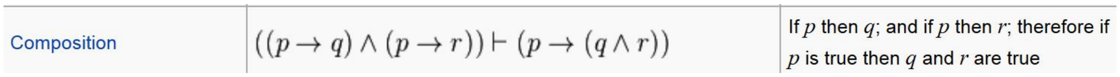
Week 1

Lecture Points:

- Propositional Logic
    - Syntax (atoms, ¬ ∧∨, formulas)
    - Semantics (worlds)
    - Satisfaction (worlds satisfies formulas)
- Association rules
    - Confidence, support
    - Frequent itemsets
    - Downward closure
    - A-PRIORI Algorithm
    - Multiple support levels
- Probabilistic logic
    - Interpretations
    - Extended syntax (probability annotations)
    - Consistency (identifying constraints in KB)
    - Entailment (Given KB , and query formula q, find probability of q)
    - Linear programming base algorithm for content
    - Word sampling

Questions:

1) **Given atoms: $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{c}$; formula $\boldsymbol{f_1 = a \rightarrow b}$ and $\boldsymbol{f_2 = a \rightarrow c}$; and a world $\boldsymbol{W}$ that $\boldsymbol{W \vDash f_1}$ and $\boldsymbol{W \vDash f_2}$, show whether the world would satisfy formula $\boldsymbol{f_3 = a \rightarrow (b \wedge c)}$**



Composition	$((p \to q) \wedge (p \to r)) \vdash (p \to (q \wedge r))$	If p then q; and if p then r; therefore if p is true then q and r are true

2) **What is the shortage of min-support? And how it can be resolved?**

   Min-support condition may miss important but not frequent itemset

   Solution: adjusting min-support to a lower value and incorporating more restrictive min-confidence while generating rules.

3) **Given atoms $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{c}$ and words, $\boldsymbol{W_1, W_2}$and $\boldsymbol{W_3}$ *such that***

   $W_1 \vDash a \wedge b \wedge c, W_2 \vDash a \vee b \vee c\ , W_3 \vDash\ a \rightarrow (b \wedge c), W_4 \vDash\ a \rightarrow c$

   Which worlds satisfy the following formula?

   i. $\boldsymbol{f = b}$**. Answer** **(W1)**
   ii. $\boldsymbol{f = a} \rightarrow c$ **Answer** **(W3 and W4)**
   iii. $\boldsymbol{f = a} \wedge c$ **Answer** W2
   iv. $\boldsymbol{f = \neg a} \vee c$ **Answer** **(W3 and W4)**
   v. $\boldsymbol{f =}\ a \vee b \vee c$ **Answer** **W1 and W2**

4) **In probabilistic logic, if there are a set of sentences and a set of possible worlds, and we want to check for an entailment for a new logical sentence, explain how linear programing can be utilized.**

   Linear programming can be used to find bounds of V matrix (rows representing each sentence and columns representing a set of worlds) if V is small enough. LP helps determine the bounds of convex hull by which entailment can holds if the projected vector probability of the sentence that we are testing its entailment is contained in that convex hull. The approximation of probability can be done by vector projection, and by using probability distribution we can compute the probability as a scaler value, and that value has to meet the constraints where sum of probabilities equal to 1.

Week 3 and 4

Contents:

- Social network, some kind of relations that can be formed as influence relation between nodes.
- SIR (monotonic) and SIS (non-monotonic) from (Physics)
- Expected number of infected is #P hard
- Centrality measure and decomposition and shell number.
- Tipping Model and Target Set Selection: a node adopt some type of behavior if threshold if k number of his friends do, where k assigned based on historical data or assumption, and this can create a cascade effect. Outcome can be computed in PTIME, but it is NP hard to find set of individual who maximize the outcome, but we resort to a heuristic approach to find seeds.
- Kempe-Kleinberg-Tardos (KKT) Framework Part I & 2
- IC generalize SIR
- Linear threshold model generalize the tipping model, by allowing uniform probability distribution over threshold per node
- Computing under IC or LT optimum is #P hard, finding seed set to maximize diffusion is NP hard
- Both are submodular => we can use greedy algorithm and lazy evaluation
- MIIA is used for IC, and SIMPATH is used for LTM
- Experimental Results about above models in current publications:
- Damon Centola's:
    - Random network: {lower clustering coefficient, and smaller diameter}
    - Lattice network{higher clustering coefficient, and larger diameter}
    - Studied effects in social network (controlled environment).
    - Experiment random graph: cluster coefficient is low.
    - Lattice model higher clustering coefficient=> encourage diffusion more than random graph.
- Ugander's study:
    - Facebook ego network where friends/communities cause diffusion.
- Zhang et al.'s study:
    - His result counters Ugander's. His dataset is directed graph (due to semantic meaning of edges)
- Data driven approach (helps to assign edge weight values for networks before applying IC or LTM as an example)
    - Expectation Maximization

