Using Feature Selection with Bagging and Rule Extraction in Drug Discovery*

Ulf Johansson[1], Cecilia Sönströd[1], Ulf Norinder[2],
Henrik Boström[3], and Tuve Löfström[1]

[1] CSL@BS Research Group, School of Business and Informatics,
University of Borås, Sweden
{ulf.johansson,cecilia.sonstrod,tuve.lofstrom}@hb.se
[2] AstraZeneca R&D Södertälje, Sweden
ulf.norinder@astrazeneca.com
[3] Department of Computer and Systems Sciences
Stockholm University, Sweden
henrik.bostrom@dsv.su.se

Abstract. This paper investigates different ways of combining feature selection with bagging and rule extraction in predictive modeling. Experiments on a large number of data sets from the medicinal chemistry domain, using standard algorithms implemented in the Weka data mining workbench, show that feature selection can lead to significantly improved predictive performance. When combining feature selection with bagging, employing the feature selection on each bootstrap obtains the best result. When using decision trees for rule extraction, the effect of feature selection can actually be detrimental, unless the transductive approach *oracle coaching* is also used. However, employing oracle coaching will lead to significantly improved performance, and the best results are obtained when performing feature selection before training the opaque model. The overall conclusion is that it can make a substantial difference for the predictive performance exactly how feature selection is used in conjunction with other techniques.

Keywords: Feature Selection, Bagging, Rule Extraction.

1 Introduction

When performing predictive modeling, feature selection is used for two main reasons. First of all, high dimensionality data sets rule out some techniques, simply because of the computational cost. Second, irrelevant attributes are detrimental for most machine learning algorithms, making feature selection a standard preprocessing technique, often used to increase predictive performance.

* This work was supported by the INFUSIS project (www.his.se/infusis) at the University of Skövde, Sweden, in partnership with the Swedish Knowledge Foundation under grant 2008/0502.

G. Phillips-Wren et al. (Eds.): Advances in Intel. Decision Technologies, SIST 4, pp. 413–422.
springerlink.com

For some machine learning techniques, typically where several models are built either in sequence or in parallel, the feature selection could, however, be applied at different stages. With this in mind, it is interesting to investigate, *if*, *when* and *how* feature selection should be applied to the two common data mining techniques *bagging* and *rule extraction*.

The simplest way to select a subset of features is to use variable ranking on the input features, typically based on correlation with the target attribute, error rate or some information theoretic measure. However, since variable ranking methods do not consider dependencies between features, they are unable to find features that are of limited or no use by themselves, but useful together with other features. Some good examples of this phenomenon are presented in [8]. A more sophisticated, but more computationally intensive, approach is to search for subsets containing features that work well together. Exhaustive search is obviously infeasible for all but very small feature sets, so some search strategy capable of finding optimal, or near optimal, feature sets with reasonable computational effort is needed. Typical search strategies include forward selection, backward elimination, bidirectional search, best-first and beam search.

Feature selection methods are generally divided into three main approaches: wrapper, filter and embedded methods. *Wrapper* methods [11] take into account the specific machine learning algorithm to be applied for prediction and use this algorithm to evaluate the performance of candidate subsets. The machine learning algorithm is treated as a black box by the feature selection process, making this approach extremely versatile since, in principle, any feature subset selection method can be combined with any machine learning algorithm. The drawback is having to build a complete predictive model to evaluate each candidate subset considered during the feature selection process, which is not always computationally feasible. In contrast, the main idea in *filter* methods is to disconnect the feature selection process from the machine learning algorithm used for the actual prediction, which means that a subset of features is selected without evaluating the performance of that subset on the algorithm later being used for classification. Variable ranking is thus a filter method. Some of the advantages of filter methods for subset selection mentioned in [8] are that they are faster than wrapper methods and that they yield feature sets which are useful for prediction in some general sense, rather than being tailored to a particular algorithm or learning scheme. In *embedded* methods, finally, the actual feature selection is performed by the machine learning algorithm during model construction. The typical example here is a decision tree algorithm choosing different split attributes. It should, however, be noted that such methods may still benefit from first applying some of the other techniques.

