

Analyzing Gene Expression Data Using Classification Rules

Gary Livingston

Univ. of Massachusetts, Lowell
gary@cs.uml.edu

Xiao Li

Univ. of Massachusetts, Lowell
xili@cs.uml.edu

Guangyi Li

Univ. of Massachusetts, Lowell
gli@cs.uml.edu

Liwu Hao

Univ. Of Massachusetts, Lowell
lhao@cs.uml.edu

Jianping Zhou

Univ. of Massachusetts, Lowell
jzhou@cs.uml.edu

Abstract

We have applied rule induction to a publicly available adenocarcinoma gene expression dataset. The typical approach to the analysis of gene expression data is to cluster the genes. However, interpreting the resulting clusters may be difficult. With rules, the interpretation is more obvious (e.g., $(CDKN3 > 253) \implies (tumor-stage = 3)$). We used HAMB, a discovery tool developed in our lab, to learn rules for survival status, survival time, and tumor stage from this dataset. When we searched the world-wide web for publications relating our top 53 genes from our discovered rules to lung cancer, we found that 9 of them are known to be associated with lung cancer, 19 of them are known to be associated with other types of cancer, and the remaining 25 were not known to be associated with cancer. We were also able to generate classification rule sets for many patient attributes, such as survival, survival-time, etc. Our results suggest that classification rule induction may be well suited to the examination of gene expression data.

1. Introduction

Although clustering is often used to analyze gene expression datasets, the interpretation of the resulting clusters may be difficult. In contrast, we have found the interpretation of induced classification rules to be mostly straightforward (e.g., $(CDKN3 > 253) \implies (tumor-stage = 3)$). The induced rules may be examined individually to extract “nuggets” of information or used in groups to predict membership in some class or to predict the values of a “target” attribute, such as the prediction of high-risk patients, or predict tumor stage. Some researchers may avoid rule induction because it traditionally requires the user to completely set up the induction problem: creating target classes, selecting sets of attributes from which the rules will be generated, and selecting parameters for running the induction program. Moreover, manually analyzing the hundreds of rules which are typically output by rule induction programs may be overwhelming. We use a new program, HAMB [1] which is a supervisor program to a rule induction program called RL [2]. HAMB has many features which make it desirable for analyzing gene expression data: (1) the user may specify multiple “target” attributes for which rules sets will be induced, (2) HAMB automatically selects the feature set and parameters for each of the target attributes using search and heuristics, and (3) HAMB does some post processing of the induced rules, such as pruning and grouping similar rules. Moreover,

when it is available, HAMB can use domain knowledge to improve the quality of its reported rules and rule sets.

2. Methods

Our gene expression dataset was first examined by Beer et al. [3]. The dataset contains expression data for 7,129 genes for 86 lung tumor and 10 normal lung samples. The data comes with patient data containing information on tumor stage (stage 1 or stage 3), tumor size, survival time (months), survival status, tumor histological type, tumor differentiation, p53 nuclear accumulation, K-ras mutation rate, and patient demographics: age, sex and the number of years the patient smoked cigarettes.

We preprocessed the data by removing genes having more than 20% missing values and filtering genes with little change in expression. After this filtering, 485 genes remained.

For the findings presented in this paper, we instructed HAMB to find rules for many of the attributes describing the tumors, such as tumor stage, differentiation, mucosity, and for attributes describing the survival of the patients and survival time. HAMB automatically used RL to generate rule sets predicting the values of these attributes, automatically selecting the training sets, features sets, and parameter settings needed by RL. After running RL to predict each of these attributes, HAMB analyzed the rules before presenting them to the user, performing pruning to eliminate redundant rules and grouping the rules to facilitate their understanding.

We attempted to use another dataset [4] to verify our findings, but we encountered two problems that frustrated our attempts: first, many of the genes in the Beer et al. dataset were not in the second dataset, and second, the population distributions were very different, which frustrated our validation attempts. In retrospect, 10-fold cross-validation would have been better, and we are in the process of performing a cross-validation study.

However, we did verify our findings by searching the World Wide Web to identify which of HAMB's findings are known and which might be novel. We searched the web using www.google.com and an additional five web sites listing genes known to be associated with lung cancer or cancer.

