CSE 592: Data Mining

Instructor: Pedro Domingos

Program for Today

- Rule induction
  - Propositional
  - First-order
- First project

Rule Induction

Learning Sets of Rules

Rules are very easy to understand; popular in data mining.

- **Variable Size**. Any boolean function can be represented.
- **Deterministic**.
- **Discrete and Continuous Parameters**.

Learning algorithms for rule sets can be described as

- **Constructive Search**. The rule set is built by adding rules; each rule is constructed by adding conditions.
- **Eager**.
- **Batch**.

Rule Set Hypothesis Space

- **Each rule is a conjunction of tests**. Each test has the form $x_j = v$, $x_j \leq v$, or $x_j \geq v$, where v is a value for x_j that appears in the training data.

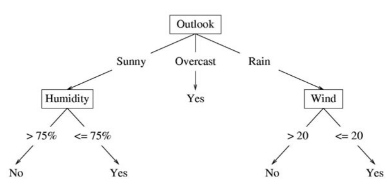
$$x_1 = Sunny \wedge x_2 \leq 75\% \Rightarrow y = 1$$

- **A rule set is a disjunction of rules**. Typically all of the rules are for one class (e.g., $y = 1$). An example is classified into $y = 1$ if *any* rule is satisfied.

$$\begin{aligned} x_1 = Sunny \wedge x_2 \leq 75\% &\Rightarrow y = 1 \\ x_1 = Overcast &\Rightarrow y = 1 \\ x_1 = Rain \wedge x_3 \leq 20 &\Rightarrow y = 1 \end{aligned}$$

Relationship to Decision Trees

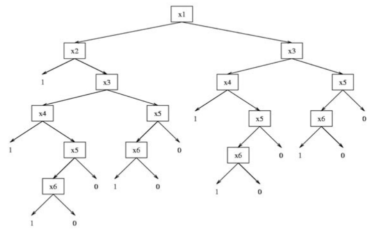
Any decision tree can be converted into a set of rules. The previous set of rules corresponds to this tree:



Relationship to Decision Trees

A small set of rules can correspond to a big decision tree, because of the *Replication Problem*.

$x_1 \wedge x_2 \Rightarrow y = 1$ $\quad x_3 \wedge x_4 \Rightarrow y = 1$ $\quad x_5 \wedge x_6 \Rightarrow y = 1$



Learning a Single Rule

We grow a rule by starting with an empty rule and adding tests one at a time until the rule "covers" only positive examples.

GrowRule(S)
$R = \{\ \}$
repeat
  choose best test $x_j \Theta v$ to add to R, where $\Theta \in \{=, \neq, \leq, \geq\}$
  $S := S-$ all examples that do not satisfy $R \cup \{x_j \Theta v\}$.
until S contains only positive examples.

Choosing the Best Test

- Current rule R covers m_0 negative examples and m_1 positive examples.
  Let $p = \frac{m_1}{m_0+m_1}$.
- Proposed rule $R \cup \{x_j \Theta v\}$ covers m_0' and m_1' examples.
  Let $p' = \frac{m_1'}{m_0'+m_1'}$.
- $Gain = m_1' [(-p \lg p) - (-p' \lg p')]$

We want to reduce our surprise (to the point where we are *certain*), but we also want the rule to cover many examples. This formula tries to implement this tradeoff.

Learning a Set of Rules (Separate-and-Conquer)

GrowRuleSet(S)
$A = \{\ \}$
repeat
  $R :=$ GrowRule(S)
  Add R to A
  $S := S-$ all positive examples that satisfy R.
until S is empty.
return A

More Thorough Search Procedures

All of our algorithms so far have used greedy algorithms. Finding the smallest set of rules is NP-Hard. But there are some more thorough search procedures that can produce better rule sets.

- **Round-Robin Replacement**. After growing a complete rule set, we can delete the first rule, compute the set S of training examples not covered by any rule, and one or more new rules, to cover S. This can be repeated with each of the original rules. This process allows a later rule to "capture" the positive examples of a rule that was learned earlier.
- **Backfitting**. After each new rule is added to the rule set, we perform a few iterations of Round-Robin Replacement (it typically converges quickly). We repeat this process of growing a new rule and then performing Round-Robin Replacement until all positive examples are covered.
- **Beam Search**. Instead of growing one new rule, we grow B new rules. We consider adding each possible test to each rule and keep the best B resulting rules. When no more tests can be added, we choose the best of the B rules and add it to the rule set.