Ensembles is a standard technique for increasing performance in predictive modeling. The core idea behind ensembles is that the combined predictions from a collection of base classifiers will have better performance than a single classifier. For this to work, the base classifiers need to be individually accurate but also diverse, which means that they commit their errors on different instances, preferably independently of each other [12]. One way of producing diversity in an ensemble is to train each base classifier on only a subset of the instances. In the standard technique *bagging* [2], each subset, called a bootstrap or bag, is of the same size as the original data set and is created by drawing from a uniform distribution with replacement, resulting in approximately 63% of all instances being present in each base classifier's particular training set.

Rule extraction is the process of generating a transparent model based on a corresponding opaque predictive model. Rule extraction has been heavily investigated in the neural network domain, and the techniques have been applied mainly to ANN models; for an introduction and a good survey of traditional methods, see [1]. When performing black-box rule extraction, the core idea is to view rule extraction as a learning task, where the target concept is the function originally learnt by the opaque model. One typical and well-known black-box rule extraction algorithm is TREPAN [5]. Naturally, the predictions from extracted models must be close to predictions from the opaque models. This criterion, called fidelity, is therefore a key part of the optimization function in most rule extracting algorithms. Most, if not all, rule extraction algorithms targeting fidelity use 0/1 fidelity, i.e., maximize the number of identical classifications. One motivation for that rule extraction may produce more accurate models than models induced directly from the data set is that a highly accurate opaque model often is a more useful representation of the data than the data set itself; i.e., the opaque model will act as a filter and smoothen irregularities caused by a few atypical instances.

But, the opaque model could also be used to generate predictions for novel instances with unknown target values, as they become available. Naturally, these newly labeled instances could then be used by the rule extraction algorithm. Despite this, all rule extraction algorithms that we are aware of use only training data (possibly with the addition of artificially generated instances) when extracting the transparent model. We have previously argued that it could be advantageous for a data miner to use test data input vectors together with predictions from the opaque model when performing rule extraction [10]. In this situation, the highly accurate opaque model, called the *oracle* since the target values it produces are treated as ground truth by the training regime of the transparent model, will coach the weaker transparent model. More specifically, the transparent model is built using a combination of standard, labeled, training data and oracle data; i.e., test data inputs together with predictions from the opaque model.

Using a coaching oracle is somewhat similar to semi-supervised self training, but there are two main differences: First we use one (stronger) classifier to label the instances, and another (weaker but transparent) classifier for the final model. Second, since the purpose is increased accuracy on a specific test set, we explicitly utilize the fact that we have the corresponding test input vectors available. Obviously, transductive learning also utilizes both labeled and unlabeled data, and the overall purpose is similar to oracle coaching; i.e., to obtain high accuracy on specific (test) instances; see e.g. [9]. But the main difference is that we explicitly focus on situations where the final model must be transparent, leading to the process described above where a stronger opaque model coaches a weaker transparent model.

Finally, it should be noted that since the test instances labeled by the oracle and then used for inducing the transparent model are the very same instances that later will be used for the actual prediction, the use of oracle data requires a sufficiently sized test set. This method is thus suitable in the very common situation where the predictive model is built for a specific situation, and the input vectors for the actual test data are available already when inducing the model. Clearly, this description matches the situation targeted in this paper where a computational chemist wants to predict biological activity for a specific set of molecules (compounds).

The main purpose of this paper is to investigate how feature selection can be used together with the two standard techniques bagging and rule extraction (possibly also utilizing oracle data in the rule extraction process), to increase predictive accuracy for opaque and transparent models, respectively.