2.1. HAMB

Livingston et al. 2003 describes HAMB, which decides for itself which discovery tasks to perform and when to perform them. HAMB uses user preferences and a small set of known relationships among the attributes in the data to automatically set up rule-induction problems for the RL program and to post-process the induced rules. HAMB's basic framework consists of an agenda-and-justification-based framework for selecting the next task to perform. Tasks refer to computational encoding of operations on items that refer to instances of the search space of possible discoveries. Tasks are performed using heuristics that create new items for further exploration and that place new tasks on the agenda. This framework has several desirable properties: (1) it facilitates the encoding of general discovery strategies using various types of background knowledge, (2) it reasons about the appropriateness of the tasks being considered, and (3) it tailors its behavior toward a user's interests by prioritizing tasks according to a plausibility function derived from an estimate of interestingness specified by the user. For example, a task in HAMB would be to "examine the predictivity of a gene toward tumor-stage. Performing this task would involve the production of new sub-tasks, such as "induce rule-set", that will cause HAMB to set up the induction task by: (1) selecting a training set of examples, (2) selecting the feature set of attributes from which the rules will be induced, (3) selecting

the parameters with which to run RL, and, finally, (4) running RL to induce the rules. HAMB then loads the induced rules and post-processes them. Application of HAMB to protein crystallization experiments has demonstrated HAMB's ability to identify patterns that are both interesting and novel [1].

HAMB's input consists of the files containing the set of cases that it will use to make its discoveries (the *discovery-database*), an optional testing set of cases (the *testing-database*), and a *domain theory* file containing domain knowledge. HAMB reports as discoveries those items with interesting relationships or properties. A property or relationship is interesting if its value exceeds a threshold provided for each relationship or property. HAMB creates a report for each relationship or property, where the items having values for that property or relationship greater than or equal to the threshold are listed in decreasing order. If the testing-database is not given to HAMB, it creates its own set of test instances comprising of a random one-third of the discovery database cases.

A major advantage of HAMB's framework is that it provides a clean separation of the discovery program from the knowledge it uses. We provide further modularity in HAMB by using domain-independent heuristics (and properties and relationships) which refer to domain- and problem-specific information that is either given in a *domain-theory* file or discovered by HAMB. While HAMB and its heuristics are general, they access domain- and problem-specific information. Therefore, HAMB is able to perform discovery using domain knowledge, allowing HAMB to tailor its behavior to the discovery problem and to evaluate the cases given to it to make discoveries from and the resulting discoveries in a domain context. *Thus, HAMB is able to examine a database using a variety of knowledge specific to the domain from which the data were taken.* In contrast, most other knowledge discovery, data mining, and machine learning programs are only capable of using one or two types of domain knowledge. Results of experiments with HAMB, reported below, demonstrate that it uses domain knowledge effectively to evaluate its discoveries and to avoid reporting a large number of uninteresting rules.

2.2. Evaluation of HAMB's use of domain knowledge

In an earlier study of HAMB's ability to use domain knowledge [1], we performed a lesion study to evaluate the effectiveness of some of HAMB's heuristics that use domain knowledge. This evaluation used 500 cases randomly selected from a database of protein crystallization experiments [5], with each of the cases being described using approximately 200 attributes. To perform this study, we removed portions of the knowledge given to HAMB or disabled the portions of HAMB that use the knowledge.

The unmodified version of HAMB with the complete crystal-growing knowledge was also run on this set of cases, as was a version of HAMB that used no domain knowledge.

The types of knowledge used by HAMB that we tested are:

- *Synonyms.* If an attribute is found in the feature set that is synonymous (as either discovered by HAMB or stated in the domain-theory) with another attribute in the feature set, the attribute with the lesser estimated interestingness is removed from the set of attributes used to form the discoveries. A baseline version of HAMB omitted this capability and did not eliminate synonyms. It allowed the creation of 40 (19%) more redundant rules than did HAMB. This was surprisingly low, because the data contain many similar attributes. However, HAMB's definition of redundancy is very strict, requiring either intensional (stated in the knowledge given to HAMB) or extensional equivalence (identical values for the cases); therefore only a few pairs of attributes met its strict criterion of similarity.

