

Introduction to Machine Learning

Classifier Evaluation



ML Problems: Recall

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction

Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

How to evaluate?

Training vs Generalization Error

- Training Error
 - Not very useful
 - Relatively easy to obtain low error

$$E_{train} = \frac{1}{n} \sum_{i=1}^n \underbrace{\text{error}(f_D(\mathbf{x}_i), y_i)}_{\substack{\text{same? different by how much?} \\ \text{value we predicted} \quad \text{true value}}} \quad \begin{matrix} \text{training examples} \end{matrix}$$

Empirical Risk Minimization

- Generalization Error
 - How well we do on future data

How to compute generalization error?

Estimating Generalization Error

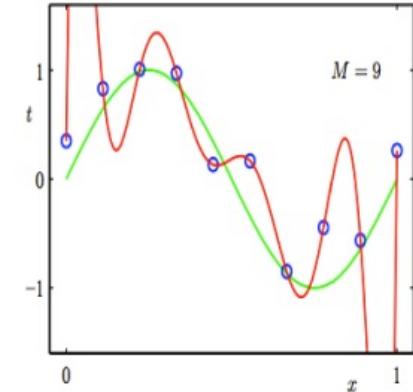
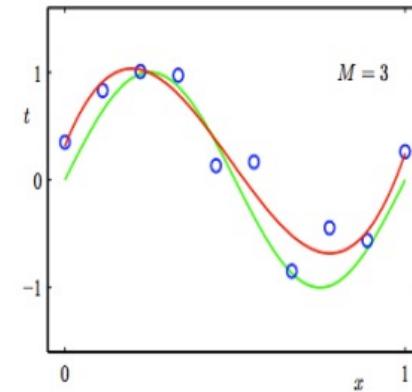
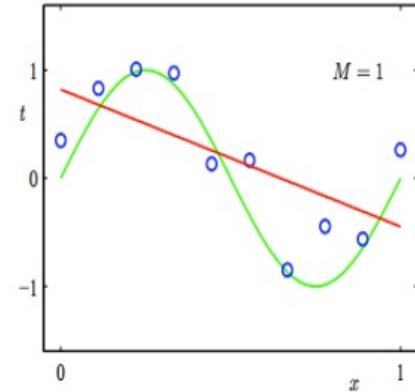
- Testing Error
 - Set aside part of training data (testing set)
 - Learn a predictor without using any of this test data
 - Predict values for testing set, compute error
 - This is an estimate of generalization error

$$E_{test} = \frac{1}{n} \sum_{i=1}^n \text{error}(f_D(\mathbf{x}_i), y_i)$$

over testing set

Underfitting and Overfitting

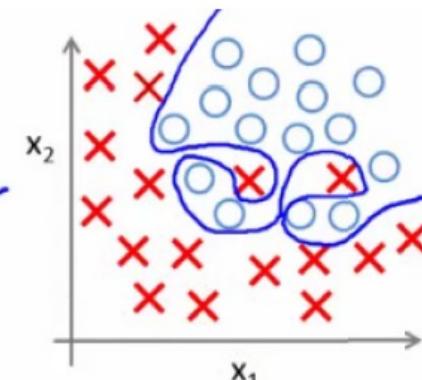
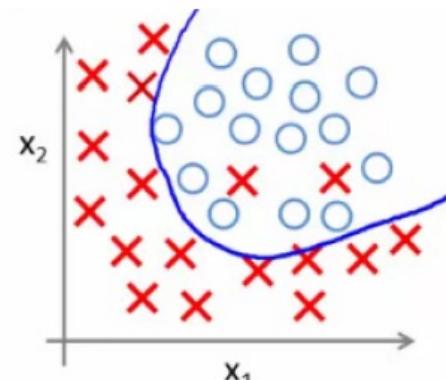
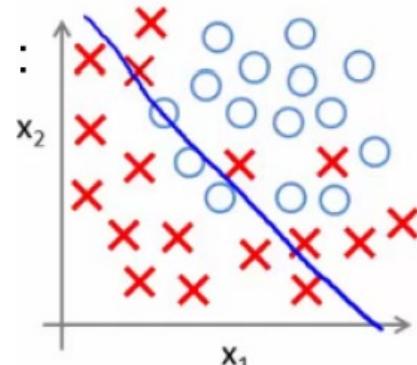
Regression