1) Give an example of a node where betweenness centrality is higher than degree centrality in a graph.
2) In SIR model, given graph G=(V,E) suppose that the probability to infect is β and $M(t)$ is the number of infectees in next time step t'.
3) Given, $S = \{x1, x2, x3, x4\}$ and $H = \{\{x1, x2\}, \{x3, x4\}, \{x2\}, \{x1, x3, x4\}\}$ we have to pick two seeds in the set H that would cover most of the elements of set$S$?
4) Show that the activation function of Tipping model is a generalized version of Jackson Yariv model?
5) Explain the relation between s-t connectivity problem and number of infectees in IC model.
6) If there is a function to compute is #P hard, and you need to approximate this function. Discuss of this function is submodular or non-submodular may or may not help in the approximation.
7) Zhang et al.'s Results counters Ugander's Results. Show what the difference and suggest a reason.

Week5

<u>Contents:</u>

- Machine learning Primer:
    - Features
    - Distance function
    - Supervised Vs. unsupervised
        - Supervised machine learning
            1. Parametric vs. nonparametric:
                - (Parametric (fixed parameters) makes stronger assumptions takes more time in learning phase but makes quick prediction)
                - Nonparametric: makes simpler assumption about given training data, its running time (in prediction phase) highly depends on the number of instances
            2. Regression vs. classification
        - Unsupervised machine learning {clustering, latent variable model}
    - Related examples to our course work and ongoing projects
- Decision Tree:
    - Hypotheses set of rules:: $f: x \to y$
    - High entropy => high uncertainty
    - Entropy and conditional entropy
    - Splitting and decreasing entropy through Information Gain (IG)
    - When stop splitting: fixed depth or fixed number of leaves
    - Overfitting
    - Building smaller decision trees
    - Random Forrest
- Testing:
    - Cross validation
    - RMSQ, precision, recall, accuracy, etc.

Questions:

- Explain how a distance function can be leveraged to reflect similarity and dissimilarity.
    - Distance function can be considered by itself as a dissimilarity function. The more distance between 2 instances, the higher dissimilarity they hold.
    - For similarity. Any dissimilarity can be transformed to a similarity function e.g.:
        - Assume $Dis(x, y)$ is a dissimilarity function, the similarity measure can takes different forms:
            1. $Sim(x, y) = -Dis(x, y)$
            2. $Sim(x, y) = 1 - Dis(x, y)$
            3. $Sim(x, y) = \frac{1}{Dis(x,y)}$

- What does overfitting means?
  It is a result of building a learning model, but that model cannot be generalized well. It gives the illusion of a good description of the data by a model, but the model will not usually be able to perform well in predicting new instances that have not been seen as it performs well in training data.
- Explain how we can deal with continuous features while building decision tree.
    - Trying heuristically different thresholds that maximize Information Gain
- Show an example that if we have a truth table of Boolean attributes, then we have a unique decision tree where each path corresponds to a row in the truth table. (in slides)
- Suggest two ways to overcome imbalance dataset which is one of main issues in machine learning:
    - Sampling from each class until reaching almost equal datasets.
    - Adding artificial samples to the minor classes until they are equal in size

Week6

Contents:

