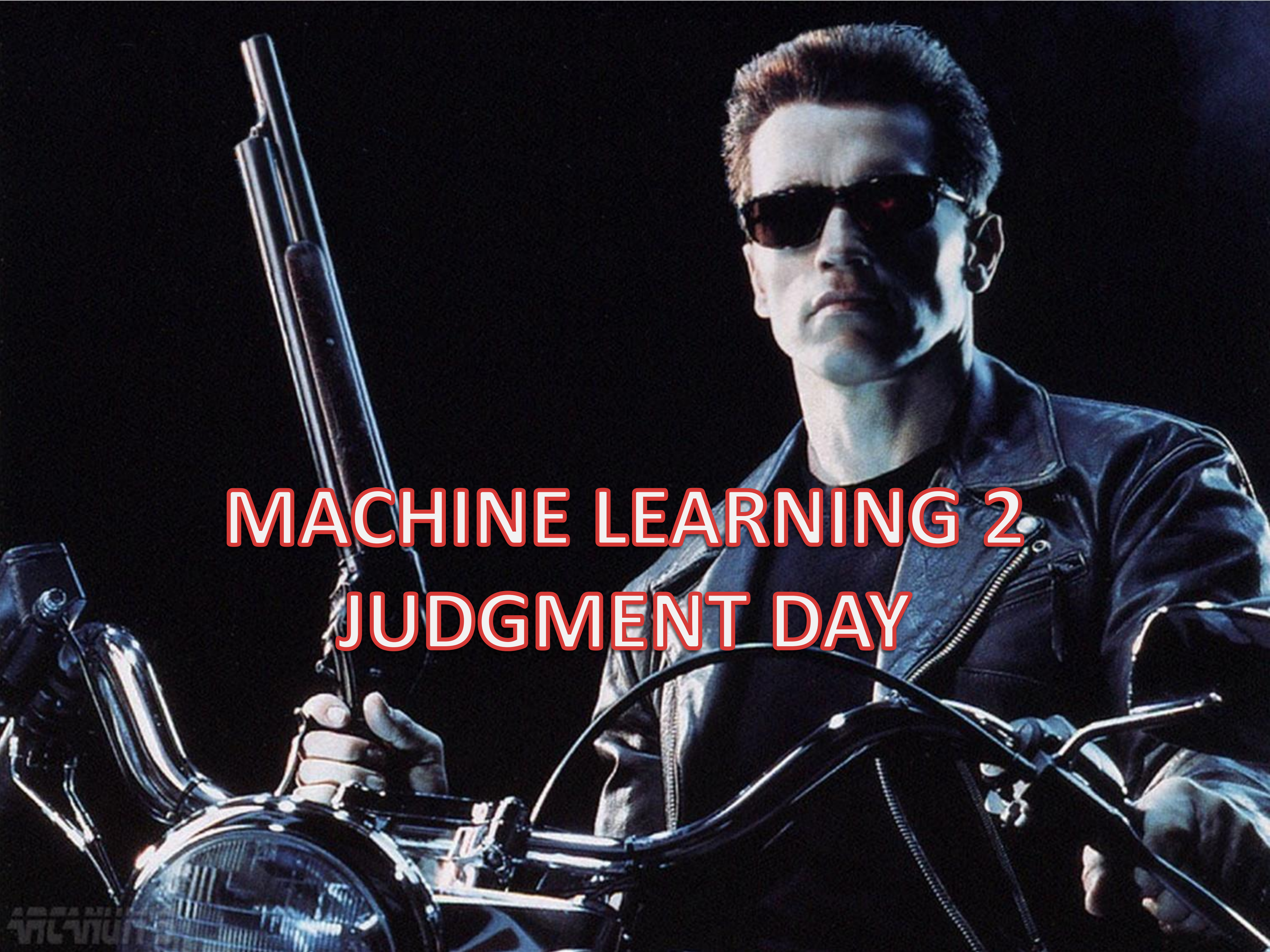


MACHINE LEARNING 2
JUDGMENT DAY

Review

- Often in PL we try to form judgments about complex human-related phenomena
- The right answer arises from a complex interaction of features
- ML can help us model this sort of thing

Examples

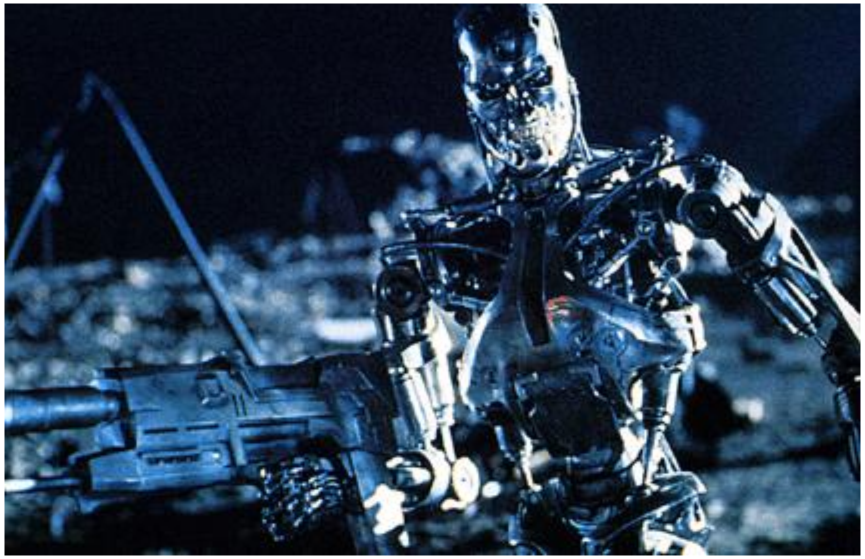
- Bug Reports
- Readability
- Documentation
- Version Control Histories

Last Time

- ML Basics
- ML (heart) PL
- K-means Clustering (Unsupervised)
- Linear Regression (Supervised)

Back to the Future

- Quick review of basics (now with formalisms)
- Advanced ML Algorithms (supervised only)
- Coercing a problem in ML
- Evaluation Techniques
- Other “useful” info



Review: Supervised Learning

Given a set of example pairs (instances, answers)

$$\mathrm{T} = \{(\vec{x}, y) \mid \vec{x} \in X, y \in Y\}$$

Find a function (in the allowed class)

$$f : X \to Y, f \in F$$

That minimizes some cost function

$$C : \langle F, \mathrm{T} \rangle \to \Re$$

Cost Function

$$C : \langle F, \mathrm{T} \rangle \rightarrow \Re$$

- C should be function of the observations. Why?

