Data-Mining: Motivation

- Humans generate increasingly huge amounts of data
- Sources of data: social media, server logs, point-of-sale terminals, medical records, etc.
- Potentially useful resource
- Caveat: large amount of raw-data is of little value without some automated techniques to extract information from it
- Distinction between *data* and *knowledge/information*

Data-Mining: Definition

- Extracting previously unknown and useful information from a corpus of data
- Accomplished by creating computer programs that can discover patterns and regularities in the data
- Problems: patterns may be uninteresting or spurious (artifact of the particular dataset), missing or corrupt values

Relationship to other fields

- Data mining is closely related to a number of other fields including...
- Statistics
- Machine Learning
- Detection and Estimation Theory
- Signals and Systems
- *Difficult to draw a precise distinction between these areas*

Distinguishing Features of Data-Mining

- High volume data
- Primarily *unsupervised* machine learning problems
- Concerned with how our solution will scale and genuinely seeking to discover new knowledge (hence *mining*)

Machine Learning

“How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?” Tom Mitchell

- Example: Classification, assignment of correct labels to previously unseen data

Classification and Standard Notation

- Classification is ultimately about discovering some function $f(x)$ that maps observations to classes
- What is the precise nature of the argument function and how is the data organized?
- Data instances: record of a database, row of a matrix
- Features: fields in a record, columns of a matrix
- Matrix view: row-dimension is the number of observations and the column dimension is the number of features

Supervised versus Unsupervised Learning Problems

- A dataset may or may not include labels that tell us what categories or classes our observations belong to
- *Supervised Learning*: typical problems are classification and regression
- *Unsupervised Learning*: typical problems are clustering or learning association rules

What is the *process* for approaching a data mining or data science problem?

- One process due to H. Mason and C. Wiggins is called OSEMN (pronounced "awesome") which is an mnemonic for *Obtain, Scrub, Explore, Model, iNterpret* (see Janssens, Jeroen. "Data Science at the Command Line." (2014))
- Organizes the activities of data science into a roughly serial process
- Scrub refers to *Data Cleaning* where data is prepared for automated techniques (taking care of NaNs, accounting for missing values, standardizing some records, selecting initial features)

Technology

- **Programming Languages**
- Python with some standard libraries for numerical computations, plotting, etc: numpy, scipy, matplotlib
- Javascript with d3 and jquery for visualization
- **Other Tools**
- Apache Spark
- Amazon EC2 (??)
- `curl` and REST Apis for data collection

Other Resources

- Machine Learning Repository at UCI
  http://archive.ics.uci.edu/ml/, currently warehouses 307 datasets in a variety of domains
- IEEE Transactions: *Pattern Analysis and Machine Intelligence*, *Knowledge and Data Engineering*, *Signal Processing*

What is Probability?

- Probability that A is true is denoted $P(A)$ generally
- Two views of probability *classical* view and the *frequentist* view

$$P_{\text{Classical}}(A) = \frac{N_A}{N} \tag{1}$$

$$P_{\text{Frequentist}}(A) = \lim_{n \to \infty} \frac{n_A}{n} \tag{2}$$

Axioms of Probability

- Implicit in probability the notion of a sample space, all possible outcomes
- Integrating P over the sample space results in 1
- $P(\top) = 1$
- $P(\perp) = 0$
- $0 \leq P(A) \leq 1$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- $P(A \vee \neg A) = 1$
- $P(A \wedge \neg A) = 0$

Conditional Probability

- The probability that A is true given that we know B is true
- $P(A|B) = \frac{P(A \wedge B)}{P(B)}$
- $P(B|A) = \frac{P(A \wedge B)}{P(A)}$
- In the context of machine learning we should consider conditional probabilities relating to observations
- Conditional probabilities are sometimes called likelihoods

Bayes Law

- *Bayes Law* gives us a formula for reversing a conditional probability
- Observe that $P(A \wedge B)$ occurred in both conditional formulas from the previous slide

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{3}$$

Rare Disease Example

- Assume that in a total population of 10,000 people 1% are afflicted with a rare condition
- Test is available where 99 % of sick patients test positive for disease and 99 % of healthy patients test negative
- Question: What is the probability that a person is sick if they test positive?
- I.e. what is $P(\text{sick}|\text{tested positive})$

Rare Disease Example cont.

