



Information-Theoretical and Combinatorial Methods in Data Mining

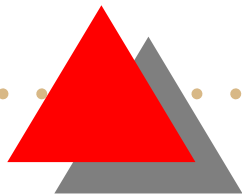
Szymon Jaroszewicz


University of Massachusetts at Boston



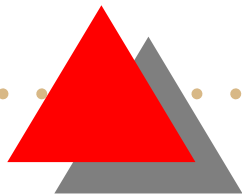
Overview

1. Axiomatizations of entropy and related information theoretical concepts with applications to decision tree induction.
2. Interestingness and pruning of association rules.
3. Bonferroni-type inequalities and their applications in data-mining.



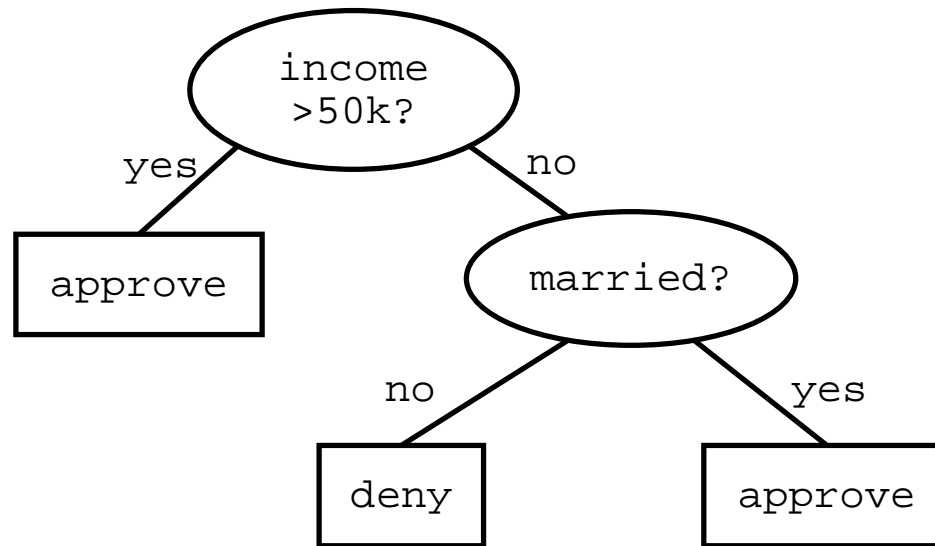


*Axiomatizations of entropy and
related information theoretical
concepts with applications to
decision tree induction*



Decision trees

Rudimentary example: credit decision



data \rightarrow decision tree is an important data mining problem

Decision Tree Construction

Usually a greedy algorithm:

- Choose an attribute for the root of the tree
- Partition the data according to the root of the tree
- Call the algorithm recursively for each subset

Splitting attribute selection problem

- **Crucial task:** choosing test attribute at each node (splitting attribute selection)
- **Most algorithms:** choose attribute which minimizes

$$H(\text{target}|\text{test}) = - \sum P(X = x_i) \sum P(Y = y_i|X = x_i) \log P(Y = y_i|X = x_i)$$

or

$$\text{Gini}(\text{target}|\text{test}) = \sum P(X = x_i) \left(1 - \sum P(Y = y_i|X = x_i)^2\right)$$

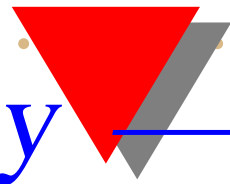


Results of Lopez de Mantaras

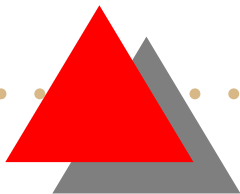
- $D(X, Y) = H(Y|X) + H(X|Y)$ is a *distance* between attributes, that is
 1. $D(X, X) = 0$
 2. $D(X, Y) = D(Y, X)$
 3. $D(X, Z) \leq D(X, Y) + D(Y, Z)$
- If used for splitting attribute selection, tends to produce smaller trees without sacrificing accuracy.

