CS 412

Aiden Sirotkine

Spring 2025

Contents

Chapter 1

CS 412

Silly data mining summer class on Coursera lmao.

Extract patterns from vast swaths of data.

Data mining happens after data analysis where you classify and cluster and find patterns.

1.1 Types of Data

There's a whole bunch of types

1.1.1 Important Characteristics of Structured Data

- Dimensionality
- Sparsity
- Resolution

- Distribution

1.1.2 Nominal

Qualitative names of things. Categorical

1.1.3 Binary

2 possible values

There are symmetric and asymmetric binary attribute. Symmetic binary attributes are attributes where both classes are equally important, like biological sex. Asymmetric attributes are those where one of the attributes is more important than the other, like having a certain disease.

1.1.4 Ordinal

Ordered, ranked

1.1.5 Others

Just know there are more and they are self explanatory

1.1.6 Numeric

These are numbers

They can be either interval scaled where you have a range and they can be in that range, so there is no true zero point, or they can be ratio-scaled, where there is a true zero-point at 0.

There is also disrete numeric data which is discrete. They can also be continuous.

1.2 Central Tendency

Mean, median, mode, midrange.

There are weighted means. Means are also very sensitive to outliers unlike the median or mode.

The median splits the two halves of the data.

The mode is whatever datapoint has the most repeated values.

1.2.1 Median

If the median falls in the mth bin, meaning between m and $m + 1$, then a prediction for the median is

$$median \approx L_m + \frac{n/2 - F_{m-1}}{f_m} \times (L_{m+1} - L_m)$$

That's a predictor for the median.

1.2.2 Mode

$$mean - mode = 3 \times (mean - median)$$

The difference between the mean and the mode is 3 times the difference between the mean and the median. This formula holds for slightly skewed distributions.

Data can be skewed which means the mode is to the left or ride of the mean by a lot.

negatively skewed means a tail to the left (decreasing value)

Distributions can be multi-modal.

Symmetric data means

$$mean = median = mode$$

1.3 Dispersion Measures

Variance, Standard Deviation, Covariance, Correlation Coefficient.

You know the equations for the first 3

$$r = \frac{Cov(a,b)}{\sigma_a \sigma_b}$$

1.3.1 Normal Distribution

They're pretty cool

65-95-99.7 rule or whatever it is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

It also mimics binary random variables taken to infinity.

1.4 Statistical Tests

Chi-Squared Test tells you if a dataset follows a certain distribution.

You need a test statistic and a significance level.

You check if the p-value is above or below the significance level.

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

1.4.1 Null Hypothesis

There is no relationship between 2 variables

1.4.2 Degrees of Freedom

(num of row - 1) * (number of columns - 1)
or
number of categories - 1
so flipping a coin has 1 degree of freedom beccause 2-1 = 1

1.5 Contingency Tables

For a table where you're finding the expected value of males and fiction, You take the total number of males and the total number of fiction and you multiply them together and then divide by the total number of people in the entire table.

1.6 Data Visualization

It just makes it easier to read and easier to see correlations.

1.6.1 Quantile

Points taken at regularly separated intervals.

Percentile is a Quantile split 100 ways

Quartile is 25th, 50th, 75th percentile for 1st, 2nd, 3rd

You know box and whiskers plots

an outlier is 1.5 * IQR

Histograms plot data frequency in bins of certain ranges.

Bar charts look like histograms but they plot categorical data.

1.6.2 Quantile Plots

You sort the data and then plot it so the x-axis is the index or percentile and the y-axis is whatever you're measure

1.6.3 QQ Plot

Scatterplot where you sort two things and put 1 variable on 1 axis and the other variable on the other axis.

Very similar to scatter plots, but scatter plots have 2 different variables on each axis, while QQ plots have the same variable, but different samples for each datapoint.

1.6.4 Pixel Visualization

It just lets you see higher dimensional data.

1.6.5 Geometric Projections

You can have 3d scatterplots or matrices or landscapes and stuff

not as important as like QQ plots.

Chapter 2

Week 2

2.0.1 Similarity Measure

0 if different, 1 if the same

2.0.2 Dissimilarity/Proximity Measure

0 distance means the same.

Distance measure

You can compute the distances between different data matrices.

2.1 Categorical Data

Names, not numbers

2.1.1 Similarity

see how many match and use that percentage.

2.1.2 Contingency table

Make a 2 by 2 matrix and increment each box depending on the match when counting out the data.

The distance measure for symmetric binary data is.

$$\begin{bmatrix} q & r \\ s & t \end{bmatrix} \qquad \frac{r+t}{q+r+s+t}$$

The diagonals are important

For asymmetric data, you just ignore the unimportant variable (buying nothing)

Given a data matrix, you can make a dissimilarity matrix.