- Solution using Bayes Law

$$P(\text{sick}|\text{tested positive}) = \frac{P(\text{tested positive}|\text{sick})P(\text{sick})}{P(\text{tested positive})} \tag{4}$$

$$P(\text{sick}|\text{tested positive}) = \frac{(99/100)(1/100)}{99/10000 + 99/10000} = 0.5 \tag{5}$$

Naive Bayes Classifier

- These observations can be used to create a (supervised) classification algorithm
- Naive Bayes has been applied extensively in text classification in particular for spam filtering
- In a text classification problem the features might be the presence or absence of certain words in the document
- Notice that this makes the problem entirely *categorical*

Naive Bayes Classifier cont.

- In the Naive Bayes algorithm we have a certain number of classes denoted C_k and observations $\mathbf{x}$ of some categorical variable
- From the training data we can determine both the prior probabilities $P(C_k)$ and the likelihoods
- $P(\mathbf{x}|C_k)$ which factors as a result of the independence (naive) assumption
- Given new data we calculate the posteriori probability for each of the classes $P(C_k|\mathbf{x})$ and assign the observation to the class with the largest posterior, this is called the MAP rule

Bayes Law and Naive Bayes Classifier

- Basic idea of Bayes law is to invert conditional probabilities
- Previously we discussed a simple classification algorithm called Naive Bayes based on this
- Allowed us to predict the class of some data instance based upon previously observed *labeled* instances

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{6}$$

Naive Bayes and Independence

- We maximize the below expression over k and assign the corresponding class to the new data instance
- The terms on the RHS of the equation can all be easily estimated from prior observations
- The central assumption here is that the features are independent of one another so that the expression factors nicely

$$P(C_k|\mathbf{x}) = P(C_k)P(x_1|C_k)P(x_2|C_k)\dots P(x_n|C_k) \tag{7}$$

Feature Selection and Independence

- Feature Selection is a crucial step in the Data Mining process
- Statistical Independence turns out to be a desirable property of a decent feature set
- Eliminate redundancy in the information that we give to our classifier
- Parsimonious with our computing resources

$$P(B|A) = P(B) \tag{8}$$

$$P(B \wedge A) = P(B)P(A) \tag{9}$$

Random Variables

- We denoted by $P(A)$ the probability that some abstract event "A" is true
- In practical terms what is more useful to associate real numbers to the events in the sample space
- This results in the concept of a *random variable*
- A random variable is *discrete* if it can take countably many values and is *continuous* otherwise

Mass and Density Functions

- Random Variables are completely specified by their probability mass functions (discrete) or density functions (continuous)
- Discrete Case

$$p(x) = P(X = x) \tag{10}$$

- Continuous Case

$$P(a \leq X \leq b) = \int_a^b f_X(x)dx \tag{11}$$

$$p(x, y),\ f_{X_1, X_2, \ldots}(x_1, x_2, \ldots) \tag{12}$$

Statistical Independence from the Joint Density

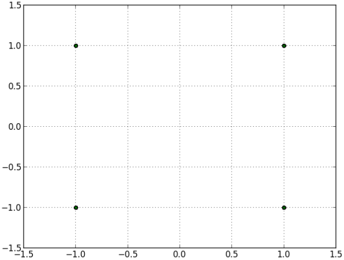
- The joint density is a complete statistical description of a set of random variables
- The condition for statistical independence is that the density or mass function factors

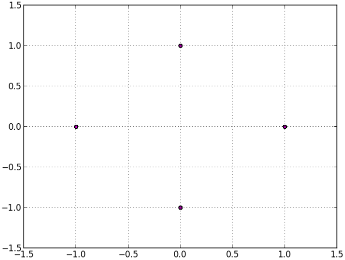
$$p(x, y) = p(x)p(y) \tag{13}$$

$$f_{X_1,X_2,\ldots}(x_1, x_2, \ldots) = f_{X_1}(x_1)f_{X_2}(x_2)\ldots \tag{14}$$

Marginal Distribution

- The marginal distribution of a random variable $f_X(x_i)$ is obtained by integrating or summing with respect to the other variables in some joint density
- Can you use the formulas we've discussed up to this point to determine the dependence of these distributions?





Contingency Table

- From the description in the previous slide we know that our variables are...
- discrete: take on countably many values
- $x, y \in -1, 0, 1$
- This means that our joint probability mass function will assign probabilities to the 9 possible pairs of values from the range
- $p(-1, -1), p(0, -1), (0, 1), \ldots, p(1, 1)$
- If we observe this from actual samples of a random variable then this is called a contingency table

Worked Example

- Take a moment to attempt to work this out...
- Think about what you need to do to test joint density for independence
- What are the values of the probability mass functions

Answer and Conclusion

- Answer: The density is independent in the first case and dependent in the second case
- What conclusions can we draw from this?
- Joint density is a *complete* description of some set of random variables, we can calculate anything we'd want to know from it
- In a practical setting we rarely have such a complete description, consider the complexity of describing the joint density as the number of arguments is increased
- Statistical Independence is an important property but it can be encoded in the data in non-obvious ways

Why is this important for Data-Mining?