predictor too inflexible:
cannot capture pattern

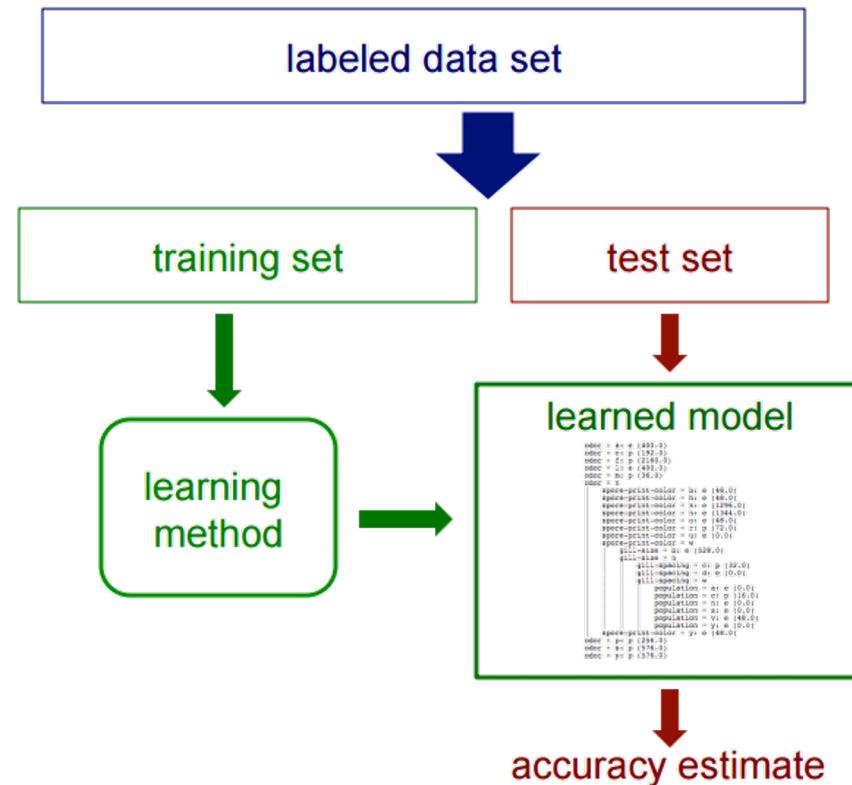
predictor too flexible:
fits noise in the data