2 Method

The data sets used are from the domain of medicinal chemistry and consist of 8 different sets of compounds, from the study of Bruce et al. [4], originally used by Sutherland [15]. In the study by Bruce et al., the two attribute sets *2.5D* and *Frags.* were used; here a further 4 attribute sets are available, thus bringing the total number of data sets to 48 (8 sets of compounds, with 6 different attribute sets). Of the six different attribute sets, three describe physical-chemical properties of the compounds and the other three molecular fingerprints. The physical-chemical attributes sets are *2.5D*, *oeSelma*, and *AZ Desc.*, while the fingerprint attribute sets are *Frags.*, *sign12*, and *ecfi 1024*. The characteristics for each combination of compound set and attribute set are summarized in Table 1 below, where *Inst.* means number of instances (compounds) in each data set.

The motivation for the use of these data sets is that they represent data mining tasks in which, on the one hand, high predictive accuracy is essential and, on the other hand, a comprehensible model is sometimes needed, since relationships found are of interest to domain experts and can also be used to guide further search for promising molecules. All data sets concern biological activity for inhibitor compounds. The continuous numerical values for activity (pIC50 for the first five data sets and pKi for the last three) in the study by Sutherland et al. were transformed by Bruce et al. into two categorical classes (active and inactive), using the median activity value as a threshold between the two classes to create a 50/50 split of active/inactive observations, since each data set showed a uniform distribution of activity values.

Table 1. Data set characteristics

			Number of attributes					
Name	Meaning	Inst.	2.5D	oeSelma	AZ Desc.	Frags.	sign12	ecfi 1024
ACE	angiotensin converting enzyme	114	56	93	196	1024	1024	332
AchE	acetylcholinesterase	111	63	93	196	774	1024	211
BZR	benzodiazepine receptor ligands	163	75	93	196	832	1024	450
COX2	cyclooxygenase-2	332	74	93	196	660	1024	573
DHFR	dihydrofolate reductase	397	70	93	196	951	1024	487
GPB	glycogen phosphorylase b	66	70	93	196	692	1024	239
THER	thermolysin	76	64	93	196	575	1024	251
THR	thrombin	88	66	93	196	527	1024	220

2.1 Experiments

For simplicity, and to allow easy replication of the experiments, the Weka [16] data mining workbench was used for the two experiments. In the first experiment, feature selection is combined with bagging and in the second experiment, the use of feature selection together with rule extraction is investigated. The choice of techniques evaluated is motivated by the domain's need for both high predictive accuracy and comprehensible models to explain relationships. It was deemed that the only feasible approach was to use a filter feature selection method, both considering computational cost and the fact that different types of predictive models would be generated. In Weka, one filter method is implemented as the *AttributeSelectedClassifier*. To evaluate candidate feature sets, the *AttributeSelectedClassifier* used *CfsSubsetEval*, which favors subsets of features with low intercorrelation, but high correlation with the target variable. The search strategy employed was best-first forward selection.

In the second experiment, an ensemble of bagged RBF networks was used as the opaque model. The transparent model was generated by using rule extraction from these ensembles to obtain decision trees. To enable evaluation of the effect of using feature selection, the ensemble models also have to be built without feature selection, i.e., for data sets with up to 1024 attributes. Using RBF networks as base classifiers was judged to be a reasonable compromise between performance and computational cost for this kind of data set. For all inbuilt Weka algorithms, the settings were left at the default values. Finally, since the data sets contain relatively few instances, all experiments were conducted using 10x4-fold cross-validation. Experiment 1 used the following three setups:

1) standard bagging in Weka of 30 RBF networks (Bag)
2) feature selection employed as a filter for bagging 30 RBF networks, i.e., in Weka terms, an *AttributeSelectedClassifier* using bagging as classifier, (FS-Bag)
3) feature selection employed as a filter for each bag, i.e., bagging 30 *AttributeSelectedClassifier* RBF networks (Bag-FS)