Axiomatizations of Entropy

background



- First axiomatization: Claude Shannon 1948
- Later years: many more axiomatizations [Mathai, Rathie]
- Axiomatization of entropy of Boolean functions [Simovici, Reischer 93]



Axiomatizations of Entropy

contributions

- Alternative axiomatization of entropy of logic functions, axiomatization of conditional entropy of Boolean functions [ISMVL'99]
- Def. of entropy for relational databases, its interactions with relational algebra, approx. functional dependencies based on entropy [in *Finite vs. infinite*, Springer-Verlag, 2000]
- Axiomatization of entropy of a partition and conditional entropy between partitions [IEEE Trans. on Inf. Theory, Jul 2002]



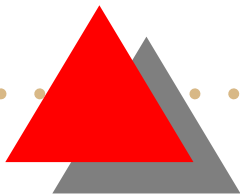
Partitions of finite sets

Let A be a finite set. Partition of A :

$$\{B_1, \dots, B_k\}$$

such that

- $B_i \cap B_j = \emptyset$ for all $1 \leq i < j \leq k$
- $\bigcup B_i = A$



Partitions and database tables

Attributes induce partitions on rows

A	...
a_1	...
⋮	⋮
a_1	...
a_2	...
⋮	⋮
a_2	...
a_3	...
⋮	⋮
a_3	...



Example: axioms for Shannon entropy of partitions

- (P1)** If $\pi, \pi' \in \text{PART}(A)$ are such that $\pi \leq \pi'$, then $0 \leq \mathcal{H}(\pi') \leq \mathcal{H}(\pi)$.
- (P2)** If A, B are two finite sets such that $|A| \leq |B|$, then $\mathcal{H}(\iota_A) \leq \mathcal{H}(\iota_B)$.
- (P3)** For every disjoint sets A, B and partitions $\pi \in \text{PART}(A)$, and $\sigma \in \text{PART}(B)$ we have:

$$\begin{aligned} & \mathcal{H}(\pi + \sigma) \\ &= \left(\frac{|A|}{|A| + |B|} \right) \mathcal{H}(\pi) + \left(\frac{|B|}{|A| + |B|} \right) \mathcal{H}(\sigma) \\ & \quad + \mathcal{H}(\{A, B\}). \end{aligned}$$

- (P4)** If $\pi \in \text{PART}(A)$ and $\sigma \in \text{PART}(B)$, then

$$\mathcal{H}(\pi \times \sigma) = \Phi(\mathcal{H}(\pi), \mathcal{H}(\sigma)).$$



Axiomatizations of generalized entropy

By changing **(P3)** to

$$\begin{aligned} & \mathcal{H}(\pi + \sigma) \\ &= \left(\frac{|A|}{|A| + |B|} \right)^\beta \mathcal{H}(\pi) + \left(\frac{|B|}{|A| + |B|} \right)^\beta \mathcal{H}(\sigma) \\ & \quad + \mathcal{H}(\{A, B\}). \end{aligned}$$

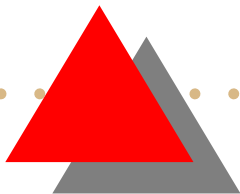
we obtain axiomatizations of a family of entropies H_β (Havrda-Charvát entropy) including important special cases:

1. Shannon entropy ($\beta = 1$)
2. Gini index ($\beta = 2$)



Our result

- $D_{\beta}(X, Y) = H_{\beta}(Y|X) + H_{\beta}(X|Y)$ is a distance between attributes
- Experiments suggest that generalized entropy distances produce small trees.



Experimental results

1. tested on 33 datasets (UCI Irvine)
2. using distance: decrease in size in 20 cases.
3. 17 cases: smallest trees for $\alpha \neq 1, 2$.
4. accuracy did not significantly change
5. best reduction: (`primary-tumor` database $\alpha = 2.5$): $2.7\times$ smaller the standard J48 algorithm (Shannon Entropy).
6. in some cases an increase in size though.
7. open problem: investigate why



*Interestingness and pruning of
association rules*

Market basket data

customer ID	beer	bread	...	diapers
101	1	0	...	1
103	0	1	...	1
107	1	1	...	1
...

