What Kinds of Relational Features Are Useful for Statistical Learning?

Amrita Saha[1], Ashwin Srinivasan[2], and Ganesh Ramakrishnan[1]

[1] Department of Computer Science and Engineering,
Indian Institute of Technology,
Bombay, India
[2] Department of Computer Science, Indraprastha Institute of Technology,
New Delhi, India

Abstract. A workmanlike, but nevertheless very effective combination of statistical and relational learning uses a statistical learner to construct models with features identified (quite often, separately) by a relational learner. This form of model-building has a long history in Inductive Logic Programming (ILP), with roots in the early 1990s with the LINUS system. Additional work has also been done in the field under the categories of propositionalisation and relational subgroup discovery, where a distinction has been made between *elementary* and *non-elementary* features, and statistical models have been constructed using one or the other kind of feature. More recently, constructing relational features has become an essential step in many model-building programs in the emerging area of Statistical Relational Learning (SRL). To date, not much work—theoretical or empirical—has been done on what kinds of relational features are sufficient to build good statistical models. On the face of it, the features that are needed are those that capture diverse and complex relational structure. This suggests that the feature-constructor should examine as rich a space as possible, in terms of relational descriptions. One example is the space of all possible features in first-order logic, given constraints of the problem being addressed. Practically, it may be intractable for a relational learner to search such a space effectively for features that may be potentially useful for a statistical learner. Additionally, the statistical learner may also be able to capture some kinds of complex structure by combining simpler features together. Based on these observations, we investigate empirically whether it is acceptable for a relational learner to examine a more restricted space of features than that actually necessary for the full statistical model. Specifically, we consider five sets of features, partially ordered by the subset relation, bounded on top by the set F_d, the set of features corresponding to definite clauses subject to domain-specific restrictions; and bounded at the bottom by F_e, the set of "elementary" features with substantial additional constraints. Our results suggest that: (a) For relational datasets used in the ILP literature, features from F_d may not be required; and (b) Models obtained with a standard statistical learner with features from subsets of features are comparable to the best obtained to date.

F. Riguzzi and F. Železný (Eds.): ILP 2012, LNAI 7842, pp. 209–224, 2013.

1 Introduction

The emerging area of statistical relational learning (SRL) is characterised by a number of distinct strands of research. Especially prominent is research concerned with the construction of models from data that use representations based on either first-order logic programs augmented with probabilities or probabilistic graphical models. The concerns here are the usual ones to do with expressive power, estimation, and inference: what kinds of probabilistic models can be constructed with one representation or the other; how can we estimate the structure and parameters in the model; how do we answer queries exactly, given data that is observed, and perhaps missing; and so on. The combination of relational learning with statistical modeling, however, has a longer history within Inductive Logic Programming (ILP), with origins at least as early as 1990, with the LINUS system [14]. Since then, there are have been regular reports in the literature on the use of ILP systems, as a tool for constructing relational features for use in statistical modeling [22].

An argument can be made that construction of relational features must necessarily require some form of first-order learning, of which ILP is an instance (for example, see [13]). Arguments in-principle aside, the literature also suggests that augmenting any existing features with ILP-constructed relational ones can substantially improve the predictive power of a statistical model [24,4,22]. There are thus good practical reasons to persist with this variant of statistical and logical learning. On the other hand there has been some work done on comparing the different kinds of propositionalisation techniques used to transform the search space from the space of first-order hypothesis to the space of propositional features which can be handled by more scalable propositional/statistical learners. [12] claims that of the two main kinds of propositionalization methods, namely logic oriented and database-oriented, both have their specific advantages. While logic-oriented methods can handle complex relational structures in the form of background knowledge and provide more expressive relational models, database-oriented models are much more scalable. According to their empirical findings, a combination of features from these two groups are necessary for learning good models.

Even within this well-trodden corner of statistical relational learning (no more, perhaps, than a "poor man's SRL"), there are some issues that remain unaddressed. To date, not much work—theoretical or empirical—has been done on what kinds of relational features are sufficient to build good statistical models. On the face of it, the features that are needed are those that capture diverse and complex relational structure. This suggests that the feature-constructor should examine as rich a space as possible, in terms of relational descriptions. One example is the space of all possible features in first-order logic, given constraints of the problem being addressed. Practically, it may be intractable for a relational learner to search such a space effectively for features that may be potentially useful for a statistical learner. Additionally, the statistical learner may also be able to capture some kinds of complex structure by combining simpler features together. For example, a statistical learner like a support vector machine or