- Example: Regression

$$C = \frac{1}{|\mathrm{T}|} \sum_{i=1}^{|\mathrm{T}|} (f(x_i) - y_i)^2$$

New: Support Vector Machines

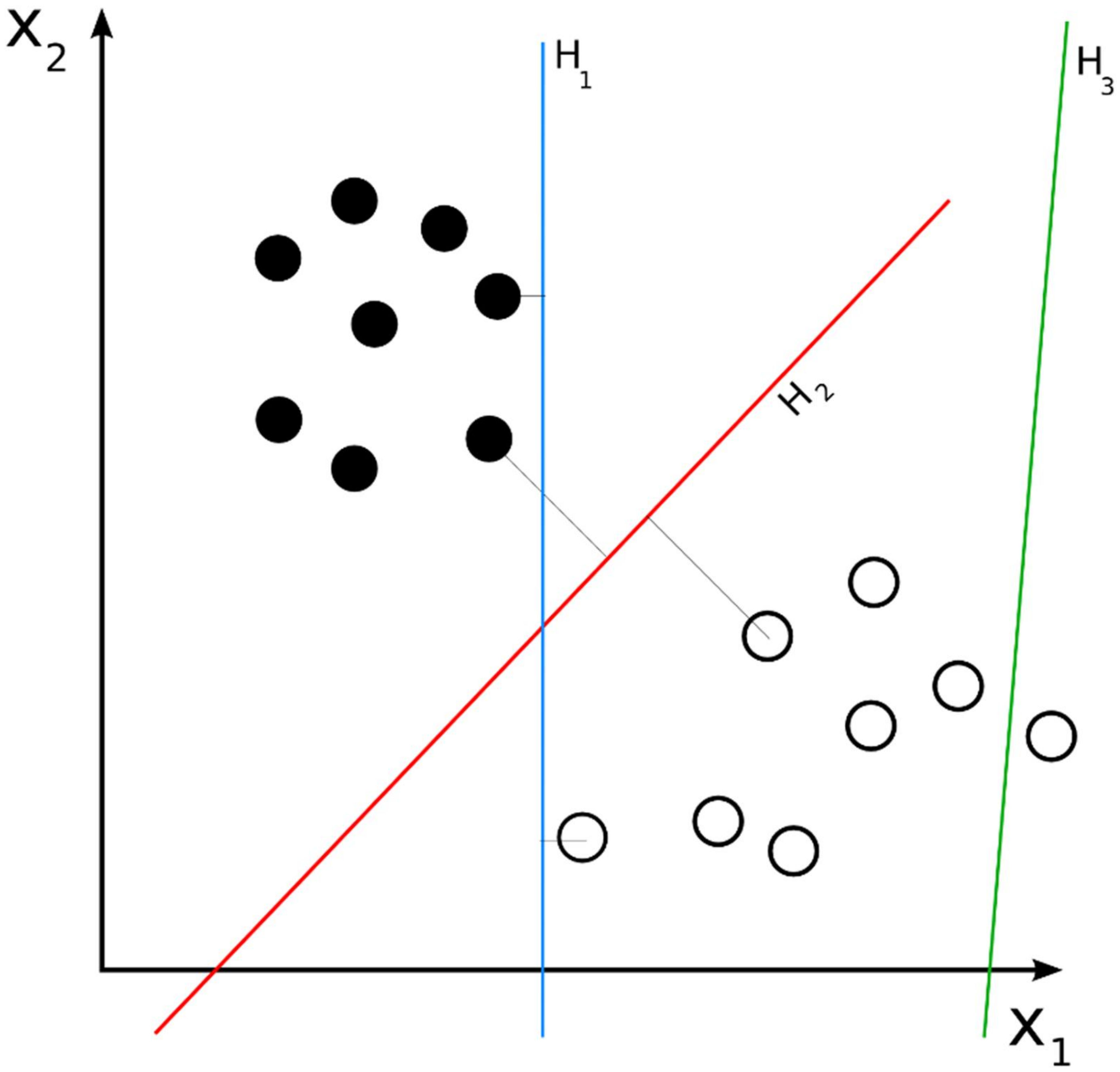
Generalized Linear Classifiers

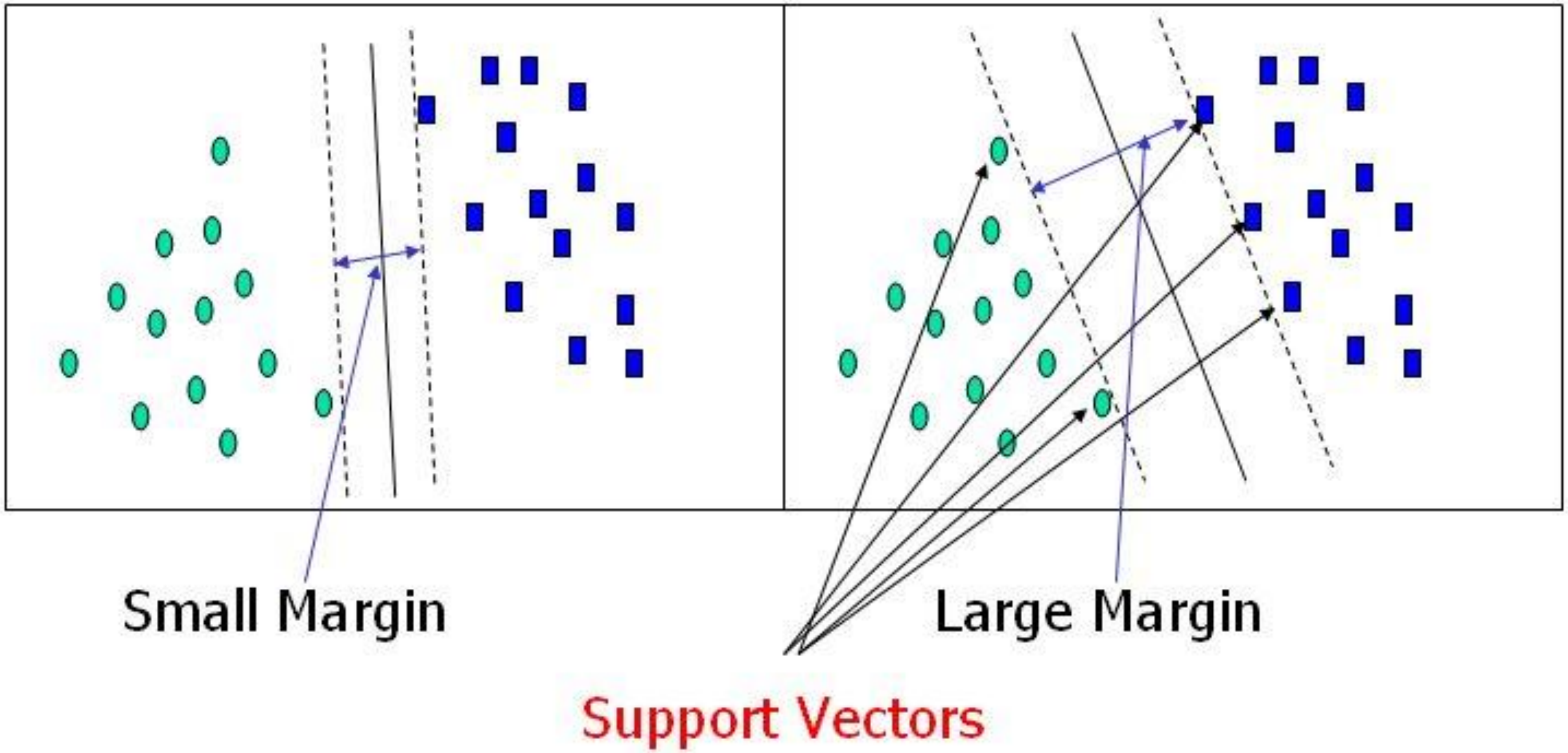
Binary Class Version:

- Given Training Data

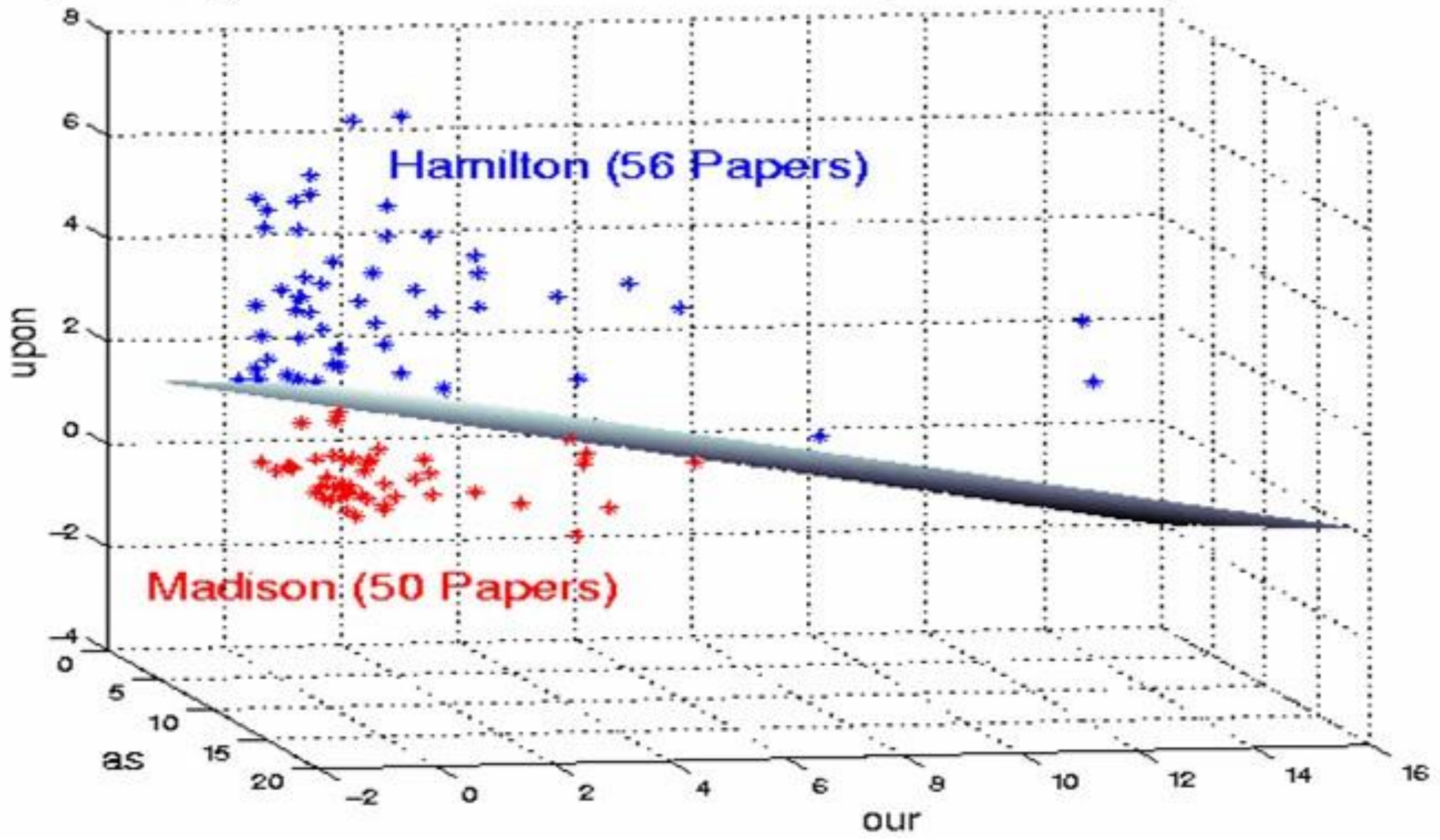
$$\mathrm{T} = \{(\vec{x}, y) \big| \vec{x} \in \Re^p, y \in \{-1,1\}\}$$

