Introduction to Pattern Recognition

Jason Corso

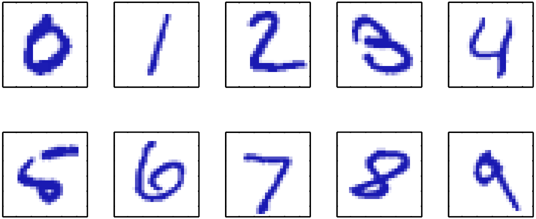
SUNY at Buffalo

17 January 2012

Flower Power
written by Brad Birch
storyboard by Craig Taillefer

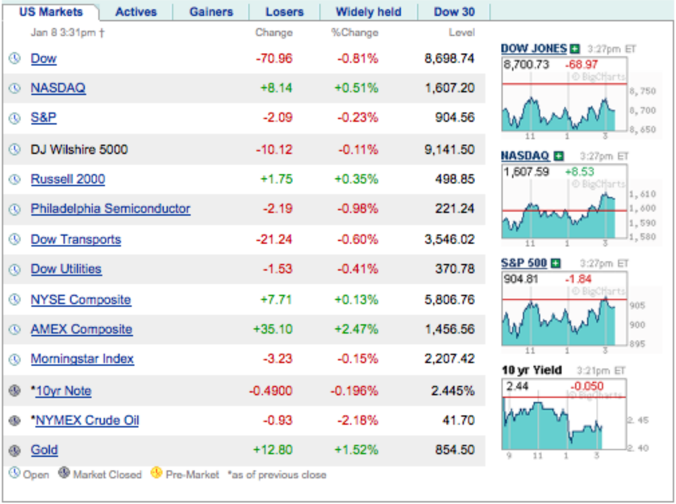
Examples of Pattern Recognition in the Real World

Hand-Written Digit Recognition



Examples of Pattern Recognition in the Real World

Computational Finance and the Stock Market



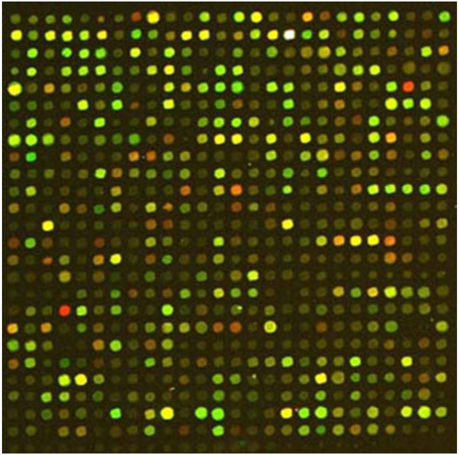
US Markets | Actives | Gainers | Losers | Widely held | Dow 30

Jan 8 3:31pm †	Change	%Change	Level
Dow	-70.96	-0.81%	8,698.74
NASDAQ	+8.14	+0.51%	1,607.20
S&P	-2.09	-0.23%	904.56
DJ Wilshire 5000	-10.12	-0.11%	9,141.50
Russell 2000	+1.75	+0.35%	498.85
Philadelphia Semiconductor	-2.19	-0.98%	221.24
Dow Transports	-21.24	-0.60%	3,546.02
Dow Utilities	-1.53	-0.41%	370.78
NYSE Composite	+7.71	+0.13%	5,806.76
AMEX Composite	+35.10	+2.47%	1,456.56
Morningstar Index	-3.23	-0.15%	2,207.42
*10yr Note	-0.4900	-0.196%	2.445%
*NYMEX Crude Oil	-0.93	-2.18%	41.70
Gold	+12.80	+1.52%	854.50

Open Market Closed Pre-Market *as of previous close

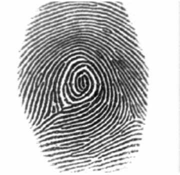
Examples of Pattern Recognition in the Real World