Classification

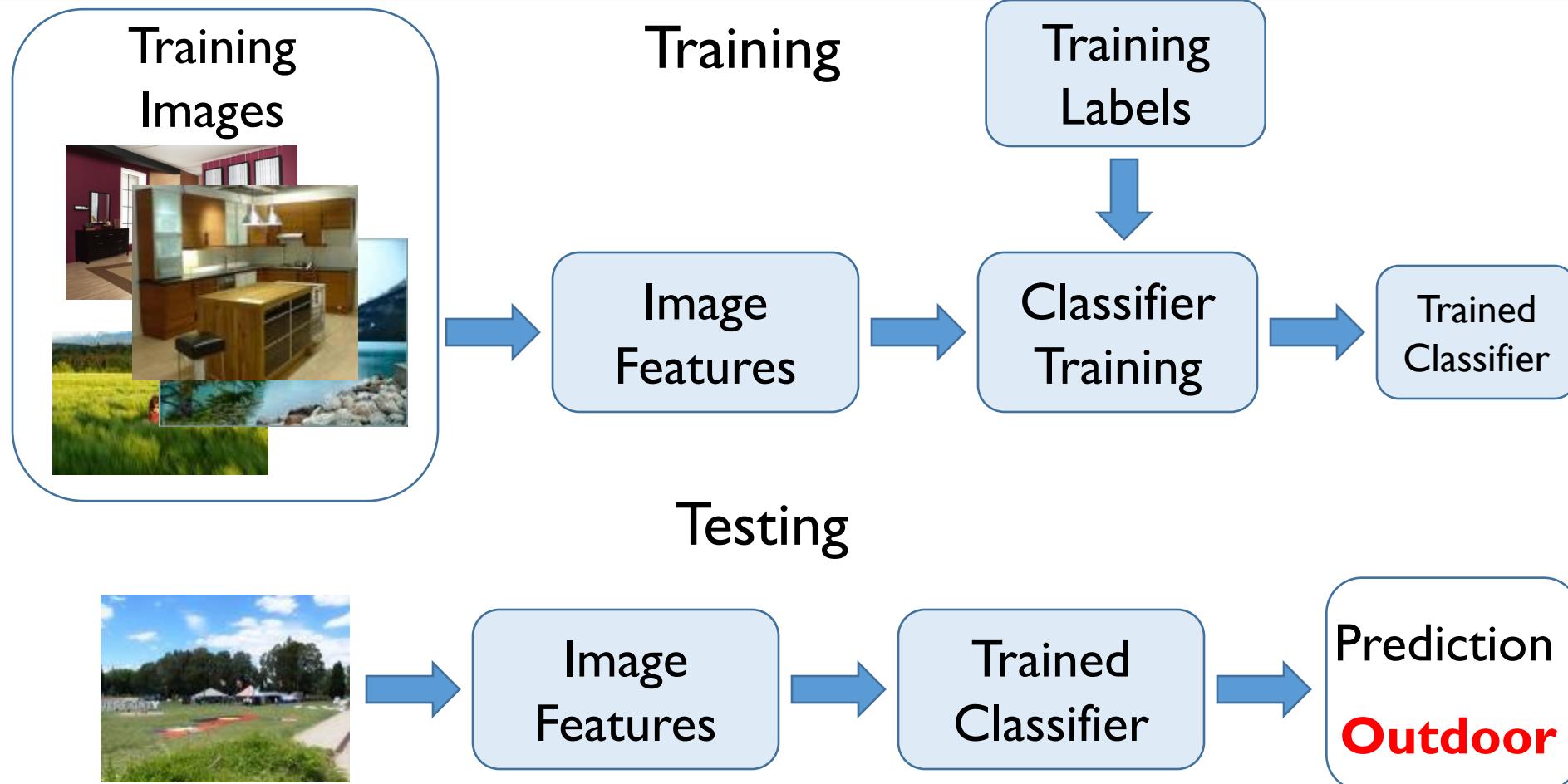


Estimating Generalization Error

- Getting an unbiased estimate of the accuracy of a learned model



Example: Image Classification



Source: Derek Hoiem

Training, Validation, Test Sets

Training set

- NB: Count frequencies, DT: Pick attributes to split on

Validation set

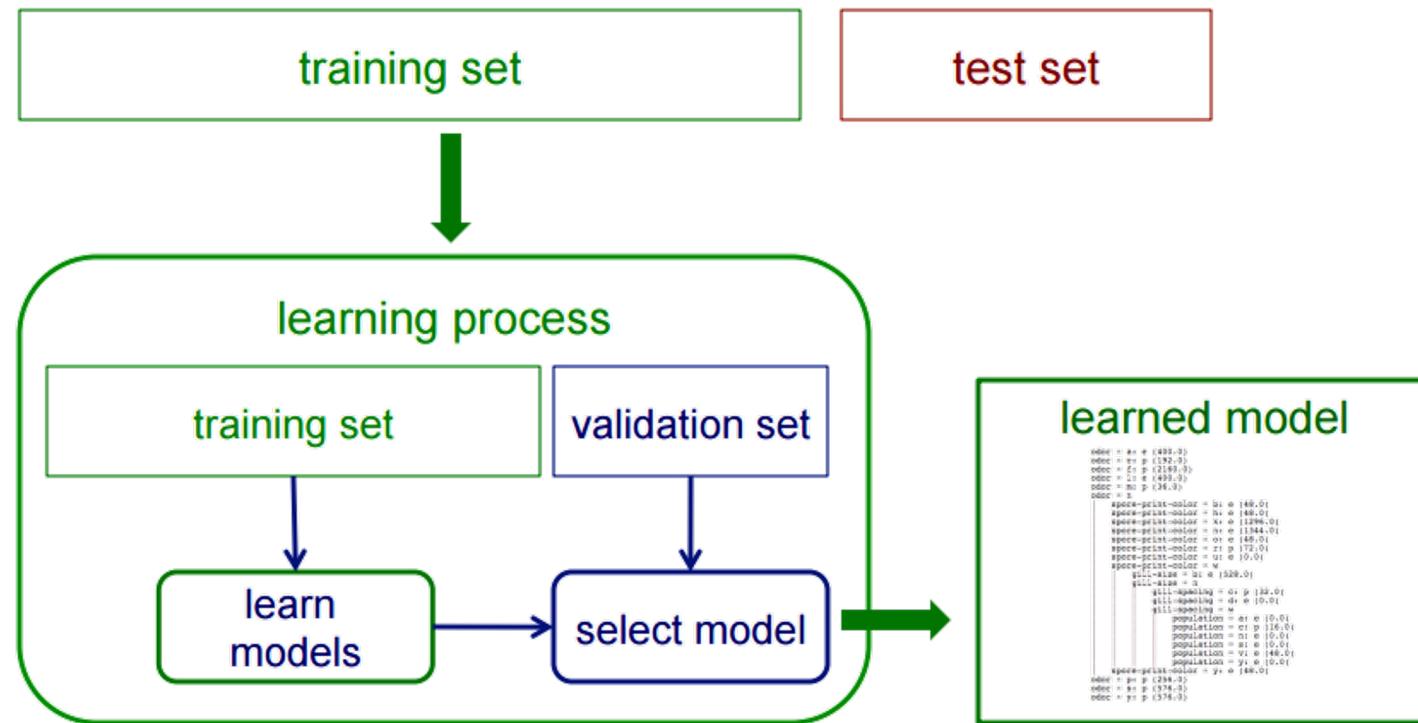
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