logistic regression that may also be able to approximate the effect of a conjunction of these features using their weighted sum. Reports in the ILP literature suggest at least 5 kinds of relational feature classes: (1) F_d: (the set of) features from definite clauses with no restrictions other than those of the domain [22,24]; (2) F_i: features from "independent" clauses that place restrictions on the sharing of existential variables [2]; (3) F_r: features denoting a class of relational subgroup that place additional restrictions on the use of existential variables in independent clauses [15]; (4) F_s: features from "simple" clauses in the sense described in [17]; and (5) F_e: features developed from the class of "elementary" clauses described in [16]. In this paper, we show certain subset relationships hold between these sets. These are shown diagrammatically in Fig. 1. When exploring whether smaller feature-spaces are adequate, we use these relationships to investigate empirically whether exploring larger sets of features adds any significant predictivity to a statistical learner. Several Statistical Relational Learning approaches in the past have focused on learning from specific classes of features and construct more complicated ones if necessary by boosting. In [10,19] it has been empirically shown in various learning settings, that boosting of weak features performs well. The same technique of boosting applied on different settings, by using differnt variants of the loss functions have been discussed repeatedly in SRL literature, for example, in [5,8,9,21]. There has also been attempt at posing the relational feature construction as a problem of combining macro-operators by statistical learner and delegating search to them [1]. In this macro-operator paradigm, it has been shown that by following a particular propositionalization technique, the trade-off between feature construction cost and learning cost can be better handled and infact the propositionalized dataset becomes PAC-learnable.

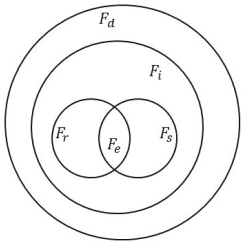

Fig. 1. Relationships between the sets of features considered in this paper

The rest of the paper is organised as follows. In Section 2 we describe the mapping between clauses and relational features. We also describe in greater detail the five feature classes F_d–F_e. In Section 3 we derive the relationships shown in Fig. 1 and enumerate some consequences that follow directly from them. Section 4 describes an empirical investigation, using standard statistical learners of the smallest size feature class that appears to be useful for constructing predictive models for several ILP benchmark datasets. Section 5 concludes the paper.

2 Feature Classes

In this paper, we will take first-order relational features simply to be first-order clauses that either maximize some objective either collectively or individually (effectively, a restatement of the distinction in [7] between strong and weakly relevant features, in optimisation terms). Specifically, we will assume that is a one-to-one correspondence between first-order relational features and definite clauses (alphabetic variants of a clause are treated as being the same clause). Formally, we adopt the same notation as in [22] for a function that maps a first-order definite clause to a feature. The set of examples provided to an ILP system can be defined as a binary relation which is a subset of the Cartesian product $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ denotes the set of individuals (*i.e.* structured objects consisting of first-order predicates) and $\mathcal{Y}$ denotes the finite set of classes. The definite clauses obtained as an output of the ILP system can be represented in the form $h_j(x, c) : Class(x, c) \leftarrow Cp_j(x)$ where $Cp_j : \mathcal{X} \rightarrow \{0, 1\}$ is a nonempty conjunction of predicates on the variable $x \in \mathcal{X}$ and c is the class variable specified in the head predicate. For convenience, we will say $Head(h_j(x, c)) = Class(x, c)$ and $Body(h_j(x, c)) = Cp_j(x)$. Given a clause $h_j(x, c)$, a first-order feature can be defined as $f_j(x) = 1$ iff $Body(h_j(x, c)) = 1$ and 0 otherwise. Constraints on $Body(h_j(x, c))$ allow us to define several kinds of features. Following [17] (with a small difference) we distinguish between "source" predicates and "sink" predicates in the language of clauses allowed in the domain. The former are those that contain at least one output argument and any number of input arguments (in the sense used by the mode declarations in [18]) and the latter are those that contain no output arguments. In a clause, thus all new existentially quantified variables should be introduced in the source literals and not in the sink literals. [1] Additionally, following [2], for a clause $h_j(x, c)$, the independent components in $Body(h_j(x, c))$ are partitions of the literals in $Body(h_j(x, c))$ into sets, such that each partition consists of a "connected" set of literals (once again, using input and output variables in the sense of [18]), and that literals across partitions only share the head variable x. We are then able to define the following kinds of relational feature classes:

The class F_d: This consists of first-order features obtained from definite clauses $h_j(x, c)$ in the functional manner described above. That is, no constraints are placed on $Body(h_j(x, c))$. Features used in [24], for example, belong to this class.

The class F_i: This is a restricted version of the class F_d, consisting of features from clauses $h_j(x, c))$ such that $Body(h_j(x, c))$ consists of exactly 1 independent component.

The class F_r: This is a restricted version of the class F_i, consisting of features obtained from clauses $h_j(x, c)$ such that $Body(h_j(x, c))$ consists of exactly 1 independent component, and with an additional constraint that all new existential variables introduced by a source literal appear in source or sink literals in $Body(h_j(x, c))$. The features in [15] are from this class, and the first-order

[1] "Structural" predicates used in [16] are thus binary source literals that introduce a single new variable, and "property" predicates are sink literals.

features described in [16] are a special case of features in F_r (the special case arising from restrictions on sources and sinks to structural and property predicates as described earlier).

The class F_s: This consists of features obtained from clauses $h_j(x, c)$ such that h_j is a simple clause in the sense defined by [17]. That is, $Body(h_j(x, c))$ contains exactly 1 sink literal.

The class F_e: This is a restricted version of the class F_r, consisting of features obtained from clauses $h_j(x, c)$ such that $Body(h_j(x, c))$ contains exactly 1 sink literal. "Elementary" features described in [16] are a special case of features in F_e.