- **Items:** binary attributes
- **Itemsets:** sets of items

Frequent Itemsets

- **Support** of an itemset I in relation ρ :

$$\text{supp}_{\rho}(I) = \frac{|\{t \in \rho : t[I] = (1, \dots, 1)\}|}{|\rho|}$$

(estimate of probability of all items present)

- Itemset I is frequent if

$$\text{supp}(I) > \text{minsupp}$$

- `Apriori` algorithm efficiently finds all frequent itemsets

Association rules

$$X \rightarrow Y$$

where X and Y are itemsets.

support and confidence:

$$\text{supp}(X \rightarrow Y) = \text{supp}(X \cup Y)$$

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

Common practice: generate all rules having some minimum support and confidence

Too many rules

contact-lenses dataset

- 5 attributes
- 24 rows
- easy to analyze 'manually'

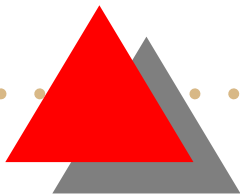


Rules for lenses data

Manually selected rules:

- tears=reduced \rightarrow no lenses
- astigmatism=no,tears=normal \rightarrow soft lenses
- astigmatism=yes,tears=normal \rightarrow hard lenses

- age=pre-presbyopic,prescription=hypermetrope,astigmatism=yes \rightarrow none
- age=presbyopic,prescription=myope,astigmatism=no \rightarrow none
- age=presbyopic,prescription=hypermetrope,astigmatism=yes \rightarrow none





Use association rules

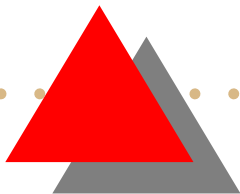
minimum support: 4.2%

no minimum confidence

- 113 rules with consequent lenses
- 890 rules total

minimum confidence: 50%:

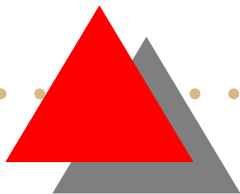
- 86 rules with consequent lenses
- 487 rules total






Solutions

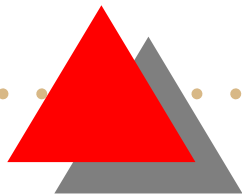
1. Rule sorting according to some measure of interestingness
2. Rule pruning





General measure of interestingness [PKDD01]

1. For rule sorting some measure of rule interestingness is necessary
2. Important measures are: Gini index, χ^2 , entropy gain.
3. We introduced a new measure $\Upsilon_{\alpha,\beta}$ which
 - has all 3 above as special cases
 - parameters α, β allow for smooth transitions between the three, creating intermediate measures with interesting properties.



Rule pruning — current approaches

Each subrule of a rule considered separately

Rule

$$AB \rightarrow Y$$

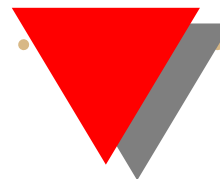
is not interesting if

$$\text{conf}(AB \rightarrow Y) \approx \text{conf}(A \rightarrow Y)$$

or

$$\text{conf}(AB \rightarrow Y) \approx \text{conf}(B \rightarrow Y)$$

Rule pruning [PAKDD02]

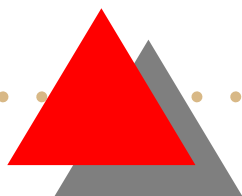


take all subrules into consideration simultaneously

- Rule $r : X \rightarrow Y$ introduces constraints

$$P(XY) = \text{supp}(r), P(X) = \text{supp}(r) / \text{conf}(r)$$

- For $A \rightarrow Y$, let \mathcal{C} : set of constraints of all its subrules.
- Find distribution $P^{\mathcal{C}}$ over $A \cup Y$ such that
 - All constraints in \mathcal{C} are satisfied
 - Entropy of $P^{\mathcal{C}}$ is maximal (**Maximum Entropy Principle**)



Rule pruning [PAKDD02]

- Estimate $\text{conf}^c(A \rightarrow Y)$ based on P^c .
- Interestingness of $A \rightarrow Y$

$$\text{inter}(A \rightarrow Y) = |\text{conf}(A \rightarrow Y) - \text{conf}^c(A \rightarrow Y)|$$

- If $\text{inter}(A \rightarrow Y) < \epsilon$ prune $A \rightarrow Y$, since its explained by its subrules.

