Machine Learning Class Project: Bayesian Network of pathologies

Matteo Matassoni (150629)
m.matassoni.1@studenti.unitn.it

1 Introduction

Bayesian model of leukemia pathologies project aims to build three different kinds of machine learning models to predict leukemia pathologies of patients. These three models are evaluated and compared for finding the best one. This research has real-world importance, in fact the learned models can be used in medical application for disease prediction. There is also and educational importance of the task: learn and apply the theory behind the machine learning field, which has been learned during class.

This machine learning problem requires to solve a binary classification task, since it is required to classify two kind of leukemia pathologies: acute lymphoblastic leukemia (ALL) and acute myelogenous leukemia (AML). Binary classification is a well-known task and some of the methods suitable for learning binary classifiers are decision trees, Bayesian networks, support vector machines, kernel machines and neural networks. This research proposes to use Bayesian networks as the models to solve this task, in particular, three different approaches are used in order to automatically learn the classification systems. The first two networks are learned using a constraint-based and a score-based approach, whereas the third network has a fixed structure and correspond to a naive Bayes classifier.

The three a graph-based models assume that the data has been discretized in two values, for simplicity a 'yes'/'no' value. Moreover, naive Bayes classifier is known to do a strong independence assumption, *i.e.* the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable.

It is expected that the first two models, in which the network structure is learned, perform better on this task than naive Bayes classifier, due to its independence assumption. Moreover, it is expected that the model learned via the score-based approach performs better than the constraint-based one.

2 Background

2.1 Leukemia

Leukemia is a type of cancer of the blood or bone marrow characterized by an abnormal proliferation of leukocytes[1]. Leukemia can affect people at any age and in 2013 the National Cancer Institute [3] estimated a number of 48,610 new cases and 23,720 of deaths in the United States. Leukemia is a broad term covering a wide range of diseases. This spectrum of diseases can be firstly divide between two types of leukemia: its acute and chronic forms. Additionally, the diseases can be subdivided according to which kind of blood cell is affected. This split divides leukemias into lymphoblastic or lymphocytic leukemias and myeloid or myelogenous leukemias. Combining these two classifications provides a total of four main categories:

Acute lymphoblastic leukemia (ALL) It is the most common type of leukemia in young children. This disease also affects adults, especially those age 65 and older.

Chronic lymphocytic leukemia (CLL) Most often affects adults over the age of 55. It sometimes occurs in younger adults, but it almost never affects children.

Acute myelogenous leukemia (AML) It occurs more commonly in adults than in children, and more commonly in men than women.

[1] also known as white blood cells

Chronic myelogenous leukemia (CML) It occurs mainly in adults; a very small number of children also develop this disease.

This research focuses on acute lymphoblastic leukemia (ALL) and acute myelogenous leukemia (AML) as the dataset contains only patients who develop these two kind of leukemia, see Section 3.1.

2.2 Expression level of genes

Living beings depend on genes, as they are responsible for protein synthesis in living cells. Genes interact with each other forming complex gene-networks. The amount of proteins synthesized in a certain time is called expression level. Measuring gene expression is an important part of many fields of science, since the ability to quantify the level at which a particular gene is expressed within a cell, tissue or organism can give a huge amount of information. For example measuring gene expression can indicate the presence of a certain pathology. Expression level of genes from pathological cells are often measured relative to healthy cells. Discovering interactions among genes which correlate with a pathology can help elucidating its characteristics.

The expression level of genes can be measured by DNA microarrays. A DNA microarray is a collection of microscopic DNA spots attached to a solid surface. Scientists use DNA microarrays to measure the expression levels of large numbers of genes simultaneously. DNA microarrays is typically used in comparing the gene expression profiling of a ill patient with a healthy one to identify which genes are responsible of a certain pathology.

3 Data and Task

3.1 The Data

Leukemia dataset consists of 72 samples: 25 samples of acute myelogenous leukemia (AML) and 47 samples of acute lymphoblastic leukemia (ALL). The source of the gene expression measurements was taken from 63 bone marrow samples and 9 peripheral blood samples. Gene expression levels in these 72 samples were measured using high density oligonucleotide microarrays. Each sample contains 5147 gene expression levels.

The dataset can be download at `http://disi.unitn.it/~passerini/teaching/2013-2014/MachineLearning/Projects/dataset_1_2_3.gz`.

3.2 The Task

The task can be subdivided into several subtasks: preprocessing, feature selection, learn Bayesian Network and classification. Details about these procedures are given respectively in Section 3.2.1, 3.2.2, 3.2.3 and 3.2.4.

3.2.1 Preprocessing

Expression levels depend strongly on the gene considered, thus they need to be normalized. Fo our purposes, normalization is the process of discretize genes expression level by choosing a threshold in order to maximize some attribute quality criterion. In this paper, the *information gain* (1) is used as attribute quality criterion.

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \quad (1)$$

Where $Values(A)$ is the set of possible values taken by A, S_v is the subset of S taking value v at attribute A and $H(S)$ is the information entropy (2).