3 Relationships between Feature Classes

Some of the relationships between feature classes are evident from the definitions. That is: $F_r \subseteq F_i \subseteq F_d$. The following additional statements hold

- $F_s \subseteq F_i$
- $F_s \not\subseteq F_r$ and $F_r \not\subseteq F_s$
- $F_e = F_r \cap F_s$

Thus, the feature classes exhibit the following hierarchical structure:

$$F_e = (F_r \cap F_s) \subseteq F_r \subseteq F_i \subseteq F_d$$
$$F_e = (F_r \cap F_s) \subseteq F_s \subseteq F_i \subseteq F_d$$

Given these subset relationships between feature classes, it is also of some importance to consider whether there exists a way of reconstructing, using logical operations, every feature in a superset class by combining features from a subset (or smaller) class (clearly, the reverse is always possible: a feature in a subset class can always be constructed from a feature in the superset class). The interest here is of course that if such logical relationships hold, then features in the larger class may be approximated by statistical learners using weighted combinations of features from the smaller class. The following logical relationships hold between the feature classes:

- Every feature in F_i can be constructed from features in F_s
- Every feature in F_d can be constructed from features in F_i

Thus, every feature in F_d can be constructed from F_s. In addition:

- Not every feature in F_i can be constructed from features in F_r
- Not every feature in F_r can be constructed from features in F_e
- Not every feature in F_s can be constructed from features in F_e

The proofs of the relations have been elaborated in Appendix A. We now evaluate empirically the utility of features from the different classes, proceeding from the smallest (F_e) to the largest (F_d).

4 Empirical Evaluation

4.1 Aims

Our aim is to obtain empirical evidence of the smallest feature class that is found to yield good statistical models. Specifically, we mean that we wish to find using a set of well-studied benchmark data sets and popular statistical learners, whether there is a feature class such that adding features from larger classes yields no significant increases in predictive accuracy.

4.2 Materials

Domains. We use biochemical datasets that have been used in a range of papers in the ILP literature. These are tabulated below.These datasets have been used widely: see, for example

Dataset	No. of instances for positive class	No. of instances for negative class
Alz (Amine) [25]	343	344
Alz (Acetyl) [25]	443	443
Alz (Memory) [25]	321	321
Alz (Toxic) [25]	443	443
Carcin [25]	182	155
DSSTox [25]	220	356
Mut(188) [25]	125	63

Fig. 2. Datasets used in the paper

4.3 Algorithms and Machines

The statistical learners used in the paper are these:

- A support vector machine(SVM). Specifically, we examine SVMs with both the L1-norm, also known in literature as L1-SVM (the LibLINEAR implementation) and the L2-norm, known as L2-SVM,(the LibSVM implementation) on the weight vector used as the regularizer. The L1-norm is known to induce sparsity on the feature space, forcing a form of feature selection which is important when the number of features is large.
- Logistic regression. Specifically, we consider a standard version of this technique and a faster and more efficient variant called SMLR (Sparse Multinomial Logistic Regression) [11]
- Rule ensemble learning using maximum likelihood estimation (MLRules) [3]. This employs a greedy approach of maximising likelihood to construct an ensemble of rules from the features. Like logistic regression, the weights of the rules are derived from the conditional probability distribution learnt. So it can be thought of as a generalization of Logistic Regression, where the space explored to construct rule ensembles can be considered as an approximation of conjunctions of rules/features.

All statistical learners here, compute the decision function as a weighted linear combination of the features (which can be thought of as an approximation to a logical conjunction). Features in each feature class are constructed using the ILP system Aleph [23]. It is possible to enforce the constraints associated with each feature class in a straightforward manner as part of background knowledge provided to this system.

All experiments were conducted on a machine equipped with a 8-core Intel i7 2.67GHz processors and 8 gigabytes of random access memory.

4.4 Method

The methodology adopted for providing the statistical learners features from feature classes F_e to F_d is as follows:

1. Select a total ordering on the classes $F_e \prec \cdots \prec F_d$ that is consistent with the partial ordering imposed by subset relationships between these sets.
2. For each dataset and each statistical learner:
   (a) For sets S_j from smallest (F_e) to the largest (F_d), in the total ordering $\prec$:
   (b) Let $F_0 = \emptyset$
      i. Repeat R times:
         A. Obtain a set of features F from S_j
         B. $F_j = F_{j-1} \cup F$
         C. Obtain an estimate of the predictive accuracy A_j of the statistical learner with features F_j
      ii. Obtain the mean predictive accuracy $\overline{A_j}$ across the R repeats
3. Determine the smallest set S_k after which there are no significant changes in mean predictive accuracy $\overline{A_k}$

The following details are relevant:

1. There are only 2 total orderings possible, given the subset relationships between the feature classes: $F_e \prec_1 F_r \prec_1 F_s \prec_1 F_i \prec_1 F_d$ and $F_e \prec_2 F_s \prec_2 F_r \prec_2 F_i \prec_2 F_d$. According to our incremental algorithm, here F_s in the 5th column, F_s in $\prec_1$ and F_r in $\prec_2$ actually refer to sets of features from $F_r \cup F_s$, because of their order of appearance. The results we report here are with $\prec = \prec_1$. Our conclusions do not change with either $\prec_1$ or $\prec_2$.
2. All predictive accuracies are estimated using 10-fold cross-validation.
3. The results are averaged over $R = 4$ repetitions. Larger values of R would result in smaller standard errors of the mean estimate.
4. Statistical learners have parameters that require optimisation. It has been shown elsewhere [25] that using default values of parameters can result in suboptimal models, which can clearly confound the conclusions that can be drawn here. For each set of training data in a cross-validation run, we set aside some small part of the training data as a "validation" set, and use this to tune the parameters of the statistical learner. The "best" parameter value that results is then used to construct a model on the training data; and then evaluated on

the test dataset of that cross-validation run. This does not necessarily yield the best model possible, but accuracies are usually higher than those obtained with the default settings for the parameters.

