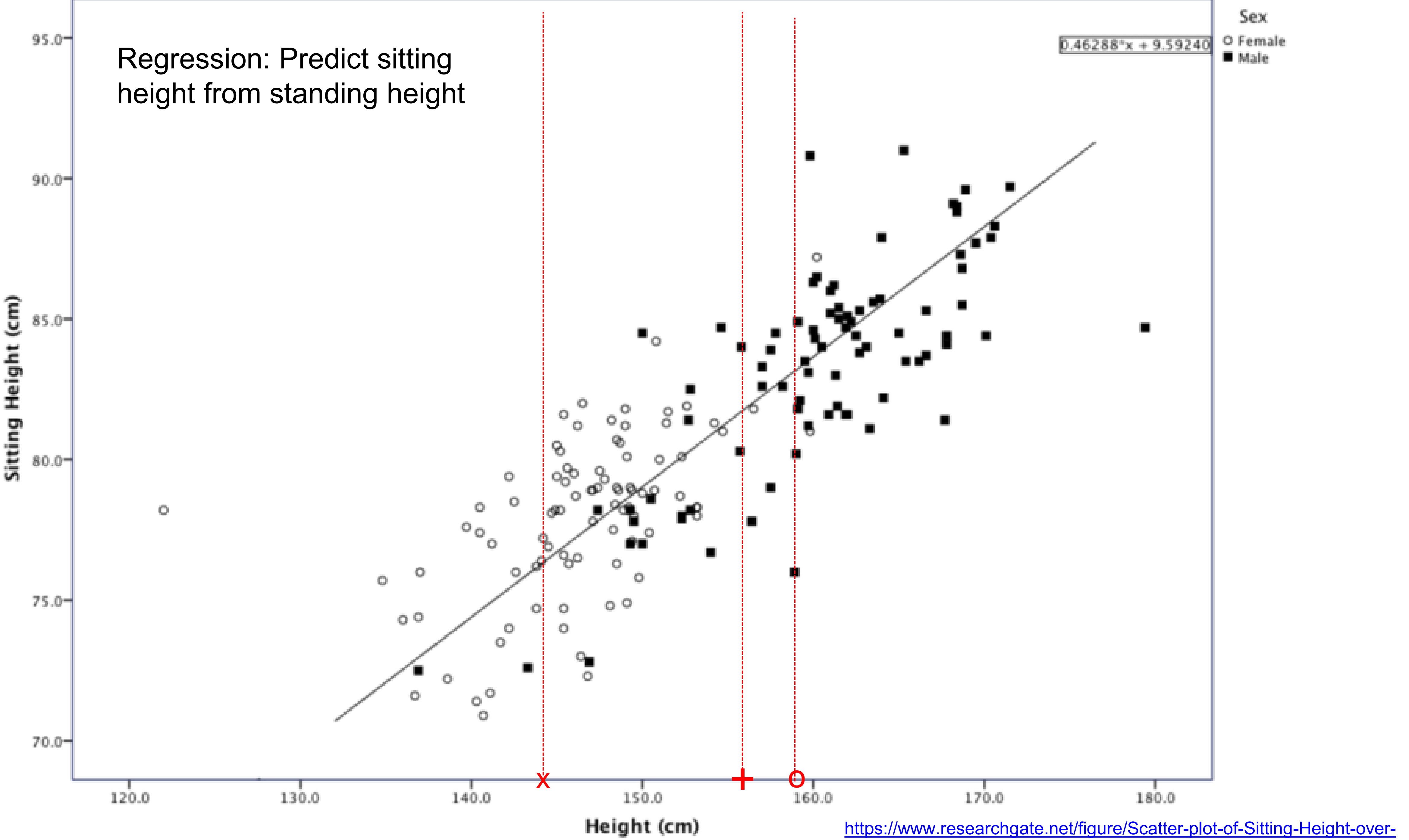
KNN Classification vs Regression

- For classification, prediction is usually the mode or most common class of the returned labels
- For regression, prediction is usually the arithmetic mean (average, informally) of the returned values

Classification: Predict sex from standing/sitting heights



https://www.researchgate.net/figure/Scatter-plot-of-Sitting-Height-over-Height-for-males-and-females_fig3_301724988



https://www.researchgate.net/figure/Scatter-plot-of-Sitting-Height-over-Height-for-males-and-females_fig3_301724988