- *Uninformative attributes or values.* The knowledge given to HAMB may contain information about attributes and values that are meaningless to the user. Examples of uninformative attributes or values might be *misc* or *missing*. HAMB's heuristics use this knowledge to avoid inducing rules containing uninteresting features (either in the left-hand side or right-hand side of a rule). A version of HAMB with this capability omitted allowed the generation of 300 (141%) additional uninteresting rules.
- *Known associations.* HAMB uses some of the knowledge given to it to remove attributes that have a known association (by causation, definition, association, etc.) with the current target attribute. HAMB also removes attributes that are discovered to be extensionally equivalent to the target attribute. The version of HAMB used to test these heuristics omitted this use of knowledge and allowed the generation of 2,897 (1,367%) additional non-novel rules.

The regular version of HAMB reported 212 rules in this experiment, whereas the baseline version that used no domain knowledge reported 3,936 rules. Thus, HAMB was able to use chemical and crystal-growing knowledge to avoid the creation of 3,724 uninteresting rules. While the number of interesting rules is about the same in each case, the number of uninteresting rules shown to the user is much lower when using the regular version of HAMB, causing the percentage of interesting rules shown to the user to be much higher.

3. Results

One of HAMB's methods for post processing rules is to group them into *rule families*. These are groups of rules where changing the value of one attribute on a rule's left-hand side yields in a consistent change in the value being predicted. The consistency of the rules in the family increases confidence in the rules. Table 1 presents some of the stronger rule families HAMB discovered for differentiation of the tumor, P53 nuclear accumulation, and tumor stage. Rules 1–3, 4 and 5, 6 and 7, 8–10, and 11–13 form 4 rule families. The p-values for all rules in Table 1 are less than 0.00001. When we validated rules 11–13 using the Meyerson data [4], the accuracy of the three rules used together to predict tumor stage was 59%, with 60% coverage. The p-values of these rules on the Meyerson data were <0.0001, 0.119, and 0.0115, respectively, which clearly suggests that the trend exists in the Meyerson data. Used together, rules 1–3 use CDKN3 to predict tumor differentiation with 60% accuracy and 60% coverage, rules 4 and 5 use KLF5 to predict P53 nuclear accumulation with 72% accuracy and 41% coverage, rules 6 and 7 use PSG5 to predict tumor stage with 66% accuracy and 40% coverage, rules 8–10 use KRT6A to predict tumor stage with 59% accuracy and 35% coverage, and rules 11–13 use CP to predict tumor stage with 59% accuracy and 60% coverage.

Table 2 summarizes the rule families HAMB discovered. *Target* indicates the variable being predicted; *# Rules* indicates the number of rules in the rule family; *Corr* indicates the correlation with the severity of the disease, *Acc*Cov* is the product of Accuracy and Coverage; *WWW.LC* indicates whether or not we could verify that the gene is thought to be associated with lung cancer; and *WWW.C* indicates whether or not we could verify that the gene is thought to be associated with any type of cancer.

An advantage of using rule induction is that rule set pruning can be used to simplify the induced rule sets, often increasing accuracy and coverage at the same time. Table 3 presents one of the better rule sets for predicting patient death. The accuracy of this rule set on the Beer et al. data is 75%, with a coverage of 70%. Due to difficulty with using the Meyerson data we could not use it to verify any of these rule sets.

Table 1. Rule families discovered by HAMB for predicting tumor differentiation, P53 nuclear accumulation, and tumor stage. *TP* (true positives) is the number of cases correctly predicted; *FP* (false positives) is the number of cases incorrectly predicted; *SENS* is a rule's sensitivity ($TP/\text{all positives}$); and *PPV* is a rule's positive predictive value ($TP/(TP + FP)$).

ID	RULE	TP	FP	SENS	PPV
1	(CDKN3 MIN) ==> (DIFFERENTIATION WELL)	20	18	0.43	0.53
2	(CDKN3 MED) ==> (DIFFERENTIATION MODERATE)	30	8	0.37	0.79
3	(CDKN3 MAX) ==> (DIFFERENTIATION POOR)	20	20	0.48	0.5
4	(KLF5 MAX) ==> (P53_NUCL_ACCUM -)	40	0	0.29	1
5	(KLF5 MIN) ==> (P53_NUCL_ACCUM +)	16	22	0.5	0.42
6	(PSG5 MIN) ==> (STAGE 0)	12	24	0.6	0.33
7	(PSG5 MAX) ==> (STAGE 1)	38	2	0.28	0.95
8	(KRT6A MAX) ==> (STAGE 3)	18	22	0.47	0.45
9	(KRT6A MIN) ==> (STAGE 0)	12	26	0.6	0.32
10	(KRT6A MED) ==> (STAGE 1)	38	0	0.28	1
11	(CP MIN) ==> (STAGE 0)	12	26	0.6	0.32
12	(CP MAX) ==> (STAGE 3)	20	20	0.53	0.5
13	(CP HIGH) ==> (STAGE 1)	36	2	0.27	0.95

