Mining the Semantic Web:

the Knowledge Discovery Process in the SW

Claudia d'Amato
Department of Computer Science
University of Bari
Italy

Grenoble, January 24 - EGC 2017 Winter School

Knowledge Disovery: Definition

Knowledge Discovery (KD)

“the process of automatically searching large volumes of data for *patterns* that can be considered ***knowledge*** about the data” [Fay'96]

Knowledge

awareness or *understanding of facts*, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning

What is a Pattern?

An expression **E** in a given language L *describing* a subset F_E of facts F.

E is called pattern if it is simpler than enumerating facts in F_E

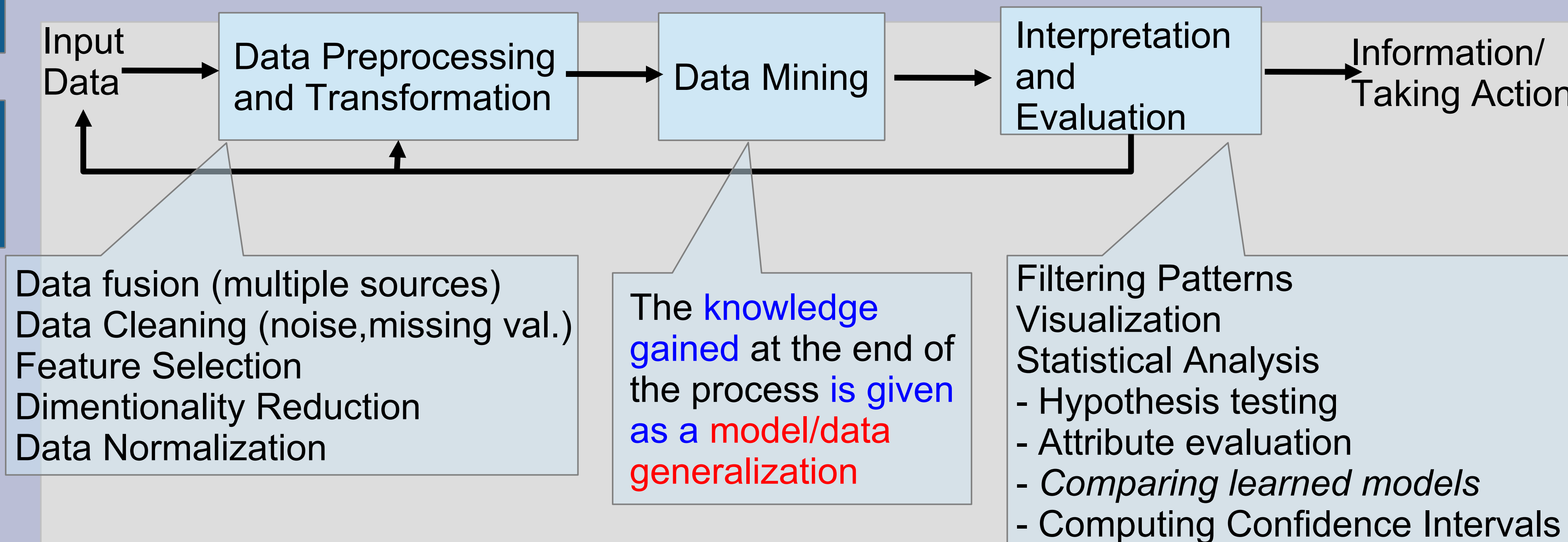
Patterns need to be:

- New – *Hidden in the data*
- Useful
- Understandable

Knowledge Discovery and Data Minig

- KD is often related with Data Mining (DM) field
- DM is one step of the "Knowledge Discovery in Databases" process (KDD)[Fay'96]
- DM is the *computational process of discovering patterns in large data* sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and databases.
- **DM goal:** extracting information from a data set and transforming it into an understandable structure/representation for further use

The KDD process



The most labourous and time consuming step

CRISP-DM (Cross Industry Standard Process for Data Mining) *alternative process model* developed by a consortium of several companies

All data mining methods use ***induction-based learning***

The KDD process

Data fusion (multiple sources)
Data Cleaning (noise,missing val.)
Feature Selection
Dimentionality Reduction
Data Normalization

The knowledge gained at the end of the process is given as a model/data generalization

Filtering Patterns
Visualization
Statistical Analysis
- Hypothesis testing
- Attribute evaluation
- Comparing learned models
- Computing Confidence Intervals

The most labourous and time consuming step

CRISP-DM (Cross Industry Standard Process for Data Mining) *alternative process model* developed by a consortium of several companies

All data mining methods use ***induction-based learing***

Data Mining Tasks...