KNN Classification Demo

- http://vision.stanford.edu/teaching/cs231n-demos/knn/

Comments on K-NN

- Simple: an excellent baseline and sometimes hard to beat
  - Naturally scales with data: it may be the only choice when you have one example per class, and is still often achieves good performance when you have many
  - Higher K gives smoother functions
- Can be slow… but there are tricks to speed it up, e.g.
  - can be precomputed

    $\underset{i}{\operatorname{argmin}} \|x_i - x_t\|_2 = \underset{i}{\operatorname{argmin}} \left(x_i^T x_i - 2x_i x_t + x_t^T x_t\right) = \underset{i}{\operatorname{argmin}} \left(x_i^T x_i - 2x_i x_t\right)$
  - FAISS
  - Approximate search like FLANN or LSH
- No training time (unless you learn a distance function)
- With infinite examples, 1-NN provably has error that is at most twice Bayes optimal error (but we never have infinite examples)

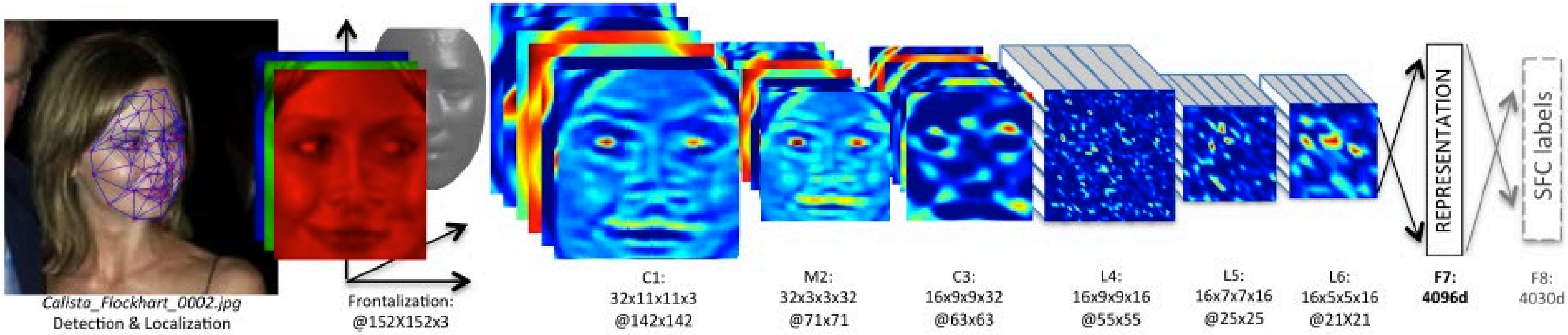
KNN Usage Example: Deep Face

DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Yaniv Taigman Ming Yang Marc'Aurelio Ranzato
Facebook AI Research
Menlo Park, CA, USA
{yaniv, mingyang, ranzato}@fb.com

Lior Wolf
Tel Aviv University
Tel Aviv, Israel
wolf@cs.tau.ac.il

CVPR 2014



1. Detect facial features
2. Align faces to be frontal
3. Extract features using deep network while training classifier to label image into person (dataset based on employee faces)
4. In testing, extract features from deep network and use nearest neighbor classifier to assign identity

- Performs similarly to humans in the LFW dataset (labeled faces in the wild)
- Can be used to organize photo albums, identifying celebrities, or alert user when someone posts an image of them
- If this is used in a commercial deployment, what might be some unintended consequences?
- This algorithm is used by Facebook (though with expanded training data)

KNN Summary

- Key Assumptions
  - Samples with similar input features will have similar output predictions
  - Depending on distance measure, may assume all dimensions are equally important
- Model Parameters
  - Features and predictions of the training set
- Designs
  - K (number of nearest neighbors to use for prediction)
  - How to combine multiple predictions if $K > 1$
  - Feature design (selection, transformations)
  - Distance function (e.g. L2, L1, Mahalanobis)
- When to Use
  - Few examples per class, many classes
  - Features are all roughly equally important
  - Training data available for prediction changes frequently
  - Can be applied to classification or regression, with discrete or continuous features
  - Most powerful when combined with feature learning
- When Not to Use
  - Many examples are available per class (feature learning with linear classifier may be better)
  - Limited storage (cannot store many training examples)
  - Limited computation (linear model may be faster to evaluate)