Testing set

- Run multiple trials and average

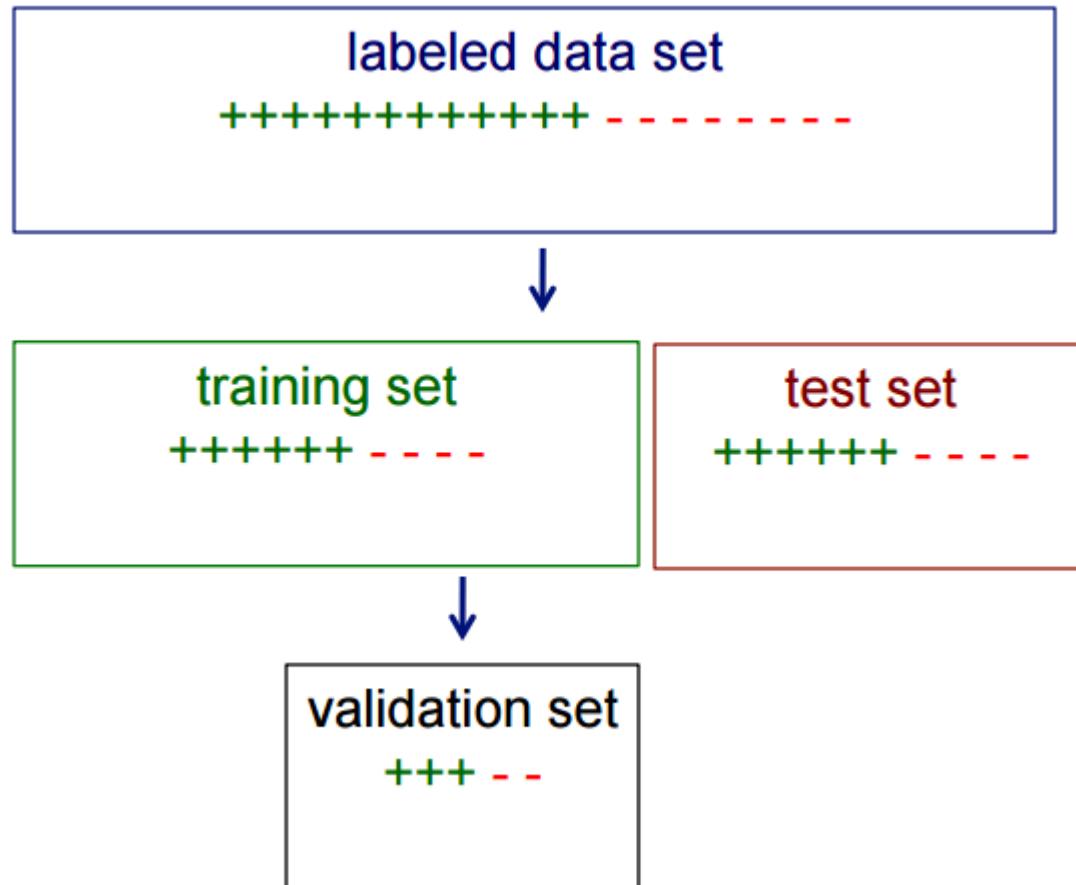
Use of Validation Sets

- If we want unbiased estimates of accuracy during the learning process:



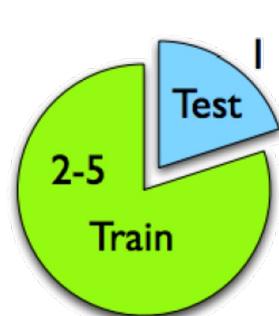
Stratified Sampling

- When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set
- This can be done via **stratified sampling**: first stratify instances by class, then randomly select instances from each class proportionally.