- **Predictive Tasks**: predict the value of a particular attribute (called target or dependent variable) based on the value of other attributes (called explanatory or independent variables)

Goal: learning a model that minimizes the error between the predicted and the true values of the target variable

- Classification → *discrete* target variables
- Regression → *continuous* target variables

...Data Mining Tasks...

Examples of Classification tasks

- Predict customers that will respond to a marketing compain
- Develop a profile of a “successfull” person

Examples of Regression tasks

- Forecasting the future price of a stock

… Data Mining Tasks...

- **Descriptive tasks:** discover patterns (correlations, clusters, trends, trajectories, anomalies) summarizing the underlying relationship in the data
  - **Association Analysis:** discovers (*the most interesting*) patterns describing strongly associated features in the data/relationships among variables
  - Cluster Analysis: discovers groups of closely related facts/observations. Facts belonging to the same cluster are more similar each other than observations belonging other clusters

...Data Mining Tasks...

Examples of Association Analysis tasks

- Market Basket Analysis
  - Discoverying interesting relationships among retail products. To be used for:
    - Arrange shelf or catalog items
    - Identify potential cross-marketing strategies/cross-selling opportunities

Examples of Cluster Analysis tasks

- Automaticaly grouping documents/web pages with respect to their main topic (e.g. sport, economy...)

… Data Mining Tasks

- Anomaly Detection: identifies facts/observations (Outlier/change/deviation detection) having characteristics significantly different from the rest of the data. A good anomaly detector has a high detection rate and a low false alarm rate.
    - Example: Determine if a credit card purchase is fraudolent → **Imbalance learning setting**

Approaches:

- *Supervised:* build models by using input attributes to predict output attribute values
- *Unsupervised:* build models/patterns without having any output attributes

The KDD process

Data fusion (multiple sources)
Data Cleaning (noise,missing val.)
Feature Selection
Dimentionality Reduction
Data Normalization

The most labourous and time consuming step

The knowledge gained at the end of the process is given as a model/ data generalization

Filtering Patterns
Visualization
Statistical Analysis
- Hypothesis testing
- Attribute evaluation
- Comparing learned models
- Computing Confidence Intervals

CRISP-DM (Cross Industry Standard Process for Data Mining) *alternative process model* developed by a consortium of several companies

All data mining methods use ***induction-based learing***

A closer look at the Evalaution step

Given

- DM task (i.e. Classification, clustering etc.)
- A particular problem for the chosen task

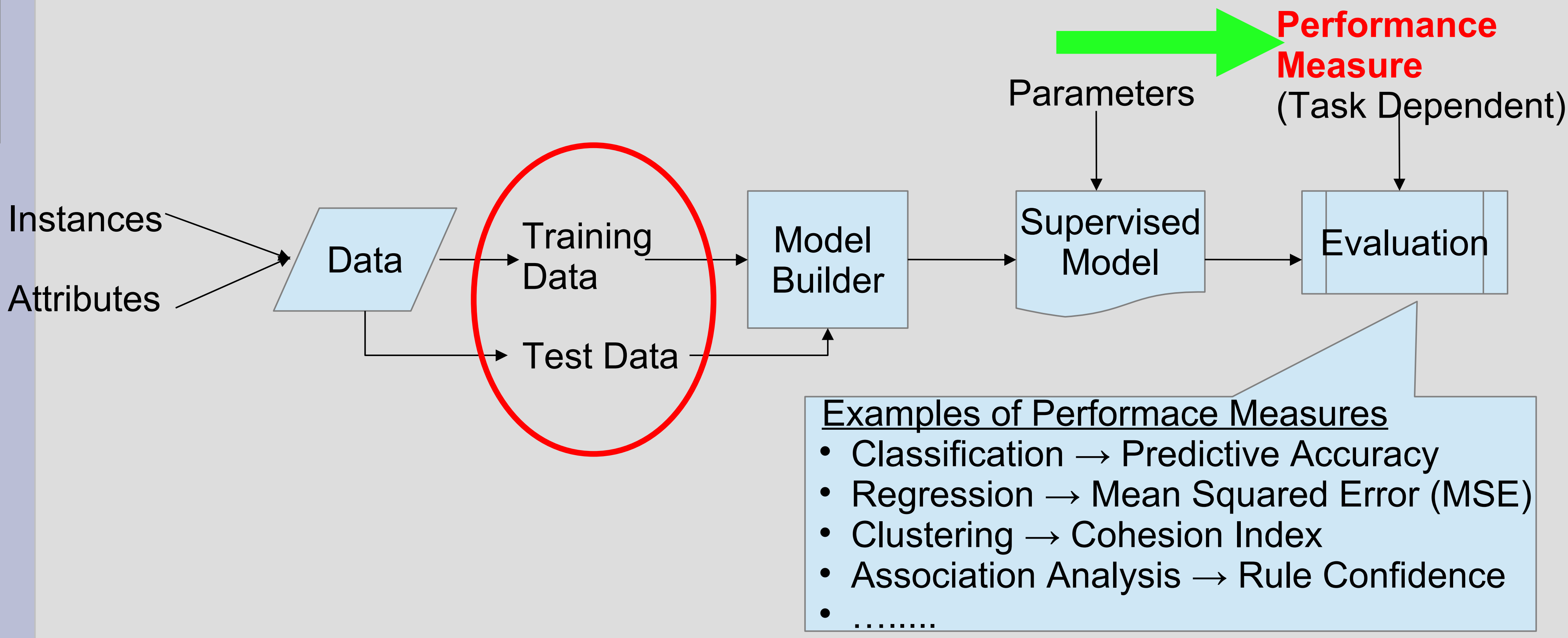
Several DM algorithms can be used to solve the problem