Bioinformatics and Gene Expression Analysis

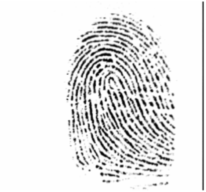


Examples of Pattern Recognition in the Real World

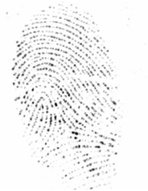
Biometrics



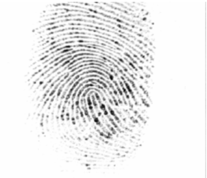
High contrast print



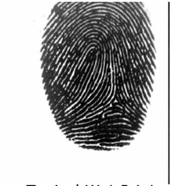
Typical dry print



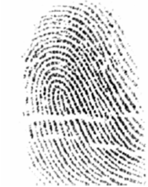
Faint print



Low contrast print



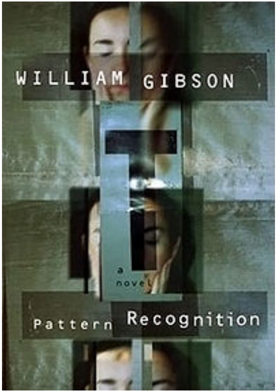
Typical Wet Print



Creases

Examples of Pattern Recognition in the Real World

It is also a Novel by William Gibson!



Do let me know if you want to borrow it!

Example: Sorting Fish

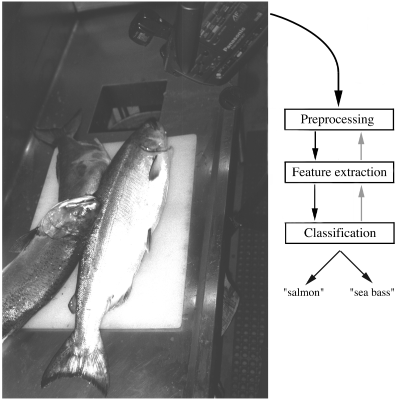
Salmon

Sea Bass

Example: Sorting Fish

Pattern Recognition System Requirements

- Set up a camera to watch the fish coming through on the conveyor belt.
- Classify each fish as salmon or sea bass.
- Prefer to mistake sea bass for salmon.

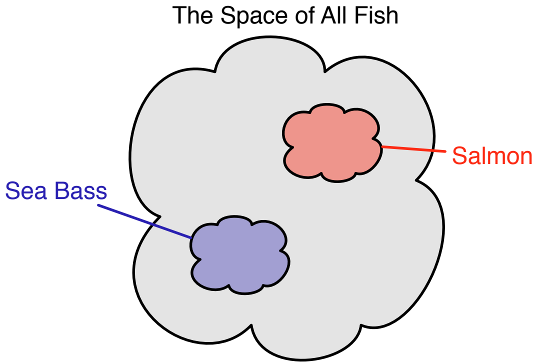


A Note On Preprocessing

- Inevitably, preprocessing will be necessary.
- *Preprocessing* is the act of modifying the input data to simplify subsequent operations without losing relevant information.
- Examples of preprocessing (for varying types of data):
  - Noise removal.
  - Element segmentation;
    - Spatial.
    - Temporal.
  - Alignment or registration of the query to a canonical frame.
  - Fixed transformation of the data:
    - Change color space (image specific).
    - Wavelet decomposition.
  - Transformation from denumerable representation (e.g., text) to a 1-of-B vector space.
- **Preprocessing** is a key part of our Pattern Recognition toolbox, but we will talk about it directly very little in this course.

Patterns and Models

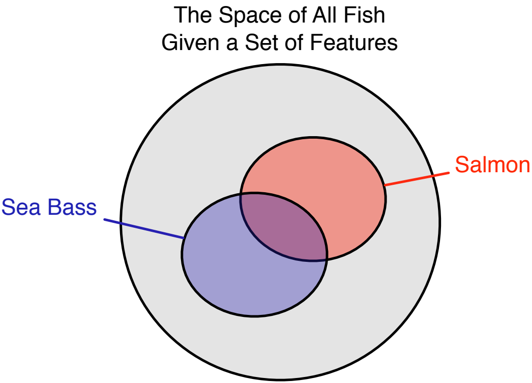
Ideal State Space



- Clear that the populations of salmon and sea bass are indeed distinct.
- The *space of all fish* is quite large. Each dimension is defined by some property of the fish, most of which we cannot even measure with the camera.