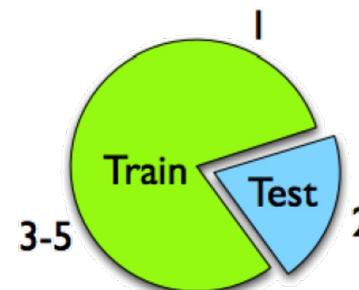


Model Selection

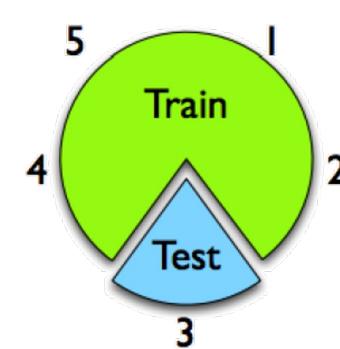
- Resubstitution
- K-fold cross-validation



Fold 1



Fold 2



Fold 3

- Leave-one-out
 - N-fold cross-validation

Cross-Validation: Example

- Suppose we have 100 instances, and we want to estimate accuracy with cross validation

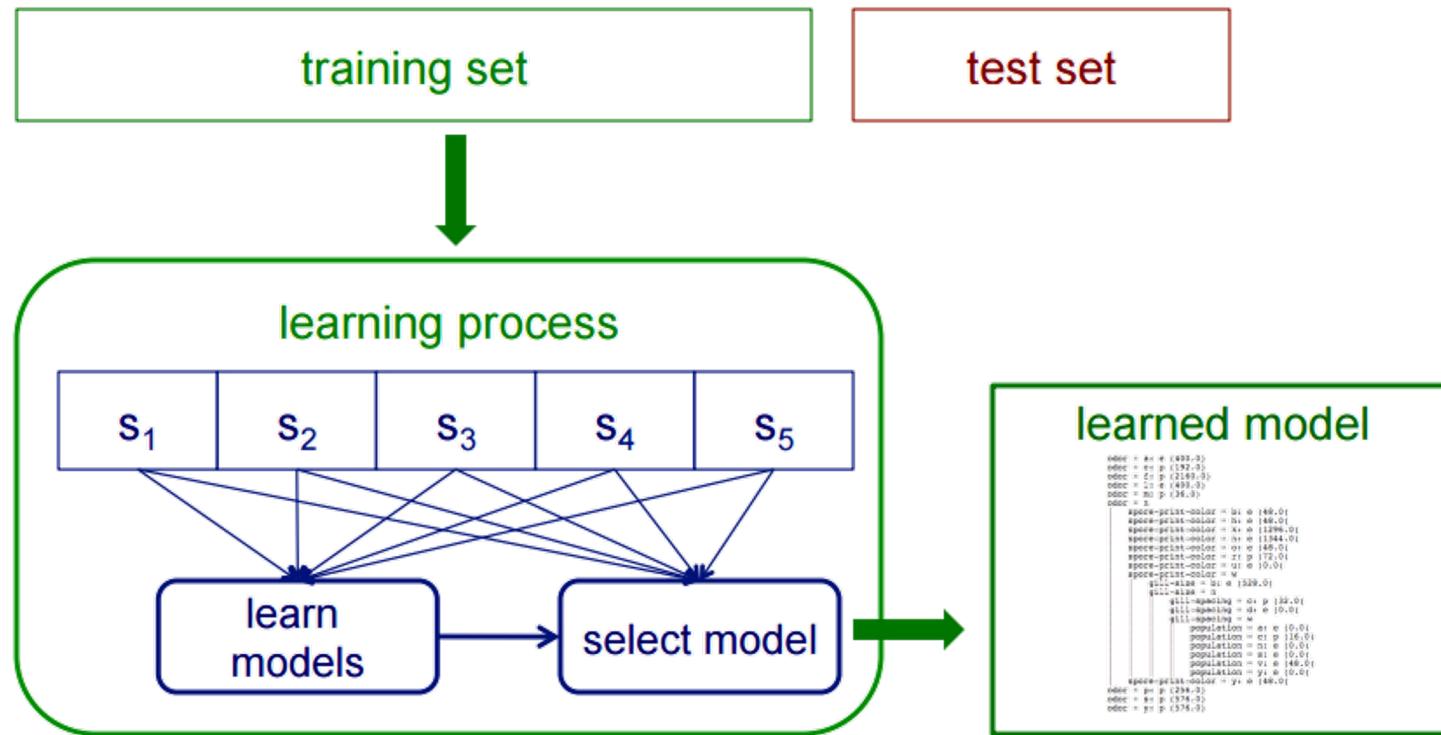
iteration	train on	test on	correct
1	s ₂ s ₃ s ₄ s ₅	s ₁	11 / 20
2	s ₁ s ₃ s ₄ s ₅	s ₂	17 / 20
3	s ₁ s ₂ s ₄ s ₅	s ₃	16 / 20
4	s ₁ s ₂ s ₃ s ₅	s ₄	13 / 20
5	s ₁ s ₂ s ₃ s ₄	s ₅	16 / 20

$$\text{Classification Accuracy} = 73/100 = 73\%$$

Note: Whenever we use multiple training sets, as in CV and random resampling, we are evaluating a learning method as opposed to an individual learned model