- We want to give the *hyperplane* that divides the points having y=-1 from y=1 that has "maximal margin"

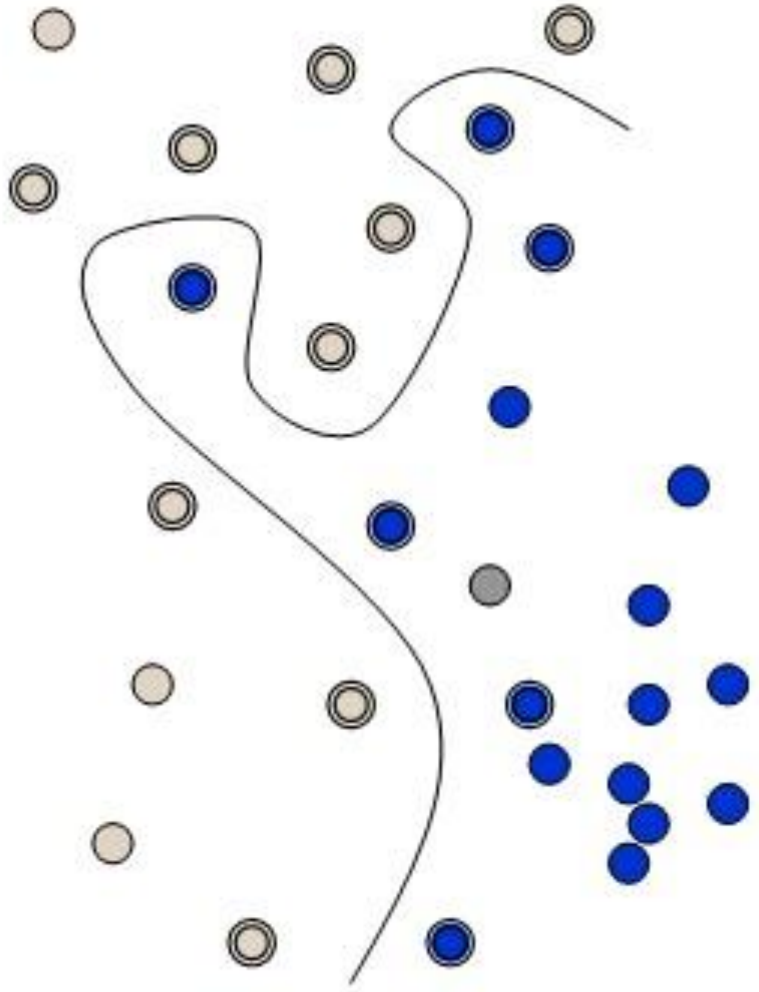

X_2
H_1
H_3
H_2
X_1


Small Margin
Large Margin
Support Vectors

Separating Plane for the Federalists Papers – 1788 (Bosch–Smith)

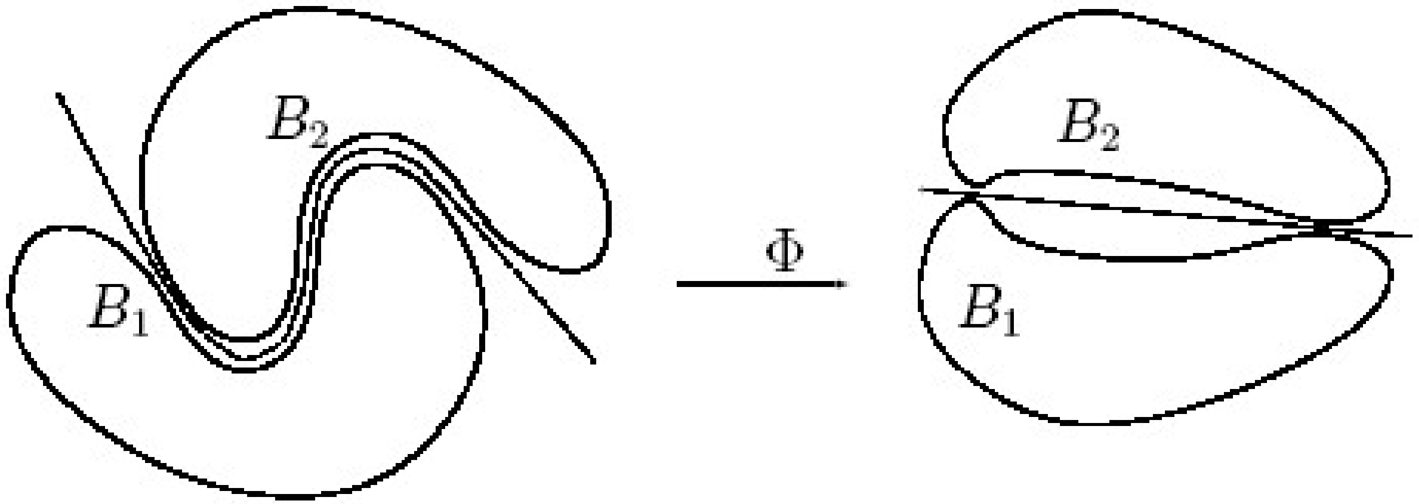


When Straight Lines go Crooked



When Straight Lines go Crooked

- Rather than fitting nonlinear curves to the data, SVM handles this by using a *kernel function* to map the data into a different space

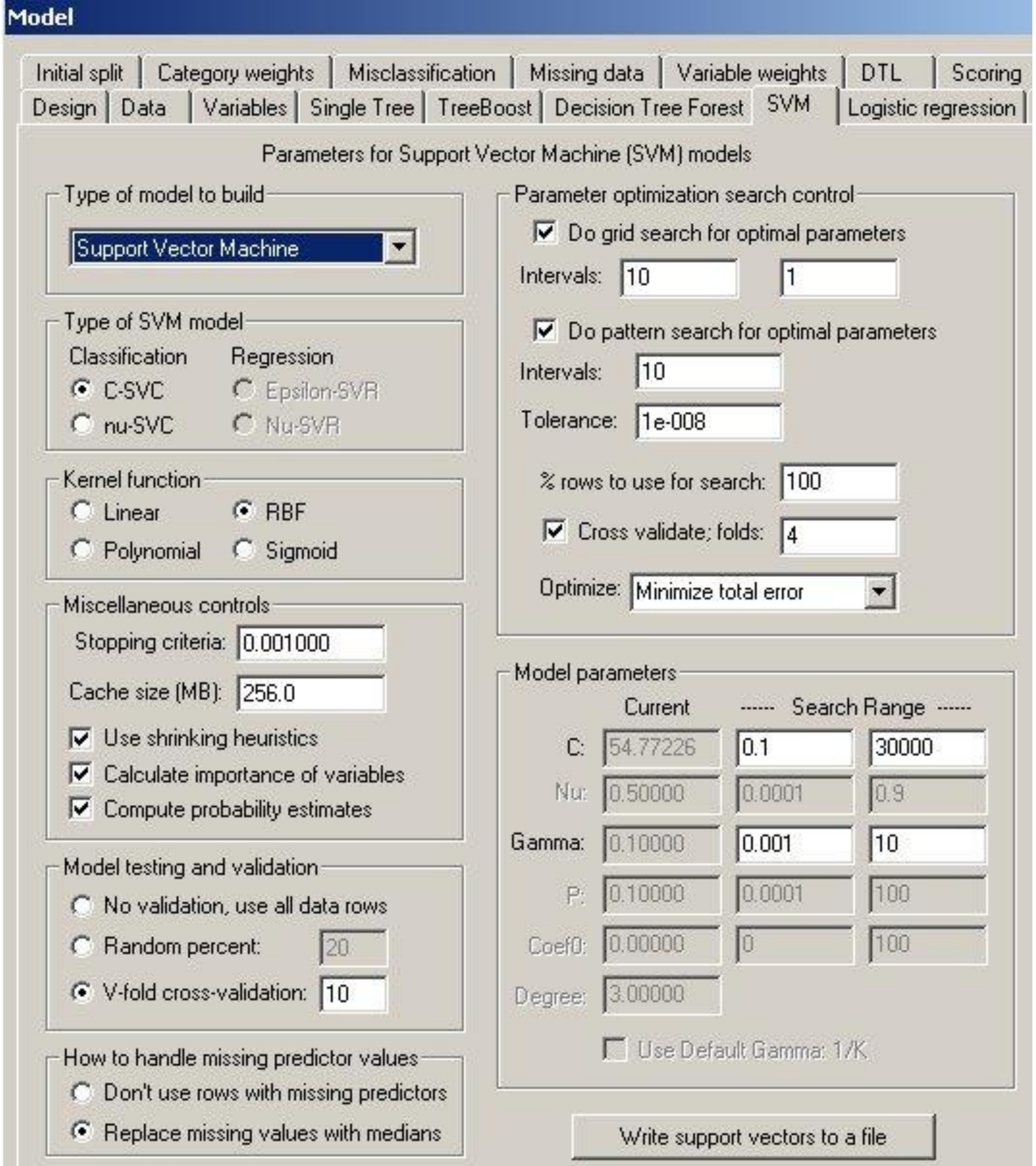


When Perfect Separation is not possible

- SVM models have a cost parameter, *C*, that controls the trade off between allowing training errors and forcing rigid margins.