Example

assoc. rule	confidence
$\emptyset \rightarrow Y$	0.5
$A \rightarrow Y$	0.3
$B \rightarrow Y$	0.7

Suppose: $\text{conf}(AB \rightarrow Y) = 0.3$

Is $AB \rightarrow Y$ interesting?

Let's use our approach

Example

Subrules introduce constraints \mathcal{C} :

$$\emptyset \rightarrow Y, \text{ conf} = 0.5$$

$$P(Y) = 0.5$$

$$A \rightarrow Y, \text{ supp} = 0.15, \text{ conf} = 0.3$$

$$P(A) = 0.5, P(AY) = 0.15$$

$$B \rightarrow Y, \text{ supp} = 0.35, \text{ conf} = 0.7$$

$$P(B) = 0.5, P(BY) = 0.35$$

Example

The MaxENT distribution is

$$P = \begin{pmatrix} 000 & 001 & 010 & 011 \\ 0.105 & 0.105 & 0.045 & 0.245 \\ & 100 & 101 & 110 & 111 \\ & 0.245 & 0.045 & 0.105 & 0.105 \end{pmatrix},$$

So we expect

$$\text{conf}^c(AB \rightarrow Y) = 0.5$$

contact-lenses *revisited*

antecedent → lenses

tears=reduced → none

astigmatism=no,tears=normal → soft

astigmatism=yes,tears=normal → hard

prescription=myope,astigmatism=yes → hard

prescription=myope,tears=normal → hard

prescription=hypermetrope,astigmatism=yes,tears=normal → none

age=pre-presb.,prescription=hypermetrope,astigmatism=yes → none

age=presbyopic,prescription=myope,astigmatism=no → none

age=presbyopic,prescription=hypermetrope,astigmatism=yes → none

... → ...



*Bonferroni-type inequalities and
their applications in data-mining*





Inspiration

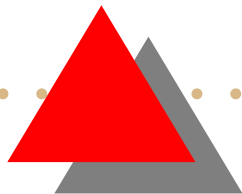
H. Manilla et al. [1996,2001]: Use frequent itemsets to get support (size) of arbitrary queries, e.g.:

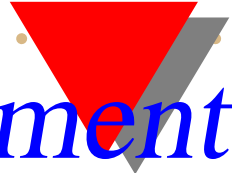
$$\text{supp}(\bar{A}\bar{B}) = 1 - \text{supp}(A) - \text{supp}(B) + \text{supp}(AB)$$

(inclusion-exclusion principle)

Questions:

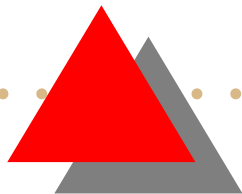
- How to obtain such a formula for arbitrary function?
- Guarantee of accuracy if some supports are unknown?





A more general statement [SIAM DM conf. 2002]

- Boolean Algebra: $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$
- Set of variables: $A = \{a_1, \dots, a_n\}$
- $\text{pol}(A)$ the free Boolean algebra on A consists of polynomials:
 - $\mathbf{0}$, $\mathbf{1}$, and each a_i belong to A ;
 - if $p, q \in \text{pol}(A)$, then $\bar{p}, (p \vee q), (p \wedge q) \in \text{pol}(A)$.





Measures on Boolean Algebras

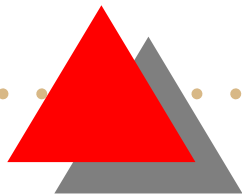
A **measure** on a Boolean Algebra
 $(B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$: $\mu : B \rightarrow [0, \infty]$ s.t.

$$\mu(x \vee y) = \mu(x) + \mu(y)$$

if $x \wedge y = \mathbf{0}$.

Example:

Support supp is a measure on $\text{pol}(A)$





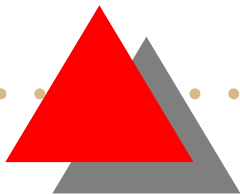
Question 1 rephrased

For any $p \in \text{pol}(A)$ and some measure μ express $\mu(p)$ in terms of measures of positive conjunctions

Examples:

$$\mu(a_1 \oplus a_2) = \mu(a_1) + \mu(a_2) - 2\mu(a_1 \wedge a_2)$$

$$\mu(\bar{a}_1 \wedge \bar{a}_2) = \mu(\mathbf{1}) - \mu(a_1) - \mu(a_2) + \mu(a_1 \wedge a_2)$$