2.1.3 Minkowski Distance

$$d = \sqrt[p]{|x_{i1} - x_{j1}|^p + \dots}$$

A special case is the manhattan distance (take just the component vectors and sum them. Taxicab distance)

The supremum distance is the largest distance between any 2 individual dimensions.

2.1.4 Standardization

Make all data have a mean of 0 and a standard deviation σ

$$z = \frac{x - \mu}{\sigma}$$

2.1.5 Mean Absolute Deviation

$$S_f = \frac{1}{n}(|x_{1f} - m_f| + \ldots)$$

2.2 Ordinal Data

Standardize with

$$z = \frac{r_i - 1}{r_{last} - 1}$$

The -1 makes sure the thing starts at 0

You can also use a weighted average

2.2.1 Cosine Similarity

Angle between 2 vectors

$$sim(d_1, d_2) = \cos(\theta) = \frac{d_1 \cdot d_2}{|d_1| * |d_2|}$$

2.3 Probability Data

2.3.1 KL Divergence

The difference between 2 probability distributions over the same variable X

Measures the information lost when using q to approximate p

$$D_{KL}(p(x)||q(x)) = \sum_{x \in X} p(x) \ln\left(\frac{p(x)}{q(x)}\right)$$

If q has a probability 0 for anything, just replace it with a very small ϵ and subtract ϵ/c from the c other probabilities.

2.3.2 Information Gain

Another way to think about it is the expected number of bits needed to sample from p starting at q

The amount of information gained from when one goes from prior probability q to posterior probability p

2.4 Data Cleaning

There are many tools to get rid of data discrepancies

2.4.1 Noisy Data

2.4.2 Binning

Sort data and put it into equal frequency bins and then turn each datapoint into the mean/median whatever of the bin.

2.4.3 Regression

Fit the datapoints along a curve

2.4.4 Clustering

Remove outliers

2.4.5 Semi-supervised

Use AI and a human to get rid of weird values.

2.4.6 Data Integration

Get the same data from multiple sources to try and get the most complete picture of the datapoints.

Clean the data with the mean/median/mode or the most recent data or something.

2.4.7 Redundancy

Don't count the same data multiple times.

2.5 Data Reduction

2.5.1 Linear Regression

Assume the data fits a model, find the parameters, get rid of the data.

2.5.2 Histograms, Clustering

You can also do a stratified sample which means you actively try to match the global probability distribution.

2.5.3 Data cube aggregation

You put all the data into a cube of nominal data and remove all unnecessary data.

2.5.4 Data compression

2.6 Data Transformation

Smoothing, Normalization, Discretization.

2.6.1 Min-Max Normalization

$$v' = \frac{v - min}{max - min}(newmax - newmin) + newmin$$

There's also z-score normalization

2.7 Dimensionality Reduction

When dimensions increase, the data becomes more and more sparse.

2^d possible combinations of d attributes

2.7.1 Supervised and Nonlinear

Feature selection and feature generation (make new features out of the existing data that might be useful)

2.7.2 Unsupervised and linear

PCA and feature extraction.

2.7.3 PCA

Take the k most important eigenvectors and eigenvalues and use just the most important eigenvectors to reconstruct/analyze the entire dataset.

Use the explained and cumulative variance to figure out how many eigenvectors to keep.

Chapter 3

Week 3

3.0.1 Supervised vs Unsupervised Learning

Supervised learning has labels already built in, while unsupervised learning clusters data based on patterns only.

3.0.2 Classification vs Regression

Classification guesses discrete or nominal labels.

Regression models continuous valued functions.

training set trains, validation set also trains, testing set is just for tests

3.1 Decision Trees

you can order data by just splitting points up based on their traits. Can handle all types of variables.

You do it recursively, splitting the data into the most meaningful groups.

NON-PARAMETRIC - no assumption of the data distribution.

unstable decision boundaries, and very sensitive to noise. Accuracy might not be perfect because of the simplicity. Perfect trees are NP hard and overfitting is common.

3.1.1 When to stop

all sorted, no attributes, or no datapoints

3.2 Splitting Measures

There's post pruning and pre-pruning and whatnot

there's information gain and entropy and stuff

3.2.1 Entropy

A measure of uncertaintly associated with a number

Entropy is always between 0 and 1

$$H(Y) = -\sum_{y \in Y} p_y \ln(p_y)$$

Higher entropy = higher uncertainty

You find the entropy for the whole system, and you change the decision tree based on if the entropy goes up or down for the whole system.