Non-binary SVM

- *one against many* - k models where each class is split out and all of the other classes are merged
- *one against one* - $k(k$-1)/2 models - all pair wise combinations.



Parameters for Support Vector Machine (SVM) models

Type of model to build: Support Vector Machine

Type of SVM model
Classification: [✓ C-SVC] nu-SVC
Regression: Epsilon-SVR Nu-SVR

Kernel function: Linear [✓ RBF] Polynomial Sigmoid

Miscellaneous controls
Stopping criteria: 0.001000
Cache size (MB): 256.0
✓ Use shrinking heuristics
✓ Calculate importance of variables
✓ Compute probability estimates

Model testing and validation
No validation, use all data rows
Random percent: 20
[✓ V-fold cross-validation: 10]

How to handle missing predictor values
Don't use rows with missing predictors
[✓ Replace missing values with medians]

Parameter optimization search control
✓ Do grid search for optimal parameters
Intervals: 10 1
✓ Do pattern search for optimal parameters
Intervals: 10
Tolerance: 1e-008
% rows to use for search: 100
✓ Cross validate; folds: 4
Optimize: Minimize total error

Model parameters

	Current	------ Search Range ------	
C:	54.77226	0.1	30000
Nu:	0.50000	0.0001	0.9
Gamma:	0.10000	0.001	10
P:	0.10000	0.0001	100
Coef0:	0.00000	0	100
Degree:	3.00000		

Use Default Gamma: 1/K

SVM Tradeoffs

Advantages

- Hypothesis has an explicit dependence on data (support vectors)
- Few tuning parameters

Disadvantages

- No clear estimate of “confidence” to go along with classification. Next: A probabilistic model…

A Probabilistic Model

- What's the probability of class C, given feature vector x?

$$P(C \mid x_{1,} \ldots x_n)$$

- The problem: If features can take on r different values, there are r^n different probabilities to consider!
- Bayes' Theorem to the rescue…

Bayes’ Theorem

- Bayes' theorem describes the way in which one's beliefs about observing 'A' are updated by having observed 'B'.

$$p(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Baysean Example

- Suppose 615 has 90% civilians and 10% terminators as students.
  - About half of terminators wear sunglasses.
  - None of the civilians wear sunglasses.
- You see a (random) student who is NOT wearing sunglasses. What is the probability this student is a terminator?

Baysean Example

Baysean Example

- Call event *A* the student is a terminator and the event *B* that the student is NOT wearing sunglasses.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.5 \cdot 0.1}{0.9} = 0.055$$

Applying Bayes' Theorem to ML

$$P(C \mid x_{1,} \ldots x_n) = \frac{P(C)P(x_{1,} \ldots x_n \mid C)}{P(x_{1,} \ldots x_n)}$$

- Denominator is effectively constant.
- Assuming conditional independence

$$P(x_i, x_j \mid C) = P(x_i \mid C)P(x_j \mid C)$$

- We can say:

$$P(C \mid x_{1,} \ldots x_n) = P(C)\prod_{i=1}^{n} P(x_i \mid C)$$

The Baysean Classifier

$$P(C \mid x_{1,} \ldots x_n) = P(C)\prod_{i=1}^{n} P(x_i \mid C)$$

- With k classes, and features r different values
  - There are k + n·k·r probabilities

Baysean Advantages

- Fast to train, fast to classify
- Well suited for categorical features
- Good for combining models:
  - Suppose we have two kinds of feature vectors, x_1 and x_2 Rather than building a large monolithic model, it might be more practical to learn two separate classifiers, $P(C|x_1)$ and $P(C|x_2)$ and then to combine them.

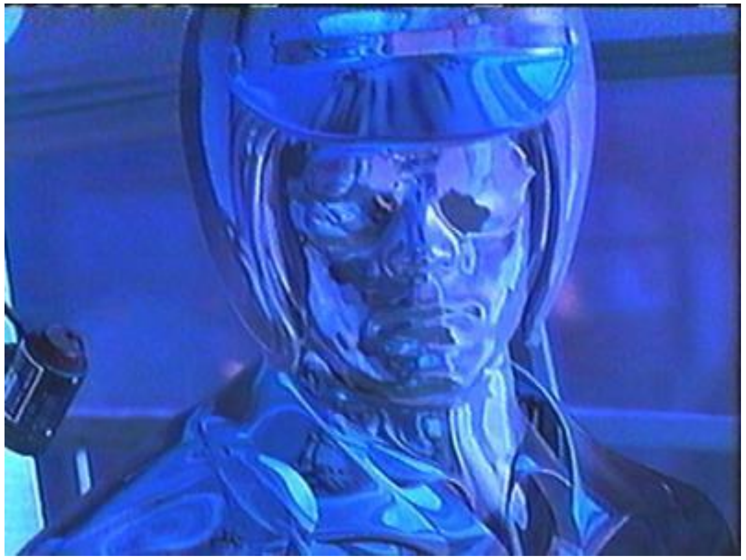
$$P(C \mid x_1, x_2) = \frac{P(C \mid x_1) P(C \mid x_2)}{P(C)}$$

Baysean Disadvantages

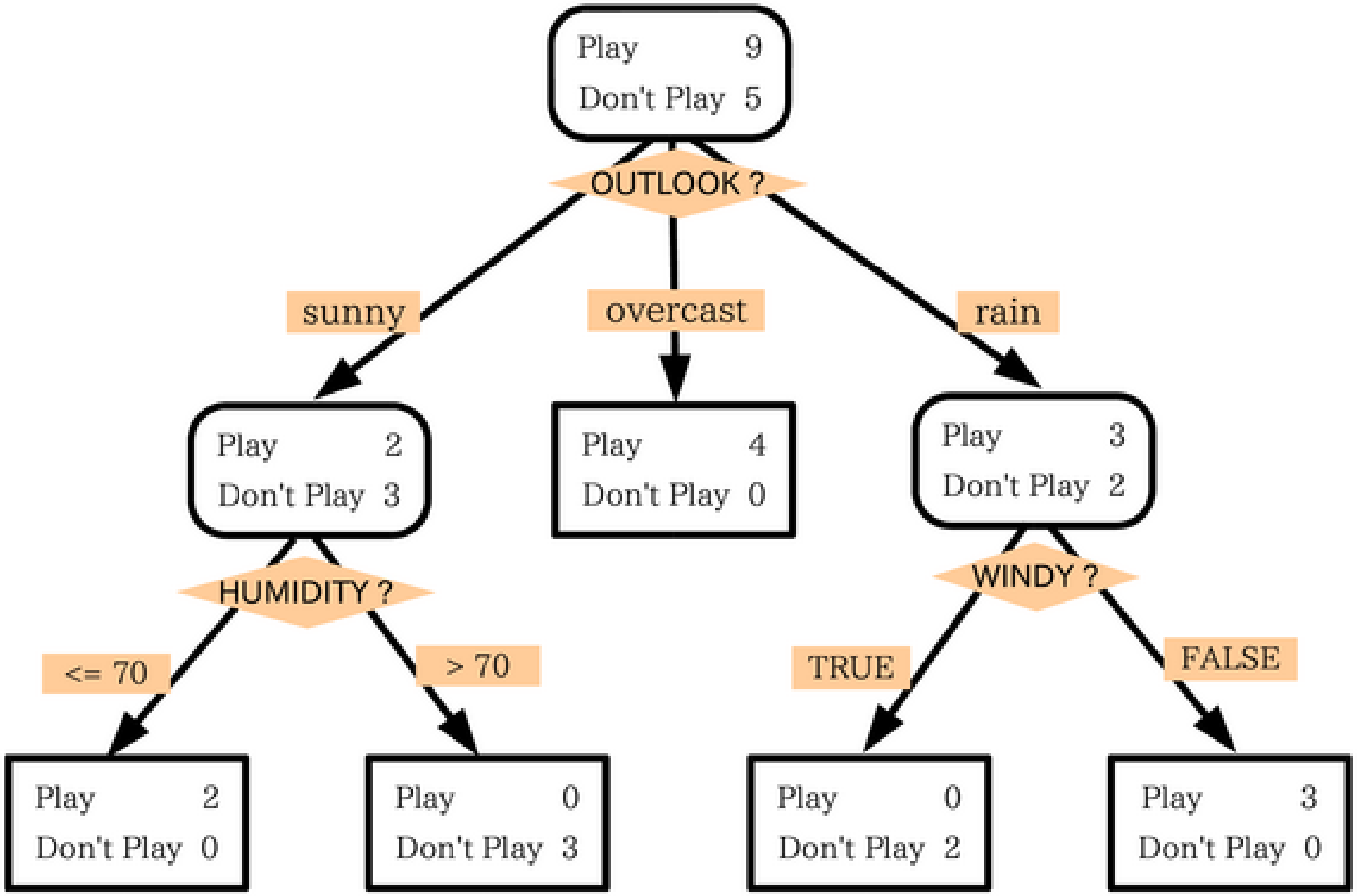
- If conditional independence isn’t true (at least most of the time), it may not make for a very good model