1) **How to assess the performance of an algorithm?**
2) How to compare the performance of different algorithms solving the same problem?

Evaluating the Performance of an Algorithm

Assessing Algorithm Performances

Components for **supervised learning** [Roiger'03]



Test data missing in unsupervised setting

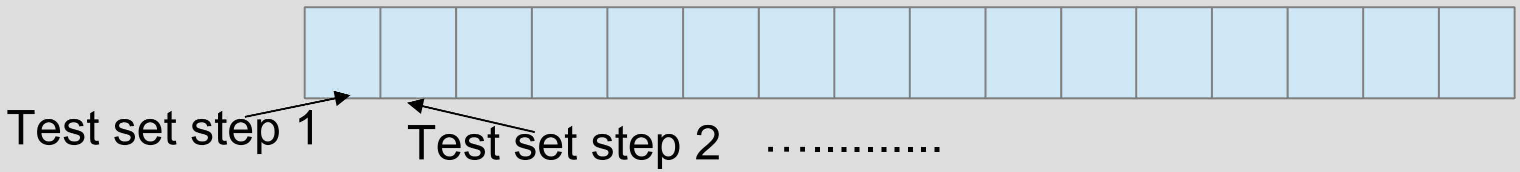
Supervised Setting: Building Training and Test Set

Necessary to predict performance bounds based with whatever data (independent test set)

- Split data into **training** and **test set**
  - The ***repeated and stratified k-fold cross-validation*** is the most widly used technique
  - *Leave-one-out* or *bootstrap* used for small datasets
- Make a model on the training set and evaluate it out on the test set [Witten'11]
  - e.g. Compute predictive accuracy/error rate

K-Fold Cross-validation (CV)

- First step: split data into *k* subsets of equal size
- Second step: use each subset in turn for testing, the remainder for training



- Subsets often *stratified* → reduces variance
- Error estimates averaged to yield the overall error estimate

- Even better: **repeated** stratified cross-validation
  - E.g. 10-fold cross-validation is repeated 15 times and results are averaged → reduces the variance

Leave-One-Out cross-validation

- Leave-One-Out → a particular form of cross-validation:
  - Set number of folds to number of training instances
  - I.e., for *n* training instances, build classifier *n* times
  - The results of all n judgement are averaged for determining the final error estimate
- Makes best use of the data for training
- Involves no random subsampling
- There's no point in repeating it → the same result will be obtained each time

The bootstrap

- CV uses sampling *without replacement*
  - The same instance, once selected, cannot be selected again for a particular training/test set

- *Bootstrap* uses sampling *with replacement*
  - Sample a dataset of ***n*** instances ***n*** times *with replacement* to form a new dataset
  - Use <u>this new dataset</u> as the <u>training set</u>
  - Use <u>the remaining instances</u> not occurring in the <u>training set for testing</u>
  - Also called the *0.632 bootstrap* → *<u>The training data will contain approximately 63.2%</u> of the total instances*

Estimating error with the bootstrap

The error estimate of the true error on the test data will be very pessimistic

- Trained on just ~63% of the instances

- Therefore, combine it with the resubstitution error:

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

- The resubstitution error (error on training data) gets less weight than the error on the test data
- **Repeat the bootstrap procedure** several times with different replacement samples; **average the results**

Comparing Algorithms Performances
For Supervised Aproach

Comparing Algorithms Performance

Frequent question: which of two learning algorithms performs better?