Y is the groups at each of the leaf nodes of the tree.

Information is the difference in the entropies before and after the split.

3.2.2 Conditional Entropy

You just look at the leaf nodes

$$H(Y|Patron) = \sum_{\text{leaf nodes after split}} p(x)H(Y|X = x)$$

You can have negative information gain but that means you do something stupid.

Information gain is biased towards attributes with a large number of values.

Use gain ratio instead

$$InformationGain = Entropy(pre) - Entropy(post)$$
$$GainRatio = \frac{Gain}{SplitInfo(A)}$$
$$SplitInfo(A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \ln\left(\frac{|D_j|}{|D|}\right)$$

Whatever has the biggest gain ratio can be used for our splitting measure.

3.2.3 Gini Index (Impurity)

The Lower the Better

Used in binary trees.

For a dataset D with n classes

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

Where p_j is the relative frequency of class j in D

For a split

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

And you just take the difference between the two.

Do some math and the Gini index is the probability of an error by random assignment.

Can be done with a multi-way split, but that wasnt its first use.

3.3 Comparisons

Information Gain: biased towards multivalued attributes

Gain Ratio: Tends to prefer unbalanced splits where 1 partition is much larger than the other

Gini Index: Biased to multi-valued attributes. Has difficulty when number of classes are large. favors equal sized partitions.

3.4 Lecture 10: Back to Decision Trees

3.4.1 Prepruning

Early stop. End when splits stop improving error

3.4.2 Post-pruning

Make the full minimized tree, then prune until cross-validation error is minimized.

Generally better then pre-pruning.

3.4.3 Random Forest

Make a whole bunch of decision trees and aggregate the predictions.

3.5 Baye's Theorem

Bayesian stats are based on belief because it usually is for events that cannot be repeated.

$$p(H|X) = \frac{p(X|H)p(H)}{p(X)} \propto p(X|H)p(H)$$

$p(X|H)$ is what we just observed, or the likelihood and $p(H)$ is the prior probability, and we are obtaining the posterior probability.

$p(X)$ is just the scaling factor.

Chapter 4

Week 4

4.1 Baye's Theorem with Multiple Hypotheses and Sequential Evidence

4.1.1 Sequential Evidence

$$p(H|X_1, X_2) = \frac{P(X_2|H, X_1)p(H|X_1)}{p(X_1, X_2)}$$

You just do bayes theorem twice I think

Assume X_1, X_2 are conditionally independent given H

$$p(H|X_1, X_2) = \frac{P(X_2|H)p(H|X_1)}{p(X_1, X_2)}$$

Then it becomes easier.

4.2 Naive Baye's Classifier

ASSUME FEATURES ARE CONDITIONALLY INDEPENDENT

Compute categorical features with frequency counts.

Continuous features likely use a Gaussian.

4.2.1 Sequential

You can use Baye's theorem to calculate the probability of an outcome given new information.

4.3 Linear Regression

Use math to map datapoints to a continuous formula.

Minimize the loss function to make a linear predictor

least-squares regression is the math used in lin alg. solves analytically.

4.4 Perceptron

Linear Classifier where you classify the top and the bottom of the classifier.

The line is adjusted if it misclassifies something

4.5 Logistic Regression

Uses a sigmoid function to classify between 2 classes.

If its below the sigmoid, its one class. Above is the other.

Returns a certain probability function that relates to the sigmoid.

$$\sigma = \frac{1}{1 + e^{-z}}$$

Minimize the negative log-likelihood.

4.5.1 Pros

Can handle many features. Fast, good. Robust, Interpretable.

Only works well if the decision boundary is linear.

4.6 Generative vs Discriminative Classifiers

Generative make a thing, Discriminative make a decision boundary and figure out everything else from there.

Chapter 5

Week 5

5.1 Model Evaluation

5.1.1 Confusion Matrix

It's a matrix of the number of actual traits in a dataset vs the number of predicted traits in a dataset, and it's used to examine how accurately a model models data.

true positive, false negative
false positive, true negative
Sensitivity = True Positive / Positive
Specificity = True Negative / Negative
Precision = True Positive / Predicted Positive
Recall = True Positive / Positive
Accuracy = True Positive + True Negative / All

5.1.2 F Measure

Gives weights to recall

$$F_s = \frac{(B^2 + 1)P \times R}{B^2 P + R}$$

B = 1 means equal weight

In some cases precision might be more or less useful than recall.

5.1.3 ROC Curves

True positive rate TP/P on the y-axis and false positive rate FP/N on the bottom.

Receiver Operating Characcteristic.