5. The feature-construction procedure employed by the ILP engine requires an upper-bound k on the number of features produced for each class label in the example data. For experiments reported here, $k = 500$. There is a randomised element to the feature-construction procedure in Aleph, which can been seen as selecting (non-uniformly) upto k features, from the set of all possible features allowed. The selection of features is controlled by parameters that correspond to precision and recall in the data mining literature. For experiments here, these values have been deliberately left low (and hence, easy to obtain for the ILP engine). The intent is that the statistical learner should be able to combine these to obtain higher values.
6. We have elected to assess the utility of features from each feature class by examining the improvements in mean accuracy by augmenting features already present from feature classes earlier in the ordering $\prec$. The alternative of simply comparing accuracies with a new set of features drawn from the from each feature class X in the ordering would have confounded matters, since this set could contain features from a class $Y \subseteq X$. It would not then be known whether the increases in accuracy, if any, are due to features from a class X, or those from class Y.
7. It is known that for large R, mean accuracies are distributed normally. For small R, it is known that the sampling distribution of the mean is a t-distribution with $R-1$ degrees of freedom. This is used in statistical comparisons when needed (as will be seen, it is often evident whether differences are significant, and no undue statistical testing is needed).

4.5 Results and Discussion

The comparative performance of the models is shown in Fig. 3. The principal result from this tabulation is this: broadly, there is little value in including features from F_d (the largest feature class considered here). Examining the table in greater detail, we are able to obtain the number of "wins" for each feature class (that is, the number of times the highest predictive accuracy results from using features in that class). A cross-tabulation of this against the learners is shown in Fig. 4:

The data in Figs. 3 and 4 suggest that it is sufficient to consider features from F_i (that is, features from clauses containing one independent component). Now, while it is possible to reconstruct every feature in F_d exactly as a simple logical conjunction, of some features in F_i, it is also possible to reconstruct several (but not all) features in F_d by simple logical conjunction of select features in F_r, F_s and F_e. This appears to be exploited by all of the statistical learners here, since it is by no means necessary for any of them that features from F_d are consistently required to produce the best results. This is reinforced further, if rather than considering outright wins, we consider a model as being good enough if there no (statistically) significant difference to the best model. Then, the performance tallies can be summarized as in Fig. 5.

Data	Learner	Using Features From				
		F_e	F_r	F_s	F_i	F_d
Alz(Amine)	L2-SVM	66.14±0.84	65.71±0.69	67.23±0.86	**81.62±0.92**	81.45±0.61
	L1-SVM	65.50±0.15	65.84±0.14	64.91±1.65	78.34±4.18	**80.31±1.57**
	LR	65.42±0.0	65.42±0.0	64.38±0.0	79.32±3.93	**81.28±0.31**
	SMLR	65.42±0.0	66.05±0.22	67.12±0.38	81.20±2.56	**81.63±2.77**
	MLRules	64.39±0.0	64.39±0.0	68.67±0.0	**82.32±1.18**	82.22±0.35
Alz(Acetyl)	L2-SVM	69.42±0.33	68.52±0.70	68.59±2.11	**74.40±0.80**	72.97±0.47
	L1-SVM	69.85±0.05	68.55±0.0	66.20±0.0	**74.06±0.91**	72.78±1.44
	LR	69.82±0.0	69.82±0.0	71.93±0.0	73.87±0.44	**74.16±0.24**
	SMLR	70.36±0.11	70.32±0.14	72.17±0.12	**73.34±1.38**	73.20±0.65
	MLRules	69.59±0.0	69.59±0.0	70.95±0.0	71.67±0.47	**72.44±0.52**
Alz(Memory)	L2-SVM	59.80±0.22	61.09±0.45	63.46±1.07	**68.39±1.15**	68.07±1.75
	L1-SVM	56.99±1.29	62.89±1.07	64.33±0.25	**71.29±1.04**	67.96±0.41
	LR	59.07±0.08	59.07±0.08	65.87±0.0	**70.31±1.54**	69.30±0.55
	SMLR	59.69±0.0	58.44±1.32	66.34±0.73	70.10±1.26	**71.83±1.67**
	MLRules	58.83±0.19	58.83±0.19	64.76±0.08	**70.13±1.14**	67.91±0.64
Alz(Toxic)	L2-SVM	70.72±1.0	71.97±1.99	77.17±0.82	**83.01±0.73**	82.04±2.40
	L1-SVM	70.06±0.0	71.73±0.0	80.94±3.15	82.59±0.70	**82.83±2.21**
	LR	72.84±0.0	72.09±1.51	77.91±0.0	81.65±0.79	**83.32±0.87**
	SMLR	74.50±2.35	74.07±0.79	76.76±0.16	82.59±0.39	**84.21±0.69**
	MLRules	73.18±0.0	73.18±0.0	77.44±0.0	**84.50±0.44**	84.19±0.56
Carcin	L2-SVM	60.59±1.58	**62.15±1.75**	59.49±1.75	62.03±1.28	59.85±1.67
	L1-SVM	**61.21±0.0**	59.71±0.0	58.66±3.10	60.26±0.71	59.21±1.29
	LR	**59.00±0.0**	**57.71±0.16**	54.01±2.79	51.80±1.37	52.52±3.49
	SMLR	**60.88±0.0**	60.17±0.38	59.51±2.91	58.80±3.42	57.94±2.84
	MLRules	**59.39±0.25**	59.20±0.15	58.83±1.96	57.94±2.31	56.64±0.94
DSSTox	L2-SVM	69.36±1.15	69.86±1.19	72.21±1.22	**73.12±0.94**	70.90±3.15
	L1-SVM	69.25±0.25	69.00±0.25	70.77±1.60	**71.91±1.77**	69.21±1.34
	LR	70.49±0.0	70.49±0.0	**72.18±1.71**	71.01±0.87	70.32±0.47
	SMLR	72.24±0.0	**72.29±0.09**	70.71±2.62	70.95±2.94	71.02±1.38
	MLRules	71.55±0.0	71.55±0.0	**72.85±1.43**	72.73±1.52	72.16±1.09
Mut(188)	L2-SVM	–	65.80±1.25	85.84±1.36	**86.49±0.28**	84.63±1.58
	L1-SVM	–	63.70±0.30	84.92±0.65	85.27±0.31	**85.52±0.48**
	LR	–	67.12±0.0	74.85±1.74	77.48±3.38	**77.61±2.88**
	SMLR	–	73.44±0.0	**88.06±1.57**	85.83±1.81	84.10±2.02
	MLRules	–	71.86±0.0	85.04±0.76	**86.50±1.40**	85.70±1.55