Probability Estimates From Small Numbers

When m_0 and m_1 are very small, we can end up with

$$p = \frac{m_1}{m_0 + m_1}$$

being very unreliable (or even zero).

Two possible fixes

- **Laplace Estimate**. Add 1/2 to the numerator and 1 to the denominator:
  $$p = \frac{m_1 + 0.5}{m_0 + m_1 + 1}$$
  This is essentially saying that in the absence of any evidence, we expect $p = 1/2$, but our belief is very weak (equivalent to 1/2 of an example).
- **General Prior Estimate**. If you have a prior belief that $p = 0.25$, you can add any number k to the numerator and $4k$ to the denominator.
  $$p = \frac{m_1 + k}{m_0 + m_1 + 4k}$$
  The larger k is, the stronger our prior belief becomes.

Many authors have added 1 to both the numerator and denominator in rule learning cases (weak prior belief that $p = 1$).

Learning Rules for Multiple Classes

What if rules for more than one class?

Two possibilities:

- Order rules (decision list)
- Weighted vote (e.g., weight = accuracy $\times$ coverage)

Learning First-Order Rules

Why do that?

- Can learn sets of rules such as
  $Ancestor(x, y) \leftarrow Parent(x, y)$
  $Ancestor(x, y) \leftarrow Parent(x, z) \wedge Ancestor(z, y)$
- The PROLOG programming language:
  programs are sets of such rules

First-Order Rule for Classifying Web Pages

[Slattery, 1997]

course(A) $\leftarrow$
    has-word(A, instructor),
    $\neg$ has-word(A, good),
    link-from(A, B),
    has-word(B, assign),
    $\neg$ link-from(B, C)

Train: 31/31, Test: 31/34

FOIL (First-Order Inductive Learner)

Same as propositional separate-and-conquer, except:

- Different candidate specializations (literals)
- Different evaluation function

Specializing Rules in FOIL

Learning rule: $P(x_1, x_2, \ldots, x_k) \leftarrow L_1 \ldots L_n$

Candidate specializations add new literal of form:

- $Q(v_1, \ldots, v_r)$, where at least one of the v_i in the created literal must already exist as a variable in the rule.
- $Equal(x_j, x_k)$, where x_j and x_k are variables already presentin the rule
- The negation of either of the above forms of literals

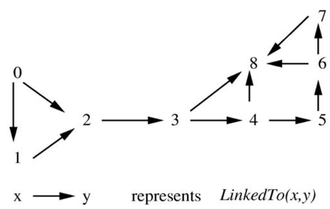
Information Gain in FOIL

$$Foil\_Gain(L, R) \equiv t \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right)$$

Where

- L is the candidate literal to add to rule R
- p_0 = number of positive bindings of R
- n_0 = number of negative bindings of R
- p_1 = number of positive bindings of $R + L$
- n_1 = number of negative bindings of $R + L$
- t = no. of positive bindings of R also covered by $R + L$

FOIL Example



Target function:

- *CanReach(x,y)* true iff directed path from x to y

Instances:

- Pairs of nodes, e.g $\langle 1, 5 \rangle$, with graph described by literals *LinkedTo(0,1)*, $\neg$ *LinkedTo(0,8)* etc.

Hypothesis space:

- Each $h \in H$ is a set of Horn clauses using predicates *LinkedTo* (and *CanReach*)

Induction as Inverted Deduction

Induction is finding h such that

$$(\forall \langle x_i, f(x_i) \rangle \in D) \; B \wedge h \wedge x_i \vdash f(x_i)$$

where

- x_i is ith training instance
- $f(x_i)$ is the target function value for x_i
- B is other background knowledge

So let's design inductive algorithm by inverting operators for automated deduction.