Advanced ML Algorithms

- Support Vector Machines
- Baysean
- Decision Trees
- Artificial Neural Networks



Decision Trees



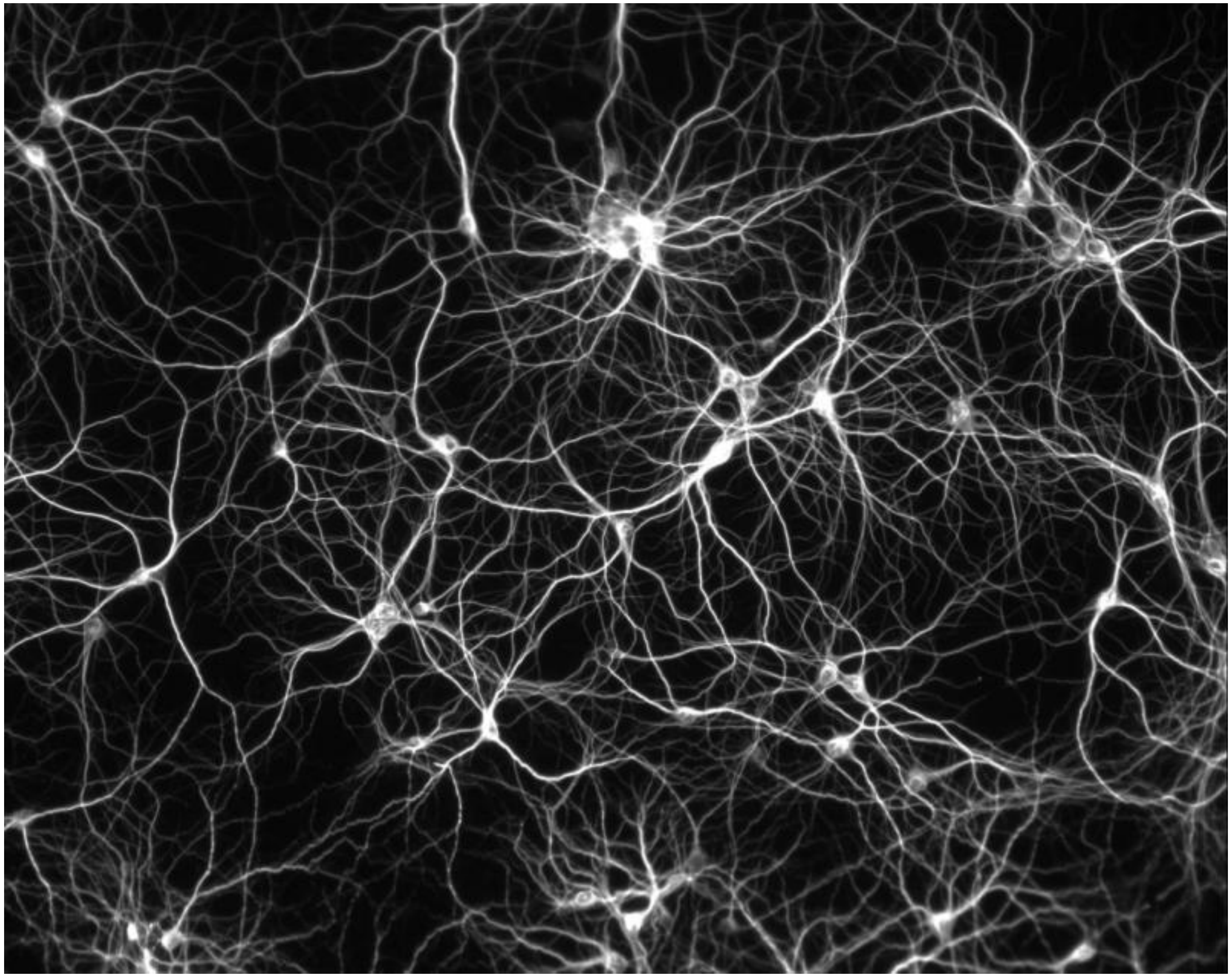
Automatic Decision Tree Building

- Most algorithms are variations on a top-down, greedy search through the space of possible decision trees.
- Search through the attributes of the training instances and pick the attribute that best separates the given examples.
- If the attribute perfectly classifies the training sets then stop.
- Otherwise recursively operate on the partitioned subsets to get their "best" attribute.

Decision Tree Advantages

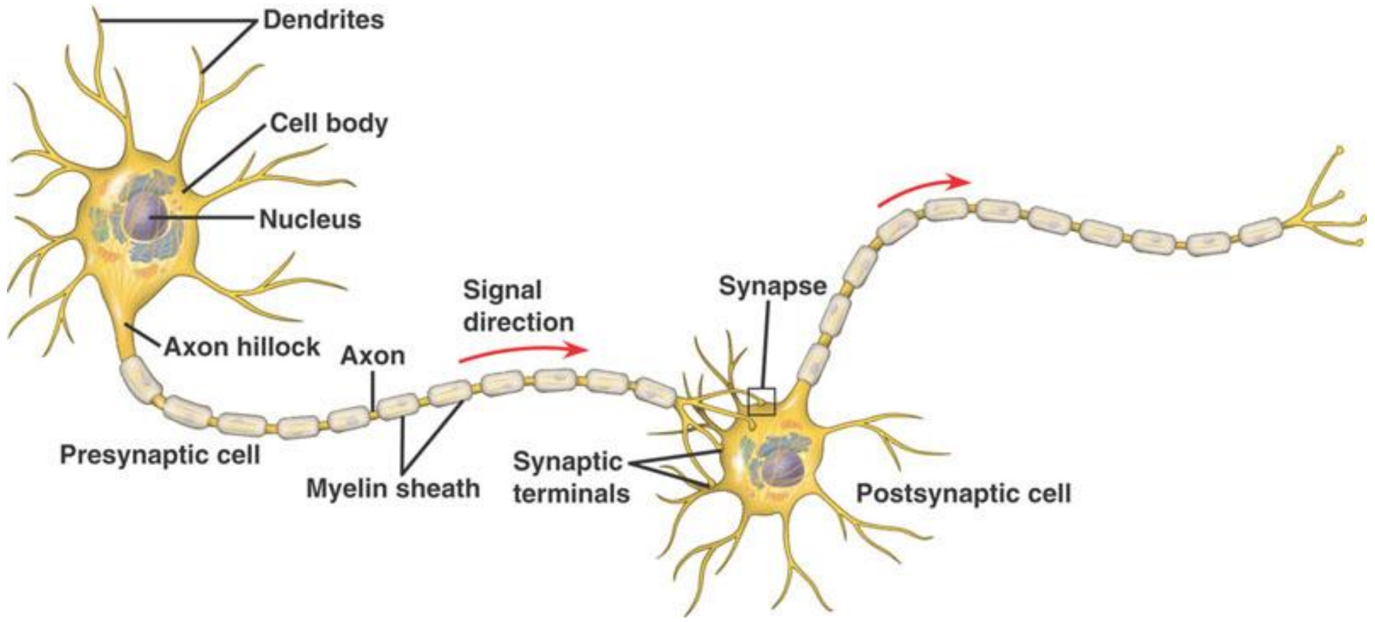
- Simple to understand and interpret.
- Able to handle both numerical and categorical data.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Decisions are fast