Note: this is domain dependent!

Obvious way: compare the error rates computed by the use of k-fold CV estimates

Problem: variance in estimate on a single 10-fold CV

Variance can be reduced using repeated CV

However, we still don't know whether the results are reliable

Significance tests

- Significance tests tell how confident we can be that there really is a difference between the two learning algorithms
- ***Statistical hypothesis test* exploited** → used for testing a statistical hypothesis
  - *Null hypothesis*: there is no significant ("real") difference (between the algorithms)
  - *Alternative hypothesis*: there is a difference
- Measures how much evidence there is in favor of rejecting the null hypothesis for a specified level of significance
  - Compare two learning algorithms <u>by comparing e.g. the average error rate *over several cross-validations*</u> (see [Witten'11] for details)

DM methods and SW: A closer Look

DM methods and SW: a closer look

- Classical DM algorithms originally developed for propositional representations
- Some upgrades to (multi-)relational and graph representations defined

Semantic Web: characterized by

- Rich/expressive representations (RDFS, OWL)
  - How to cope with them when applying DM algorithms?
- Open world Assumpion (OWA)
  - DM algorithms grounded on CWA
  - Are metrics for classical DM tasks still applicable?

Exploiting DM methods in SW: Problems and Possible Solutions

Classification

Exploiting DM methods in SW: Problems and Possible Solutions...

- Approximate inductive instance retrieval
  - assess the class membership of the individuals in a KB w.r.t. a query concept [d'Amato'08, Fanizzi'12, Rizzo'15]
- (Hyerarchical) Type prediciton
  - Assess the type of instances in RDF datasets [Melo'16]
- Link Prediction
  - Given an individual and a role *R*, predict the other individuals *a* that are in *R* relation with [Minervini'14-'16]

Regarded as a classification task → (semi-)automatic ontology population

…Exploiting DM methods in SW: Problems and Possible Solutions...

Classification task → assess the class membership of individuals in an ontological KB w.r.t. the query concept

What is the value added?

- Perfom some form of reasoning on inconsistent KB
- Possibly induce new knowledge not logically derivable

State of the art classification methods cannot be straightforwardly applied

- generally applied to feature vector representation

→ upgrade expressive representations

- implicit Closed World Assumption made

→ cope with the OWA (made in DLs)

…Exploiting DM methods in SW: Problems and Possible Solutions...

Problem Definition

Given:

- a populated ontological knowledge base KB = (T ,A)
- a query concept Q
- a training set with {+1, -1, 0} as target values (OWA taken into account)

Learn a classification function f such that: $\forall a \in$ Ind(A):

- f (a) = +1 if a is instance of Q
- f (a) = -1 if a is instance of ¬Q
- f (a) = 0 otherwise

…Exploiting DM methods in SW: Problems and Possible Solutions...

Dual Problem

- given an individual $a \in Ind(A)$, determine concepts $C_1, \ldots., C_k$ in KB it belongs to

the multi-class classification problem is decomposed into a set of ternary classification problems (one per target concept)