Induction as Inverted Deduction

"Pairs of people $\langle u, v \rangle$ such that child of u is v"

$$\begin{aligned} f(x_i) &: \; Child(Bob, Sharon) \\ x_i &: \; Male(Bob), Female(Sharon), Father(Sharon, Bob) \\ B &: \; Parent(u,v) \leftarrow Father(u,v) \end{aligned}$$

What satisfies $(\forall \langle x_i, f(x_i) \rangle \in D) \; B \wedge h \wedge x_i \vdash f(x_i)$?

$$\begin{aligned} h_1 &: \; Child(u,v) \leftarrow Father(v,u) \\ h_2 &: \; Child(u,v) \leftarrow Parent(v,u) \end{aligned}$$

Induction as Inverted Deduction

We have mechanical *deductive* operators $F(A, B) = C$, where $A \wedge B \vdash C$

Need *inductive* operators

$$O(B, D) = h \text{ where } (\forall \langle x_i, f(x_i) \rangle \in D) \; (B \wedge h \wedge x_i) \vdash f(x_i)$$

Induction as Inverted Deduction

Positives:

- Subsumes earlier idea of finding h that "fits" training data
- Domain theory B helps define meaning of "fit" the data

$$B \wedge h \wedge x_i \vdash f(x_i)$$

- Suggests algorithms that search H guided by B

Induction as Inverted Deduction

Negatives:

- Doesn't allow for noisy data. Consider

$$(\forall \langle x_i, f(x_i) \rangle \in D)\ (B \wedge h \wedge x_i) \vdash f(x_i)$$

- First order logic gives a *huge* hypothesis space H
  → Overfitting
  → Intractability of calculating all acceptable h's

Deduction: Resolution Rule

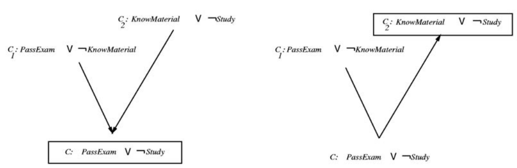
$$\begin{array}{ccc} P & \vee & L \\ \neg L & \vee & R \\ \hline P & \vee & R \end{array}$$

1. Given initial clauses C_1 and C_2, find a literal L from clause C_1 such that $\neg L$ occurs in clause C_2
2. Form the resolvent C by including all literals from C_1 and C_2, except for L and $\neg L$. More precisely, the set of literals occurring in the conclusion C is

$$C = (C_1 - \{L\}) \cup (C_2 - \{\neg L\})$$

where $\cup$ denotes set union, and "$-$" is set difference

Inverting Resolution



Inverted Resolution (Propositional)

1. Given initial clauses C_1 and C, find a literal L that occurs in clause C_1, but not in clause C.
2. Form the second clause C_2 by including the following literals

$$C_2 = (C - (C_1 - \{L\})) \cup \{\neg L\}$$

First-Order Resolution

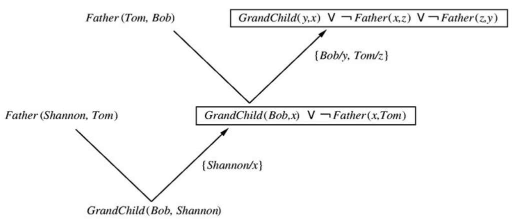
1. Find a literal L_1 from clause C_1, literal L_2 from clause C_2, and substitution θ such that $L_1\theta = \neg L_2\theta$
2. Form the resolvent C by including all literals from $C_1\theta$ and $C_2\theta$, except for $L_1\theta$ and $\neg L_2\theta$. More precisely, the set of literals occurring in the conclusion C is

$$C = (C_1 - \{L_1\})\theta \cup (C_2 - \{L_2\})\theta$$

Inverting First-Order Resolution

$$C_2 = (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{\neg L_1 \theta_1 \theta_2^{-1}\}$$

Cigol



Progol

PROGOL: Reduce comb explosion by generating the most specific acceptable h

1. User specifies H by stating predicates, functions, and forms of arguments allowed for each
2. PROGOL uses sequential covering algorithm.
   For each $\langle x_i, f(x_i)\rangle$
   - Find most specific hypothesis h_i s.t.
     $B \wedge h_i \wedge x_i \vdash f(x_i)$
     - actually, considers only k-step entailment
3. Conduct general-to-specific search bounded by specific hypothesis h_i, choosing hypothesis with minimum description length

Rule Induction: Summary

- Rule grown by adding one antecedent at a time
- Rule set grown by adding one rule at a time
- Propositional or first-order
- Alternative: inverse resolution

First Project: Clickstream Mining

Overview

- The Gazelle site
- Data collection
- Data pre-processing
- KDD Cup
- Hints and findings

