PhD Dissertation

Document Classification Models based on Bayesian networks

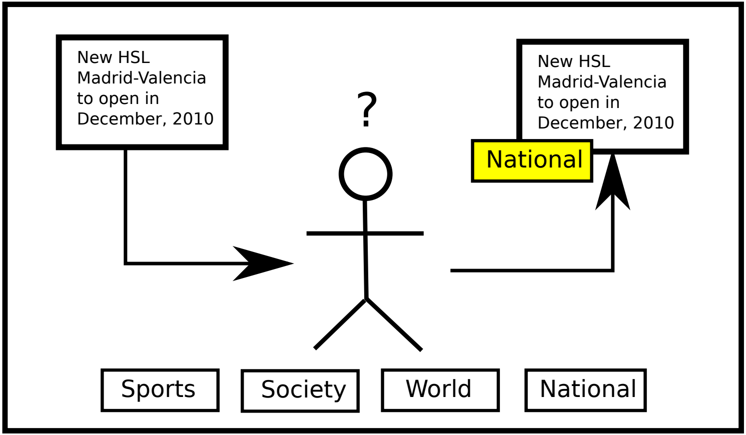
Alfonso E. Romero

April 27, 2010

Advisors: Luis M. de Campos, and
Juan M. Fernández-Luna
Department of Computer Science and A.I.
University of Granada

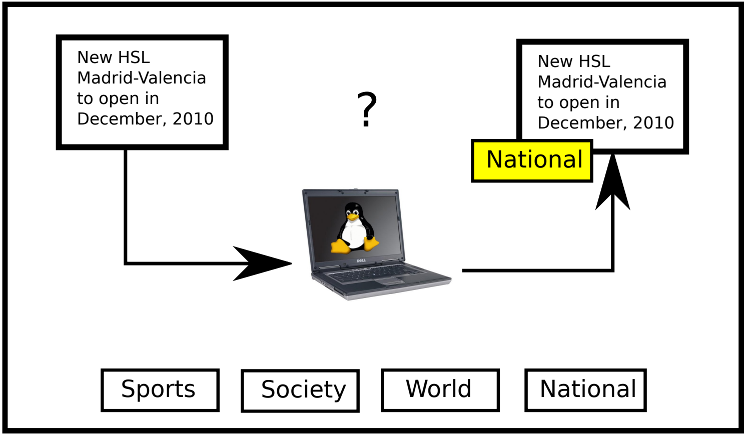
A brief overview: the problem I

We shall solve problems in **document categorization...**



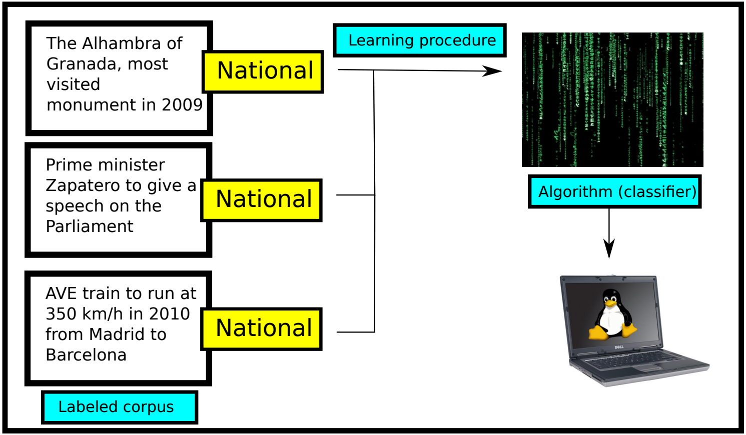
A brief overview: the problem II

... in particular, automatic **document categorization**...



A brief overview: the problem III

... concretely, supervised **document categorization.**



A brief overview: the methods I

Which **learning method** shall we use?

Neural networks

Support Vector Machines

k-$\mathcal{NN}$methods

Bayesian networks

and probabilistic methods

Decision trees

Evolutive algorithms

A brief overview: the methods II

The **answer**:

Neural networks

Support Vector Machines

***k*-$\mathcal{NN}$ methods**

Bayesian networks

and probabilistic methods

Decision trees

Evolutive algorithms

A brief overview: the methods III

But, **why**?

- Strong **theoretical foundation** (probability theory).
- Models for (uncertain) **knowledge representation.**
- Great **success in related tasks** (IR).
- **Our background** at the group UTAI.

Outline

1. Text Categorization

2. Bayesian networks

3. An OR Gate-Based Text Classifier

4. Automatic Indexing From a Thesaurus Using Bayesian Networks

5. Structured Document Categorization Using Bayesian Networks

6. Final Remarks

Outline

⇒ Text Categorization

2. Bayesian networks

3. An OR Gate-Based Text Classifier

4. Automatic Indexing From a Thesaurus Using Bayesian Networks

5. Structured Document Categorization Using Bayesian Networks

6. Final Remarks

Supervised Text Categorization I

Provided

1. Set of **labeled** documents $\mathcal{D}_{Tr}$ (training).
2. $\mathcal{C}$, set of categories/labels.

The goal is to build a model f (**classifier**) capable of predicting categories (of $\mathcal{C}$) of documents in $\mathcal{D}$.