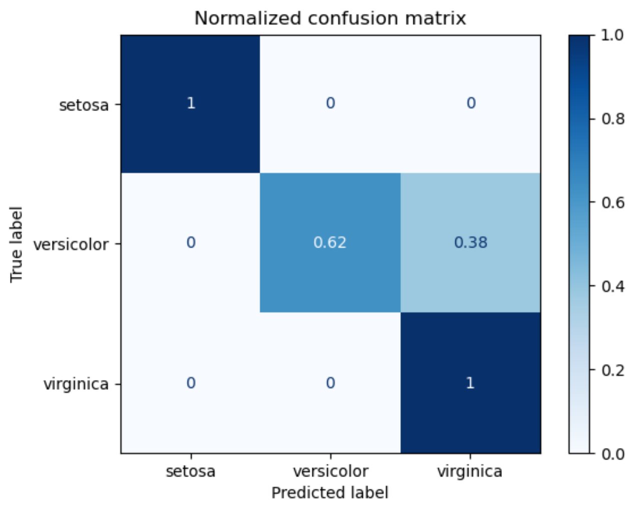
3 minute break

T/F (and why): 1-NN will never have higher training error than 3-NN for classification and regression.

T/F (and why): For 1-NN classification, you cannot remove a training sample without affecting at least some portion of the decision boundary.

How do we measure and analyze classification error?

- Classification error: $\frac{1}{\mathrm{N}} \sum_i f(X_i) \neq y_i$
  - Percent of examples that are wrongly predicted
- Confusion matrix: joint/conditional distribution of predicted and true labels
  - Can be a count or probability
  - Practice varies whether “Predict” or “True” is the y-axis. Need to label.



https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Measuring error example

True	Predicted
Y	Y
N	Y
Y	Y
Y	N
N	N
N	Y
N	N

Classification Error:

Confusion Matrix:

Count

		Predicted	
		N	Y
True	N		
	Y		

P(predicted | true)

		Predicted	
		N	Y
True	N		
	Y		

How do we measure and analyze regression error?

- Root mean squared error $\sqrt{\frac{1}{N}\sum_i (f(X_i) - y_i)^2}$

- Mean absolute error $\frac{1}{N}\sum_i |f(X_i) - y_i|$

- R^2: $1 - \frac{\sum_i (f(X_i) - y_i)^2}{\sum_i (y_i - \bar{y})^2}$ (unexplained variance) / (total variance)

- RMSE/MAE are unit-dependent measures of accuracy, while R^2 is a unitless measure of the fraction of explained variance