- In Classification tasks we are attempting to learn some function $f(x)$ that is defined over the space of possible data instances
- Ideally, $f(x)$ maps the data instance to the correct class
- In a majority situations we don't arrive at a closed form solution for $f(x)$, it is optimized iteratively with respect to some criteria (objective function)
- Because there may be many local minima the algorithms that we use can be affected in sometimes unpredictable ways
- As a rule of thumb it is often desirable to eliminate redundancy in our feature set

Other Concepts of Dependency

- Linear Dependencies in random variables are easier to detect than general statistical independence
- For this we first need to talk about expectation $E[X]$
- $E[X]$ can be thought of in two ways
- The arithmetic mean of a large number of observations of X
- As a quantity defined from the probability distribution

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx \tag{15}$$

Defining Other Quantities

- Many statistics about a random variable are expressed in terms of expectations: moments, covariance, variance, correlation
- We can use the expectation for an arbitrary formula $g(X)$

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx \tag{16}$$

Measures of dispersion

- Dispersion measures how much a r.v. spreads away from its typical or *central* value
- The expectation is also called the *mean* μ and is a measure of central tendency
- The variance describes the width of the distribution around μ

$$\sigma^2 = E[(X - \mu)^2] \tag{17}$$

- In Data-Mining this quantity is useful because we may need to standardize or data or features before applying a learning algorithm, computing the variance allows us to *normalize* features

Measures of Dependence

- Using expectation we can also define quantities that characterize how likely two variables are to change together
- For this we use the *covariance*

$$\sigma_{xy}^2 = E[(X - \mu_x)(Y - \mu_y)] \tag{18}$$

- In the data mining context we will often want to compute the variance or standard deviation because we may want to *normalize* and center the values of all of our features prior to learning
- Cross-tabulation or contingency table is infeasible for a continuous r.v.

Verifying our Results with Code

- Without the ability to program Data Science is moot
- We need some environment where we can compute statistics, explore the data with simple plots or visualizations, apply different learning algorithms
- Typical solution for this would be either a scripting language or a domain specific language such as Matlab, Octave, Mathematica, R, or SPSS
- As mentioned in the previous class we will use Python for mainline work

Python for Data Analysis

- Python with one of its shell environments and some libraries basically replicate what is available from Matlab
- Advantages: general purpose language, not tied to proprietary environment, code will be more portable, numerous bindings API for other frameworks
- `numpy`: basic array data type
- `scipy`: scientific computing
- `matplotlib`: reproduces the style of plotting from Matlab, can be run interactively
- `pandas`: importing data

Code for the Example

- Importing packages and calling into them

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats

g = np.random.random_integers(0,3,100)
```

Working with arrays

- We also have more convenient Matlab style array indexing

```
s1 = np.array([[-1,1],[1,-1],[1,1],[-1,-1]])
s2 = np.array([[0,1],[0,-1],[1,0],[-1,0]])

x1 = s1[g,0]
y1 = s1[g,1]

H1,xedges,yedges = np.histogram2d(x1,y1,bins=[-1.5, 0., 1.5])

x2 = s2[g,0]
y2 = s2[g,1]

H2,xedges,yedges = np.histogram2d(x2,y2,bins=[-1.5,-0.5,0.5,1.5])
```

Accessing Library Routines

- These lines perform a χ^2 test on our contingency table data
- A χ^2 test is a goodness-of-fit test of data to probability distributions

```
chi2, p, dof, ex = scipy.stats.chi2_contingency(H1)
print p

chi2, p, dof, ex = scipy.stats.chi2_contingency(H2)
print p
```

Simple Visualizations

- `matplotlib.pyplot` gives us a state machine interface for creating graphics that allows us to manipulate markers, colors, add grids, ticks, label axes, and add titles
- also enable this to run interactively through the shell

```
plt.figure(1)
plt.scatter(x1,y1,20,'g')
plt.grid()

plt.figure(2)
plt.scatter(x2,y2,20,'m')
# plt.plot(g)
# plt.plot(x1,'k')
plt.grid()

plt.show()
```