Fig. 3. Mean predictive accuracies of statistical models including features from F_e to F_d. "–" indicates no features in this class were possible given the domain constraints. Here, refers to the F_s in the incremental order $\prec_1$, hence it actually denotes the set of features from $F_r \cup F_s$. Same notation has been followed in Fig. 4. and Fig. 5.

Feature Class	Number of Wins					Total Wins
	L1-SVM	L2-SVM	LR	SMLR	MLRules	
F_e	1	0	1	1	1	4/35
F_r	0	1	1	1	0	3/35
F_s	0	0	1	1	1	3/35
F_i	3	6	1	1	4	15/35
F_d	3	0	4	3	1	11/35

Fig. 4. Number of outright wins for a feature class. This is number of occasions out of the total number of possible occasions (*i.e.* 35) on which a statistical learners achieves the highest mean predictive accuracy using features from that class.

Feature Class	Number of Good Enough Models					Total No. of Good Models
	L1-SVM	L2-SVM	LR	SMLR	MLRules	
F_e	2	1	2	2	2	9/35
F_r	2	2	1	4	4	13/35
F_s	3	2	1	4	4	14/35
F_i	7	7	6	7	7	34/35
F_d	4	5	6	7	7	25/35

Fig. 5. Number of good enough models (out of all possible models *i.e.* 35), using a feature class. A model is taken to be good enough if its predictive accuracy is not statistically different to the model with the highest predictive accuracy.

Data	Statistical Model	ILP Model With Parameter Selection & Optimization
Alz (Amine)	82.32±1.18	80.20
Alz (Acetyl)	74.16±0.24	77.40
Alz (Memory)	71.83±1.67	67.40
Alz (Toxic)	84.50±0.44	87.20
Carcin	62.15±1.75	59.10
DSSTox	73.12±0.94	73.10
Mut(188)	88.06±1.57	88.30

Fig. 6. Comparison of mean predictive accuracies of statistical models against the ILP models constructed with parameter selection and optimisation (see [25])

These tabulations do suggest that of the classes $F_e \ldots F_d$, the class F_i may be the most useful. But how good are the models obtained with F_i, when compared against the ILP models reported in the literature? Fig. 6 shows a comparison of the best statistical models against the predictive accuracies of the ILP models obtained after parameter selection and optimisation [25].

Finally, although not relevant to the aims of the experiment here, we note some exceptional behaviour on the "Carcin" dataset, in which there is a fairly consistent trend of decreasing predictive accuracies as we progress from F_e to F_d. An examination of re-substitution (training-set) accuracies for this data shows the opposite trend, suggesting that these data may be especially prone to overfitting.

We also experimented with $\prec = \prec_2$ and found that the results were quite similar with the ones with $\prec = \prec_1$ tabulated here. In interest of space, those results have been omitted.