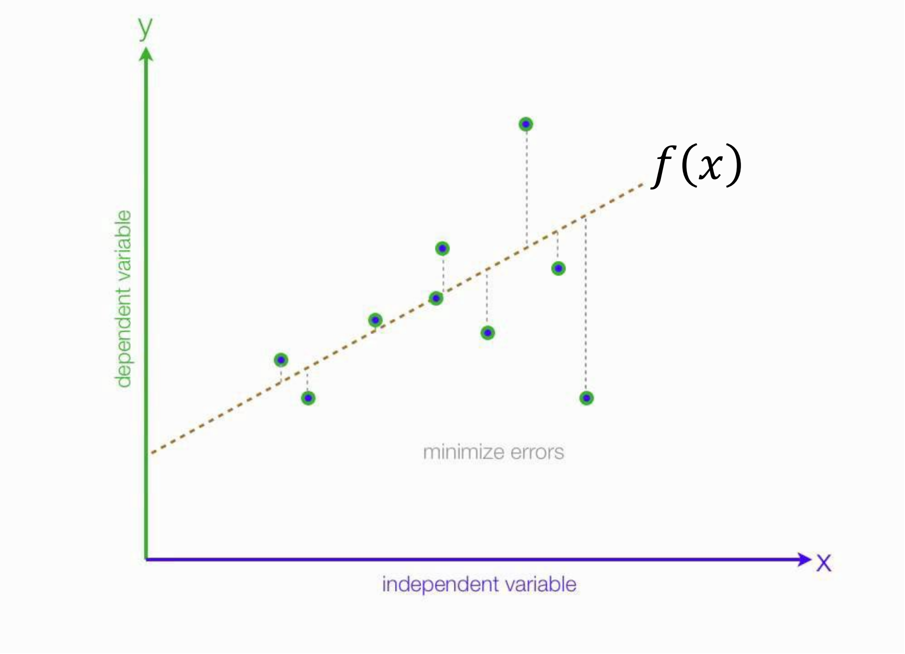


Fig: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

Error and Bias Variance Trade-off

When model parameters are fit to a *training set* and evaluated on a *test set*

- **Training error**: The error on the training set
- **Test error**: The error on the test set
- **Generalization error**: test error – training error

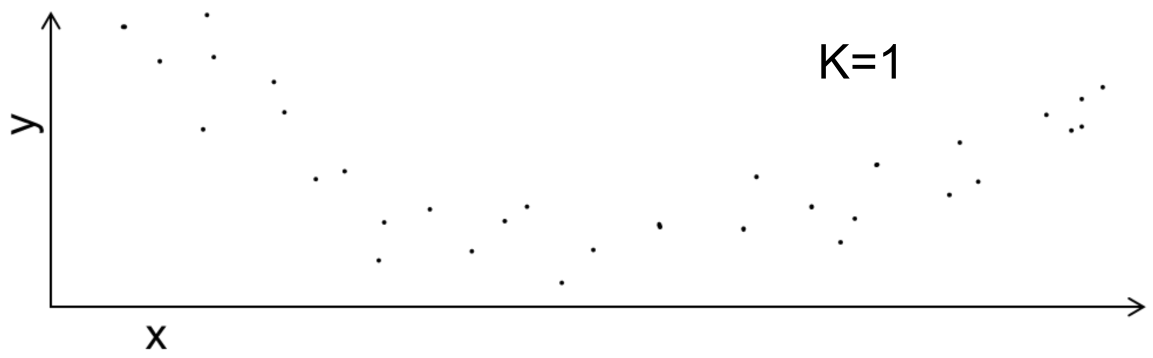
Test error has three important sources in common ML settings:

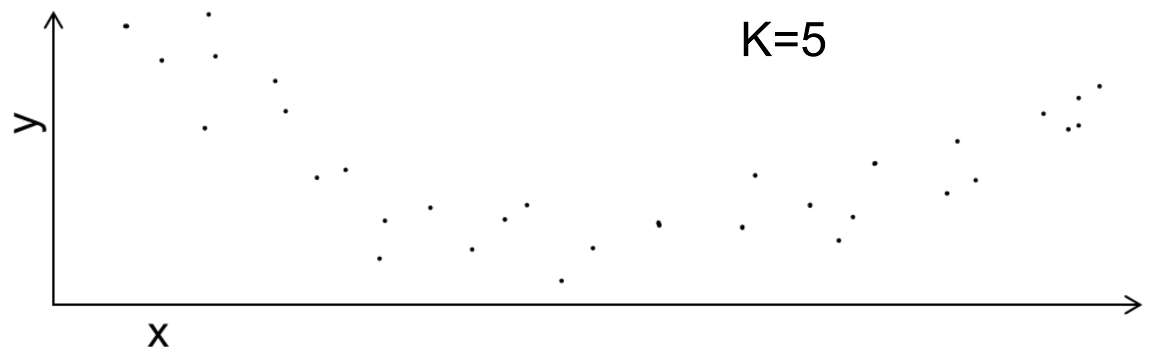
- **Intrinsic**: sometimes it is not possible to achieve zero error given available features (e.g. handwriting, weather prediction)
  - Bayes optimal error: The error if the true function P(y|x) is known
- **Model Bias**: the model is limited so that it can't fit perfectly to the true data distribution (e.g. there will be error, even if you have infinite training data)
- **Model Variance**: given finite training data, different parameters and predictions would result from different samplings of data