Patterns and Models

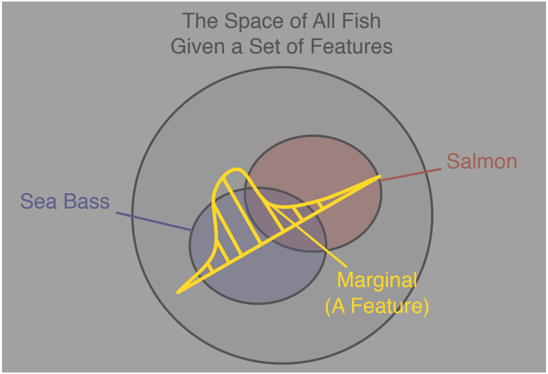
Real State Space



- When we choose a set of possible features, we are projecting this very high dimension space down into a lower dimension space.

Patterns and Models

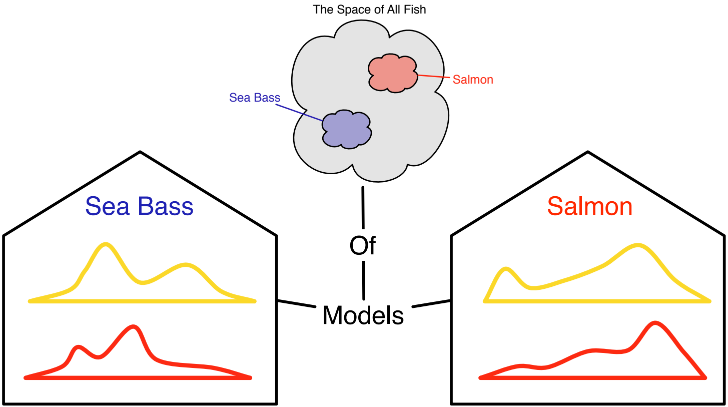
Features as Marginals



- And indeed, we can think of each individual feature as a single marginal distribution over the space.
- In other words, a projection down into a single dimension space.

Patterns and Models

Models



- We build a model of each phenomenon we want to classify, which is an approximate representation given the features we've selected.

Patterns and Models

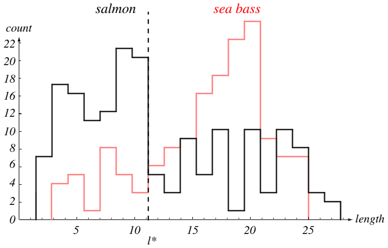
Models

The overarching goal and approach in pattern classification is to hypothesize the class of these models, process the sensed data to eliminate noise (not due to the models), and for any sensed pattern choose the model that corresponds best. -DHS

Selecting Feature(s) for the Fish

- Suppose an expert at the fish packing plant tells us that length is *the best* feature.
- We **cautiously trust** this expert. Gather a few examples from our installation to analyze the length feature.
  - These examples are our **training set**.
  - Want to be sure to gather a representative population of them.
  - We analyze the length feature by building histograms: **marginal distributions**.

Histogram of the Length Feature

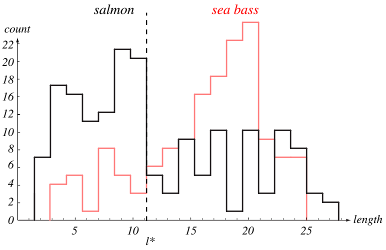


Selecting Feature(s) for the Fish

- Suppose an expert at the fish packing plant tells us that length is *the best* feature.

- We **cautiously trust** this expert. Gather a few examples from our installation to analyze the length feature.
  - These examples are our **training set**.
  - Want to be sure to gather a representative population of them.
  - We analyze the length feature by building histograms: **marginal distributions**.