1) Recommendations- what to predict? A rating prediction for a given item OR an Item prediction for a given user
2) Content based recommendation (solely based on what a user have previously bought), so new items can be recommended to that user based on items exist in his/her purchase history alone (called "user profile"). Different machine learning techniques can be applied such KNN since feature space can be already normalized such as rating or different features about items themselves that TF-IDF can be utilized as well. Then a model can recommend top items similar to those existing in the user's purchase profile.
3) Collaborative filtering
    - A user may like what other similar users likes
        - Need to capture users behaviors in purchase to recommend an item for a user
        - Identify/recommend product for a given user by predict a rating for a given product and select top k items.
        - Scalability issue here with the **user based CF**, then we can resort to **item based** (processing similarity among items themselves) to predict rating for an item.
4) Association Rule:
    - Just prediction items (no predicting rating)
    - Some extracted rules may not be customized for a given user
        - Sol: perform new search (depth first search) for items that a user purchased within a window of time on the **Frequent Itemset Graph**, and then look for one level below. The items showed in that level can be the head (recommended items) for that user, and we could again prioritize based on the confidence.
        - Searching the graph is practical since commonality intersection between rule and body is already inherited in the Frequent Itemset Graph.
5) Matrix factorization:
    - Some latent variables (we don't care about them)
    - Help to generate 2 smaller more generalized matrices such that its product results is very similar to the original matrix M (user X item), which later can help for predicting rates.
    - Optimization of Error function using gradient descent
    - Stochastic gradient descent to improve performance
        - May differ that it would have faster iteration than normal gradient descent since in each iteration just some of components (randomly) got updated. This will lead to larger number of iteration until it converge, but it will generally faster as a whole if it is compared with
    - It can predict rating for any missing entry in the matrix M

Questions

1) **Explain how frequent itemset graph would help to recommend new items if rules are already generated and stored in DB.**

   Given an active user session window w, sorted in lexicographic order, a depth-first search of the Frequent Itemset Graph is performed to level |w|. If a match is found, then the children of the matching node n containing w are used to generate candidate recommendations.

   Ref:

   http://facweb.cs.depaul.edu/mobasher/research/papers/WIDM01/node4.html

2) **Stochastic gradient descent generally improves gradient descent.**

   a. **Explain how it operates:**
      It operates as gradient descent to find local minimum value by approximation, but it does not update all components in each iteration, it instead randomly selects a subset of components, so they can be updated.

   b. **Compare number of iteration until convergence:**

      Since not every component will be updated in each iteration for the stochastic gradient descent. Number of iteration is much higher in stochastic than gradient descent.

   c. **Compare computation time for one iteration.**
      Stochastic gradient descent takes less time in each iteration, and this yields to shorter time to convergence even though number of iteration in the stochastic more than the gradient descent.

   d. **Learn the importance of Convergence Rate.**
      Better to be as a decreasing function based on iteration that help converging. High values for convergence rate would cause diverge, and low values will cause slow learning.
      Ref (page 16: *rakaposhi.eas.asu.edu/cse494/notes/s07-filtering-class.ppt*)

3) **What are the major issues with Collaborative filtering? Suggest an approach to resolve it.**

- The Cold Start Problem

  - CF cannot function effectively because items' (or users') vectors do not have sufficient numbers of rated items to find vectors similar to them.

  - Hybrid approach that includes: content based and collaborative filtering.

- Data Sparsity

  - When there are a large number of items to be ranked, most of the entries can be zero, resulting in a very sparse matrix. Finding similar items using this sparse matrix is challenging for many algorithms.

- First Rater Problem
  Cannot recommend an item that has not been previously rated.
  New items
  Esoteric items

Week7

Contents:

The focus is now on ***unsupervised** learning* (where data have no target attribute/class) to find some intrinsic structure without advance knowledge. Meanwhile, ***supervised learning*** **discovers patterns in the data that relate data attributes with a target (class) attribute**

- ***Unsupervised** learning* (where data have no target attribute/class) to find some intrinsic structure
- ***supervised learning*** **discovers patterns in the data that relate data attributes with a target (class) attribute**

Most popular supervised technique:

- K-Mean: a partitioning clustering algorithm that iteratively calculate centroids and compute data point (or instances) membership to k-clusters
    - In each iteration k centroids are updated, and datapoints' membership to a cluster are updated too
    - Weakness
        - Sensitive to seeds initial centroids (seeds) selection
        - Sensitive to outlier
        - Number of cluster has to be determined and it has to be given to the algorithm as an input before running it.
        - Simple and easy to implement.
    - Hierarchical Clustering
        - Dendogram incapsulation
        - (Generally)Quadratic run time

Data Standardization:

- Data generalization distances
    - Standardize for attribute value
    - Different scales
- Intervals
- Ratio discrete elements
- How to describe clusters (representative points… the closer to the centroids tend to be more representative)