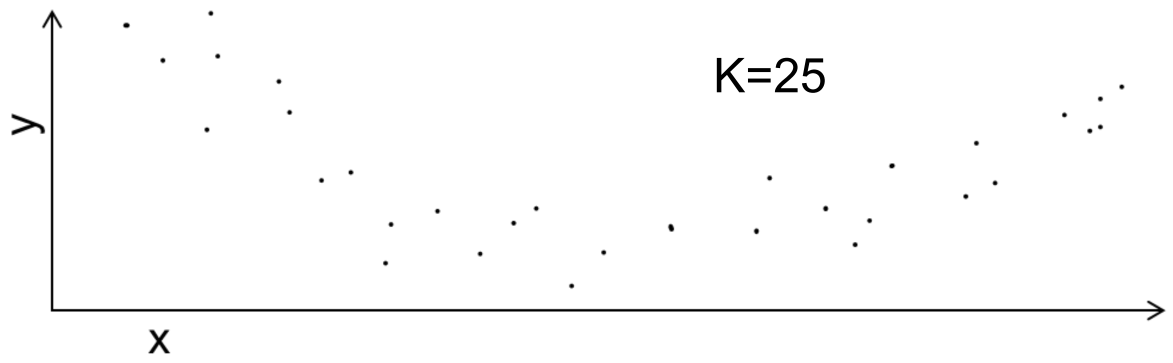
A more complex or specific model will have

- Lower bias: better fit to training set
- Higher variance: more uncertainty in best parameters, so more generalization error