Different **kinds of labeling**:

- $f : \mathcal{D} \rightarrow \{c, \overline{c}\}$ (binary).
- $f : \mathcal{D} \rightarrow \{c_1, c_2, \ldots, c_n\}$ (multiclass).
- $f : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$ (multilabel).

A multilabel problem reduces to $|\mathcal{C}|$ binary problems $\mathcal{C} = \{c, \overline{c}\}$. We often change the codomain from $\{0, 1\}$ (**hard classification**) to $[0, 1]$ (**soft classification**).

Supervised Text Categorization II

Document representation:

As in **Information Retrieval**

Stopwords removal + stemming + Vector representation (Frequency, binary or tf-idf).

From (preprocessed) document to vector

term ⇔ dimension

Example (beginning of John Milton's "Lost Paradise"):

Of Mans First Disobedience, and the Fruit
Of that Forbidden Tree, whose mortal tast
Brought Death into the World, and all our woe,
With loss of Eden, till one greater Man...

Of **Mans** First **Disobedience**, and the **Fruit**
Of that **Forbidden Tree**, whose **mortal tast**
Brought Death into the **World**, and all our **woe**,
With **loss** of **Eden**, till one **greater Man**...

2	1	1	1	1	1	1	1	1	1	1	1	1	1
man	obedience	fruit	forbid	tree	mortal	tast	bring	death	world	woe	loss	eden	great

Supervised Text Categorization III
What are the particularities of the problem?

It differs from a “classic” Machine Learning problem in:

- **High dimensionality** (easily > 10000).
- Very **unbalanced** datasets.
- $|\mathcal{C}| \gg 0$.
- Sometimes, there is a **hierarchy in the set** $\mathcal{C}$.
- Sometimes, **explicit relationships among documents** are given.

Supervised Text Categorization IV
Evaluation

How to measure the correctness of **documents assigned to set of categories**?

- **Binary/multiclass:**
  - **Hard categorization** *Precision:* $\frac{TP}{TP+FP}$ and *Recall:* $\frac{TP}{TP+FN}$, F_1: $\frac{2PR}{P+R}$.
  - **Soft categorization** Precision/Recall BEP.
- **Multilabel: micro** and **macro** averages.
- Also, average precision on the 11 std. recall points (**category ranking**).
- **Standard corpora**: Reuters, Ohsumed, 20 NG…

Outline

1. Text Categorization

⇒ Bayesian networks

3. An OR Gate-Based Text Classifier

4. Automatic Indexing From a Thesaurus Using Bayesian Networks

5. Structured Document Categorization Using Bayesian Networks

6. Final Remarks

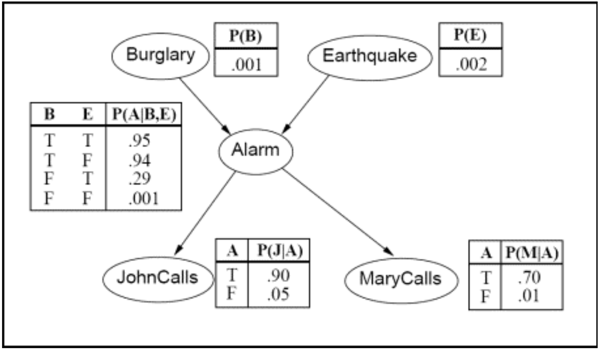
Bayesian networks I

Definition and characteristics

A set of **random variables** $X_1, \ldots, X_N$ in a DAG, verifying $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \Rightarrow$ the graph represents independences.

Causal interpretation.

Learning and **inference** methods available.



The problem

- One value **for each configuration of the parents.**
- General case: **exponential number of parameters on the number of parents**.

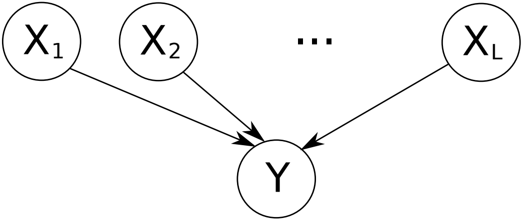
The solution

1. Few parents per node **(not realistic in text).**
2. Write the probability of a node as a deterministic function of the configuration **(canonical model).** Set of parameters with **linear size** on the number of parents.

Bayesian networks: canonical models
Components and examples of canonical models

- $\mathbf{X} = \{X_i\}$: parents **(causes)**, Y: child **(effect).**
- X_i in $\{x_i, \overline{x_i}\}$, Y_i in $\{y_i, \overline{y_i}\}$ (occurrence or not).

1. **Noisy-OR gate model**: $p(y|\mathbf{X}) = 1 - \prod_{X_i \in R(\mathbf{x})}(1 - w_{OR}(X_i, Y))$.
2. **Additive model:** $p(y|\mathbf{X}) = \sum_{X_i \in R(\mathbf{x})} w_{add}(X_i, Y)$.



Outline

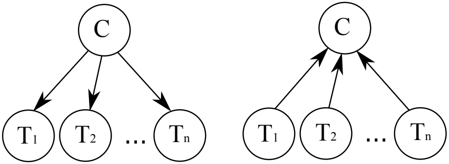
1. Text Categorization
2. Bayesian networks

⇒ An OR Gate-Based Text Classifier

4. Automatic Indexing From a Thesaurus Using Bayesian Networks
5. Structured Document Categorization Using Bayesian Networks
6. Final Remarks

An OR Gate-Based Text Classifier
Why using OR gates?