For the rule extraction experiment, we decided to use the readily available J48, which is the Weka implementation of the standard decision tree algorithm C4.5 [14] as rule extractor, instead of a specialized rule extraction algorithm. Naturally, feature selection can be performed either *before* the opaque model is trained (the rule extractor uses the same reduced feature set as the opaque model) or *after* the opaque model is trained, i.e., the feature set is reduced based on the predictions from the opaque model. It must be noted that both these feature reductions are based on training data only. If, however, oracle data is used, the feature selection algorithm can also utilize the test set instances together with the predictions from the oracle. More specifically, the setups utilizing oracle data used the oracle (here an ensemble consisting of 30 bagged RBFs) to label the test data, and then used this test data together with the standard training data for **both** the feature selection and the actual tree induction. In Experiment 2, a total of eight setups were compared, three of which used oracle data. The setups were:

1) standard J48 decision trees induced directly from the data set; this represents a baseline comparison against a simple transparent model (J48)
2) feature selection employed as a filter for J48 decision trees, i.e., an *AttributeSelectedClassifier* using J48 as the classifier (FS-J48)
3) J48 trees obtained using rule extraction from 30 bagged RBF networks (Ens-J48)
4) feature selection employed as a filter for rule extraction, i.e., an *AttributeSelectedClassifier* using rule extraction from a 30-bag RBF ensemble as classifier (FS-Ens-J48)
5) feature selection employed for each bag before rule extraction, i.e., bagging 30 *AttributeSelectedClassifier* RBF networks and then using rule extraction (Ens-FS-J48)
6) same setup as 3) above, but also utilizing oracle data during the rule extraction (EnsO-J48)
7) same setup as 4) above, but also utilizing oracle data during the rule extraction (FS-EnsO-J48)
8) same setup as 5) above, but also utilizing oracle data during feature selection and rule extraction (EnsO-FS-J48)

3 Results

Due to the large number of data sets, most results have to be reported aggregated over the different attribute sets, treating the physical-chemical attribute sets as one group and the fingerprint attribute sets as another group. Predictive performance is evaluated as accuracy and AUC, where the reported numbers for each data set is, of course, averaged over the 10x4 folds.

Table 2 below shows a summary of all accuracy results for Experiment 1, aggregated over the different attribute sets. As can be seen from the table, the setup employing feature selection separately for each bag clearly obtains the best performance, whereas feature selection as a filter for the whole bagging process is actually slightly worse than not using feature selection at all. To determine if there are any statistically significant differences, we use the statistical tests recommended by Demšar [6] for comparing several classifiers over a number of data sets, i.e., a Friedman test [7], followed by a

Table 2. Experiment 1 - Accuracy

	Bag		FS-Bag		Bag-FS	
	Avg. acc	Avg. rank	Avg. acc	Avg. rank	Avg. acc	Avg. rank
2.5D	.741	2.00	.730	2.50	.746	1.50
oeSelma	.741	2.75	.748	2.13	.766	1.13
AZ Desc	.751	2.13	.750	2.50	.768	1.38
Mean Phys-Chem	**.745**	**2.29**	**.743**	**2.38**	**.760**	**1.33**
Frags	.721	2.13	.709	2.50	.730	1.38
sign12	.730	1.88	.710	2.50	.735	1.63
ecfi	.700	2.63	.702	2.25	.725	1.13
Mean Fingerprint	**.717**	**2.21**	**.707**	**2.42**	**.730**	**1.38**

Nemenyi post-hoc test [13]. With three classifiers and 24 data sets, the critical distance (for $\alpha = 0.05$) is 0.68, so based on these tests, all differences between Bag-FS and the other two techniques are significant for both groups of attribute sets. These significant differences between the three techniques also hold for AUC. The result of these tests should, however, be treated with some care in this study, as it is not obvious that sets of compounds represented by different feature sets can be considered to be independently selected data sets. Hence, the statistical tests employed here should rather be seen as approximate tests.

Table 3 below shows the detailed accuracy and AUC results for the oeSelma attribute set. This table illustrates that the superior overall performance of Bag-FS is due to consistently obtaining slightly higher accuracy and AUC for almost every data set.