Histogram of the Length Feature

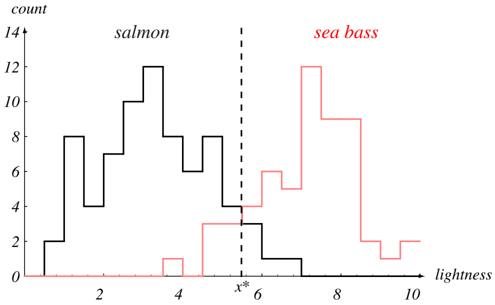


- But this is a disappointing result. The sea bass length does exceed the salmon length on average, but clearly not always.

Selecting Feature(s) for the Fish

Lightness Feature

- Try another feature after inspecting the data: **lightness**.

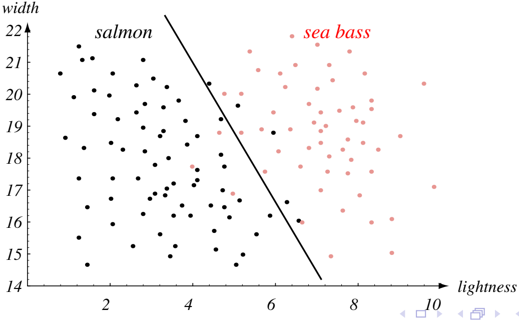


- This feature exhibits a much better separation between the two classes.

Feature Combination

- Seldom will one feature be enough in practice.
- In the fish example, perhaps lightness, x_1, and width, x_2, will jointly do better than any alone.
- This is an example of a 2D feature space:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad . \tag{1}$$



Curse Of Dimensionality

- The two features obviously separate the classes much better than one alone.
- This suggests adding a third feature. And a fourth feature. And so on.
- Key questions

Curse Of Dimensionality

- The two features obviously separate the classes much better than one alone.
- This suggests adding a third feature. And a fourth feature. And so on.
- Key questions
  - How many features are required?

Curse Of Dimensionality

- The two features obviously separate the classes much better than one alone.
- This suggests adding a third feature. And a fourth feature. And so on.
- Key questions
  - How many features are required?
  - Is there a point where we have **too many** features?

Curse Of Dimensionality

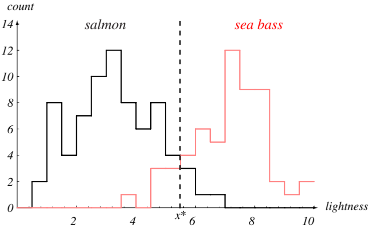
- The two features obviously separate the classes much better than one alone.
- This suggests adding a third feature. And a fourth feature. And so on.
- Key questions
  - How many features are required?
  - Is there a point where we have **too many** features?
  - How do we know beforehand which features will work best?

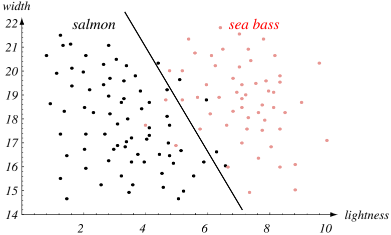
Curse Of Dimensionality

- The two features obviously separate the classes much better than one alone.
- This suggests adding a third feature. And a fourth feature. And so on.
- Key questions
  - How many features are required?
  - Is there a point where we have **too many** features?
  - How do we know beforehand which features will work best?
  - What happens when there is feature redundance/correlation?

Decision Boundary

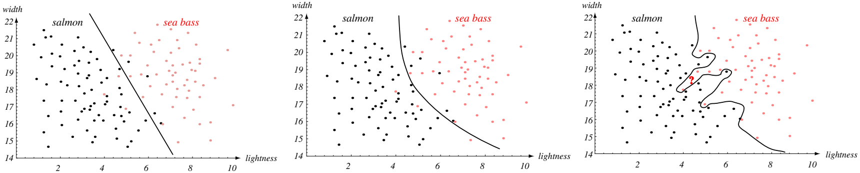
- The **decision boundary** is the sub-space in which classification among multiple possible outcomes is equal. Off the decision boundary, all classification is unambiguous.