- The OR gate is a **simple** model (and fast for inference).
- It has gained great **success in knowledge representation**.
- **Discriminative** classifier (models directly $p(c_i|d_j)$) (NB generative).
- Seems natural the **term are causes** and **category the effect**.



An OR Gate-Based Text Classifier
The model

Equations for the OR gate classifier

Probability distributions:

$$
\begin{aligned}
p_i(c_i|pa(C_i)) &= 1 - \prod_{T_k \in R(pa(C_i))} (1 - w(T_k, C_i)) \,, \\
p_i(\overline{c}_i|pa(C_i)) &= 1 - p_i(c_i|pa(C_i)).
\end{aligned}
$$

Inference:

$$
\begin{aligned}
p_i(c_i|d_j) &= 1 - \prod_{T_k \in Pa(C_i)} \left(1 - w(T_k, C_i)\, p(t_k|d_j)\right) \\
&= 1 - \prod_{T_k \in Pa(C_i) \cap d_j} (1 - w(T_k, C_i))\,.
\end{aligned}
$$

Model **characterized** by the $w(T_k, C_i)$ formula.

Weight estimation formulae

Weight $w(T_k, C_i)$ means

$\hat{p}_i(c_i | t_k, \bar{t}_h \forall T_h \in Pa(C_i),\ T_h \neq T_k)$.

1. **ML:** $w(T_k, C_i)$ as $\hat{p}(c_i | t_k)$, $w(T_k, C_i) = \frac{N_{ik}+1}{N_{\bullet k}+2}$ (using Laplace).
2. **TI:** assuming term independence, given the category, $w(T_k, C_i) = \frac{N_{ik}}{nt_i N_{\bullet k}} \prod_{h \neq k} \frac{(N_{i\bullet} - N_{ih})N}{(N - N_{\bullet h})N_{i\bullet}}$.

Notation: N number of words in the training documents. $N_{\bullet k}$ times that term t_k appears in training documents ($N_{\bullet k} = \sum_{c_i} N_{ik}$), $N_{i\bullet}$ is the total number of words in documents of class c_i ($N_{i\bullet} = \sum_{t_k} N_{ik}$), M vocabulary size.

An OR Gate-Based Text Classifier
Pruning independent terms

- The model can be improved **pruning terms** which are **independent with the category.**
- We run an **independence test for each pair term/category**, at a certain **confidence level.** Only terms which pass it are kept.
- The size of the **parent set is highly reduced**, but classification is often **improved.**

An OR Gate-Based Text Classifier

Experiments I: Experimental setting

- We compare a **Multinomial NB**, a **Rocchio**, **OR TI**, **OR ML**, and both OR with **pruning** at levels $\{0.9, 0.99, 0.999\}$.
- We made experiments on **Ohsumed-23**, **Reuters** and **20 NG**.
- **Soft categorization**, evaluated with macro/micro BEP, 11 avg std prec, and macro/micro $F_1@\{1, 3, 5\}$.

Results (# of wins on 9 measures):

- **Reuters:** OR-TI-0.999 (7), OR-TI (1), OR-ML-0.999 (1).
- **Ohsumed:** OR-TI-0.999 (8), OR-ML-0.99 (1).
- **20 NG:** OR-TI (2), OR-ML (2), OR-TI-0.999 (1), NB (4).

An OR Gate-Based Text Classifier

Conclusions

- A new text categorization model, based on noisy OR gates.
- Simple and computationally affordable.
- Results improves Naïve Bayes noticeably.

Future work

- Use more advanced noisy OR models (leaky).
- Use another canonical models.
- Explore another alternatives for term pruning.

Outline

1. Text Categorization
2. Bayesian networks
3. An OR Gate-Based Text Classifier

⇒ Automatic Indexing From a Thesaurus Using Bayesian Networks

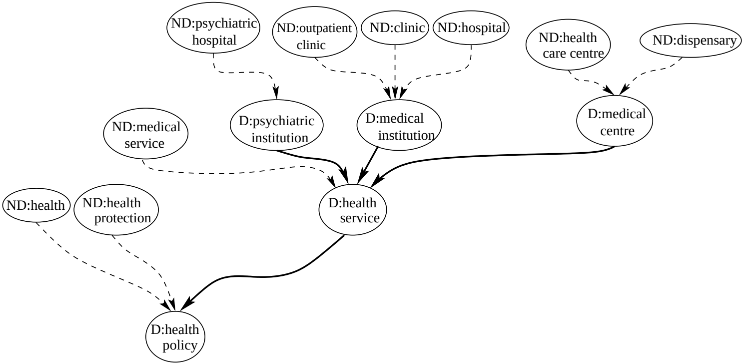
5. Structured Document Categorization Using Bayesian Networks
6. Final Remarks

Automatic Indexing From a Thesaurus Using Bayesian Networks: Thesauri I

Definitions

A thesaurus

A list of terms representing *concepts*, grouped those with the same meaning, with *hierarchical relationships* among them.