Artificial Neural Networks



ANNs

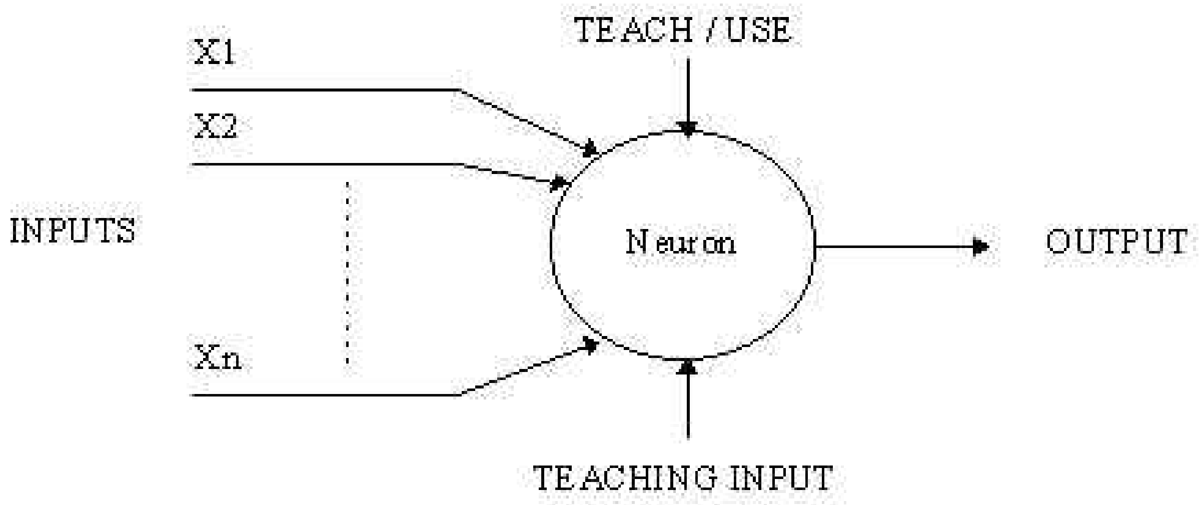
- Idea: simulate biological neural networks



- Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

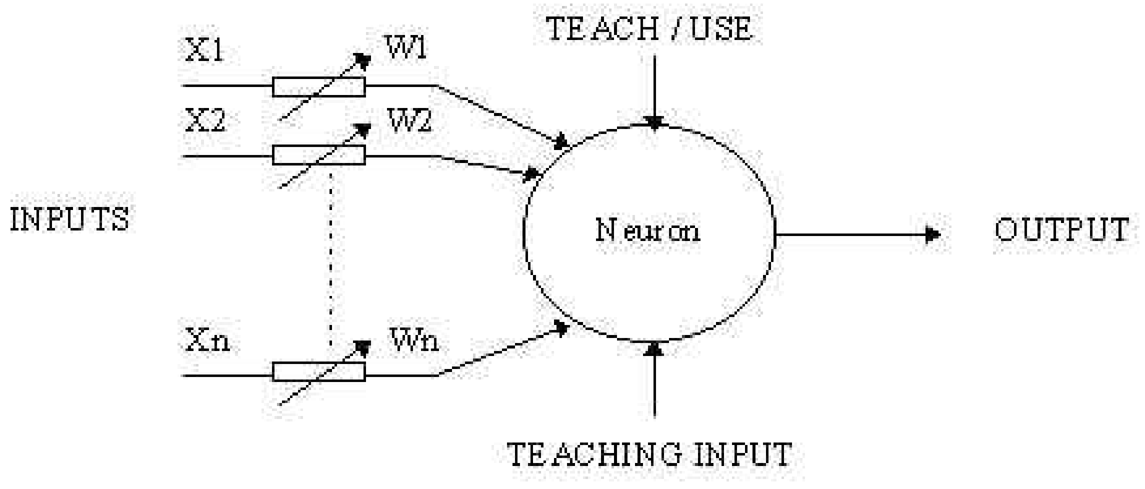
A Simple Firing Rule

- Take a collection of training patterns for a node.
- Patterns not in the collection cause the node to fire if, on comparison, they have more input elements in common with the 'nearest' pattern in the firing set than with the 'nearest' pattern in the non-firing set.



A More Complicated Neuron

- Weight the inputs.
- These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.



Back Propagation

Back Propagation

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a *scaling factor*, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error..

ANN Advantages

- Online learning
- Large dataset applications
- Their simple implementation and the existence of mostly local dependencies exhibited in the structure allows for fast, parallel implementations in hardware.

ANN Disadvantages

- Model is opaque
- Can take a long time to train
- Have to re-train whole thing when new features are added

You must choose…

How would you design a …

- Spam Filter (classify strings as Spam/Not Spam)
- Alarm System for a nuclear power plant.
- Terminator (Learn to act like a human, terminate John Connor)

…or not? Boosting

- Weak learner: classifier which is only slightly correlated with the true classification.
- Strong learner: a classifier that is arbitrarily well-correlated with the true classification.
- Can a set of **weak learners** create a single **strong learner**?
- Most boosting algorithms consist of iteratively weighting weak learners in some way that is usually related to the weak learner's accuracy.
- After a weak learner is added, the data is reweighted: examples that are misclassified gain weight and examples that are classified correctly lose weight.

Boosting & Cake Eating

- Can only work when the different learners have learned different things
- Combining is non-trivial
- Boosted decision trees a probably most popular

Now: Practical Stuff

- Formulating a problem for ML (creating features)
- Evaluating a model
- External Validity
- Evaluating Features
- Other things to think about
- Where to go for code

Forming a model

- ML is a recipe for solving a certain type of problem:

$$\mathrm{T} = \{(\vec{x}, y) \,|\, \vec{x} \in X, y \in Y\}$$
$$f : X \to Y, f \in F$$
$$C : \langle F, \mathrm{T} \rangle \to \Re$$

- Trick is to formulate a predictive model by choosing the right “features”

Features

```
/**
 * Moves this unit to america.
 *
 * @exception IllegalStateException If the
move is illegal.
 */
public void moveToAmerica() {

   if (!(getLocation() instanceof Europe))
{
      throw new IllegalStateException("A
unit can only be "
         + "moved to america from
europe.");
   }

   setState(TO_AMERICA);

   // Clear the alreadyOnHighSea flag
   alreadyOnHighSea = false;
}
```

feature	value
Comments	2
Function calls	3
LOC	19
Has Exception?	TRUE

Direct Model Evaluation

- How closely does the model conform to the data?
- This is non-trivial
- Major Issues
  - Bias
  - Statistical Significance
  - Over-fitting

Evaluation Strategy 1: Correlation

- Suppose we have a vector of correct answers X, and model output Y
- Question: How similar are X and Y?
- Pearson Product Moment Correlation Coefficient : r
- If r^2 is 0.90, then 90% of the variance of *Y* can be "accounted for" by the linear relationship between *X* and *Y*
- But what if the data is categorical?

Evaluation Strategy 2: f-measure

- You have a binary classifier for "will this report be resolved in <= 30 days"
- You have 27,984 reports with known answers
  - C = correct set of reports resolved in 30 days
  - R = set of reports the model returns

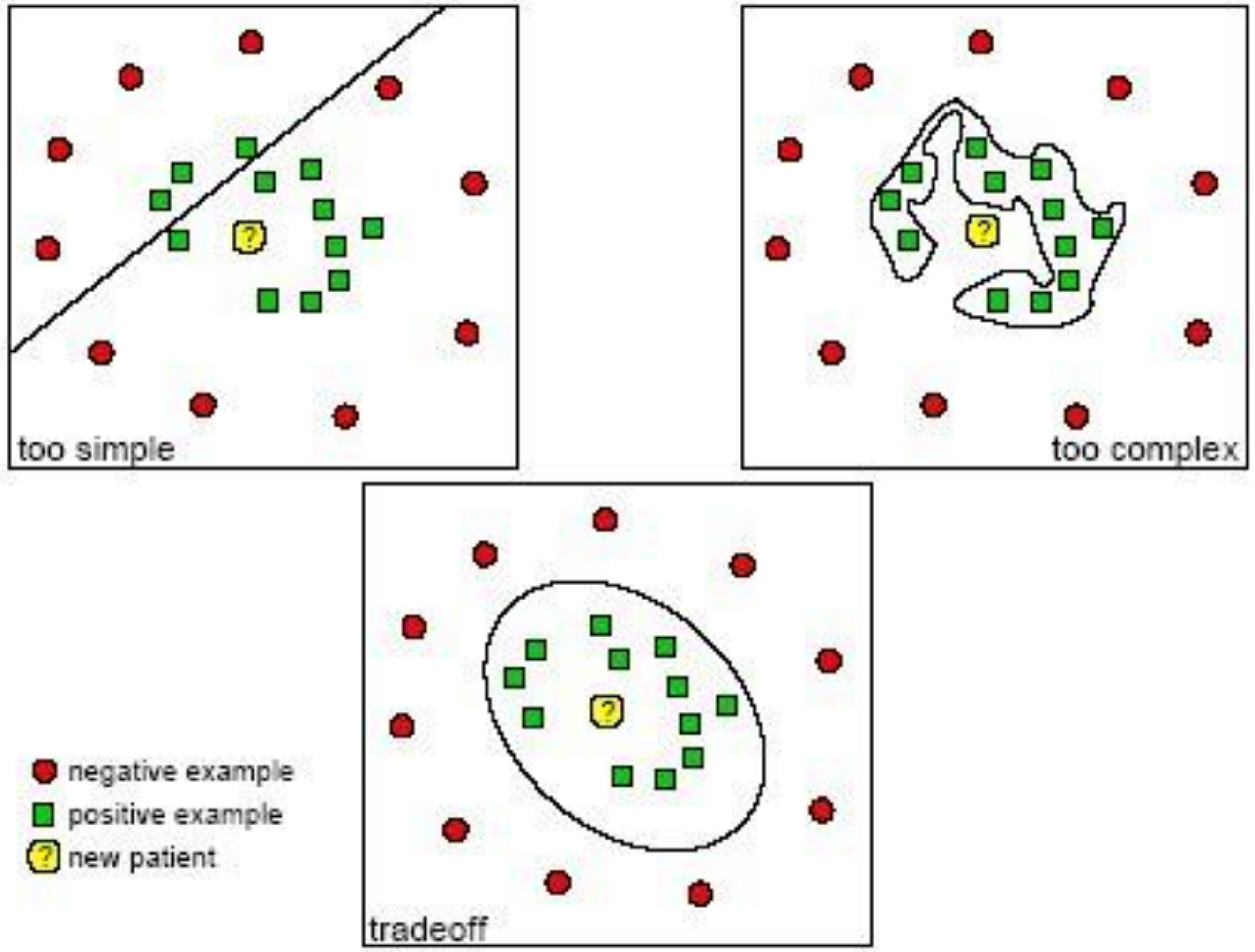
Precision = $\frac{|C \cap R|}{|R|}$ Recall = $\frac{|C \cap R|}{|C|}$