It shows the true positive rate over the false positive rates, and as most models increase their true positive rate, their false positive rate also increases because they're just saying everything is true.

You can use the ROC curves of different classifiers to see what works best.

The area under the curve represents the quality of the model.

5.2 Classifier Evaluation

5.2.1 Holdout Method

Holdout Method is just training set and a validation set.

5.2.2 Cross-validation

Cross-Validation is the same as the holdout method but you do it multiple times over so that every part of the set becomes a validation set at some point.

You split it into a bunch

5.2.3 Leave-One-Out

self-explanatory.

5.2.4 Bootstrap Method

sample tuples with replacement.

The most common is the 0.632 bootstrap, where that much ends up in the training set and the rest is the validation set.

5.2.5 Parameters and Statistical Tests

Accuracy, Speed, Robustness, Scalability, Interpretability.

You can do a t-test to compare models.

5.3 Lazy Learning

Stores training data and waits until given a test tuple.

less training time, more predicting time.

example: Nearest Neighbor.

You need proper indexing/decent algorithms because lazy learning can be computationally expensive.

Must commit to a single hypothesis for the whole dataset.

5.3.1 KNN

Look at the nearest points and do a majority vote and boom youve classified it.

5.4 Ensemble Methods

5.5 Bagging and Random Forest

EACH TREE HAS RANDOMLY SELECTED FEATURES.

Bagging is training a bajillion different models with different training sets and put em all together.

boot + agg(regating models) = bagging

Random forest is bagging + a decision tree. You have different features for some models and they're selected randomly.

The different models have different features.

5.6 Boosting

Sequentially put a datapoint through classifiers with each one mending the mistakes of the last.

5.6.1 Adaboost

Adaptive boosting.

Assign initial weights to a dataset, fix the incorrect classifications by changing the weights. Repeat until classifier is good.

5.6.2 Gradient Boosting

Incrementally add procedurally weaker classifiers to the model to decrease the loss function.

You need a differentiable loss function.

5.6.3 Regression Tree

decision tree but for continuous variables.

Chapter 6

Week 6

6.1 Class Imbalance

Alot of methods assume both classes have equal error cost. This however is not true for many real-life samples (rare disease diagnosis, credit card fraud.)

You can oversample from the minority and undersample from the majority.

6.1.1 Threshold Moving

Increase the chance of classification for the minority to decrease the chance of the false negatives (Better to be safe than sorry).

6.1.2 Class Weight Adjusting

Make false negatives more penalizing.

6.2 Bayesian Belief Network

Given a di-graph of dependent probabilities, find the total probability of an outcome given the data.

6.2.1 Training

If the structure is known, and the variables are observable, compute the Conditional Probability Table entries, and estimate the probabilities analytically.

If the structure is known, but the variables are hidden, use the training data to predict the unknown probabilities/parameters. Use gradient ascent to maximize probabilities.

6.2.2 Plate Notation

You index graphs that are all related and put them all into 1 square.

6.3 Support Vector Machines

You use a non-linear mapping to map your data into a higher dimension such that you only need a linear hyperplane to separate your classes.

You can also do the mapping using a kernel function.

6.3.1 IF-THEN Rules

Seeing what previous traits correlation to other classifications.

6.3.2 Size Ordering

Assign highest priority to triggering rules that have the "toughest" requirement (with the most attribute tests)

6.3.3 Class-Based Ordering

Decreasing order of prevalence or misclassification cost per class.

6.3.4 Rule-based Ordering

Rules are organized into one long priority list according to some measure or expert opinion.

6.3.5 Rule Induction

Rules are mutually exclusive and exhaustive which means no two rules are triggered by the same tuple, and all combinations of attributes have a rule.

Start with a list of 0 rules.
Add a rule
Remove the point classified by the rule.
repeat until terminating condition.

6.4 Pattern-Based Classification

Learn patterns

6.5 Semi-Supervised Learning

Use labeled AND unlabeled data to train a classifier.

6.5.1 Self-Training

Train with the labeled data. Predict the unlabeled data. The most confident predictions get added to the labeled data. Repeat.

6.5.2 Co-Training

Use two features with mutually independently features. Give the most confident predictions to THE OTHER List to not reinforce errors.

6.5.3 Active Learning

The learner queries a human expert to have only the most informative data for training. It specifically queries the least confident data in the set.

You can also use a learning curve

6.5.4 Transfer Learning

Use a separate classifier to help train with unlabeled data.

6.5.5 Weak Supervision

Use noisier less high quality data to train the data instead of our fancy labeled training data.

You just ask people or use wikipedia for your weak data.