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \quad (2)$$

More formally the discretization procedure can expressed by the below pseudo-code.

function $discretize(example)$
  $thresholds \leftarrow candidate\_t(examples)$
  $best \leftarrow \emptyset$
  $max_{ig} \leftarrow -\infty$
  $classes \leftarrow$ classes of example
  for $t \in thresholds$ **do**
    $disc\_attribute \leftarrow discretize(example, t)$
    $ig \leftarrow IG(classes, disc\_attribute)$
    if $ig > max_{ig}$ **then**

$max_{ig} \leftarrow ig$
$best \leftarrow disc\_attribute$
end if
end for
return $best$
end function

Where the procedure for calculating the candidate thresholds is:

function $candidate\_t(examples)$
$T \leftarrow \emptyset$
sort examples according to their continuous attribute values
for $i \in [0, examples.length - 1)$ **do**
$(x_1, y_1) \leftarrow example[i]$
$(x_2, y_2) \leftarrow example[i + 1]$
if $y_1 \neq y_2$ **then**
$T \cup \left\{ \frac{x_1 + x_2}{2} \right\}$
end if
end for
return T
end function

3.2.2 Feature selection

Most of the genes are expected to be uncorrelated with the pathologies (uninformative features). Feature selection is performed on genes according to their *information gain* (1). The best 24 out of 5147 genes are chosen as informative features when sorted by their information gain value. Section 4.2 clarifies the choice of selecting only a number of 24 informative genes.

3.2.3 Learning Bayesian Networks

3.2.3.1 Learning the structure Suppose that there is an unknown Bayesian network BN over the universe U that produces the sample cases D. The BN gives you a distribution $P_{BN}(U)$ and the dataset D gives you another distribution $P_D^{\#}(U)$. The task is to construct a Bayesian network M for which $P_{BN}(U)$ is close to $P_D^{\#}(U)$. The two most known and used methods for learning Bayesian network structure are, namely, the constraint-based and the score-based approach.

3.2.3.2 Learning the parameters Assume the structure of the Bayesian network is given (either by domain experts, learned as described in Paragraph 3.2.3.1 or a simple fixed structure as in naive Bayesian classifier). Moreover, a dataset of examples $D = \{\mathbf{x}(1), ..., \mathbf{x}(N))\}$ is given, where each example $\mathbf{x}(i)$ is a configuration for all or some variables in the model. The task is to estimate the parameters of the model – the conditional probability distribution (CPD) – from the dataset D.

3.2.4 Classification

We consider a binary classification problem where instances $(\mathbf{x}, y)$ are drawn from a distribution P over $\mathbf{X} \times Y$, with Y a finite discrete set of labels, *i.e.* $Y = \{\text{“}AML\text{”}, \text{“}ALL\text{”}\}$. The task is predicting the class y of examples given the input $\mathbf{x}$, with $y \in Y$ and $\mathbf{x} \in \mathbf{X}$.

4 The Models

4.1 Existing Models

Cho and Won [1] used Multi-layer perceptron, k-nearest neighbour, support vector machine (SVM) and structure adaptive self–organizing for classification on the same Leukemia cancer dataset among two others. Also, they have combined the classifiers to improve the performance of classification.

4.2 Proposed Models

In this section, we present the three types of model that we have trained to predict the two types of disease (AML vs ALL): a Bayesian network classifier learned with a constraint-based approach, another Bayesian network classifier with structure learned instead with a score-based approach and a naive Bayes classifier.

The models are learned using the Hugin Lite software[2]. Being a trial version, the Hugin Lite software is limited to handle a maximum of 50 states and learn from maximum of 500 cases. This constraint has forced us to use in our graphical models only 25 node variables (24 genes plus the label class).

[2] available at `http://www.hugin.com/productsservices/demo/hugin-lite`

4.2.1 Bayesian network classifier: constraint-based

Constraint-based learning tests conditional independencies on the data and construct a model satisfying them. To learn this model, Hugin Lite is used as it provides two constraint-based algorithms: the PC algorithm and the NPC algorithm. The NPC (Necessary Path Condition) is an extension of the PC version; both these two algorithms are used as alternatives.

4.2.2 Bayesian network classifier: score-based

Score-based learning assigns a score to each possible structure and defines a search procedure looking for the structure maximizing its score. Hugin Lite is used to learn this model, as it provides a *Greedy Search-And-Score* structure learning algorithm. As the Hugin GUI Help [2] states:

> The greedy search-and-score algorithm for learning the structure of a Bayesian network uses a score function to evaluate the goodness of a candidate network structure. The algorithm performs a search through the space of possible network structures and returns the structure with the highest score. The operators used to perform the search given a current candidate are add an arc, remove an arc and reverse an arc.
>
> [T]he user can choose between using the *Akaike Information Criterion* (AIC) or the *Bayesian Information Criterion* (BIC) as the score function. The user can specify an upper limit on the number of parents of each node in the network structure.