Automatic Indexing From a Thesaurus Using Bayesian Networks: Thesauri II

Indexing with a thesauri

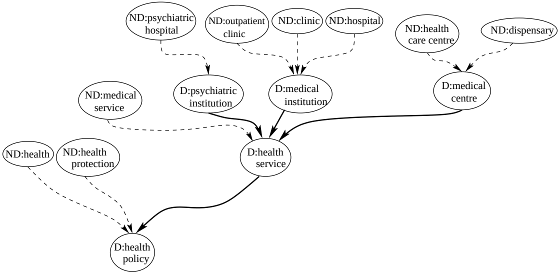
- **Problem:** associating descriptors (keywords) to documents (scientific, medical, legal,...).
- **Manually:** expensive and time consuming work (due to *thousand of descriptors!* [EUROVOC $>$ 6000]). Besides:
  - How many descriptors should we assign?
  - Which descriptor should assign in the hierarchy?

We propose an automatic solution

1. TC problem (categories $\equiv$ descriptors).
2. Makes use of the meta-information of the thesaurus.
3. Unsupervised and supervised case.
4. Based on BNs.

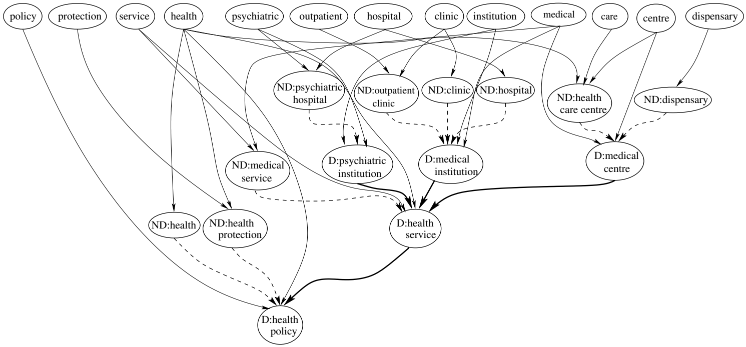
Automatic Indexing From a Thesaurus Using Bayesian Networks: Unsupervised case I

From the thesaurus...



Automatic Indexing From a Thesaurus Using Bayesian Networks: Unsupervised case II

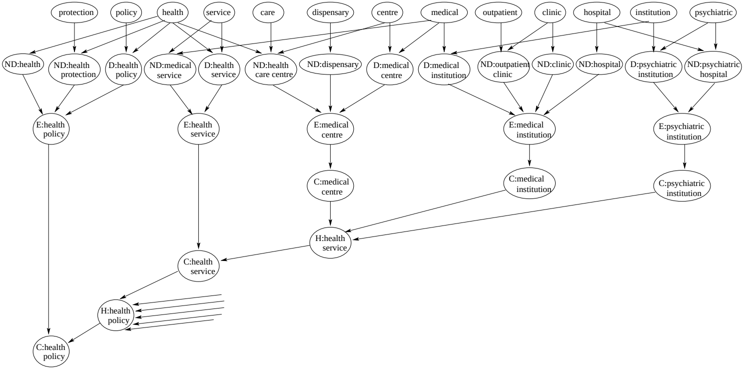
... to the Bayesian network.



- **Binary variables** (relevant/not relevant), for each *term*, *descriptor* and *non descriptor*.
- **Problem:** information of different nature mixed in descriptor nodes!

Automatic Indexing From a Thesaurus Using Bayesian Networks: Unsupervised case III

Introducing concept and virtual nodes



- New *concept nodes*, C.
- New Hierarchy nodes, H_C.
- New Equivalence nodes, E_C.

Automatic Indexing From a Thesaurus Using Bayesian Networks: Unsupervised case IV

Probability distributions

We already have the structure, but we have to **specify the distributions** on each node.

Many parents $\Rightarrow$ **canonical models**.

- D and ND nodes (SUM):
  $p(d^+|pa(D)) = \sum_{T \in R(pa(D))} w(T, D)$.
- H_C nodes (SUM):
  $p(h_c^+|pa(H_C)) = \sum_{C' \in R(pa(H_C))} w(C', H_C)$.
- E_C nodes (OR):
  $p(e_c^+|pa(E_C)) = 1 - \prod_{D \in R(pa(E_C))} (1 - w(D, C))$.
- C nodes (OR):

$$p(c^+|\{e_c, h_c\}) = \begin{cases} 1 - (1 - w(E_C, C))(1 - w(H_C, C)) & \text{if } e_c = e_c^+, h_c = h_c^+ \\ w(E_C, C) & \text{if } e_c = e_c^+, h_c = h_c^- \\ w(H_C, C) & \text{if } e_c = e_c^-, h_c = h_c^+ \\ 0 & \text{if } e_c = e_c^-, h_c = h_c^- \end{cases}$$

Automatic Indexing From a Thesaurus Using Bayesian Networks: Unsupervised case V
Inference

Classification is seen as **inference.**

- Given a **document** d, we set the term variables $T \in d$ to t^+, t^- otherwise.
- **Exact propagation** is carried out:
  - First, to **descriptor** nodes.
  - Then, to E_C nodes.
  - Following a **topological order**, probabilities $p(h_c^+|pa(H_C))$ are computed after their parents $p(c^+|pa(C))$ are set.
- Final $p(c^+|pa(C)), \forall C$ values are **returned.**