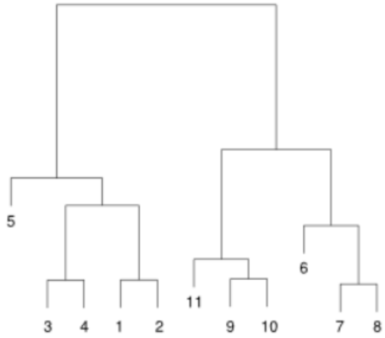
Community Structure in networks:

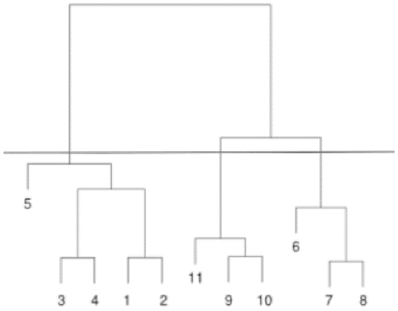
- Component based: simple threshold … not always applicable
- Not component based
    - Modularity and Maximization
    - Greedy approach (not scalable)
    - Louvain Approach: better in scalability and stable results
    - Label Propagation (another approach)

- All above are sensitive to initial community definition (the default: each node is considered as a community by its own)

Questions:

1. How to describe obtained clusters in term of inner distance between cluster's elements and clusters themselves?
    a. Inner distance is objected to be minimized (Intra-cluster cohesion).
    b. Distance between clusters is objected to be maximize (inter-cluster separation).
2. Explain why sum of squared error (SSE) is the best technique as convergence criteria in k-mean algorithm
    - SSE is the most one used, since it is derivative and it can be utilized as an objective function to be minimized, and it also its value change can be checked and tracked in each iteration.
    - Most implementation will use SSE as a convergence criteria check.
3. K-mean is sensitive to selecting initial centroids. Suggest an approach to overcome this limitation
    a. Choose centroids as farthest apart as possible from each other.
    b. Try multiple initialization (different seed sets) and select a best one that for instance would give the lowest SSE.
4. K-mean is sensitive to outliers. Suggest an approach to overcome this limitation.
    a. Removing that outliers manually
    b. Sampling from data which leads outliers to have a lower chance to be selected for computed centroids, after computing centroids by running k-mean, all data then can be assigned clusters by checking nearest centroids through distance function.
    c. Similar to b, but by using other statistical techniques.
5. Giving the following dendrogram tree find 3 clusters and each belonging datapoints.





This can be done by finding a cut line that gives desired number of clusters.

Clusters are:

Cluster1={5,3,4,1,2}, Cluster2={ 11,9 ,10} and Cluster3={ 6,7, 8}

6. Community finding via modularity is similar to clustering. Show one similarity and one difference.

Both have a global objective function to optimize:

- Clustering: minimize inner cluster distance, and maximize intra cluster distance.
- Community finding: maximize inner community edge density, and minimize edge density inter- community.

The difference is that we may obtain different communities where (datapoints) in a given community are not necessarily near by each other's by using any similarity measure.

Week8

<u>Content:</u>

- **Naïve Base Classification**
    - Documents, as bag of words, can be transferred into a numerical space such as term frequency (TF). This TF can be normalized by the inverse document frequency (TF-IDF) which aim to keep unimportant words (e.g. stop words) having a lower weight than other terms.
    - Features in Naïve Bayes are assumed to be **<u>independent</u>** (for simplicity) which helps to build a simple and highly effective model (from practical view)
    - From training data, we can simply compute prior (fraction of class labels in training data) and then estimate posterior.

$$\Pr(C=c_j \mid A_1=a_1,\dots,A_{|A|}=a_{|A|}) \longrightarrow \text{posterior}$$

$$=\frac{\underbrace{\Pr(A_1=a_1,\dots,A_{|A|}=a_{|A|} \mid C=c_j)}_{\text{likilhood}}\underbrace{\Pr(C=c_j)}_{\text{Prior}}}{\underbrace{\Pr(A_1=a_1,\dots,A_{|A|}=a_{|A|})}_{\text{normalization consitant}}}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

-
    - We assign a test data with a class label having maximum posterior (known as MAP: maximum posterrior). We do not need to calculate dominator since it is normalization constant for each $\Pr(C_j|A)$

$$c=\underset{c_j}{\operatorname{argmax}}\Pr(c_j)\prod_{i=1}^{|A|}\Pr(A_i=a_i \mid C=c_j)$$

-
    - Problems:
        - Numeric attributes can be discretized "binning"
        - Zero counts -> smoothing
        - Missing value-> ignored or assigning 0.5 each
        - Correlated or related attribute may affect accuracy since it violated independence assumption
- **Probabilistic framework and Mixture Model**
    - Main focus of document classification as if classes are topics
    - Better to view topics as a multinomial **PDF** of **topics** over <u>words</u> and <u>documents</u> called "a mixture model"
    - Assuming we know topics/classes.