Cross-Validation: Example

- Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best k in k-NN)



Evaluation Measures

- Classification
 - How often we classify something right/wrong
- Regression
 - How close are we to what we're trying to predict
- Ranking/Search
 - How correct are the top-k results?
- Clustering
 - How well we describe our data (Not straightforward)

Regression Error: Root Mean Square Error

Regression consider real valued output

Regression

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

Classification Error: Accuracy

In 2-class problems:

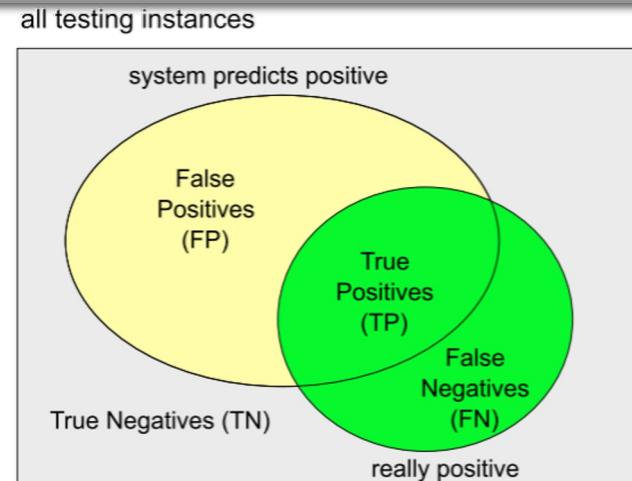
		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Classification Performance Measures

Predict positive?

	Yes	No
Yes	TP	FN
No	FP	TN



- Classification Error: $\frac{errors}{total} = \frac{FP + FN}{TP + TN + FP + FN}$
 - Accuracy = 1 - Error: $\frac{correct}{total} = \frac{TP + TN}{TP + TN + FP + FN}$
 - False Alarm = False Positive rate = $FP / (FP+TN)$
 - Miss = False Negative rate = $FN / (TP+FN)$
 - Recall = True Positive rate = $TP / (TP+FN)$
 - Precision = $TP / (TP+FP)$
- meaningless if classes imbalanced

- True Positive Rate also called “**Sensitivity**”
 - “**Specificity**” = $1 - \text{False Alarm}$
 - “**Sensitivity**” = Probability of a positive test given a patient has the disease
 - “**Specificity**” = Probability of a negative test given a patient is well
- always report in pairs, e.g.: Miss / FA or Recall / Prec.
-

Classification Error: Beyond Accuracy

For multi-class problems?

Confusion Matrix

		activity recognition from video									
		predicted class									
		actual class									
bend	predicted class	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	actual class	100	0	0	0	0	0	0	0	0	0
jack		0	100	0	0	0	0	0	0	0	0
jump		0	0	89	0	0	0	11	0	0	0
pjump		0	0	0	100	0	0	0	0	0	0
run		0	0	0	0	89	0	11	0	0	0
side		0	0	0	0	0	100	0	0	0	0
skip		0	0	0	0	0	0	100	0	0	0
walk		0	0	0	0	0	0	0	100	0	0
wave1		0	0	0	0	0	0	0	0	67	33
wave2		0	0	0	0	0	0	0	0	0	100

Courtesy: vision.jhu.edu

Is accuracy adequate?