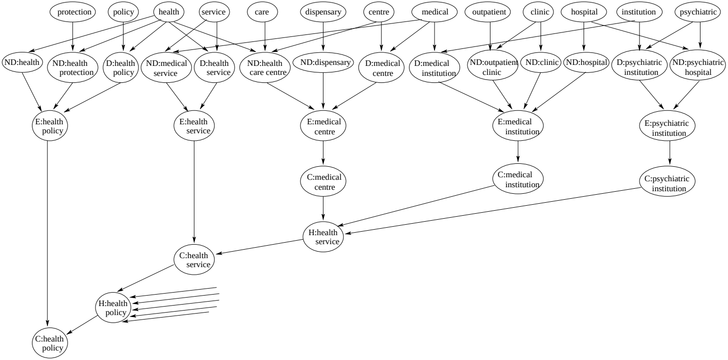
Automatic Indexing From a Thesaurus Using Bayesian Networks: Supervised model I

Changes from the unsupervised case

- Concept also receives **information from labeled documents** in the training set.
- We add a **training node** T_C as new parent of the concept one. The node is an OR gate ML classifier.
- Distributions of C nodes are **modified consequently**.

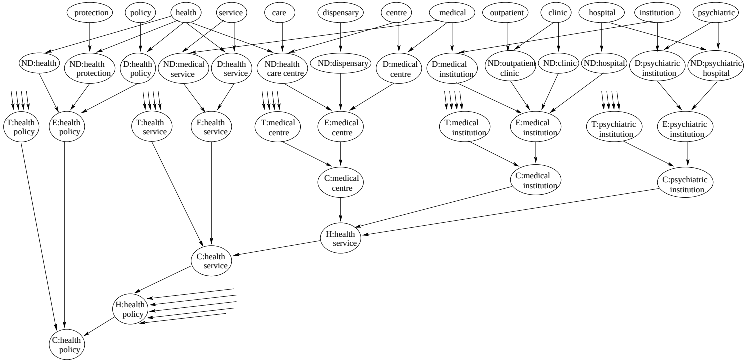
Automatic Indexing From a Thesaurus Using Bayesian Networks: Supervised model II

Graphically: unsupervised



Automatic Indexing From a Thesaurus Using Bayesian Networks: Supervised model II

Graphically: supervised



Automatic Indexing From a Thesaurus Using Bayesian Networks: Experiments I

Description of the collection

- Database from the Andalusian Parliament at Spain, containing **7933 parliamentary resolutions**.
- Classified on the **Eurovoc** thesaurus (**5080 categories**).
- From **1 to 14 descriptors** assigned (average 3.8).
- Each initiative **1 to 3 lines** of text.

Experimentation

1. **Unsupervised:** our model Vs. VSM and HVSM.
2. **Supervised:** our model Vs. SVM, Rocchio and NB.

Micro-macro BEP, F1@5, AV Prec.

Automatic Indexing From a Thesaurus Using Bayesian Networks: Experiments II

Unsupervised experiments

Models	Micro BEP	Macr BEP	Av. prec.	Micro F1@5	Macro F1@5
BN, 0.8, 0	0.26244	0.20394	0.29967	0.30811	0.17661
BN, 0.9, 0	**0.28241**	0.20234	0.30700	**0.31419**	0.18419
BN, 0.8, 0.8	0.26068	0.21208	0.30500	0.30845	0.17521
BN, 0.9, 0.9	**0.26881**	**0.20903**	**0.31321**	**0.31473**	**0.18433**
BN, 0.9, 1.0	0.26636	**0.20880**	**0.31261**	0.31381	0.18265
BN, 1.0, 1.0	0.25584	0.20768	0.27870	0.30963	**0.18865**
VSM	0.15127	0.18772	0.18061	0.20839	0.17016
HVSM	0.13326	0.17579	0.17151	0.20052	0.14587

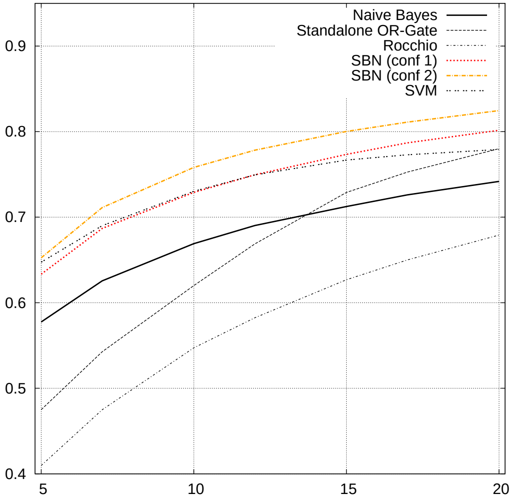
Automatic Indexing From a Thesaurus Using Bayesian Networks: Experiments III