  - Clustering documents to simply obtain topics suffered from the assumption that document must only belong to one topic and we cannot extend our view that a document is distributed over topic by some given PDF.
- LSI using SVD
  - Dimension reduction
  - Document share frequently co-occurring terms (even semantically) will be more similar in SVD
  - Complexity
- LDA
  - After training a model, it can classify unseen document by returning a PDF of topics.

Questions:

1) What is a main issue if we elect to use clustering approach on documents to observe topics?
Clustering will divide document into clusters where each cluster may represent topic. This ignore that fact that document can be generated by different topics.

2) Compare pLSA with LDA.
Both can return set of topics, but pLSA did not extend to handle unseen documents, the former can be estimated once (by for instance maximum likelihood estimation). The LDA uses Bayes estimation by which it would consider prior parameters and better estimate posterior for unseen document (or test document), and continually can updating prior.

3) Cosine similarity is well known gauge for computing similarity between any vector in space
   a) Show whether this measure is symmetric it is symmetric cosine(x,y) = cosine(y,x)
   b) Show the output range [-1,1]
   c) Show whether its output is invariant to shift in input. (shifting will still be variant in output)
   d) Show whether its output is invariant to input negation (as binary input). (negation will still show variance in output)

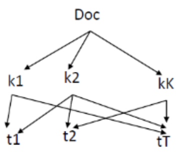
4) Compare cosine similarity with Pearson correlation, and under what circumstance both measure are equally likely (assuming input range is $\mathbb{R}$) (means value for both vector equal to zero)

Week9

Contents:

PLSI/PLSA: Probabilistic latent semantic (indexing/ analysis)

Statistical technique for two mode and co-occurrence data (t, d) ("t": terms and "d": documents) it can be thought as



The model has two parameters to estimate:

(1) p(t|k): it returns terms distribution for a given topic

(2) p(k|doc): it returns topic distribution given a document.

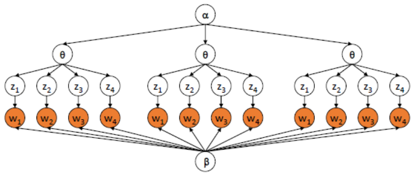
Given algorithm page (14), this algorithm aims to better estimate the parameters: p(t|k) and p(k|doc):

The log likelihood is monotonic (for each next iteration the formula below is an increasing function)

$$\sum_{d=1}^{N} \log P(d) = \sum_{d=1}^{N} \sum_{t=1}^{T} X(t,d) \log \sum_{k=1}^{K} P(t \mid k) P(k \mid d)$$

LDA: Latent Dirchilet Allocation

- Documents are considered as bag of words regardless words position in each documents
- Corpus: is a collection of documents.
- LDA aims to model documents in the corpus as Dirichlet distribution by estimating posteriors, so each document can be considered as if it generated from different set of topics.
- It is Similar to pSLI, but it can handle unseen document by introducing a method of sampling θ form Dirchelet distribution where θ is a distribution over topics. This layer theta would make variable z 'topics' conditionally dependent on θ. w (words/terms) variables are distributed and conditional dependent on z. (if alpha is 1, it means we have uniform Dirichlet distribution equivalent to pLSA)
- Return better topics comparing with the other SVD and pSLI
- Different usage for obtained parameters, but the most common usage is by getting histogram/world cloud of p(w|z,beta) where beta in practice is (a fixed value "data dependent")



  - Main assumptions:

  - Documents can be viewed as if they generated by mixture models, and our job is to try to come up with models that resemble this generating process.
  - Also these documents can be viewed as a mixture of different topics (in other words, topics are distributed over documents where each topic can be viewed as a list of the most thematic coherent words that come with that topic). Both Topics and documents can be modeled by Dirichlet distribution where α is the model prior per-document topic distribution, and β is the Dirichlet prior on the per topic word distribution
  - Topics can be characterized by a distribution over words, and after building and training LDA model, these words can be drawn by given any topic.
  - Words/terms are the atomic units of LDA, and it is the only variable are observed.
  - Both documents and words can be observed in corpus, but topics are not (i.e. "hidden variable" that our job to find them). To infer the hidden structure of topics. We can use a generative process that use Dirichlet distribution and learn posterior as we observe more and morewords contained in documents. This is a way to model multinomial distribution for both documents, words, and topics.
  - As we estimate Dirichlet posterior in each iteration. Drawing θ's tend to cone from (as geometric speaking), and more similar to the previous iteration. This can be tested by KULL measure to check minimum variance as a convergence test which indicates parameters stabilization.
  - To conclude generative model tries to answer the following simple question "What is the topic distribution that can generate documents its content?". Therefore, we need to infer underlying topic structural

Semi-supervised learning

Issue: labeling all data (in big data) is costly and unfeasible.

Semi-supervised help partial labeling, and give labels for unlabeled data.

- Slides gave a nice example for how to Incorporate unlabeled data and labeled ones in EM
- EM algorithm is continually updating mixture component parameters in the M (Maximization) step, and assigning labels for unlabeled instances in the E (Expectation) step (labels for unlabeled data can be changed in each iteration).

Questions:

(1) Define and give example of

a. Supervised learning-> data are labeled e.g. classification regression
b. Unsupervised learning-> unlabeled data -> clustering
c. Semi supervised learning-> mixed labeled and unlabeled

(2) Highlight the main difference between pLSI and LDA.

pSLA cannot scale as number of documents are increasing because its parameters depend on no. of documents.

pSLA could not extend to handle unseen documents and classify them, but LDA can do since it separates documents in corpus from the LDA model.

LDA is still complicated in terms of its dependents on some parameters initialization (i.e. α, β)

(3) LDA separates documents from its model. List some advantage of this model.

(1) Better for scalability where parameters do not depend on the number of documents.

(2) To classify or to obtain topic distribution for unseen documents.

(4) You aim to obtain a histogram for words based on a given topic. What probability will you use for this purpose.

$p(Z|W)$

(5) In your own words, what the difference between E step and M step, and suggest a way for convergence test.

In E step, classifier is used to estimate membership of each data point $P(C_j|d_j, \theta)$.

In M step, We estimate classifier parameters θ given $P\left(C_j \middle| d_j, \theta\right)$ From E step. We use maximum posterior parameters estimation to find θ $\mathbf{arg}_\theta$ max $P(D|\theta)P(\theta)$. This expression: $P(D|\theta)P(\theta)$ is Bayes rule, and it is proportional to $P(\theta|D)$

Conducting Iteratively over step E and M until parameter θ stabilized, and thus EM finds local maximum solution.

Week10

Contents:

Semi supervised learning.

Labeled data is costly, and unlabeled are plentiful.

E.g. EM with underlying Bayesian classifier to initially estimate prior parameters on labeled data, EM iteratively maximizes likelihood of the parameters. This lead to classify unlabeled data, enhance classification performance, and also classifying unlabeled effectively.

Co-training:

Two classifiers can learn from same data.

Also features can be divided into 2 sets, say x1 and x2, where both have to be conditionally independent features.

Learning from positive (Positive Unlabeled 'PU' Learning or one class learning)

It tries to enhance classification by focusing on recall all positive samples unlabeled data.

Different 2 step strategies for finding positive samples:

(1) Identify a set of reliable negative documents from the unreliable set (spy technique or 1-DNF, Rocchio, Naïve Bayesian method) … All this method wills always find the reliable negative by either considering the lowest probability if (Naive) or the farthest from positive samples if cosine Rocchio

(2) Build a sequence of classifiers by iteratively applying a classification algorithm and then selecting a good classifier. Applying EM or SVM iteratively page 44

Performance measure

$r^2/\Pr[f(x)=1]$ Where $f(x)$ is a label given by classifier {1, -1}.

Questions:

1) Explain how unlabeled data can increase classification performance?

More pattern can be discovered from unlabeled data can lead to a better classification

2) Give some examples of applications domain that positive learning is more suitable than traditional classifications where positive and negative are not scarce?

This might be referred as “one class classification”. It is very common in biology where the main concern is about “scarce” positive samples. This kind of machine learning approach is widely applied in anomaly & outlier detection, and direct marketing

3) Define spy learning and how it is practically used to find more positive samples in unlabeled data.

Following steps from slide: 41 to 43 will result in set Q containing only positive samples from unlabeled data.