The Gazelle Site

- Gazelle.com was a legwear and legcare web retailer.
- Soft-launch: Jan 30, 2000
- Hard-launch: Feb 29, 2000
  with an Ally McBeal TV ad on 28th
  and strong $10 off promotion
- Training set: 2 months
- Test sets: one month
  (split into two test sets)

Data Collection

- Site was running Blue Martini's Customer Interaction System version 2.0
- Data collected includes:
  - Clickstreams
    - Session: date/time, cookie, browser, visit count, referrer
    - Page views: URL, processing time, product, assortment (assortment is a collection of products, such as back to school)
  - Order information
    - Order header: customer, date/time, discount, tax, shipping.
    - Order line: quantity, price, assortment
  - Registration form: questionnaire responses

Data Pre-Processing

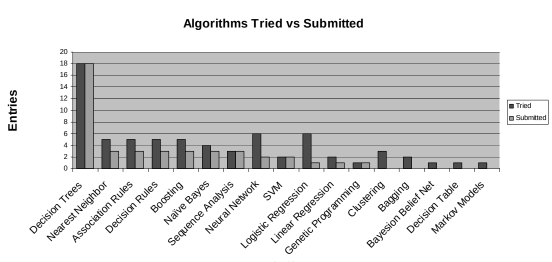
- Acxiom enhancements: age, gender, marital status, vehicle type, own/rent home, etc.
- Keynote records (about 250,000) removed. They hit the home page 3 times a minute, 24 hours.
- Personal information removed, including: Names, addresses, login, credit card, phones, host name/IP, verification question/answer. Cookie, e-mail obfuscated.
- Test users removed based on multiple criteria (e.g., credit card) not available to participants
- Original data and aggregated data (to session level) were provided

KDD Cup Questions

1. Will visitor leave after this page?
2. Which brands will visitor view?
3. Who are the heavy spenders?
4. Insights on Question 1
5. Insights on Question 2

KDD Cup Statistics

- 170 requests for data
- 31 submissions
- 200 person/hours per submission (max 900)
- Teams of 1-13 people (typically 2-3)



Decision trees most widely tried and by far the most commonly submitted

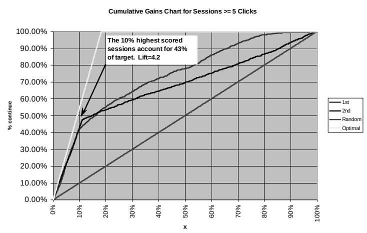
Note: statistics from final submitters only

Evaluation Criteria

- Accuracy (or score) was measured for the two questions with test sets
- Insight questions judged with help of retail experts from Gazelle and Blue Martini
- Created a list of insights from all participants
  - Each insight was given a weight
  - Each participant was scored on all insights
  - Additional factors: presentation quality, correctness

Question: Who Will Leave

- Given set of page views, will visitor view another page on site or leave?
  Hard prediction task because most sessions are of length 1.
  Gains chart for sessions longer than 5 is excellent.

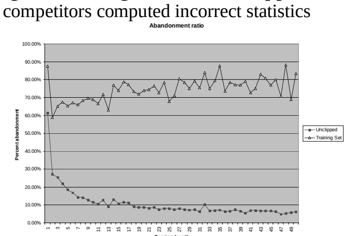


Insight: Who Leaves

- Crawlers, bots, and Gazelle testers
  - Crawlers hitting single pages were 16% of sessions
  - Gazelle testers: distinct patterns, referrer file://c:\...
- Referring sites: mycoupons have long sessions, shopnow.com are prone to exit quickly
- Returning visitors' prob. of continuing is double
- View of specific products (Oroblue,Levante) causes abandonment - Actionable
- Replenishment pages discourage customers. 32% leave the site after viewing them - Actionable

Insight: Who Leaves (II)

- Probability of leaving decreases with page views
  Many many “discoveries” are simply explained by this.
  E.g.: “viewing 3 different products implies low abandonment”
- Aggregated training set contains clipped sessions
  Many competitors computed incorrect statistics



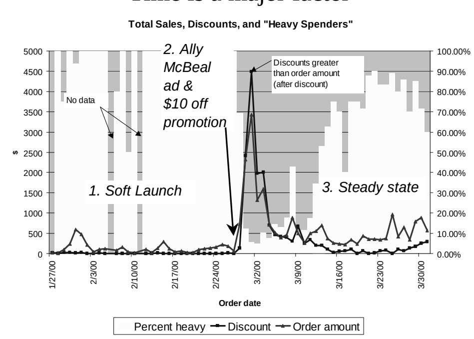
Insight: Who Leaves (III)