Table 3. Experiment 1 - oeSelma attribute set

	Accuracy			AUC		
	Bag	FS-Bag	Bag-FS	Bag	FS-Bag	Bag-FS
ACE	.858	.868	.867	0.92	0.93	0.94
AchE	.698	.708	.730	0.76	0.76	0.78
BZR	.712	.718	.747	0.80	0.81	0.84
COX2	.758	.769	.776	0.84	0.83	0.84
DHFR	.799	.801	.806	0.88	0.88	0.89
GPB	.714	.756	.769	0.80	0.85	0.87
THER	.692	.688	.724	0.77	0.75	0.79
THR	.700	.676	.708	0.75	0.73	0.79
Mean	**.741**	**.748**	**.766**	**0.82**	**0.82**	**0.84**
Avg Rank	**2.75**	**2.13**	**1.13**	**2.25**	**2.38**	**1.00**

Summarizing Experiment 1, it is clear that when using feature reduction together with bagging on these medicinal chemistry data sets, the only reasonable choice is to use a separate feature selection for each bag, which unfortunately is also the most computationally costly procedure. However, this extra cost is clearly justified given the significant improvement in predictive performance obtained by the ensemble. Turning to Experiment 2, Table 4 below shows a summary of the accuracy results.

The most important observation is that the use of oracle coaching clearly led to more accurate transparent models. Especially the setup using feature selection before building the oracle model (FS-EnsO-J48) consistently produced very accurate models. As a matter of fact, using the statistical tests described above, the critical distance for eight setups and 24 data sets is 2.14, so this setup obtained significantly higher accuracies than all setups not utilizing oracle data on the physical-chemical data sets. Looking at fingerprint data sets, the picture is almost identical since the FS-EnsO-J48 setup again obtained the lowest mean rank, even if the difference in accuracy when compared to standard J48 is not statistically significant using this test. It could be noted, however, that any pairwise test (e.g. a standard sign test) would result in a significant difference.

Another interesting observation is that if no oracle data is used, the best option is, for most feature sets, actually to use standard tree induction directly on the data; i.e., no feature selection and no rule extraction. This is of course a very good result for J48, showing that the algorithm is well capable of finding and utilizing the most important attributes, without the help of explicit feature selection. It should be noted that this result does not necessarily reflect poorly on rule extraction in general. As a matter of fact, more specialized rule extraction algorithms, explicitly maximizing fidelity, have often been shown to outperform decision tree induction. When oracle data is used, it clearly becomes better to perform feature selection, preferably before building the oracle. So, summarizing the accuracy results, the use of oracle data when building comprehensible models improved accuracy significantly, and the best results overall were obtained by performing feature selection before the oracle was trained.

Looking at the mean AUC results in Table 5 below, the first impression is probably that most setups result in similar AUCs. This is, however, somewhat deceptive since