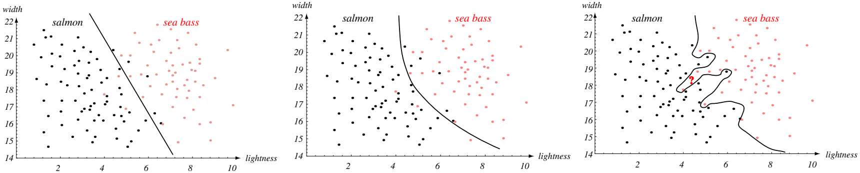
Bias-Variance Dilemma

- Depending on the available features, complexity of the problem and classifier, the decision boundaries will also vary in complexity.



Bias-Variance Dilemma

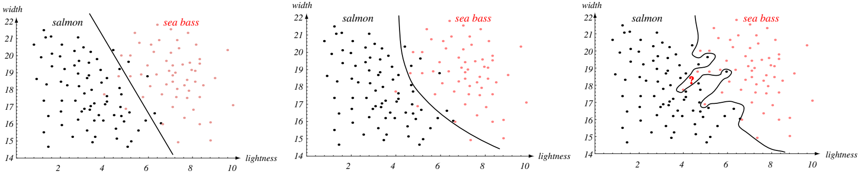
- Depending on the available features, complexity of the problem and classifier, the decision boundaries will also vary in complexity.



- Simple decision boundaries (e.g., linear) seem to miss some obvious trends in the data — **variance**.

Bias-Variance Dilemma

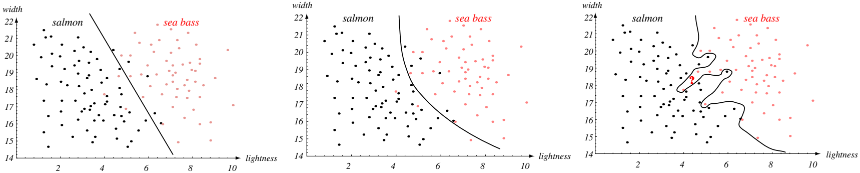
- Depending on the available features, complexity of the problem and classifier, the decision boundaries will also vary in complexity.



- Simple decision boundaries (e.g., linear) seem to miss some obvious trends in the data — **variance**.
- Complex decision boundaries seem to lock onto the idiosyncracies of the training data set — **bias**.

Bias-Variance Dilemma

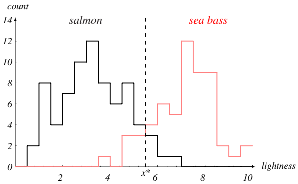
- Depending on the available features, complexity of the problem and classifier, the decision boundaries will also vary in complexity.



- Simple decision boundaries (e.g., linear) seem to miss some obvious trends in the data — **variance**.
- Complex decision boundaries seem to lock onto the idiosyncracies of the training data set — **bias**.
- A central issue in pattern recognition is to build classifiers that can work properly on novel query data. Hence, **generalization** is key.
- Can we predict how well our classifier will generalize to novel data?

Decision Theory

- In many situations, the consequences of our classifications are not equally costly.
- Recalling the fish example, it is acceptable to have tasty pieces of salmon in cans labeled sea bass. But, the converse is not so.
- Hence, we need to adjust our decisions (decision boundaries) to incorporate these varying costs.
- For the lightness feature on the fish, we would want to move the boundary to smaller values of lightness.



- Our underlying goal is to establish a decision boundary to minimize the overall cost; this is called **decision theory**.

Pattern Recognition

- First in-class quiz: can you define Pattern Recognition?