The methodology of experimentation discussed above, constructs a total ordering of the feature classes by cumulatively augmenting the features already obtained from the feature classes earlier in the ordering $\prec$. But this process of incrementally adding features can increase the size of the hypothesis space. Increasing the size of the hypothesis space naturally increases the possibility that models perform better simply due to chance effects. We investigate for this by sampling a fixed number of features from each feature- class only, without cumulatively augmenting the already present features generated by its subsumed feature classes. The second set of experiments based on this setting has been elaborated in Table 7 and the comparative study of the performance of the different feature

Data	Learner	Using Features From					
		F_e	F_r	F_s	$F_{r\cup s}$	F_i	F_d
Alz(Amine)	L2-SVM	65.27	66.15	79.54	78.29	**79.56**	77.82
	L1-SVM	63.95	68.19	76.91	**78.12**	77.38	74.61
	LR	65.56	66.45	79.98	**80.91**	79.13	79.47
	SMLR	67.12	67.75	79.10	78.76	**79.33**	79.17
	MLRules	65.42	66.0	80.73	80.62	**80.74**	78.74
Alz(Acetyl)	L2-SVM	66.83	67.02	69.37	67.55	**73.53**	72.54
	L1-SVM	65.13	65.28	69.22	67.55	**73.53**	72.54
	LR	66.75	67.17	67.55	68.01	**73.53**	72.54
	SMLR	66.68	68.14	68.09	**71.01**	69.88	70.2
	MLRules	66.30	67.56	69.96	**72.22**	69.60	70.35
Alz(Memory)	L2-SVM	59.8	61.21	64.94	63.38	67.46	**69.3**
	L1-SVM	59.8	61.21	64.94	63.38	67.46	**69.3**
	LR	59.8	61.21	64.94	63.38	67.46	**69.3**
	SMLR	60.88	59.17	67.73	**71.01**	68.04	67.17
	MLRules	60.411	59.78	69.91	**72.56**	68.52	66.63
Alz(Toxic)	L2-SVM	70.47	74.86	77.08	76.52	**82.06**	81.59
	L1-SVM	75.67	75.55	80.28	79.4	**84.68**	81.40
	LR	70.47	74.86	77.08	76.52	**82.06**	81.59
	SMLR	71.27	58.85	78.94	**82.78**	79.03	81.43
	MLRules	70.33	55.90	80.33	**82.22**	79.79	81.43
Carcin	L2-SVM	58.81	**63.87**	59.29	58.41	63.23	57.58
	L1-SVM	58.81	**63.87**	59.29	58.41	63.23	57.58
	LR	58.81	**63.87**	59.29	58.41	63.23	57.58
	SMLR	**61.16**	60.50	59.07	55.43	59.37	59.32
	MLRules	**60.52**	55.26	56.64	54.23	57.29	55.67
DSSTox	L2-SVM	70.32	69.69	71.25	71.25	**73.65**	66.87
	L1-SVM	70.32	69.69	71.25	71.25	**73.65**	66.87
	LR	70.32	69.69	71.25	71.25	**73.65**	66.87
	SMLR	71.77	**73.67**	72.01	71.06	69.63	71.48
	MLRules	66.98	69.10	71.84	71.11	**72.48**	72.47
Mut(188)	L2-SVM	–	–	85.14	85.14	**86.66**	82.45
	L1-SVM	–	–	85.14	85.14	**86.77**	82.45
	LR	–	–	85.14	85.14	**86.66**	82.45
	SMLR	–	–	**73.43**	**73.43**	64.95	72.72
	MLRules	–	–	74.43	74.43	65.47	**76.93**

Fig. 7. Mean predictive accuracies of statistical models including features from F_e to F_d. "–" indicates no features in this class were possible given the domain constraints. Here, refers to the $F_{r\cup s}$ actually denotes the set of features from $F_r \cup F_s$. Same notation has been followed in Fig. 4. and Fig. 5.

Feature Class	Total Number of Outright Wins	Total Number of Good Models
F_e	2/35	3/35
F_r	4/35	3/35
F_s	1/35	9/35
$F_{r\cup s}$	8/35	16/35
F_i	**16/35**	**27/35**
F_d	4/35	17/35

Fig. 8. Number of outright wins for a feature class and the Number of good enough models generated from features obtained from those classes

classes in terms of number of outright wins and number of good enough statistical models constructed, has been presented in 8.

5 Concluding Remarks

In this paper we have explored the relationship between several feature spaces that have been reported in ILP literature and examined empirically whether it is possible to construct good statistical models using features from smaller spaces. The intuition underlying this is that statistical models may often be able to approximate the effect of more elaborate features by weighted combinations of simpler features. Our results suggests that the class F_i, consisting of features constructed from clauses containing exactly one independent component seems to be particularly useful. This makes some sense: a linear combination of multiple features from F_i can approximate the reconstruction of a full first-order feature, since no variable sharing is required between such features. In fact, this leads us to hypothesize that statistical learners like [6] that perform conjunctive feature learning will not perform any better than learners using weighted combinations of features from F_i, and will incur a greater computational cost. Further, this also leads us to believe that weighted linear combinations of first order clauses in relational learning models such as MLN [20], Relational Markov Networks [26] and the boosting techniques like [10,19] could be efficiently and effectively approximated by weighted linear combinations of clauses from simpler classes such as F_i and is part of our ongoing work. On the other hand the main focus of this paper has been directed at learning a discriminative classification model whereas the general Statistical Relational Learning also addresses a generative model learning setting. It would be interesting to see how this study of learning from specific feature classes can make the generative learning more efficient.