Chapter 7

Week 7

Patterns are items or structures that occur frequently together.

7.1 Pattern

A set of items, subsequences, or substructures that occur frequently together in a dataset.

Pattern discovery is finding those patterns in a very large dataset.

7.1.1 Types of Pattern Discovery

- Sequential Patterns
- Classification
- Clustering

7.1.2 Transactional Database

A data that has transactions, and each transaction has a (sometimes ordered) set of items.

Example: grocery list.

a k-itemset is a set of k items that occur in a transaction and could be a pattern.

7.2 Support and confidence

The support is the number of transactions that contain the k-itemset.

The confidence in a pattern existing is the conditional probability that both X and Y are in a transaction given X is already in the transaction.

$$c = \frac{sup(X, Y)}{sup(X)}$$

Where the support is just the frequency of the pattern.

patterns can have minimum support labeled minsup.

You can also use a minimum confidence.

7.3 Closed Pattern

A closed pattern X is a pattern that does not have a larger encompassing pattern with the same support as X.

Closed patterns are lossless compression (This means if you know all the closed patterns, you can figure out all the possible other patterns in the dataset AND THEIR SUPPORT.).

7.3.1 Maximal Pattern

if X is frequent and there exist no super-patterns of X.

The differnce between maximal and closed is that closed patterns have no super-pattern with THE SAME SUPPORT, while maximal patterns have no frequent super-pattern at all. Closed patterns can have super-patterns AS LONG AS THE SUPPORT IS SMALLER.

Maximal patterns are LOSSY compression. We know it's frequent but we do not know the support of it.

7.4 Apriori Algorithm

Find sequences that are infrequent and get rid of them and then just make bigger and bigger patterns from there.

Make candidate patterns that might or might not be frequent.

Prune larger patterns if all their subsets are not also frequent.

- Find all frequent-1 itemsets (scan)
- Generate frequent 2-itemsets if all the subsets are also frequent

- scan for frequency
- find larger candidate itemsets from the subsets
- repeat

7.5 Improvements

7.5.1 Partitioning

Any item that is frequent in the total database must be frequent in at least 1 partition of the total database.

SPECIFICALLY, THE SUPPORT IS NOW A FRACTION BASED ON THE PARTITION SIZE.

7.5.2 Hashing

Put patterns together and then if the sum of the supports is below a certain threshold, then none of the items in that set of patterns can be frequent.

7.5.3 Pruning

pruning is removing candidates that dont have ALL subsets frequent.

7.6 Vertical Data Format

ECLAT = Equivalent CLass Transformation method.

For every item a, you can see which indexes contain a.

Usually it's the other way around (index 1 contains a, b, c or whatever).

It allows you to discover patterns in a new way.

Only keep track of the differences in Tids (transaction ids)

7.7 FPGrowth

Apriori is a breadth-first search of pattern discovery

FPGrowth is a depth-first search.

- find 1-frequent items
- sort them in descending order
- perform another scan to find ordered frequency items in the dataset.
- that's your data structure

There's a whole bunch of tree-traversal that I don't fully get.

You basically just recursively order the most frequent patterns in exceedingly deeper and deeper trees.

There are like conditional databases which are essentially all the things that exist assuming m exists for some m in the database.

- Find all the frequent 1-itemsets

- remove all the non-frequent items from each transaction and order them by support.
- Build the FP-tree by inserting transactions 1 by 1 and incrementing counts/adding branches as necessary.

7.8 Mining Closed Patterns (can probably ignore)

CLOSET+

7.8.1 Merging

If Y is in every transaction in X, then Y is merged with X.

Probably not important

Chapter 8

Week 8

8.1 Pattern Evaluation

Pattern mining will give you many patterns, but many are useless.

8.1.1 Interestingness Measures

Numbers that tell you how interesting a pattern is.

8.1.2 Objective vs Subjective

Objective can be calculated with a formula every time, while subjective measures are dependent on the database.

Objective = support, confidence, correlation, lift, chi-squared

Subjective measures are silly

8.1.3 Support-Confidence Limitations

Support and confidence are interestingness measures.

If your data is skewed, support and confidence might be high without any correlation.

8.1.4 Lift

Measure of correlation in 2 events.

Less than 1 is negatively correlated

Greater than 1 is positively correlated.

$$lift(B,C) = \frac{s(B \cup C)}{s(B) \times s(C)}$$

Where s is the support, which we know how to measure.

8.1.5 χ^2

Tells you if datapoints follow a distribution.

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

Both lift and χ^2 are good if you do NOT have a bunch of null datapoints.