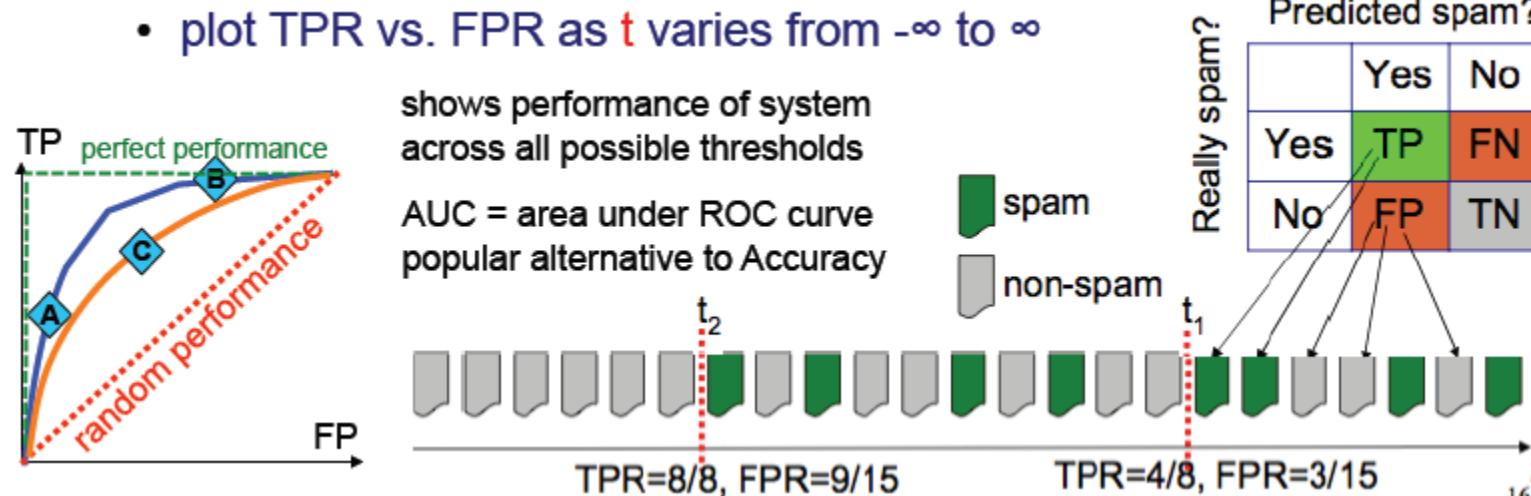
- Accuracy may not be useful in cases where
 - There is a large class skew
 - Is 98% accuracy good if 97% of the instances are negative?
 - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
 - Consider a medical domain in which a false positive results in an extraneous test but a false negative results in a failure to treat a disease
 - We are most interested in a subset of high-confidence predictions

Utility and Cost

- Sometimes, there is a cost for each error
 - E.g. Earthquake prediction
 - False positive: Cost of preventive measures
 - False negative: Cost of recovery
- Detection Cost (Event detection)
 - $\text{Cost} = C_{FP} * FP + C_{FN} * FN$
- F-measure (Information Retrieval)
 - $F1 = 2/(1/\text{Recall} + 1/\text{Precision})$

ROC Curves

- Many algorithms compute “confidence” $f(x)$
 - Threshold to get decision: spam if $f(x) > t$, non-spam if $f(x) \leq t$
 - Threshold to determine error rates
- Receiver Operating Characteristic (ROC)