Acknowledgements. Ashwin Srinivasan is supported by a Ramanujan Fellowship of the Government of India. He is also a Visiting Professor at the Department of Computer Science, Oxford University; and a Visiting Professorial Fellow at the School of CSE, University of NSW, Sydney.

A Appendix A

A.1 Proofs of Results from Section 3

Relationship between F_e and F_s : $F_e \subseteq F_s$.

Proof. Every feature from the class F_e also belongs to the class F_s, since it is minimal (*i.e.* cannot be decomposed into smaller features that share only global variable), and contains a single sink (*i.e.* cannot be decomposed into smaller features that share only a local variable). On the other hand, a feature from the class F_s may not belong to the class F_e, since the former may have some unused output variables, For example, in the trains problem originally proposed by Ryzhard Michalski, $eastbound(A) \leftarrow hasCar(A, B)$ is a valid clause from F_s but it cannot belong to class F_e.

Relationship between F_s and F_r : $F_s \not\subset F_r$. And the reverse is neither true, *i.e.* $F_r \not\subset F_s$.

Proof. Every feature from the class F_s may not belong to the class F_r, attributed to the same reason of local-variable reusal property that the latter must satisfy, a property that need not hold for the former. The same clause mentioned above as an example, *viz.*, $eastbound(A) \leftarrow hasCar(A, B)$, will not yield a feature from F_r (or for that matter, from F_e), though it is a valid clause from F_s. Also the reverse is not true *i.e.* every feature from F_r may not belong to F_s, since the former can have any number of property nodes *i.e.* sinks, which is not possible in the case of the latter.

Relationship between F_e, F_r and F_s : $F_e = (F_r \cap F_s)$.

Proof. By the very definition of a feature from class F_e, it must belong to class F_r and since it cannot have more than one sink, it must also be a valid feature from class F_s.

Relationship between F_r and F_i : $F_r \subseteq F_i$

Proof. Every feature from class F_r also belongs to class F_i because of the minimal property, but the reverse is not true *i.e.* not every feature from class F_i is a valid feature from class F_r again because of the variable reusal property. For example, $eastbound(A) \leftarrow hasCar(A, B)$ is a valid feature from class F_i but cannot belong to class F_r because of the unused output variable C introduced by the structural predicate, $hasCar(A, B)$. Also there may not exist any property predicate in the clause from class F_i as in this example.

Relationship between F_s and F_i : $F_s \subseteq F_i$.

Proof. By the very definitions of features from class F_s and F_i, it can be seen that every feature from class F_s is a valid feature from class F_i but the reverse is not true.

Relationship between F_i and F_d : *i.e.* $F_i \subseteq F_d$

Proof. It is obvious that every feature from class F_i is a first-order definite feature. But the reverse is not true, since there exist clauses such as the following one from the trains problem,
$eastbound(A) \leftarrow hasCar(A, B), hasCar(A, C), short(B), closed(C)$
that are not independent (since they can be decomposed further into independent components).

A.2 Reconstruction Property of Feature Classes

1. Every full first-order feature from definite clauses *i.e.* from F_d can be constructed from features from F_s.

   Proof. The one-to-one mapping from clauses to features allows any feature f_d from F_d to be inverted to a definite clause c_d. Also [17] states that every definite clause can be constructed from simple causes. So given a set of simple clauses S_s that can reconstruct the first order definite clause c_d, the mapping can be used to construct the set of features corresponding to the set S_s of simple clauses used to reconstruct c_d and hence feature f_d.

2. Every feature from F_i can be constructed from features from F_s.

   Proof. Since every feature from F_i is also a full first-order feature (owing to the subset relation) and every full first-order feature from F_d can be reconstructed from features from F_s, it follows that every feature from F_i can be reconstructed from features from F_s by performing a combination of logical conjunction and (possibly) variable unification.

3. Every feature from F_r can be constructed from features from F_s.

   Proof. Similar to the above justification of (2), since each feature from F_r is also a feature from F_d, it can be constructed from features from F_s by performing their logical conjunction and (possibly) variable unification.

4. Every feature from F_e can be constructed from features from F_s.

   Proof. This follows from the subset relation $F_e \subseteq F_s$. In general, any feature of a subclass can be constructed from its superclass. This is just stating the obvious.

5. Every full first-order feature from F_d can be constructed from features from F_i.

   Proof. This follows from (1), since every full first-order feature from F_d can be constructed from features from F_s and the latter is a subset of F_i. In this case the reconstruction is much simpler since it is equivalent to logical conjunction without any variable unification required.

6. Not every feature from F_r can be constructed from features from F_e.

   Proof. Because of the redefined structural predicates that can introduce any non-zero number of new variables, there can be features from F_r that reuse these newly introduced variables separately in different property predicates. In that case no single property predicate will be sufficient to satisfy the variable-reusal property. The clauses corresponding to these features from F_r cannot be reconstructed from any number of clauses obtained by the inverse-mapping from features from F_e. For example, the feature corresponding to the clause $eastbound(A) \leftarrow hasCarFollowsCar(A, B, C), short(B), closed(C)$ is a valid feature belonging to F_r which cannot be constructed from features obtained from F_e class alone.