Both the AIC and BIC score functions are used as alternatives.

4.2.3 Naive Bayes classifier

This classifier learns from training data the conditional probability of each attribute A_i given the class label C. Classification is then done by applying Bayes rule to compute the probability of C given the particular instance x described by a conjunction of attribute values $A_1, ..., A_n$, and then predicting the class with the highest posterior probability. This computation is rendered feasible by making a strong independence assumption: all the attributes A_i are conditionally independent given the value of the class C. When represented as a Bayesian network, a naive Bayesian classifier has the simple fixed structure depicted in Figure 1. This network captures the main assumption behind the naive Bayesian classifier, namely, that every attribute (every leaf in the network) is independent from the rest of the attributes, given the state of the class variable (the root in the network). To learn this model Hugin Lite is used. Firstly, the fixed structure is created manually in Hugin Lite, then we run E-M algorithm feeding it with the training data set to learn the parameter of the network – the conditional probability distribution (CPD).

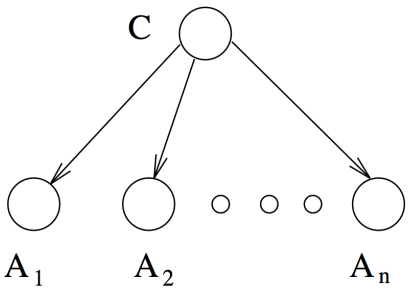

Figure 1. The structure of the naive Bayes network

5 Experiments

5.1 Experimental Hypotheses

It is expected that the two models in which the network structure is learned, perform better on this task than naive Bayes classifier, due to its independence assumption. Moreover, it is expected that the model learned via the score-based approach performs better than the constraint-based one.

5.2 Experimental setup

The dataset is split randomly into a training and a test set, keeping the same proportion among classes. The 70% of ALL and AML – respectively 32 ALL and

17 AML (59 cases in total) – belongs to the training set and the remaining 30%: 15 ALL and 8 AML in the test set (23 cases in total).

For each model the confusion matrix has been obtained during classification test using the "Analysis Wizard" provided by the Hugin Lite software. Then, using the confusion matrix accuracy (3), precision (4), recall (5) and F_1 (6) measures has been calculated for each model.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Prec = \frac{TP}{TP + FP} \tag{4}$$

$$Rec = \frac{TP}{TP + FN} \tag{5}$$

$$F_1 = \frac{2(Prec * Rec)}{Prec + Rec} \tag{6}$$

Where the larger response value – ALL – represents the "positive" result.

5.3 Experimental results

Classifiers	Prec	Rec	Acc	F_1
score-based$_{BIC}$	1.00	1.00	1.00	1.00
score-based$_{AIC}$	1.00	1.00	1.00	1.00
Naive Bayes	1.00	1.00	1.00	1.00
constraint-based$_{PC}$	0.83	1.00	0.87	0.91
constraint-based$_{NPC}$	1.00	0.80	0.87	0.89

Table 1. Experimental results

As it can be seen in Table 1, the models with the best accuracy are: score-based learned BN classifier (both the AIC and BIC alternatives) and naive Bayes classifier, with an accuracy of 1.00; whereas the worst are the constraint-based learned BN classifiers (NPC and PC) with an accuracy of 0.87. The network learned via the PC algorithm has a greater F_1 measure than the NPC one. This is interesting since the Hugin GUI Help [2] states that:

> [G]enerally, it is recommended to use the NPC algorithm, as the resulting graph will be a better map of the (conditional) independence relations represented in the data. In particular, when the data set is small, the NPC algorithm should be the one preferred. The NPC algorithm, however, has longer running times than the PC algorithm.

As we believed, the models learned with the score-based approach performs better than the constraint-based ones, but the experimental results contradict the hypothesis that the worst model is the naive Bayes classifier. Its performance is somewhat surprising, since its strong independence assumption is clearly unrealistic.

6 Conclusion

Despite their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked as well as score-based learned BN classifiers in the leukemia context. Also, we have found that constraint-based learned BN classifiers are worse than the other models. Naive Bayes classifier and score-based learned BN classifiers have an accuracy and F_1 measure of 1.00, thus they can be really used in medical application for predicting AML vs ALL. In this paper we have proposed Bayesian networks to solve our binary classification task and the experimental results has shown that they are indeed a proper way of solving classification in our context.

References

[1] Sung-Bae Cho and Hong-Hee Won. Machine learning in dna microarray analysis for cancer classification. In *Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics 2003 - Volume 19*, APBC '03, pages 189–198, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. ISBN 0-909-92597-6. URL `http://dl.acm.org/citation.cfm?id=820189.820213`.

[2] HUGIN Expert A/S. Hugin GUI Help. URL `http://download.hugin.com/webdocs/manuals/Htmlhelp/`.

[3] National Cancer Institute. Leukemia Home Page. URL `http://www.cancer.gov/cancertopics/types/leukemia`.