8.2 Pattern Evaluation Part 2

8.3 Null-Invariant Measures

These measures are the same whether or not you count null transactions.

ALL RANK [0, 1]

8.3.1 AllConf(A, B)

$$\frac{s(A \cup B)}{\max(s(A), s(B))}$$

8.3.2 Jaccard(A, B)

$$\frac{s(A \cup B)}{s(A) + s(B) - s(A \cup B)}$$

8.3.3 Cosine(A, B)

$$\frac{s(A \cup B)}{\sqrt{s(A) \times s(B)}}$$

8.3.4 Kulcynski(A, B)

$$\frac{1}{2}\left(\frac{s(A \cup B)}{s(A)} + \frac{s(A \cup B)}{s(B)}\right)$$

8.3.5 MaxConf(A, B)

$$\max \frac{s(A \cup B)}{s(A)}, \frac{s(A \cup B)}{s(B)}$$

8.3.6 Differences

Kulc is the most steady, but it does not change as dynamically as the others, so it is best paired with a second measure.

Null invariant means that all transformations done with them are also null-invariant.

Null transactions do not affect the measure in the slightest.

8.3.7 Imbalance Ratio

$$\frac{|s(A) - s(B)|}{s(A) + s(B) - s(A \cup B)}$$

measures the imbalance of two itemsets A and B in rule implications.

This + Kulcynski works pretty well

8.4 Mining Diverse Patterns

There's a big ass lecture

You can partition your data before looking at patterns.

8.5 Multilevel Association Rules

You can partition your data in a hierarchy.

Lower level patterns naturally will have a lower support because they're subpatterns of bigger patterns.

How do we figure out the best minsup for all levels?

8.5.1 Multi-Dimensional Associations

You might be working with data from many different dimensions like age, weight, products, etc. and you need to figure out different support threshholds for each.

8.5.2 Static Discretization

Turn a range of data-points into 1 discretized value.

age = 18-20, 20-22, etc etc

8.5.3 Clustering

Connect age clusters together based on other values.

8.5.4 Z-Test

How to figure out if Quantitative patterns are actually interesting.

8.5.5 Shared Multilevel Mining

Use the lowest possible minsup of all the larger patterns.

8.5.6 Negative Patterns

There's a formula for both support and a null-invariant thing.

$$sup(A \cup B) << sup(A) \times sup(B)$$

If that is true, then A and B are negatively correlated.

This does NOT work well when there are many null transactions.

You can use the null-invarient method with Kulc ¡ some epsilon to figure out of your items are negatively correlated.

8.6 Mining Compressed Patterns

Closed patterns are not compressable, and max patterns lose too much data.

8.6.1 Pattern Distance Measuring

$$Dist(A, B) = 1 - \frac{|T(A) \cap T(B)}{|T(A) \cup T(B)}$$

Where $T(A)$ is the set of transaction id's that contain A.

8.6.2 Delta Cover

There's also a distance metric for patterns

delta clusters are patterns that can be made from each other and have a distance less than delta.

Cluster all patterns that are δ close to each other.

8.7 Constraint-Based Mining

8.7.1 Meta Rules

if constraint and constraint then pattern

find predicate patterns with min-sup

8.7.2 Anti-Monotonic

if an itemset violates a constraint C, then so does all of its supersets.

sum of prices $< k$ is anti-monotonic because prices are always positive.

8.7.3 Monotonic

if an itemset satisfies a constraint C, then so does all of its supersets.

sum $> k$ is monotonic because prices are always positive.

8.7.4 Data anti-monotonic

If a data entry does not satisfy a constraint C, then no superset containing that entry satisfies C.

min(X) $< k$ is data anti-monotonic because that datapoint will be the new smallest minimum.

8.7.5 Succinct

can a constraint c be made true by directly manipulating the data with a function.

8.7.6 Convertible

Can a set be converted to monotonic or anti-monotonic based on just ordering the dataset.

average $< k$ is convertible because if the data is in descending order, then the average will always be decreasing.

Chapter 9

Week 9

9.1 Sequence-Based Data Mining

Gapped vs non-gapped means the direction matters more vs like a stochastic pattern.

(ab) are in-ordered while ab(cd)e are ordered (except for c and d)

unordered items are listed alphabetically.

9.1.1 Generalized Sequential Patterns (GSP)

Apriori method of finding sequential patterns

- Find all length 1 frequent patterns
- calculate candidates with their subsets
- scan to find frequent 2-patterns
- repeat forever

9.1.2 How to test Joining

Sequence joining and item joining (if the last item is part of an unordered set)

9.1.3 SPADE

Sequential Pattern mining in Vertical Data Format

for each element (a), you get both a sequence ID and and element ID.