f-measure= $\frac{2 \cdot P \cdot R}{P + R}$

f-measure and Bias

- Say you have 100 instances
- 50 yes instances, 50 no instances, at random
  - "Flip Fair Coin": Prec=0.5, Rec=0.5, F=0.5
  - "Always Guess Yes": Prec=0.5, Rec=1.0, F=0.66
- 70 yes instances, 30 no instances, at random
  - "Flip Fair Coin": Prec=0.7, Rec=0.5, F=0.58
  - "Flip Biased Coin": Prec=0.7, Rec=0.7, F=0.7
  - "Always Guess Yes": Prec=0.7, Rec=1.0, F=0.82
- May want to subsample to 50-50 split for evaluation purposes

Underfitting and Overfitting



Cross Validation to test for Over-fitting

- Idea: Don’t evaluate on the same data you trained on.

- N-Fold Cross-Validation
  - Partition instances into n subsets
  - Train on 2..n and test on 1
  - Train on 1, 3..n and test on 2, etc.

More on Evaluation

- Given one day's worth of features, our best f-Measure for predicting “resolved within 30 days” was 0.76, and the industrial practice baseline was 0.73. Win?
- f-measure assumes false positives and false negatives are equally bad
  - For bug reports, missing a bug report is much worse than triaging an invalid one
- IR metrics are good, but relating your results back to the real world is key:
  - “For the purposes of comparison, however, if *Triage is $30 and Miss is $1000, using our model as a filter saves between five and six percent of the development costs for this data set.”*

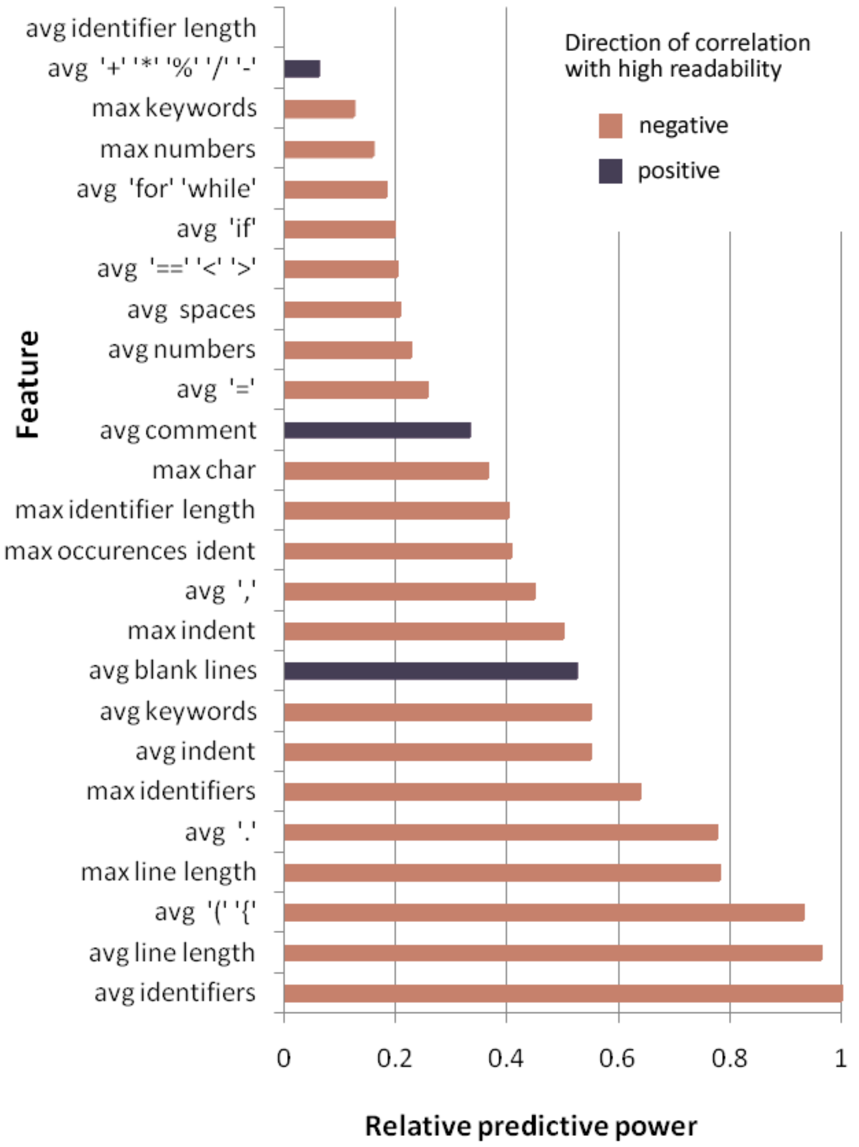
External Validity

- Sometimes we want to explore how well a model correlates with some external truth (e.g., to show utility)
- Solution:
  - Discrete Case: f-measure
  - Continuous Case: Pearson
  - Hybrid: Bins + Kendall’s Tau

Feature Importance

- Often we want to understand which features are most predictive
- Three Basic Techniques
  - Inspect Model Directly (Not always possible)
  - Leave-one-out analysis (Train on all features but one)
  - Singleton feature analysis (Train on each feature individually)