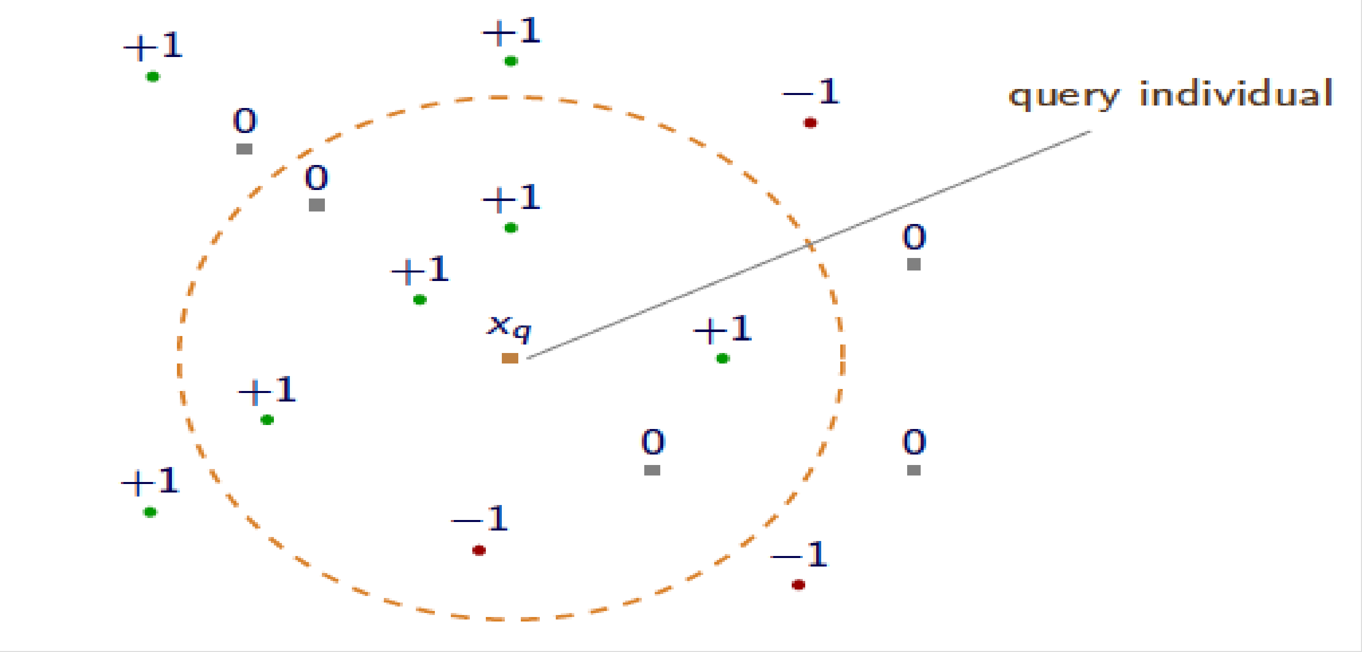
…Exploiting DM methods in SW: Problems and Possible Solutions…

Example: Nearest Neighbor based Classification

Query concept: `Bank` k = 7

Training set with Target values: {+1, 0, -1}

Similarity Measures for DLs [d'Amato et al. @ EKAW'08]



$f(x_q) \leftarrow +1$

…Exploiting DM methods in SW: Problems and Possible Solutions...

Evaluating the Classifier

- Inductive Classification compared with a standard reasoner
- Registered mismatches: Ind. {+1,-1} - Deduction: no results
- Evaluated as mistake if **precision** and **recall** used while it could turn out to be a correct inference if judged by a human

Defined new metrics to distinguish induced assertions from mistakes [d'Amato'08]

M Match Rate

C Comm. Err. Rate

O Omis. Err. Rate

I Induct. Rate

		Reasoner		
		+1	0	-1
Inductive Classifier	+1	**M**	**I**	**C**
	0	**O**	**M**	**O**
	-1	**C**	**I**	**M**

...Exploiting DM methods in SW: Problems and Possible Solutions...

Pattern Discovery

...Exploiting DM mthods in SW: Problems and Possible Solutions

- Semi-automatic ontology enrichment [d'Amato'10,Völker'11, Völker'15,d'Amato'16]
  - exploiting the evidence coming from the data → discovering hidden knowledge patterns in the form of relational association rules
  - new axioms may be suggested → existing ontologies can be extended

Regarded as a pattern discovery task

Associative Analysis: the Pattern Discovery Task

Problem Definition:

Given a dataset

find

- all possible *hidden pattern* in the form of Association Rule (AR)
- having <u>support</u> and <u>confidence</u> greater than a minimum thresholds

Definition: An AR is an implication expression of the form X → Y where X and Y are <u>disjoint</u> itemsets

An AR expresses a co-occurrence relationship between the items in the antecedent and the concequence <u>not a causality relationship</u>

Basic Definitions

- An *itemset* is a finite set of assignments of the form $\{A_1 = a_1, \ldots, A_m = a_m\}$ where A_i are attributes of the dataset and a_i the corresponding values

- The support of an itemset is the number of istances/tuples in the dataset containing it.

Similarily, support of a rule is $s(X \rightarrow Y) = |(X \cup Y)|$;

- The confidence of a rule provides how frequently items in the consequence appear in instances/tuples containing the antencedent

$c(X \rightarrow Y) = |(X \cup Y)| / |(X)|$ (seen as $p(Y|X)$)

Discoverying Association Rules: General Approach

Articulated in two main steps [Agrawal'93, Tan'06]:

1. Frequent Patterns Generation/Discovery (generally in the form of *itemsets*) wrt a minimum frequency (support) threshold
    - *Apriori* algortihm → The most well known algorithm
    - the most expensive computation;
2. Rule Generation
    - Extraction of all the *high-confidence* association rules from the discovered frequent patterns.