Table 4 presents a summary of the better rule sets HAMB induced to predict the following targets: patient death, differentiation, K-ras mutation, P53 nuclear accumulation, and survival time in months. We sort our rule sets by the sum of the positive predictive values of the rule sets for all values of the target attributes. This sorting helps identify rules sets that predict well for more than just one value.

4. Related Work

While association rule mining methods [6][7] and support vector machines [8] have been used to mine gene expression data, association rules are more difficult to understand than the simple predictive rules presented here, and the statistical basis upon which support vector machines are based is often not sufficient for the small sample size of gene expression data. Moreover, these methods are not as suited to the incorporation of domain knowledge as is our method.

5. Conclusion

We have shown that using rule induction to examine gene expression data produces rules that are easily interpreted. Moreover, rule set pruning techniques, which are well-developed, may be applied to the induced rule sets to simplify them. In addition, pruning often improves accuracy with little loss in coverage. Applying post processing to group induced rules and to evaluate the generated rule sets further enhances the comprehensibility of induced rules and rule sets. Not only do individual rules

provide “nuggets” of useful information, but rule sets may be used as classifiers, which have many uses, such as identifying high-risk patients.

6. References

- [1] Livingston, G., Rosenberg, J. and Buchanan, B. (2003). An Agenda- and Justification-Based Framework for Discovery Systems, *Journal of Knowledge and Information Systems* 5(2), to appear.
- [2] Provost, F. J. and Buchanan, B. G. (1995), Inductive Policy: The Pragmatics of Bias Selection, *Machine Learning* 20(1): 35–61.
- [3] Beer, D., Kardia, S., Huang, C., Giordano, T., Levin, A., Misek, D., Lin, L., Chen, G., Gharib, T., Thomas, D., Lizyness, M., Kuick, R., Hayasaka, S., Taylor, J., Iannettoni, M., Orringer, M., and Hanash, S. (2002), Gene-Expression Profiles Predict Survival of Patients with Lung Adenocarcinoma, *Nature* 8(8): 816 – 824.
- [4] Bhattacharjee, A., Richards, W. G., Staunton J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J., and Meyerson, M. (2001), Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinoma Subclasses, *Proceedings of the National Academy of Science USA*. 98(24):13790-5.
- [5] Gililand, G. L., Tung, M., and Ladner, J. (1996). The Biological Macromolecule Crystallization Database and Protein Crystal Growth Archive. *Journal of Research of the National Institute of Standards and Technology* 101(3): 309–320.
- [6] Creighton, C. and Hanash, S. (2003). Mining Gene Expression Databases for Association Rules. *Bioinformatics* 19(1):79-86.
- [7] Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J. F. and Gandrillon, O. (2002). Strong-Association-Rule Mining for Large-Scale Gene-Expression Data Analysis: a Case Study on Human SAGE Data. *Genome Biology* 3(12).
- [8] Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares M. Jr., and Haussler, D. (2000). Knowledge-based Analysis of Microarray Gene Expression Data by using Support Vector Machines. *Proceedings of the National Academy of Science USA* 97(1):262-267.

Table 2. Summary of rule families discovered by HAMB. Please refer to the text for an explanation of the columns.