Supervised experiments

Models	Micro BEP	Macr BEP	Av. prec.	Micro F1@5	Macro F1@5
Naïve Bayes	0.42924	0.17787	0.61840	0.50050	0.20322
Rocchio	0.34158	0.35796	0.43516	0.40527	0.33980
OR gate	0.40338	0.44855	0.56236	0.41367	0.24629
SVM	**0.63972**	0.47890	0.69695	0.57268	**0.40841**
SBN 0.0, 0.9	0.54825	0.43361	0.66834	0.54066	0.33414
SBN 0.0, 0.8	0.55191	0.43388	0.67149	0.54294	0.33781
SBN 0.0, 0.5	0.55617	0.43269	0.67571	0.54578	0.34088
SBN 0.0, 0.1	0.55735	0.43282	0.67761	0.54652	0.34228
SBN 0.9, 0.0	0.55294	0.47207	0.65998	0.56940	0.36761
SBN 0.8, 0.0	0.57936	0.47820	0.68185	**0.58163**	**0.38589**
SBN 0.5, 0.0	**0.58372**	**0.48497**	**0.70176**	0.57875	0.38009
SBN 0.1, 0.0	0.56229	0.46171	0.68715	0.55390	0.35123
SBN 0.8, 0.1	0.57887	0.47809	0.68187	**0.58144**	**0.38610**
SBN 0.5, 0.1	**0.58343**	**0.48487**	**0.70197**	**0.57887**	0.38146
SBN 0.5, 0.5	0.58285	**0.48716**	**0.70096**	0.57859	0.37868
SBN 0.8, 0.8	0.56801	0.47946	0.67358	0.57508	0.37300
SBN 0.9, 0.9	0.53963	0.47200	0.64957	0.56278	0.35742
SBN 1.0, 1.0	0.49084	0.45875	0.59042	0.53235	0.32173

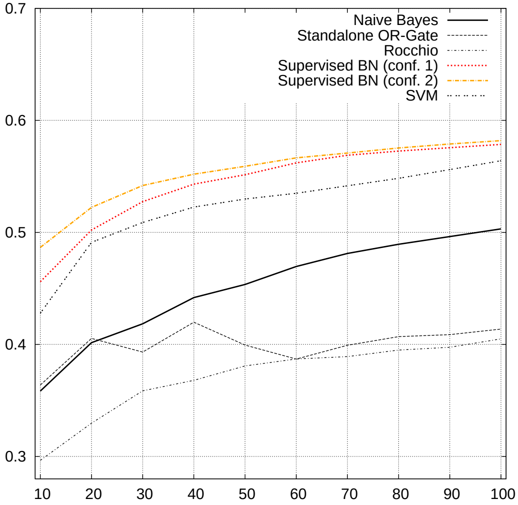
Automatic Indexing From a Thesaurus Using Bayesian Networks: Experiments IV

Supervised experiments: Micro Recall for incremental number of categories



Automatic Indexing From a Thesaurus Using Bayesian Networks: Experiments V

Supervised experiments: Micro F_1 at five computed for incremental percentage of training data



Automatic Indexing From a Thesaurus Using Bayesian Networks

Conclusions

- A BN-based model for document classification indexed from a thesaurus.
- Very good results in unsupervised (300% VSM results).
- Outstanding results in supervised (above SVM).

Future work

- Consider associative relationships.
- Take context into account.
- Test on other thesauri (MeSH, Agrovoc), build testing corpora.

Outline

1. Text Categorization
2. Bayesian networks
3. An OR Gate-Based Text Classifier
4. Automatic Indexing From a Thesaurus Using Bayesian Networks
5. ⇒ Structured Document Categorization Using Bayesian Networks
6. Final Remarks

What is **structure** in document collections?

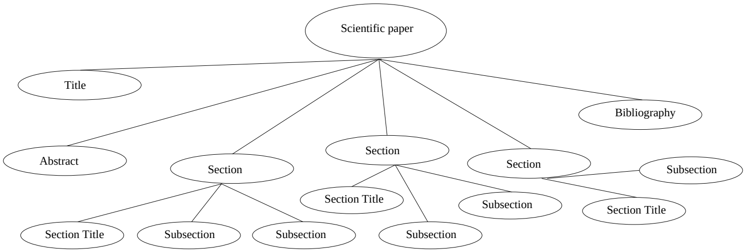
- **Internal structure** (inside each document): XML ("structured") documents .
- **External structure** (outside documents, graph in the collection): linked-based collections.

Outline of this part:

1. Contributions in *Structured TC* (XML).
2. A model for *link-based document categorization* (**multiclass**).
3. A model for *link-based document categorization* (**multilabel**).

Structured Document Categorization Using Bayesian Networks I

Structured Documents



Transformations I

Structured TC: same as “plain” TC, corpora of structured documents.

Our approach

Convert with transformations structured documents to plain documents, and test plain classifiers on them.

Structured Document Categorization Using Bayesian Networks III

Transformations II

```
<book>
 <title>El ingenioso hidalgo Don Quijote
 de la Mancha</title>
 <author>Miguel de Cervantes Saavedra
 </author><contents>
   <chapter>Uno</chapter>
    <text>En un lugar de La Mancha de
    cuyo nombre no quiero acordarme...
 </text> </contents>
</book>
```

Figure: “Quijote”, XML Fragment used for examples, with header removed.

Structured Document Categorization Using Bayesian Networks III

Transformations III

```
El ingenioso hidalgo Don Quijote de la
Mancha Miguel de Cervantes Saavedra Uno
En un lugar de La Mancha de cuyo nombre
no quiero acordarme...
```

Figure: "Quijote", with "only text" approach.