Pattern Recognition

- First in-class quiz: can you define Pattern Recognition?
- DHS: Pattern recognition is the act of taking in raw data and taking an action based on the "category" of the pattern.
- DHS: Pattern classification is to take in raw data, eliminate noise, and process it to select the most likely model that it represents.
- Jordan: The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying data into different categories.

Types of Pattern Recognition Approaches

- Statistical
  - Focus on statistics of the patterns.
  - The primary emphasis of our course.
- Syntactic
  - Classifiers are defined using a set of logical rules.
  - Grammars can group rules.

Feature Extraction and Classification

- **Feature Extraction** — to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories.
    - Invariant features—those that are invariant to irrelevant transformations of the underlying data—are preferred.
- **Classification** — to assign an category to the object based on the feature vector provided during feature extraction.

Feature Extraction and Classification

- **Feature Extraction** — to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories.
    - Invariant features—those that are invariant to irrelevant transformations of the underlying data—are preferred.
- **Classification** — to assign an category to the object based on the feature vector provided during feature extraction.
- The perfect feature extractor would yield a representation that is trivial to classify.
- The perfect classifier would yield a perfect model from an arbitrary set of features.
- But, these are seldom plausible.

Feature Extraction and Classification

Classification Objective Functions

- For classification, there are numerous underlying objective functions that we can seek to optimize.
- **Minimum-Error-Rate** classification seeks to minimize the the error rate: the percentage of new patterns assigned to the wrong category.
- **Total Expected Cost**, or **Risk** minimization is also often used.
- Important underlying questions are
  - How do we map knowledge about costs to best affect our classification decision?
  - Can we estimate the total risk and hence know if our classifier is acceptable even before we deploy it?
  - Can we bound the risk?

No Free Lunch Theorem

- A question you're probably asking is **What is the best classifier?**
- Any ideas?

No Free Lunch Theorem

- A question you're probably asking is **What is the best classifier?**
- Any ideas?
- We will learn that indeed no such generally **best** classifier exists.
- This is described in the **No Free Lunch Theorem**.
  - If the goal is to obtain good generalization performance, there are no context-independent or usage-independent reasons to favor one learning or classification method over another.
  - When confronting a new pattern recognition problem, appreciation of this thereom reminds us to focus on the aspects that matter most—prior information, data distribution, amount of training data, and cost or reward function.

Analysis By Synthesis

- The availability of large collections of data on which to base our pattern recognition models is important.
- In the case of little data (and sometimes even in the case of much data), we can use **analysis by synthesis** to test our models.
- Given a model, we can randomly sample examples from it to analyze how close they are to
    - our few examples and
    - what we expect to see based on our knowledge of the problem.

Classifier Ensembles

- Classifier combination is obvious – get the power of multiple models for a single decision.
- But, what happens when the different classifiers disagree?
- How do we separate the available training data for each classifier?
- Should the classifiers be learned jointly or in silos?
- Examples
  - Bagging
  - Boosting
  - Neural Networks (?)

SO MANY QUESTIONS...

Schedule of Topics

1. Introduction to Pattern Recognition
2. Tree Classifiers *Getting our feet wet with real classifiers*
    1. Decision Trees: CART, C4.5, ID3.
    2. Random Forests
3. Bayesian Decision Theory *Grounding our inquiry*
4. Linear Discriminants *Discriminative Classifiers: the Decision Boundary*
    1. Separability
    2. Perceptrons
    3. Support Vector Machines
5. Parametric Techniques *Generative Methods grounded in Bayesian Decision Theory*
    1. Maximum Likelihood Estimation
    2. Bayesian Parameter Estimation
    3. Sufficient Statistics

6. Non-Parametric Techniques
    1. Kernel Density Estimators
    2. Parzen Window
    3. Nearest Neighbor Methods
7. Unsupervised Methods *Exploring the Data for Latent Structure*
    1. Component Analysis and Dimension Reduction
        1. The Curse of Dimensionality
        2. Principal Component Analysis
        3. Fisher Linear Discriminant
        4. Locally Linear Embedding
    2. Clustering
        1. K-Means
        2. Expectation Maximization
        3. Mean Shift
8. Classifier Ensembles (Bagging and Boosting)
    1. Bagging
    2. Boosting / AdaBoost