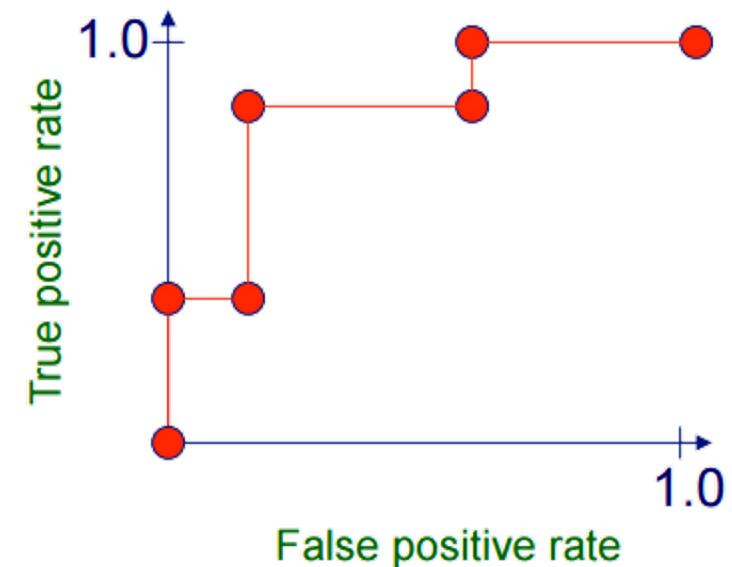


ROC Curve:Algorithm

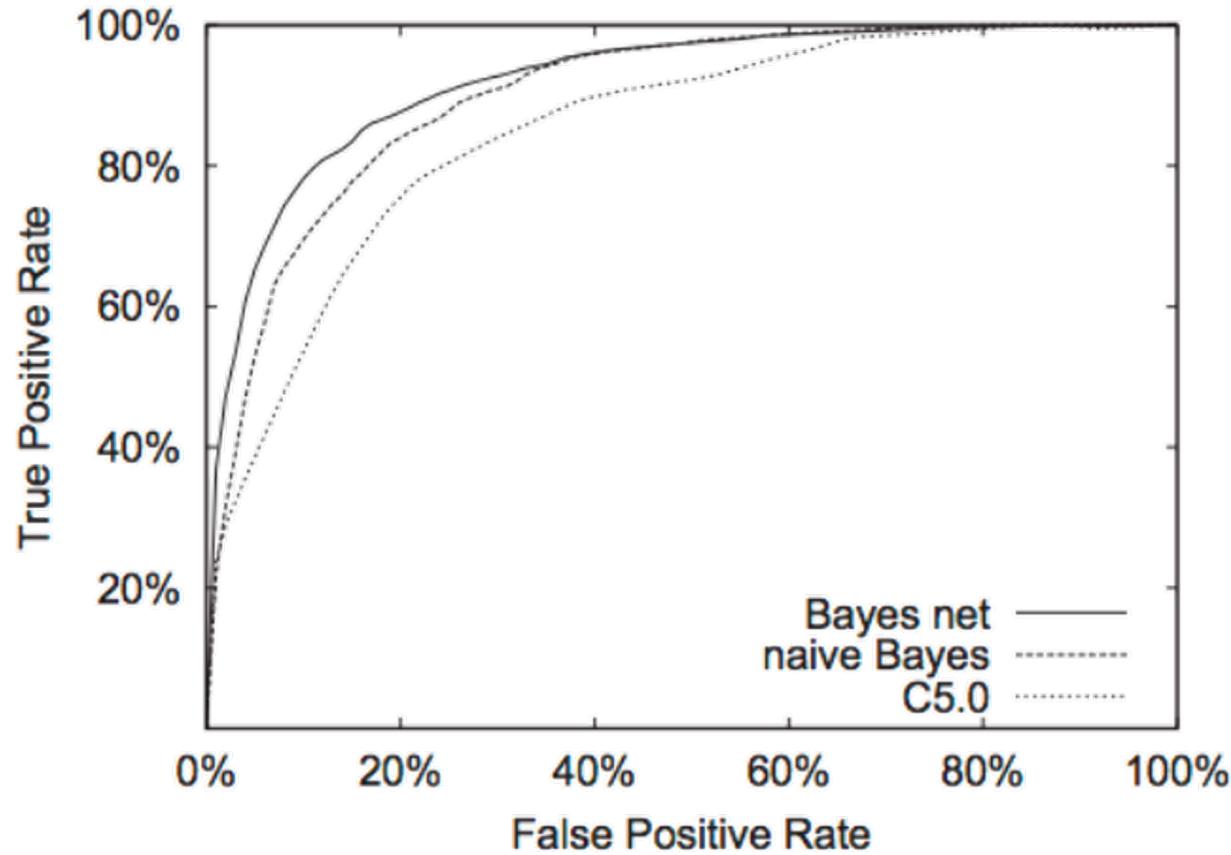
- Sort test-set predictions according to confidence that each instance is positive
- Step through sorted list from high to low confidence
 - Locate a threshold between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)
 - Compute TPR, FPR for instances above threshold
 - Output (FPR,TPR) coordinate

Plotting an ROC Curve

instance	confidence positive	correct class
Ex 9	.99	+
Ex 7	.98	TPR= 2/5, FPR= 0/5
Ex 1	.72	TPR= 2/5, FPR= 1/5
Ex 2	.70	+
Ex 6	.65	TPR= 4/5, FPR= 1/5
Ex 10	.51	-
Ex 3	.39	TPR= 4/5, FPR= 3/5
Ex 5	.24	TPR= 5/5, FPR= 3/5
Ex 4	.11	-
Ex 8	.01	TPR= 5/5, FPR= 5/5



ROC Curve: Example



Courtesy: Bockhorst et al., Bioinformatics 2003

Recall: Precision-Recall

		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

$$\text{recall (TP rate)} = \frac{\text{TP}}{\text{actual pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

F-score: Harmonic mean of precision and recall
 $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$

Summary

- Rigorous statistical evaluation is extremely important in experimental computer science in general and machine learning in particular
- How good is a learned hypothesis?
- How close is the estimated performance to the true performance?
- Is one hypothesis better than another?
- Is one learning algorithm better than another on a particular learning task?

References

- Key References
 - Chapter 19, EA Introduction to ML, 2nd Edn
 - Chapter I (Sec I.I-I.5), Pattern Recognition and Machine Learning, Bishop
- Other Recommended References
 - http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf (Tutorial on Performance Evaluation of Classifiers)
 - Chapter 5 ('Evaluating Hypotheses'), Machine Learning by Tom Mitchell
 - <http://www.cs.cmu.edu/~tom/mlbook.html>