Structured Document Categorization Using Bayesian Networks III

Transformations IV

```
title_El title_ingenioso title_hidalgo
title_Don title_Quijote title_de
title_la title_Mancha author_Miguel
author_de author_Cervantes
author_Saavedra chapter_Uno text_En
text_un text_lugar text_de text_La
text_Mancha text_de text_cuyo text_nombre
text_no text_quiero text_acordarme...
```

Figure: “Quijote”, with “tagging_1”.

Our contribution

`title: 1, author: 0, chapter: 0, text: 2`

```
El ingenioso hidalgo Don Quijote de
la Mancha En En un un lugar lugar de
de La La Mancha Mancha de de cuyo cuyo
nombre nombre no no quiero quiero
acordarme acordarme...
```

Figure: “Quijote”, with “replication” method, using values proposed before.

Structured Document Categorization Using Bayesian Networks III

Experimentation

- Experiments on the **INEX 2007** XML dataset.
- **96611** documents, **21** categories, **50%** training/test split.
- **Replication** improves **macro** measures on Naïve Bayes a lot.
- Other transformations are **not useful here**.

Method	Reduction	Selection?	μBEP	MBEP	μF1	MF1
Naïve Bayes	Only text	no	**0.76160**	0.58608	0.78139	0.64324
Naïve Bayes	Only text	$\geq$ 2 docs.	0.72269	0.67379	0.77576	0.69309
Naïve Bayes	Only text	$\geq$ 3 docs.	0.69753	0.67467	0.76191	0.68856
Naïve Bayes	Repl. (id=2)	None	0.76005	0.64491	**0.78233**	0.66635
Naïve Bayes	Repl. (id=2)	$\geq$ 2 docs.	0.71270	0.68386	0.61321	**0.73780**
Naïve Bayes	Repl. (id=2)	$\geq$ 3 docs.	0.70916	0.68793	0.73270	0.65697
Naïve Bayes	Repl. (id=3)	None	0.75809	0.67327	0.77622	0.67101
Naïve Bayes	Repl. (id=4)	None	0.75921	0.69176	0.76968	0.67013
Naïve Bayes	Repl. (id=5)	None	0.75976	**0.70045**	0.76216	0.66412
Naïve Bayes	Repl. (id=8)	None	0.74406	0.69865	0.72728	0.61602
Naïve Bayes	Repl. (id=11)	None	0.72722	0.67965	0.71422	0.60451

Structured Document Categorization Using Bayesian Networks III

Conclusions

- Several XML transformation (one original).
- Good results with “replication” + NB.

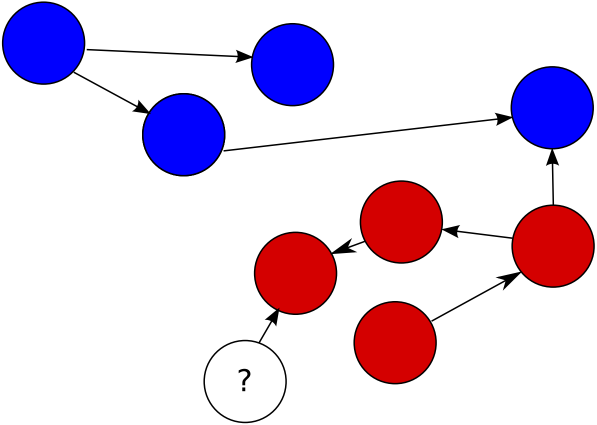
Future work

- More extensive experimentation.
- New transformations.

Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization

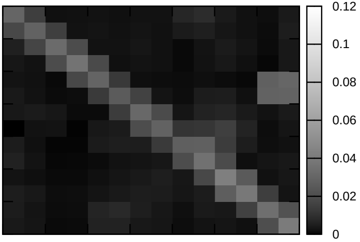
A set of documents with a **graph structure** among them.
The goal is to label a document using both its **content** and the **graph structure** (labels of the neighbors?).



Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization

Typically, scatterplots like this:



Encyclopedia regularity (*a document of category C_i tends to links documents on the same category*).

Structured Document Categorization Using Bayesian Networks IV

link-based categorization: multiclass I

Document d_0, linked to documents $d_1, \ldots, d_m$.

Random variables $C_0, C_1, \ldots, C_m$, in $\{c_0, c_1, \ldots, c_n\}$.

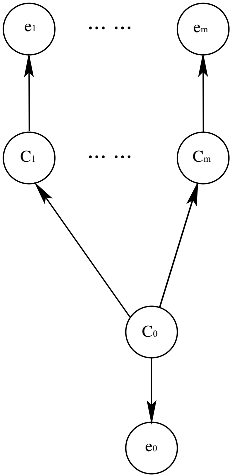
Variables e_i, **evidence** of the classification (content) of document d_i.

Given the **true class** of the document to classify (**independences**):

1. the **categories of the linked documents are independent** among each other, and
2. the **evidence about this category** due to the document content **is independent of the original category** of the document we want to classify.

Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization: multiclass II



Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization: multiclass III

With some computation:

$$p(C_0 = c_0|e) \propto p(C_0 = c_0|e_0) \prod_{i=1}^{m} \left(\sum_{c_j = \{c_0, \ldots, c_n\}} p\left(C_i = c_j | C_0 = c_0\right) \frac{p(C_i = c_j|e_i)}{p(C_i = c_j)} \right)$$

Where:

- $p(C_0 = c_0|e)$ **final evidence** that the document belongs to C_0.
- $p(C_i = c_j|e_i)$ **obtained with a "local"** (content) **classifier** (NB).
- $p(C_i = c_i)$ (prior) and $p(C_i = c_i|C_0 = c_0)$ (probability a document of C_i links another of C_0), **obtained from training data**.

Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization: multiclass IV

Experiments: INEX 2008 corpus:

- A classical *Naïve Bayes algorithm* on the flat text documents obtained **0.67674** of recall.
- *Our proposal* using the previous Naïve Bayes as the base classifier obtained **0.6787** of recall (using outlinks).
- Our model (inlinks): **0.67894** of recall.
- Our model (neighbours): **0.68273** of recall.

The model works better in a “ideal environment” (knowing the labels of all neighbors).

Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization: multiclass V

Conclusions

- A new model for classification of multiclass linked documents, based on BNs.
- Good performance in an ideal environment.

Future work

- Use a base classifier (probabilistic) with a better performance (Logistic? SVM with probabilistic outputs?).

Structured Document Categorization Using Bayesian Networks V

Linked-document categorization: multilabel I

- Previous model was not flexible. Structure of BN imposed.
- We learn the interactions among categories from data, **no fixed structure, but any which is learnt from the set of categories**.
- **Variables:** categories C_i (one for category), categories of incoming links E_j (one for category) and terms T_k (many).
- We will search for $p(c_i|e_j, d_j)$.
- **Main assumption**:

$$p(d_j, e_j|c_i) = p(d_j|c_i)\, p(e_j|c_i).$$

Structured Document Categorization Using Bayesian Networks V

Linked-document categorization: multilabel II

With a few computations:

$$p(c_i|d_j, e_j) = \frac{p(c_i|d_j)\, p(c_i|e_j) \;/\; p(c_i)}{p(c_i|d_j)p(c_i|e_j)/p(c_i) + p(\overline{c}_i|d_j)p(\overline{c}_i|e_j)/p(\overline{c}_i)}$$

- $p(c_i|d_j)$: output of a probabilistic classifier. **Any probabilistic classifier.**
- $p(c_i|e_j)$: probability of being of C_i considering the set of the categories of the incoming (known) links. **This is modeled by the BN.**

Structured Document Categorization Using Bayesian Networks V

Linked-document categorization: multilabel III

Experimentation INEX 2009 corpus: 54572 documents, test/train split of a 20/80%. 39 categories.

Measures Accuracy (ACC), Area under Roc curve (ROC), F1 measure (PRF) and Avg prec on 11 std (MAP).

- **Learning** Bayesian Network, using **WEKA** package.
  - **Hillclimbing** algorithm (easy and fast) + **BDeu** metric (3 parents max. per node).
- **Propagation**, using **Elvira**
  - Compute $p(c_i)$ (once), and $p(c_i|e_j)$ (for each document j). Exact propagation is **slow** for so many categories! $\Rightarrow$ **Importance Sampling** algorithm (approximate).

Structured Document Categorization Using Bayesian Networks V

Linked-document categorization: multilabel IV

Results

	MACC	μACC	MROC	μROC	MPRF	μPRF	MAP
N. Bayes	0.95142	0.93284	0.80260	0.81992	0.49613	0.52670	0.64097
N. Bayes + BN	**0.95235**	**0.93386**	0.80209	0.81974	**0.50015**	**0.53029**	**0.64235**
OR gate	0.92932	0.92612	0.92526	0.92163	0.45966	0.50407	0.72955
OR gate + BN	**0.96607**	**0.95588**	**0.92810**	**0.92739**	**0.51729**	**0.55116**	0.72508

Our method clearly improves both baselines.

Structured Document Categorization Using Bayesian Networks IV

Linked-document categorization: multilabel V

Conclusions

- A new model for classification of multilabel linked documents, based on BNs.
- Very flexible.
- Any learning procedure is usable.
- Very promising results

Future work

- Use different baselines.
- More extensive experimentation.

Outline

1. Text Categorization
2. Bayesian networks
3. An OR Gate-Based Text Classifier
4. Automatic Indexing From a Thesaurus Using Bayesian Networks
5. Structured Document Categorization Using Bayesian Networks

⇒ Final Remarks

Final Remarks

Summary

Most relevant contributions

- A **new text classifier** (OR gate), better than NB.
- Definition of a **new problem** (*thesaurus indexing*). **Two models**, outstanding results.
- Some **minor contributions** in XML classification (“text replication”).
- **Two models** of link-based **document categorization.** Promising results in the Multilabel one.

Final Remarks II

List of Publications supporting this work:

In the thesis:

See pages **170-173**

or...

In the www:

Visit `http://decsai.ugr.es/~aeromero`

Final Remarks III

Software:

- **DauroLab**. A toolbox for Machine Learning. Written in Java (by me!).
- **Free (*libre*) software** (GPL v3).
- `http://sf.net/projects/daurolab`.

Thank you for your attention