- People who register see 22.2 pages on average compared to 3.3 (3.7 without crawlers)
- Free Gift and Welcome templates on first three pages encouraged visitors to stay at site
- Long processing time (> 12 seconds) implies high abandonment - Actionable
- Users who spend less time on the first few pages (session time) tend to have longer session lengths

Question: “Heavy” Spenders

- Characterize visitors who spend more than $12 on an average order at the site
- Small dataset of 3,465 purchases /1,831 customers
- Insight question - no test set
- Submission requirement:
  - Report of up to 1,000 words and 10 graphs
  - Business users should be able to understand report
  - Observations should be correct and *interesting*
    average order tax > $2 implies heavy spender
    is not interesting nor actionable

Time is a major factor



Insights (II)

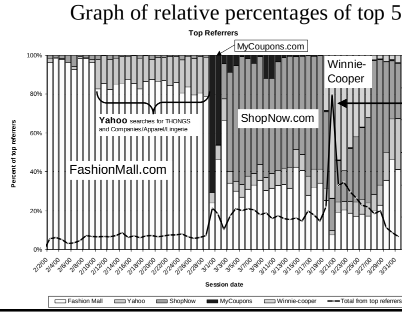
- Factors correlating with heavy purchasers:
  - Not an AOL user (defined by browser) (browser window too small for layout - poor site design)
  - Came to site from print-ad or news, not friends & family (broadcast ads vs. viral marketing)
  - Very high and very low income
  - Older customers (Acxiom)
  - High home market value, owners of luxury vehicles (Acxiom)
  - Geographic: Northeast U.S. states
  - Repeat visitors (four or more times) - loyalty, replenishment
  - Visits to areas of site - personalize differently (lifestyle assortments, leg-care vs. leg-ware)

Target segment

Insights (III)

Referring site traffic changed dramatically over time.

Graph of relative percentages of top 5 sites



Note spike in traffic

Insights (IV)

- Referrers - establish ad policy based on conversion rates, not clickthroughs
  - Overall conversion rate: 0.8% (relatively low)
  - MyCoupons had 8.2% conversion rate, but low spenders
  - FashionMall and ShopNow brought 35,000 visitors Only 23 purchased (0.07% conversion rate!)
  - What about Winnie-Cooper?

    Winnie Cooper is a 31-year-old guy who wears pantyhose and has a pantyhose site.

    8,700 visitors came from his site (!).

    Actions:
    - Make him a celebrity, interview him about how hard it is for men to buy in stores
    - Personalize for XL sizes

Common Mistakes

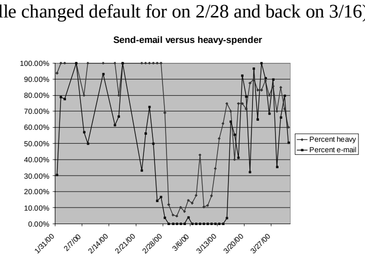
- Insights need support

  Rules with high confidence are meaningless when they apply to 4 people
- Dig deeper

  Many “interesting” insights with interesting explanations were simply identifying periods of the site. For example:
  - “93% of people who responded that they are purchasing for others are heavy purchasers.” True, but simply identifying people who registered prior to 2/28, before the form was changed.
  - Similarly, “presence of children" (registration form) implies heavy spender.

Example

- Agreeing to get e-mail in registration was claimed to be predictive of heavy spender
- It was mostly an indirect predictor of time (Gazelle changed default for on 2/28 and back on 3/16)



Question: Brand View

- Given set of page views, which product brand will visitor view in remainder of the session? (Hanes, Donna Karan, American Essentials, or none)
- Good gains curves for long sessions (lift of 3.9, 3.4, and 1.3 for three brands at 10% of data).
- Referrer URL is great predictor
  - FashionMall, Winnie-Cooper are referrers for Hanes, Donna Karan - different population segments reach these sites
  - MyCoupons, Tripod, DealFinder are referrers for American Essentials - AE contains socks, excellent for coupon users
- Previous views of a product imply later views
- Few realized Donna Karan only available > Feb 26

Project

- Implement decision tree learner
- Apply to first question (Who leaves?)
- Improve accuracy by refining data
- Report insights
- Good luck and have fun!