The way you find the patterns is by putting together incrementing element ID's

if (a) has EID 1 and (b) has EID 2, then you can put together the sequence (a, b)

You do the same thing starting from 1 and going upwards with the vertical table.

9.1.4 PreFix Span

Pattern Growth Approach

Prefixes are patterns at the beginning of a datapoint.

- Find all frequent 1-patterns
- Find the prefix projection for each pattern
- find the frequent 1-patterns inside that projected database
- make the prefix projection database of the 2-pattern
- repeat forever

9.1.5 CloSpan

Closed Sequential Pattern: Just a regular closed pattern but in sequential data.

9.1.6 Graph Pattern Data Mining

Graphs are the things you use in CS225

you have to look at the full graph patterns instead of just the unions of different frequent edges.

9.1.7 Support

If the whole graph is a subset of the datapoint, then increment the support.

9.1.8 Apriori

Use anti-monotonicity.

A graph is frequent only if all of its subgraphs are frequent.

Start with small graphs and build up.

9.1.9 gSpan (Pattern Growth)

graph span. Pattern Growth approach.

Do a depth-first search.

Start with a k-edge graph, then add a edge, check if it is frequent, and immediately add another edge if it is frequents.

Order the graph so that it's a sequence.

9.1.10 CloseGraph

A n-sized graph may have 2^n subgraphs.

only mine closed frequent graphs.

Used your mining closed graph patterns.

An n-edge frequent graph may have 2^n subgraphs.

Finding only closed frequent graphs removes this explosion problem.

Lossless compression.

Done by extending gspan.

If G_1 is always frequent when G is frequent, you can ignore G and only consider the larger graphs of G_1.

Chapter 10

Week 10

10.1 Cluster Analysis

A cluster is a collection of data objects that are similar to each other.

Objects can be classified based on just similarity.

As similar as possible in the same groups. As far as possible to other groups.

Cluster classification is a form of unsupervised learning. This is because there is no training prior to cluster classification.

What quantitative datapoints should you use to classify your dataset?

10.1.1 Considerations

- Partitioning Criteria
- Separation of Clusters

- Similarity Measure
- Clustering Space (high-dimensional vs low-dim)
- Quality (Qualititative, Numeric, Hierarchical, Ordered)
- Scalability
- Constraints
- Interpretability

10.2 Partitioning Algorithms

10.2.1 K-Partitioning

There are K clusters and you split the data into that many clusters, trying to make them as even and correct as possible.

Specifically, you are minimizing some loss function accross all the of the clusters.

10.2.2 K-Means Clustering

goated unsupervised learning method.

Each cluster is represented by the center of the points and we minimize the mean distance between the cluster and all of its points.

sensitive to noisy data and outliers.

Best used for circular-ish shapes.

Not suitable for non-convex shapes.

10.2.3 K-Medoids

Instead of using the mean, you use the most centrally located point in the cluster (the point that's closest to the mean).

Not as affected by outliers.

Use random swapping in your medoid algorithm. Because of this, it does not scale well, but is good for small datasets.

10.2.4 K-Medians

Less sensitive to outliers.

10.2.5 K-Modes

Useful for qualitative data.

dissimilarity measure is frequency-based.

10.2.6 Kernel K-Means

Use a kernel function to make more distinct clusters.

Turn the space into a higher-dimensional one such that your points are more easily separable.

10.3 Hierarchical Clustering

You make clusters within clusters.

10.3.1 Single Link Clustering

Use the nearest possible objects within the cluster to calculate distance.

This allows you to have non-circular shapes.

10.3.2 Complete Link Clusterings

Use the farthest possible objects to calculate distance.

10.3.3 Average Link

Use the mean.

10.3.4 Centroid Link

Use the medoid.

10.3.5 Agglomerative

Start with smaller clusters and put them together until you have the whole dataset.

10.3.6 Divisive Clustering

Start with a giant cluster and split until you get to single clusters.

Inverse of agglomerative clustering model (AGNES).

10.4 Probabilistic Clustering Methods

Model the data from a generative process. Turn the datapoints into stats.

Assume the data is generated by some underlying probability distributions.

Optimize the fit between the observed data and some mathematical model.

10.4.1 Parametric Mixed Models

Assume that the data is made up of multiple different distributions, but they're all of the same type (Gaussian, for example). Each of the different clusters are made up of models with different means and variances and models.

It's called parametric mixture model because its a mix of the same distributions, but with different parameters.

10.4.2 Gaussian Mixture Models

Assume each cluster matches a Gaussian distribution.