Apriori Algortihm: Key Aspects

- Uses a level-wise generate-and-test approach
- Grounded on the non-monotonic property of the support of an itemset
  - The support of an itemset never exceeds the support of its subsets
- ***Basic principle***:
  - if an itemset is frequent → all its subsets must also be frequent
  - If an itemset is infrequent → all its supersets must be infrequent too
  - Allow to sensibly cut the search space

Apriori Algorithm in a Nutshell

Goal: Finding the frequent itemsets ↔ the sets of items that satisfying the min support threshold

Iteratively find frequent itemsets with lenght from 1 to k (k-itemset)

Given a set L_{k-1} of frequent (k-1)itemset, join L_{k-1} with itself to obain L_k the candidate k-itemsets

Prune items in L_k that are not frequent (Apriori principle)

If L_k is not empty, generate the next candidate (k+1)itemset until the frequent itemset is empty

Apriori Algorithm: Example...

Suppose having the transaction table

(Boolean values considered for simplicity)

ID	List of Items
T1	{I1,I2,I5}
T2	{I2,I4}
T3	{I2,I3}
T4	{I1,I2,I4}
T5	{I1,I3}
T6	{I2,I3}
T7	{I1,I3}
T8	{I1,I2,I3,I5}
T9	{I1,I2,I3}

Apply APRIORI algorithm

...Apriori Algorithm: Example...

Itemset	Sup. Count.
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Min. Supp. 2
Pruning

Output After Pruning

Itemset	Sup. Count.
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Join for candidate generation

Itemset	Sup. Count.
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

L_2

Min. Supp. 2
Pruning

...Apriori Algorithm: Example

Apply Apriori principle

Itemset	Sup. Count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

Output After Pruning

Join for candidate generation

Itemset	Prune Infrequent
{I1,I2,I3}	No
{I1,I2,I5}	No
{I1,I2,I4}	Yes {I1,I4}
{I1,I3,I5}	Yes {I3,I5}
{I2,I3,I4}	Yes {I3,I4}
{I2,I3,I5}	Yes {I3,I5}
{I2,I4,I5}	Yes {I4,I5}

L_3

Min. Supp. 2

Pruning

Itemset	Sup. Count.
{I1,I2,I3}	2
{I1,I2,I5}	2

Output After Pruning

Join for candidate generation

Itemset	Prune Infrequent
{I1,I2,I3,I5}	Yes {I3,I5}

L_4

Empty Set

STOP

Generating ARs from frequent itemsets

- For each frequent itemset “I”
  - generate all non-empty subsets S of I
- For every non empty subset S of I
  - compute the rule *r* := “S → (I-S)”
- **If** conf(*r*) > = min confidence
  - **then** output *r*

Genrating ARs: Example...

Given:

L = { {I1}, {I2}, {I3}, {I4}, {I5}, {I1,I2}, {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}, {I1,I2,I3}, {I1,I2,I5} }.

Let us fix 70% for the Minimum confidence threshold

- Take I = {I1,I2,I5}.
- All nonempty subsets are {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}.

The resulting ARs and their confidence are:

- R1: I1 AND I2 →I5

Conf(R1) = supp{I1,I2,I5}/supp{I1,I2} = 2/4 = 50% REJECTED

...Generating ARs: Example...

Min. Conf. Threshold 70%; I = {I1,I2,I5}.

- All nonempty subsets are {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}.

The resulting ARs and their confidence are:

- R2: I1 AND I5 $\rightarrow$I2

Conf(R2) = supp{I1,I2,I5}/supp{I1,I5} = 2/2 = 100% RETURNED

- R3: I2 AND I5 $\rightarrow$ I1

Conf(R3) = supp{I1,I2,I5}/supp{I2,I5} = 2/2 = 100% RETURNED

- R4: I1 $\rightarrow$ I2 AND I5

Conf(R4) = sc{I1,I2,I5}/sc{I1} = 2/6 = 33% REJECTED

...Genrating ARs: Example

Min. Conf. Threshold 70%; I = {I1,I2,I5}.

- All nonempty subsets: {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}.

