About this class

The problem of overfitting and how to deal with it

Modifying logistic regression training to avoid overfitting

Evaluating classifiers. Accuracy, precision, recall, ROC curves

Overfitting

Many hypotheses consistent with/close to the data

With enough features and a rich enough hypothesis space, it becomes easy to find meaningless regularity in the data

Day/Month/Rain may give you a function that exactly matches the outcomes of dice rolls, but would that function be a good predictor of future dice rolls?

Linear regression vs. polynomial example

How do you decide on a particular preference?

Simpler functions vs. more complex ones. But how do we define complexity in all cases?

Simpler polynomial: lower degree Simpler decision tree: less depth Simpler linear function: lower weights?

Have to make a tradeoff between fit to training data and complexity. Sometimes we will see that we can make this mathematically explicit

Other possibilities: statistical significance, simulating unseen test data

Regularization

Prevent overfitting by creating a function that you are trying to maximize (on the training data) that explicitly penalizes model complexity

We'll see a number of different examples, but let's examine regularization in the context of logistic regression (Section 3.3 of the Mitchell chapter)

Evaluating Accuracy: Random Training/Test Splits

Divide dataset into **training set** and **test set**

Apply learning algorithm to training set, generating hypothesis h

Classify examples in test set using h, and measure percentage of correct predictions made by h (accuracy)

Repeat a set number of times.

Repeat the whole thing for differently sized training and test sets, if you want to construct a learning curve...

How do you compute confidence intervals?

Central limit theorem and sampling distribution of the mean can help! 95% confidence interval given by mean $\pm 1.96\hat{\sigma}/\sqrt{n}$

Cross-Validation

Another possible method if you have two different candidate models

Attempt to estimate accuracy of the models on simulated "test" data

Standard approach: n-fold cross validation (very typical: $n = 10$). Divide the data into n equally sized sets. Train on $n - 1$ of them and test on the nth. Repeat for all n folds

Is the accuracy of the better one then a good estimate of expected accuracy on unseen test data?

If you tune your parameters in *any* way on training data (including for model selection), you must test on fresh test data to get a good estimate!

Leave-One-Out Cross-Validation

Just what it sounds like. Train on all examples except one and then test on that one example. Repeat for all examples in the training data

Most efficient use of available data in terms of getting an estimate of accuracy

Can be horribly computationally inefficient, unless you can figure out a smart way to retrain without throwing away everything when swapping in one example for another

Confusion Matrices, Precision, and Recall

Two kinds of errors: false positives and false negatives (can generalize this to k classes as "Predict class i, actually class j"

	Pred. Negative	Pred. Positive
Act. Negative	TN	FP
Act. Positive	FN	TP

Precision: Percentage of predicted positives that were actually positive – TP / (TP + FP) (also known as Specificity)

Recall: Percentage of actual positives that were predicted positive – TP / (TP + FN) (also known as Sensitivity)

Spam example: if Spam messages are considered "positives" then recall is how much of the

Spam you catch, and precision gives a measure of how often you will label a legitimate message as Spam.

Precision and Recall are usually traded off against each other. 100% recall can be achieved by predicting everything to be positive. Extremely high precision can be achieved by predicting only the examples you are most confident of to be positive.

Important in information retrieval. What would precision and recall be in terms of searching for information on the web?

ROC Curves

Let's generalize. For any predictor, for a *given* false positive rate, what will the true positive rate be?

How do we do this? Well, just think about ranking test examples by confidence and then taking cutoffs wherever we want to test.

If one classifier dominates another at every point on the ROC curve it is better

People often use the area under the curve (AUC) as a single summary statistic to measure performance of classifiers