7. Not every full first-order feature from F_d can be constructed from features from F_r.

   Proof. Clauses corresponding to features from F_d that do not satisfy the variable-reusal property cannot be reconstructed from any number of clauses inverse-mapped from features from F_r.

   For the same reason as above, the following two properties additionally hold.

8. Not every feature from F_i can be constructed from features from F_e.
9. Not every full first-order feature from F_d can be constructed from features from F_e.

References

1. Alphonse, É.: Macro-operators revisited in inductive logic programming. In: Camacho, R., King, R., Srinivasan, A. (eds.) ILP 2004. LNCS (LNAI), vol. 3194, pp. 8–25. Springer, Heidelberg (2004)
2. Santos Costa, V., Srinivasan, A., Camacho, R., Blockeel, H., Demoen, B., Janssens, G., Van Laer, W., Cussens, J., Frisch, A.: Query transformations for improving the efficiency of ILP systems. Journal of Machine Learning Research 4, 491 (2002)
3. Dembczynski, K., Kotlowski, W., Slowinski, R.: Maximum likelihood rule ensembles. In: ICML, pp. 224–231 (2008)
4. Gottlob, G., Leone, N., Scarcello, F.: On the complexity of some inductive logic programming problems. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 17–32. Springer, Heidelberg (1997)
5. Gutmann, B., Kersting, K.: TildeCRF: Conditional random fields for logical sequences. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 174–185. Springer, Heidelberg (2006)
6. Jawanpuria, P., Nath, J.S., Ramakrishnan, G.: Efficient rule ensemble learning using hierarchical kernels. In: ICML, pp. 161–168 (2011)
7. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Machine Learning: Proceedings of the Eleventh International, pp. 121–129. Morgan Kaufmann (1994)
8. Karwath, A., Kersting, K., Landwehr, N.: Boosting relational sequence alignments. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 857–862. IEEE Computer Society, Washington, DC (2008)
9. Kersting, K., Driessens, K.: Non-parametric policy gradients: a unified treatment of propositional and relational domains. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 456–463. ACM, New York (2008)
10. Khot, T., Natarajan, S., Kersting, K., Shavlik, J.: Learning markov logic networks via functional gradient boosting. In: Proceedings of the IEEE 2011 11th International Conference on Data Mining, ICDM 2011, pp. 320–329. IEEE Computer Society, Washington, DC (2011)
11. Krishnapuram, B.I., Carin, L., Figueiredo, M.A.T., Hartemink, A.J.: Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Mach. Intell. 27(6), 957–968 (2005)
12. Krogel, M.A., Rawles, S., Zelezny, F., Flach, P.A., Lavrac, N., Wrobel, S.: Comparative evaluation of approaches to propositionalization (2003)
13. Landwehr, N., Passerini, A., De Raedt, L., Frasconi, P.: kfoil: Learning simple relational kernels. In: AAAI, pp. 389–394. AAAI Press (2006)
14. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. Routledge, New York (1993)
15. Matwin, S., Sammut, C. (eds.): ILP 2002. LNCS (LNAI), vol. 2583. Springer, Heidelberg (2003)
16. Matwin, S., Sammut, C. (eds.): ILP 2002. LNCS (LNAI), vol. 2583. Springer, Heidelberg (2003)
17. McCreath, E., Sharma, A.: LIME: A system for learning relations. In: Richter, M.M., Smith, C.H., Wiehagen, R., Zeugmann, T. (eds.) ALT 1998. LNCS (LNAI), vol. 1501, pp. 336–374. Springer, Heidelberg (1998)
18. Muggleton, S.: Inverse entailment and progol. New Generation Computing 13, 245–286 (1995)

19. Natarajan, S., Khot, T., Kersting, K., Gutmann, B., Shavlik, J.: Gradient-based boosting for statistical relational learning: The relational dependency network case. Mach. Learn. 86(1), 25–56 (2012)
20. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. 62(1-2), 107–136 (2006)
21. Tadepalli, P., Kristian, K., Natarajan, S., Joshi, S., Shavlik, J.: Imitation learning in relational domains: a functional-gradient boosting approach. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 2, pp. 1414–1420. AAAI Press (2011)
22. Specia, L., Srinivasan, A., Joshi, S., Ramakrishnan, G., das Graças Volpe Nunes, M.: An investigation into feature construction to assist word sense disambiguation. Machine Learning 76(1), 109–136 (2009)
23. Srinivasan, A.: The aleph manual (1999)
24. Srinivasan, A., Muggleton, S., Sternberg, M.J.E., King, R.D.: Theories for mutagenicity: A study in first-order and feature-based induction. Artif. Intell. 85(1-2), 277–299 (1996)
25. Srinivasan, A., Ramakrishnan, G.: Parameter screening and optimisation for ILP using designed experiments. Journal of Machine Learning Research 12, 627–662 (2011)
26. Taskar, B., Abbeel, P., Wong, M.-F., Koller, D.: Relational markov networks. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning. MIT Press (2007)