The resulting ARs and their confidence are:

- R5: I2 → I1 AND I5

Conf(R5) = sc{I1,I2,I5}/sc{I2} = 2/7 = 29% REJECTED

- R6: I5 → I1 AND I2

Conf(R6) = sc{I1,I2,I5}/ {I5} = 2/2 = 100% RETURNED

Similarily for the other sets I in L (Note: it does not make sense to consider an itemset made by just one element i.e. {I1})

On improving Discovery of ARs

Apriori algorithm may degrade significantly for dense datasets

Alternative solutions:

- FP-growth algorithm outperforms Apriori
  - Does not use the generate-and-test approach
  - Encodes the dataset in a compact data structure (FP-Tree) and extract frequent itemsets directly from it
- Usage of additional interenstingness metrics (besides support and confidence) (see [Tan'06])
  - Lift, Interest Factor, correlation, IS Measure

Pattern Discovery on RDF data sets for Making Predictions

Discoverying ARs from RDF data sets → for making predictions

Problems:

- Upgrade to Relational Representation (need variables)
- OWA to be taken into accout
- Background knowledge should be taken into account
- ARs are exploited for making predictions
    - New metrics, considering the OWA, for evaluating the results, are necessary

Proposal [Galarraga'13-'15]

- Inspired to the general framework for discovering frequent Datalog patterns [Dehaspe'99; Goethals et al'02]
- Grounded on level-wise generate-and-test approach

Pattern Discovery on RDF data sets for Making Predictions

Start: initial general pattern, single atom → role name (plus variable names)

Proceed: at each level with

- specializing the patterns (use of suitable operators)
  - Add an atom sharing at least one variable/constant
- evaluating the generated specializations for possible pruning

Stop: stopping criterion met

A rule is a list of atoms (interpreted as a conjunction) where the first one represents the head

The specialization operators represent the way for exploring the search space

Pattern Discovery on Populated Ontologies for Making Predictions

Pros: Scalable method

Limitations:

- Any background/ontological KB taken into account
- No reasoning capabilites exploited
- Only role assertions could be predictied

Upgrade: Discovery of ARs from ontologies [d'Amato'16]

- Exploits the available background knowledge
- Exploits deductive reasoning capabilities

Discovered ARs can make **concept** and role predictions

Pattern Discovery on Populated Ontologies for Making Predictions

Start: initial general pattern

- concept name (plus a variable name) or a role name plus variable names)

Proceed: at each level with:

- specializing the patterns (use of suitable operators)
  - Add a concept or role atom sharing at least one variable
- evaluating the generated specializations for possible pruning

Stop: stopping criterion met

A rule is a list of atoms (interpreted as a conjunction) where the first one represents the head

Pattern Discovery on Populated Ontologies for Making Predictions

For a given pattern all possible specializations are generated by applying the operators:

- Add a concept atom: adds an atom with a concept name as a predicate symbol and an already appearing variable as argument
- Add a role atom: adds an atom with a role name as a predicate symbol; at least one variable already appears in the pattern

The Operators are applied so that always connected and non-redundant rules are obtained

Additional operators for tanking into account constants could be similarly considered

Pattern Discovery on Populated Ontologies for Making Predictions

Language Bias (ensuring decidability)

- Safety condition: all variables in the head must appear in body
- Connection: atoms share at least one variable or constant
- Interpretation under **DL-Safety** condition: all variables in the rule bind only to known individuals in the ontology
- Non Redundancy: there are no atoms that can be derived by other atoms

Example (Redundant Rule)
Given K made by the TBox T = {Father $\sqsubseteq$ Parent} and the rule
r := Father(x) ∧ Parent(x) ⇒ Human(x)

r **redundant** since Parent(x) is entailed by Father(x) w.r.t. K.

Specializing Patterns: Example

- Pattern to be specialized: $C(x) \wedge R(x,y)$

Non redundant Concept D

Refined Patterns
$C(x) \wedge R(x,y) \wedge D(x)$
$C(x) \wedge R(x,y) \wedge D(y)$

Non redundant Role S
Fresh Variable z

Refined Patterns
$C(x) \wedge R(x,y) \wedge S(x,z)$
$C(x) \wedge R(x,y) \wedge S(z,x)$
$C(x) \wedge R(x,y) \wedge S(y,z)$
$C(x) \wedge R(x,y) \wedge S(z,y)$

Non redundant Role S
All Variables Bound

Refined Patterns
$C(x) \wedge R(x,y) \wedge S(x,x)$
$C(x) \wedge R(x,y) \wedge S(x,y)$
$C(x) \wedge R(x,y) \wedge S(y,x)$
$C(x) \wedge R(x,y) \wedge S(y,y)$