Gene	Target	# Rules	Corr	Accuracy	Coverage	Acc*Cov	WWW.LC	WWW.C
ABCA3	Time	2	+	0.513	0.406	0.208	No	No
CRYM	Diff	2	+	0.615	0.406	0.250	No	No
FABP4	Stage	2	-	0.700	0.417	0.292	No	No
FABP4	Time	2	-	0.564	0.406	0.229	No	No
FABP5	Stage	2	-	0.718	0.406	0.292	No	No
FMO2	Death	2	-	0.795	0.406	0.323	No	No
GCHFR	K-ras-mutation	2	+	0.709	0.396	0.312	No	No
GPX3	Time	2	-	0.500	0.396	0.198	No	No
HBA2	Time	2	-	1.000	0.104	0.104	No	No
KAL1	Stage	2	-	0.692	0.406	0.281	No	No
KRT6A	Stage	2	+	0.718	0.406	0.292	No	No
LAD1	Stage	2	+	1.000	0.104	0.104	No	No
NEUROD2	Death	2	+	0.763	0.406	0.323	No	No
NR0B1	Time	2	+	1.000	0.083	0.083	No	No
REG1A	Stage	2	-	0.711	0.396	0.281	No	No
RNASE1	Diff	2	-	0.513	0.406	0.208	No	No
SERPIND1	Time	2	-	1.000	0.083	0.083	No	No
SLC26A2	Stage	2	+	1.000	0.083	0.083	No	No
TPSB1	Time	2	-	1.000	0.073	0.073	No	No
ABLIM	Stage	2	-	0.789	0.396	0.312	No	Yes
ADH1	Stage	2	-	0.718	0.406	0.292	No	Yes
ADH1	Diff	2	+	0.605	0.396	0.240	No	Yes
ADH1	Time	2	-	0.538	0.406	0.219	No	Yes
AOC3	Stage	2	-	0.718	0.406	0.292	No	Yes
CPA3	Stage	2	-	0.718	0.406	0.292	No	Yes
KIAA0101	Diff	2	-	0.513	0.406	0.208	No	Yes
MMP12	Stage	2	+	1.000	0.115	0.115	No	Yes
NULL	Time	2	+	0.513	0.406	0.208	No	Yes
OSF_2	Diff	2	-	0.692	0.406	0.281	No	Yes
SCTR	Diff	2	+	0.641	0.406	0.260	No	Yes
SLPI	Stage	2	-	1.000	0.104	0.104	No	Yes
TYMS	Diff	2	-	0.513	0.406	0.208	No	Yes
BENE	Diff	2	+	0.590	0.406	0.240	Yes	Yes
CDH17	Time	2	-	0.900	0.104	0.094	Yes	Yes
CDKN3	Diff	3	+	0.603	0.604	0.365	Yes	Yes
EMP2	Diff	2	+	0.553	0.396	0.219	Yes	Yes
EMP2	Time	2	-	0.564	0.406	0.229	Yes	Yes
IGFBP6	Time	2	-	0.462	0.406	0.187	Yes	Yes

Table 3. Rule set induced by HAMB for predicting patient death. The accuracy of this rule set on the Beer et al. data is 75%, with a coverage of 70%.

Rule	TP	FP	Sensitivity	PPV	P-value
(GAGE1 HIGH) ==> (DEATH? 1)	10	9	0.42	0.53	0.0036
(GPC3 MAX) ==> (DEATH? 0)	20	0	0.29	1	0.0007
(ELN-B MAX) ==> (DEATH? 0)	20	0	0.29	1	0.0007
(ADH7 LOW) ==> (DEATH? 1)	12	8	0.5	0.6	0.0002
(FMO2 LOW) ==> (DEATH? 1)	12	7	0.5	0.63	0.0001
(F10 MIN) ==> (DEATH? 1)	13	6	0.54	0.68	0

Table 4. Summary of some of the better rule sets generated by HAMB. PPV-SUM is the total of the positive predictive values of a rule set for all of the values of the target attribute.

ID	TOTAL-PPV	TARGET	SIZE	ACCURACY	COVERAGE	ACCURACY * COVERAGE
73	1.575	DEATH	6	0.75	0.7	0.52
100	1.469	DEATH	39	0.73	1	0.73
22	2.329	DIFFERENTIATION	5	0.7	0.56	0.4
18	2.19	DIFFERENTIATION	23	0.53	0.97	0.51
42	1.68	K-RAS-MUTATION	8	0.84	0.76	0.64
4	1.597	K-RAS-MUTATION	5	0.79	0.66	0.52
7	1.579	K-RAS-MUTATION	2	0.79	0.4	0.31
47	1.421	P53_NUCL_ACCUM	2	0.72	0.41	0.29
45	1.381	P53_NUCL_ACCUM	13	0.71	0.93	0.66
88	4.168	SURVIVAL TIME	32	0.81	0.73	0.59
86	3.985	SURVIVAL TIME	37	0.72	0.9	0.65
87	3.449	SURVIVAL TIME	17	0.83	0.5	0.42