9. Graphical Models *The Modern Language of Pattern Recognition and Machine Learning*
    1. Introductory ideas and relation back to earlier topics
    2. Bayesian Networks
    3. Sequential Models
        1. State-Space Models
        2. Hidden Markov Models
        3. Dynamic Bayesian Networks
10. Algorithm Independent Topics *Theoretical Treatments in the Context of Learned Tools*
    1. No Free Lunch Theorem
    2. Ugly Duckling Theorem
    3. Bias-Variance Dilemma
    4. Jacknife and Bootstrap Methods
11. Other Items Time Permitting
    1. Syntactic Methods
    2. Neural Networks

Python

- A big change for this year. . . Everything we implement will be in Python/NumPy/SciPy
- Many suggestions from past students to give more in-class and concrete examples.

Python

- A big change for this year... Everything we implement will be in Python/NumPy/SciPy
- Many suggestions from past students to give more in-class and concrete examples.
- Why Python?
    1. Matlab is the language of the PR/ML/CVIP realm. You will get exposed to it outside of this course...
    2. Python is maturing and becoming increasingly popular for projects both within PR/ML/CVIP and beyond. So, I want to expose you to this alternate reality.
    3. Preparation with Python in 555 may be more useful to a graduate in the job-hunt than some of the 555 material itself, e.g. Google does a lot with Python.
    4. Python is free as in beer.
    5. Some of the constructs in Python are easier to work with than other high-level languages, such as Matlab or Perl.
    6. Python is cross-platform.
    7. Numpy and Scipy are available.

Python

- Introduction to Python Slides (from inventor of Python)
- Introduction to NumPy/SciPy
  - http://www.scipy.org/Getting_Started
  - http://www.scipy.org/NumPy_for_Matlab_Users
- We will use the Enthought Python Distribution as our primary distribution (version 7.2.1).
  - http://enthought.com/products/epd.php
  - Available on the CSE network. https://wiki.cse.buffalo.edu/services/content/enthought-python-distribution
  - Python 2.7
  - Packages up everything we need into one simple, cross-platform package.

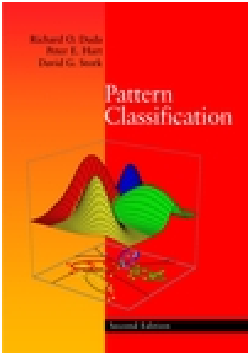
Python

- Introduction to Python Slides (from inventor of Python)
- Introduction to NumPy/SciPy
  - `http://www.scipy.org/Getting_Started`
  - `http://www.scipy.org/NumPy_for_Matlab_Users`
- We will use the Enthought Python Distribution as our primary distribution (version 7.2.1).
  - `http://enthought.com/products/epd.php`
  - Available on the CSE network. `https://wiki.cse.buffalo.edu/services/content/enthought-python-distribution`
  - Python 2.7
  - Packages up everything we need into one simple, cross-platform package.
- You will need to be Python-capable very soon. . .

Logistical Things

- Read the course webpage (now and regularly): `http://www.cse.buffalo.edu/~jcorso/t/CSE555`
- Read the syllabus: `http://www.cse.buffalo.edu/~jcorso/t/CSE555/files/syllabus.pdf`
- Read the newsgroup: `sunyab.cse.555`

THE BOOK



Logistical Things

Policy on reading and lecture notes

Lecture notes are provided (mostly) via pdf linked from the course website.
For lectures that are given primarily on the board, no notes are provided.
It is always in your best interest to attend the lectures rather than exclusively read the book and notes. The notes are provided for reference.
In the interest of the environment, I request that you do NOT print out the lecture notes.
The lecture notes linked from the website may be updated time to time based on the lecture progress, questions, and errors. Check back regularly.