Pattern Discovery on Populated Ontologies for Making Predictions

- Rule predictitng concept/role assertions
- The method is actually able to prune redundat and inconsistent rules
  - thanks to the exploitation of the background knowledge and resoning capabilities

Problems to solve/research directions:

- Scalability
  - investigate on additional heuristics for cutting the search space
  - Indexing methods for caching the results of the inferences made by the reasoner
- Output only a subset of patterns by the use a suitable interestingness measures (potential inner and post pruning)

Conclusions

- Surveyed the classical KDD process
  - Data mining tasks
  - Evaluation of algorithms
- Analized some differences of the KD process when RDF/OWL knowledge bases are considered
  - Expressive representation language
  - OWA vs. CWA
  - New metrics for evaluating the algorithms
- Analized existing solutions
- Open issues and possible research directions

References...

- [Fay'96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From Data Mining to Knowledge Discovery: An Overview. Advances in Knowledge Discovery and Data Mining, MIT Press (1996)
- [Agrawal'93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. Proc. of Int. Conf. on Management of Data, p. 207–216. ACM (1993)
- [d'Amato'10] C. d'Amato, N. Fanizzi, F. Esposito: Inductive learning for the Semantic Web: What does it buy? Semantic Web 1(1-2): 53-59 (2010)
- [Völker'11] J. Völker, M. Niepert: Statistical Schema Induction. ESWC (1) 2011: 124-138 (2011)
- [Rizzo'15] G. Rizzo, C. d'Amato, N. Fanizzi, F. Esposito: Inductive Classification Through Evidence-Based Models and Their Ensembles. Proceedings of ESWC 2015: 418-433 (2015)

...References...

- [Völker'15] J. Völker, D. Fleischhacker, H. Stuckenschmidt: Automatic acquisition of class disjointness. J. Web Sem. 35: 124-139 (2015)
- [d'Amato'16] C. d'Amato, S. Staab, A.G.B. Tettamanzi, T. Minh, F.L. Gandon. Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases.SAC 2016:333-338
- [d'Amato'08] C. d'Amato, N. Fanizzi, F. Esposito: Query Answering and Ontology Population: An Inductive Approach. ESWC 2008: 288-302
- [Tan'06] P.N. Tan, M. Steinbach, V. Kumar. Introduction to Data Mining. Ch. 6 Pearson (2006) http://www.users.cs.umn.edu/~kumar/dmbook/ch6.pdf
- [Aggarwal'10] C. Aggarwal, H. Wang. Managing and Mining Graph Data. Springer, 2010

...References...

- [Witten'11] I.H. Witten, E. Frank. Data Mining: Practical Machine Learning Tool and Techiques with Java Implementations. Ch. 5. Morgan-Kaufmann, 2011 (3rd Edition)
- [Fanizzi'12] N. Fanizzi, C. d'Amato, F. Esposito: Induction of robust classifiers for web ontologies through kernel machines. J. Web Sem. 11: 1-13 (2012)
- [Minervini'14] P. Minervini, C. d'Amato, N. Fanizzi, F. Esposito: Adaptive Knowledge Propagation in Web Ontologies. Proc. of EKAW Conferece. Springer. pp. 304-319, 2014.
- [Minervini'16] P. Minervini, C. d'Amato, N. Fanizzi, V. Tresp: Discovering Similarity and Dissimilarity Relations for Knowledge Propagation in Web Ontologies. J. Data Semantics 5(4): 229-248 (2016)
- [Roiger'03] R.J. Roiger, M.W. Geatz. Data Mining. A Tutorial-Based Primer. Addison Wesley, 2003

...References

- [Melo'16] A. Melo, H. Paulheim, J. Völker. Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification. WIMS 2016: 14
- [Galárraga'13] L. Galárraga, C. Teflioudi, F. Suchanek, K. Hose. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. Proc. of WWW 2013. http://luisgalarraga.de/docs/amie.pdf
- [Fanizzi'09] N. Fanizzi, C. d'Amato, F. Esposito: Metric-based stochastic conceptual clustering for ontologies. Inf. Syst. 34(8): 792-806 (2009)
- [Galárraga'15] L. Galárraga, C. Teflioudi, F. Suchanek, K. Hose. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. VLDB Journal 2015. http://suchanek.name/work/publications/vldbj2015.pdf