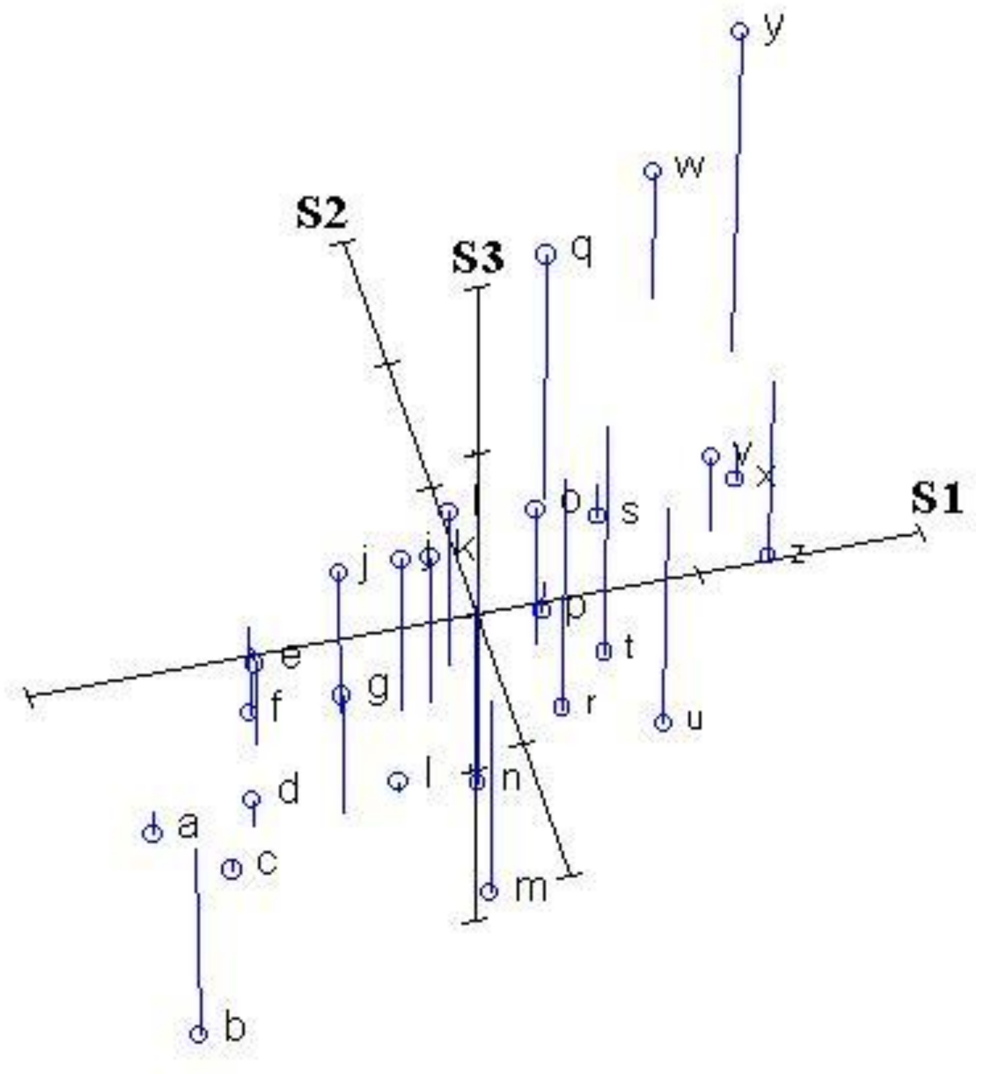
Feature Importance Example



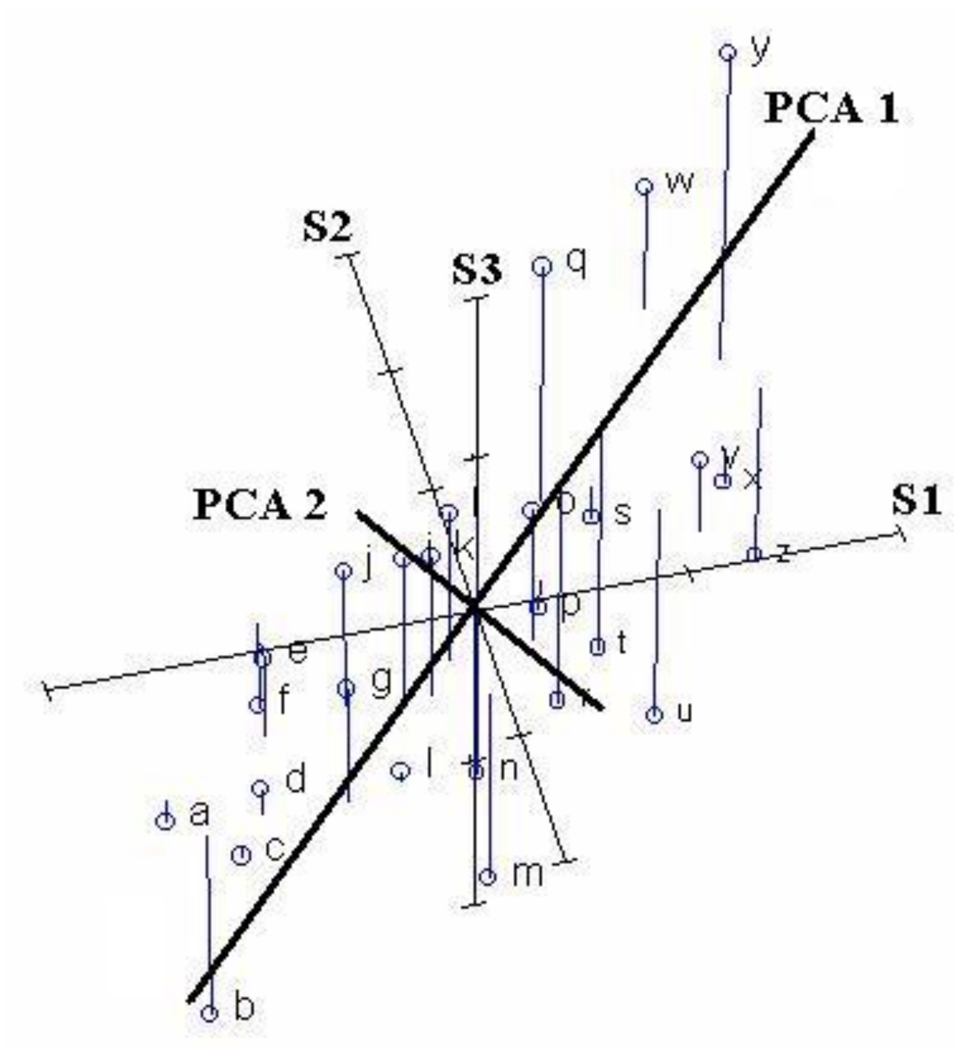
When Features Overlap

- Sometimes we want to know how many features are actually needed
- Example:
  - Feature 1: Lines of code
  - Feature 2: $\text{Lines of code} \cdot 2$
- Solution: Principle Component Analysis (PCA)
- Idea: Iteratively perform linear regressions, find out how many vectors (ideal features) are needed to account for variance in the data.

PCA



PCA



PCA

- Each PC reduces dimensionality such that the total linear variability is reduced
- Repeat until no dimensions left
- Can only account for *linear* variability

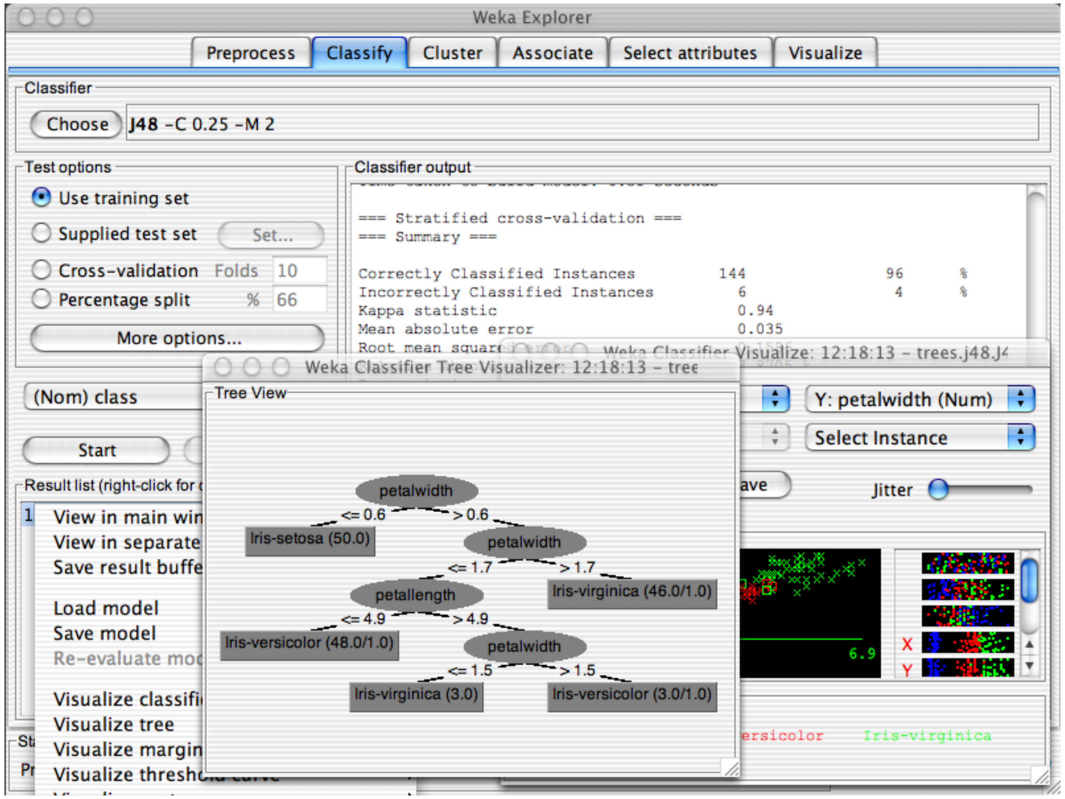
Other Concerns

- Degree of granularity in classifier output

Even More Practical

ML implementations available in the environment of your choice

- Matlab
- R
- Weka (Java)
- Mathmatica
- [Many others]



Classifier: J48 -C 0.25 -M 2

Test options: [✓ Use training set] Supplied test set Cross-validation Folds 10 Percentage split % 66

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        144               96      %
Incorrectly Classified Instances        6                4      %
Kappa statistic                         0.94
Mean absolute error                     0.035
Root mean square
```

The End?