Inclusion-exclusion type result for Exclusive-or

- p_1, p_2, \dots, p_m are Boolean polynomials
- Let

$$S_k = \sum_{i_1 \leq \dots \leq i_k} \mu(p_{i_1} \wedge p_{i_2} \wedge \dots \wedge p_{i_k})$$

- Then,


$$\mu(p_1 \oplus \dots \oplus p_m) = \sum_{k=1}^m (-2)^{k-1} S_k$$

Example

Parity function: $a_1 \oplus a_2 \oplus a_3$

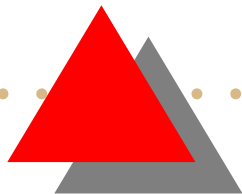
- $S_1 = \mu(a_1) + \mu(a_2) + \mu(a_3)$
- $S_2 = \mu(a_1 \wedge a_2) + \mu(a_2 \wedge a_3) + \mu(a_1 \wedge a_3)$
- $S_3 = \mu(a_1 \wedge a_2 \wedge a_3)$
- giving

$$\mu(a_1 \oplus a_2 \oplus a_3) = S_1 - 2S_2 + 4S_3$$



Every Boolean polynomial can be represented as exclusive-or of positive conjunctions

We can express a measure of any boolean polynomial in terms of measures of positive conjunctions of its variables



Bounds

Dropping terms from inclusion-exclusion we get bounds on the measure: **Bonferroni Inequalities:**

$$\sum_{k=1}^{2r} (-2)^{k-1} S_k^\mu \leq \mu(p_1 \oplus \dots \oplus p_m) \leq \sum_{k=1}^{2s+1} (-2)^{k-1} S_k^\mu,$$

for any $r, s \in \mathbb{N}$

Example

- Upper bound:

$$\mu(a_1 \oplus a_2 \oplus a_3) \leq \mu(a_1) + \mu(a_2) + \mu(a_3)$$

- Lower bound:

$$\begin{aligned} \mu(a_1 \oplus a_2 \oplus a_3) &\geq \mu(a_1) + \mu(a_2) + \mu(a_3) \\ &\quad - 2\mu(a_1 \wedge a_2) - 2\mu(a_2 \wedge a_3) - 2\mu(a_1 \wedge a_3) \end{aligned}$$

We can thus obtain bounds for support of any database query

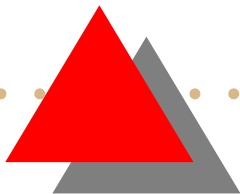


There are queries which cannot be approximated

Parity polynomial: $p_{par} = a_1 \oplus a_2 \oplus \dots \oplus a_n$

Two relations over $A = (a_1, a_2, \dots, a_n)$:

$$\begin{aligned}\rho_{odd} &= \{t \in \text{Dom}(A) : n_1(t) \text{ is odd}\}, \\ \rho_{even} &= \{t \in \text{Dom}(A) : n_1(t) \text{ is even}\},\end{aligned}$$





There are queries which cannot be approximated (cont.)

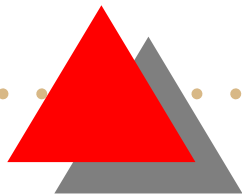
We have:

$$\text{supp}_{\rho_{\text{odd}}}(K) = \text{supp}_{\rho_{\text{even}}}(K) \text{ for all } K \subset A,$$

but

$$\text{supp}_{\rho_{\text{odd}}}(p_{\text{par}}) = 1, \text{supp}_{\rho_{\text{even}}}(p_{\text{par}}) = 0$$

One unknown itemset A can result in huge inaccuracy of $\text{supp}(p_{\text{par}})$



Tables with missing values

Allow missing values $\text{Dom}(a_i) = \{0, \mathbf{u}, 1\}$

Define $\mu^{\mathbf{u}}$ generalizing support to such tables

- With each attribute a_i associate a value $\alpha_i \in [0, 1]$
- If only one attribute a_i is missing, multiply tuple's support by α_i
- If more attributes are missing, use independence assumption

Example

$$\begin{aligned}\mu^{\mathbf{u}}(\bar{a}_1 \wedge a_2) &= \text{supp}(a_1 = \mathbf{0} \wedge a_2 = \mathbf{1}) \\ &+ (1 - \alpha_1) \text{supp}(a_1 = \mathbf{u} \wedge a_2 = \mathbf{1}) \\ &+ \alpha_2 \text{supp}(a_1 = \mathbf{0} \wedge a_2 = \mathbf{u}) \\ &+ (1 - \alpha_1) \alpha_2 \text{supp}(a_1 = a_2 = \mathbf{u})\end{aligned}$$



Properties of μ^u

Theorem:

μ^u is a measure.