You can have multivariate Gaussians.

10.4.3 Expectation-Maximization Function

It's just a lost function for datapoints matching a probability distribution.

Calculate the Maximum Likelihood Estimation with the log-likelihood.

Use Bayes' Theorem and weight changes to adapt and create your cluster model.

Chapter 11

Week 11

11.1 Density-Based Clustering

Allows you to find clusters of arbitary shape.

11.2 DBSCAN

A cluster a maximal set of density-connected points.

It's an algorithm
sensitive to parameters.

11.2.1 Directly Density Reachable

this is true if p is within the epsilon neighborhood of q.

And there's enough core points in the neighborhood.

11.2.2 Density Reachable

If there is some sequence of direct density reachable points that get your from p to q.

11.2.3 Density Connected

If there exists o such that its density reachable to both p and q. This allows p and q to both be at the border of a cluster while still connected-ish.

11.3 Evaluation of Clusters

11.3.1 Clustering Tendency

suitability of clustering on a dataset.

- Spatial Histogram
- Distance Distribution
- Hopkins Statistic (whatever that is)

11.4 Number of Clusters

Find something in between 1 cluster and every point being its own cluster.

11.4.1 Empirical Number

$$k = \sqrt{\frac{n}{2}}$$

Personally I think this is dumb.

11.4.2 Elbow Method

I TALKED ABOUT THIS IN PHYS498

Look at the whole bunch of cluster methods with different k values and look at cluster variance. At some point you can diminishing returns in your information gain.

11.4.3 Cross Validation

Partition your data and validation your clustering method against the other partitions.

11.5 Clustering Quality

There is not a consensus-perfect way to do this.

11.5.1 Extrinsic

human-derived goals (a.k.a. *ground truth*)

The qualities that we want out of extrinsic cluster evaluation are:

- Homogeneity
- Completeness
- rag bag better than alien
- small cluster prevention

Some common methods are

- Matching Based
- Information Theory Based
- Pairwise-Comparison Based

11.5.2 Matching Based: Purity

The extent that a cluster contains points only from the ground truth.

$$Purity(C_i, G_j) = \sum \frac{|C_i \cap G_j|}{|C_i|}$$

11.5.3 Information Theory Based: Conditional Entropy

$$H(G|C) = -\sum_{i=1}^{k}\sum_{j=1}^{l} \frac{|C_i \cap G_j|}{n} \log \frac{|C_i \cap G_j|}{n}$$

11.5.4 Normalized Mutual Information

$$I(C,G) = -\sum_{i=1}^{k}\sum_{j=1}^{l} p_{ij} \log \frac{p_{ij}}{p_{Ci} \cdot p_{Gj}}$$

$$NMI(C,G) = \frac{I(C;G)}{\sqrt{H(C)} + H(G)}$$

Value close to 1 means good clustering. It means the clusterings have very similar information.

11.5.5 Pairwise-Comparison Based: Jaccard Coefficient

Make a contingency table of calculated clusters and ground truth clusters.

$$J(C,G) = \frac{|C \cap G|}{|C \cup G|} = \frac{TP}{TP + FP + FN}$$

11.6 Intrinsic

quality based off of just the data (silhouette coefficient).

If clusters are compact and separated, then that is good.

11.6.1 Dunn Index

Maximum distance between 2 points that belong in the same cluster is Δ.

The minimum distance between 2 points in different clusters is δ.

$$DI = \frac{\delta}{\Delta}$$

The larger the better.

11.6.2 Silhouette Coefficient

Measures how well-separated clusters are by comparing intra-cluster and inter-cluster distances.

For each point i:

- $a(i)$ = average distance to other points in same cluster
- $b(i)$ = average distance to points in nearest neighboring cluster

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Range: $[-1, 1]$. Values close to 1 indicate good clustering (point is well-matched to its cluster). Values near 0 indicate the point is on the border between clusters. Negative values indicate the point may be in the wrong cluster.

Overall silhouette score is the mean of all $s(i)$ values.

When the value for an individual object o is 1, that is good.

If the value is negative, that is bad.

Chapter 12

Week 12

12.1 Neural Networks

They have many neurons and activation functions for their regression stuff.

12.2 Gradient Descent and Backpropagation

You take the gradient of the probability space and you change the weights of the neurons in the opposite direction of the gradient to minimize the loss function.

12.3 Convolutional NN's

You make the data space lower-dimensional with a convolution function and then run your NN on that before bringing it back to the regular dimensional space.

12.4 Recurrent NN's

You input sequential data and the previous datapoints are kept in the nodes to affect the future datapoints.