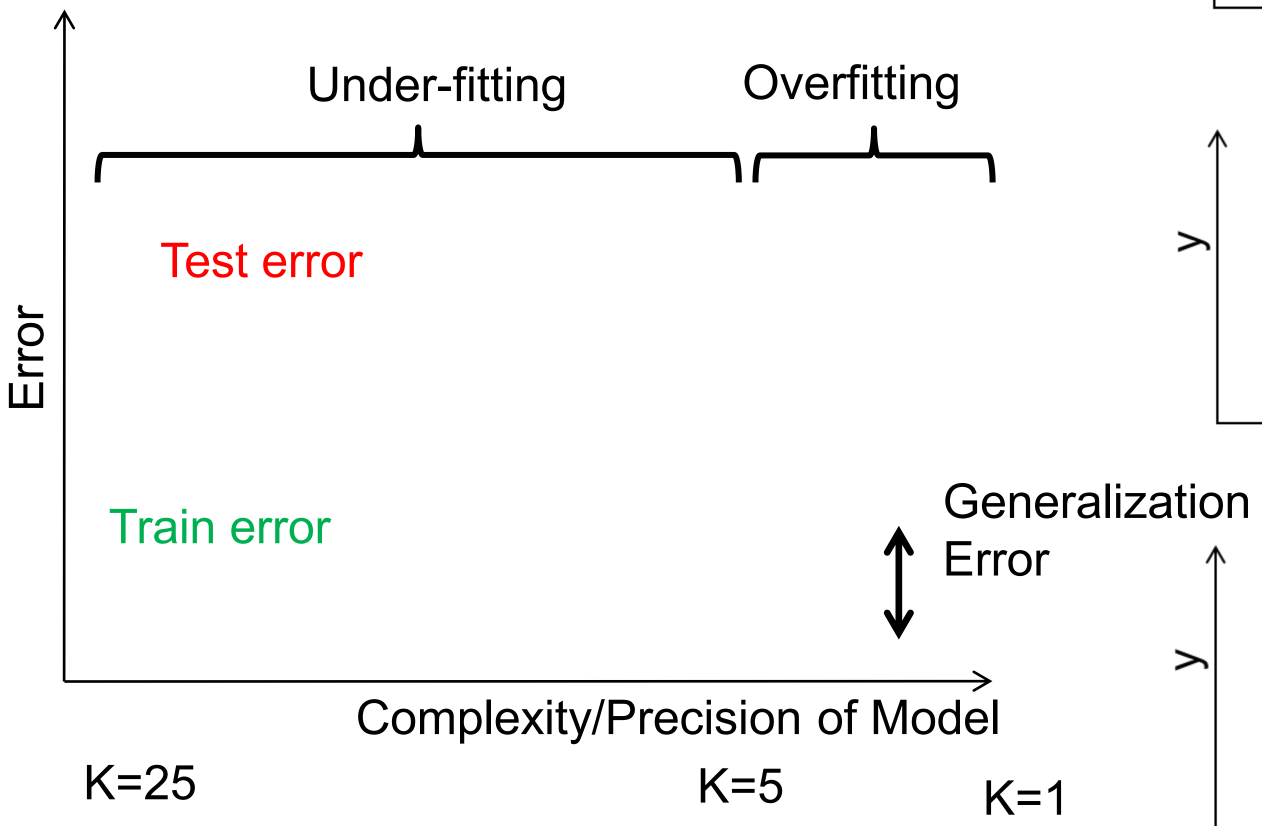
Error and Bias Variance Trade-off

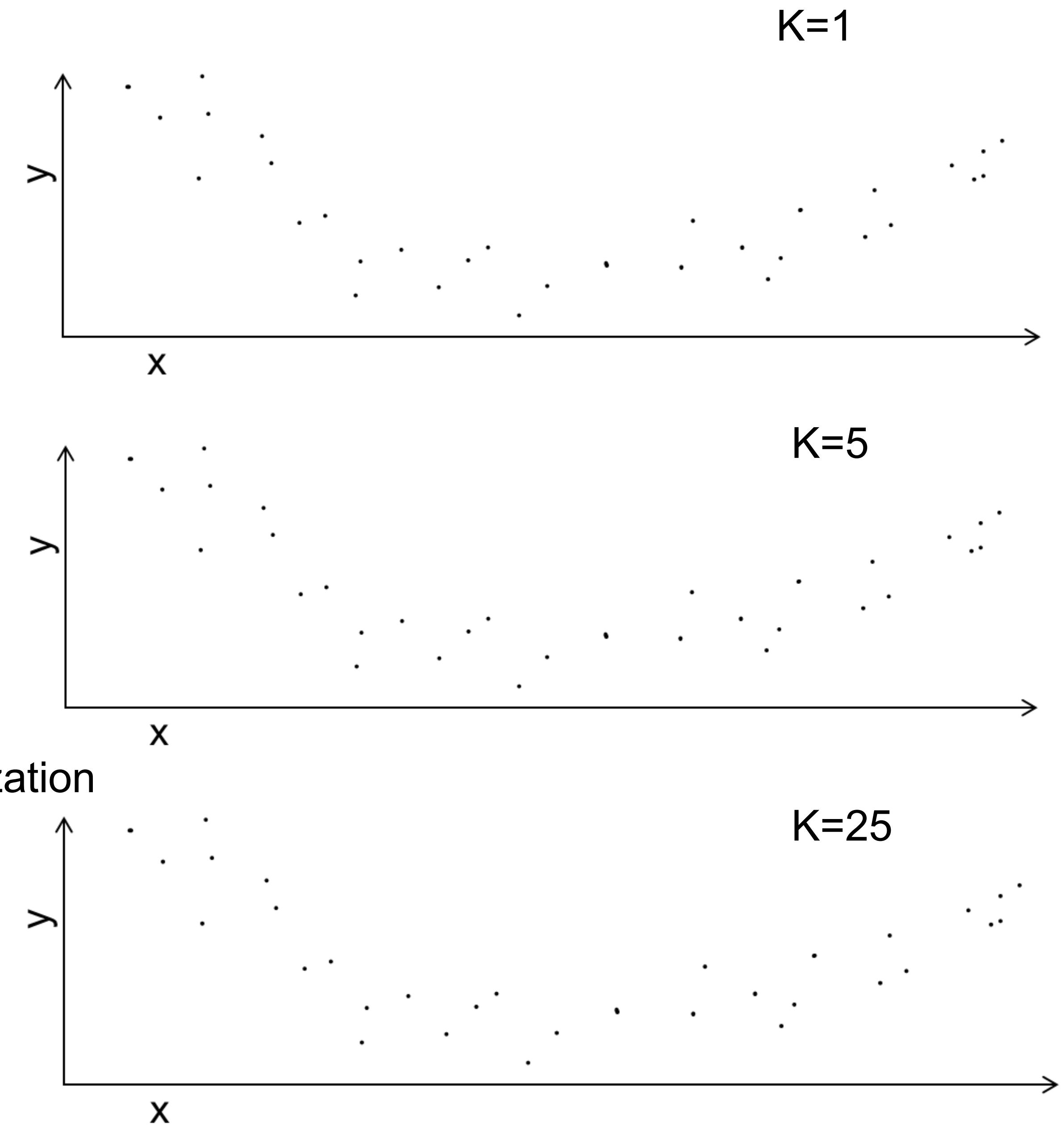






Error and Bias Variance Trade-off





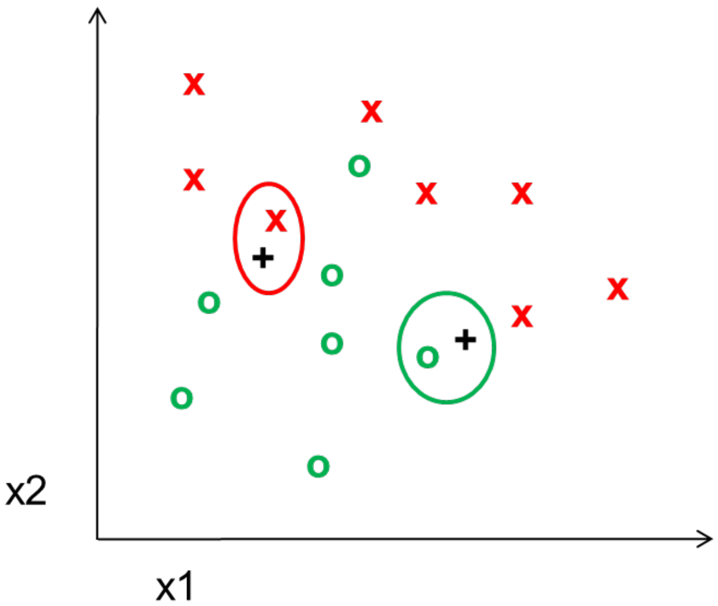
HW 1

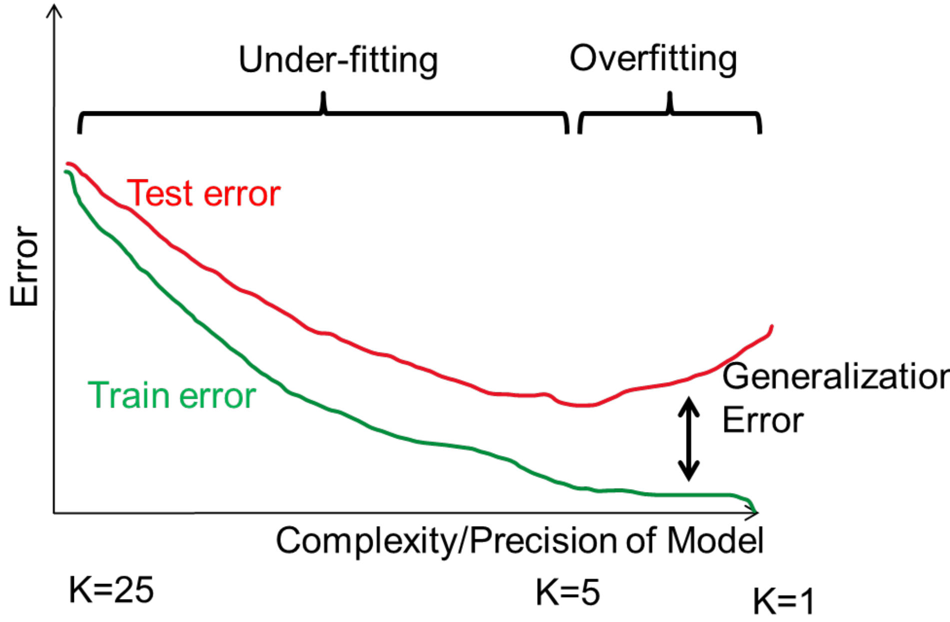
Things to remember

- KNN is a simple but effective classifier/regressor that predicts the label of the most similar training example(s)

- Larger K gives a smoother prediction function

- Test error is composed of bias (model too simple/smooth to fit data) and variance (model too complex to learn from training data)





Next week

- Dimensionality reduction
- Linear regression