Consequences:

- μ^u gives probabilistically consistent results.
- All previous results apply to μ^u



Example

a_1	a_2
1	1
1	u
0	u
0	u

$$\alpha_2 = 1$$

Example

[Ragel, Crémilleux 98]: count each itemset where it is defined


$$\text{supp}(a_1) = 0.5 < \text{supp}(a_1 \wedge a_2) = 1$$

[Nayak, Cook 01]: weighted sum of attributes in a row

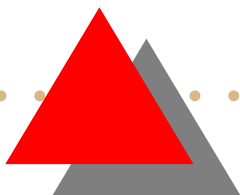
$$\text{supp}(a_1) = 0.5 < \text{supp}(a_1 \wedge a_2) = 0.75$$

but

$$\mu^u(a_1) = 0.5 \quad \mu^u(a_2) = 1 \quad \mu^u(a_1 \wedge a_2) = 0.5$$



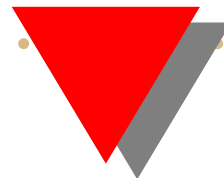
*Use supports of known itemsets to
estimate support of other itemsets
[PKDD02]*



Bonferroni-type inequalities for supports

The following inequalities hold for any $t \in \mathbb{N}$:

$$\text{supp}(a_1 a_2 \dots a_m) \leq \sum_{k=0}^{2t} (-1)^k \binom{m-k-1}{2t-k} S_k$$
$$\text{supp}(a_1 a_2 \dots a_m) \geq \sum_{k=0}^{2t+1} (-1)^{k+1} \binom{m-k-1}{2t+1-k} S_k$$



Problem:

Bonferroni-type inequalities can be applied only if supports of all itemsets up to a certain size are known — usually not the case in data-mining

Solution:

Apply Bonferroni inequalities recursively



Example

A	B	C	Frequency
0	0	0	0
0	0	1	0
0	1	0	0.10
0	1	1	0.25

A	B	C	Frequency
1	0	0	0.10
1	0	1	0.25
1	1	0	0.05
1	1	1	0.25

minsupp= 0.35

Frequent itemsets:

Itemset	Support
A	0.65
B	0.65
C	0.75
AC	0.50
BC	0.50

Want bounds on $\text{supp}(ABC)$.

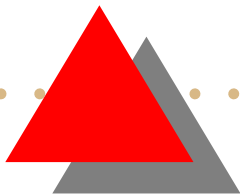


We have

$$\text{supp}(ABC) \leq 1 - \text{supp}(A) - \text{supp}(B) - \text{supp}(C) \\ + \text{supp}(AB) + \text{supp}(AC) + \text{supp}(BC)$$

but $\text{supp}(AB) \leq \text{minsupp}$ so

$$\text{supp}(ABC) \leq 1 - \text{supp}(A) - \text{supp}(B) - \text{supp}(C) \\ + \text{minsupp} + \text{supp}(AC) + \text{supp}(BC) = 0.3$$





Future research

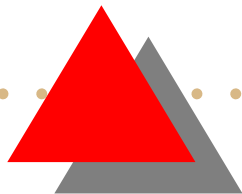


Rule pruning using background knowledge

Retain only rules which are unexpected with respect to what we already know

Representation of background knowledge

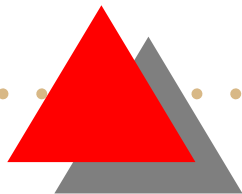
- Bayesian networks
- Loglinear models
- . . .





Other Bonferroni-type inequalities

Develop tighter bounds for sizes of specific types of queries like monotonic functions





*Application to genetic data:
ongoing research*



Thank you for your attention