Table 4. Experiment 2 - Accuracy results

		J48	FS-J48	Ens-J48	FS-Ens-J48	Ens-FS-J48	EnsO-J48	FS-EnsO-J48	EnsO-FS-J48
2.5D	Mean Acc.	.704	.698	.706	.699	.696	.742	.749	.745
	Avg Rank	**5.75**	**5.75**	**5.13**	**6.25**	**5.75**	**2.88**	**1.75**	**2.63**
oeSelma	Mean Acc.	.745	.745	.730	.736	.732	.750	.764	.753
	Avg Rank	**4.63**	**4.88**	**6.00**	**5.13**	**6.00**	**3.50**	**2.25**	**3.50**
AZ Desc.	Mean Acc.	.732	.736	.732	.736	.734	.764	.764	.757
	Avg Rank	**5.50**	**4.25**	**5.50**	**5.00**	**5.50**	**4.38**	**2.50**	**2.63**
Phys-Chem	Mean Acc.	**.727**	**.726**	**.723**	**.724**	**.721**	**.747**	**.759**	**.752**
Phys-Chem	Mean Rank	**5.29**	**4.96**	**5.54**	**5.46**	**5.75**	**3.58**	**2.17**	**2.92**
Frags.	Mean Acc.	.723	.703	.708	.697	.682	.738	.727	.731
	Avg Rank	**3.75**	**4.88**	**5.00**	**5.75**	**6.88**	**3.00**	**3.38**	**3.13**
sign12	Mean Acc.	.728	.712	.722	.707	.707	.739	.740	.740
	Avg Rank	**3.75**	**5.75**	**4.88**	**6.38**	**6.38**	**3.13**	**2.50**	**3.00**
ecfi	Mean Acc.	.709	.709	.704	.695	.701	.702	.733	.716
	Avg Rank	**5.00**	**4.00**	**4.75**	**6.00**	**5.13**	**5.50**	**1.88**	**3.75**
Fingerprint	Mean Acc.	**.720**	**.708**	**.711**	**.700**	**.696**	**.726**	**.733**	**.729**
Fingerprint	Mean Rank	**4.17**	**4.88**	**4.88**	**6.04**	**6.13**	**3.88**	**2.58**	**3.29**

Table 5. Experiment 2 - AUC results

	J48	FS-J48	Ens-J48	FS-Ens-J48	Ens-FS-J48	EnsO-J48	FS-EnsO-J48	EnsO-FS-J48
Phys-Chem Mean AUC	**0.73**	**0.74**	**0.73**	**0.73**	**0.73**	**0.76**	**0.76**	**0.76**
Phys-Chem Mean Rank	**4.71**	**4.42**	**5.21**	**5.08**	**5.04**	**3.42**	**2.83**	**2.71**
Fingerprint Mean AUC	**0.74**	**0.72**	**0.73**	**0.71**	**0.71**	**0.74**	**0.75**	**0.75**
Fingerprint Mean Rank	**3.46**	**4.79**	**4.25**	**6.08**	**5.63**	**3.71**	**2.38**	**2.75**

the mean ranks show that the setups utilizing oracle coaching actually outperformed all other setups, on a large majority of the data sets, also with regard to AUC.

4 Conclusions

We have in this paper investigated when and how feature selection should be applied to two common data mining techniques; bagging and rule extraction. When using bagging, feature selection could be used either before the data set is used to draw the bootstraps, or locally on each bootstrap. The experimental results clearly show that the best option is to perform the feature selection on each bootstrap. As a matter of fact, that setup obtained significantly higher accuracy and AUC compared to both feature selection before the bootstrapping and not using feature selection at all. In addition to being a straightforward recipe, it is also an interesting theoretical finding, since it indicates that for data sets like the ones studied here (i.e. fairly high dimensionality with at least some redundancy) the increased diversity obtained by performing the feature selection independently for each bag turned out to improve ensemble accuracy. This hence has a similar effect as random feature selection in random forests [3].

In the rule extraction experiment, the somewhat surprising result was that unless oracle data is used, the best option is actually to skip both feature selection and rule extraction and just perform standard J48 rule induction directly on the data set. However, the results also show that utilizing oracle coaching led to a significant improvement in both accuracy and AUC, with a relatively low extra computational cost. Specifically, the best setup evaluated first performed feature selection (using training data only) and then used the reduced feature set to build the opaque (oracle) model. The oracle was then used to label the test data instances, which together with the original training instances (with reduced features) were used as training instances when inducing the final J48 model. The most important observation from this experiment is that the use of oracle coaching affects the performance much more than if and how the feature selection is applied. Having said that, if oracle data is used, both setups that contain feature selection clearly outperformed the setup using unreduced feature sets, with regard to both accuracy and AUC.

References

1. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl.-Based Syst. 8(6), 373–389 (1995)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
4. Bruce, C.L., Melville, J.L., Pickett, S.D., Hirst, J.D.: Contemporary qsar classifiers compared. J. Chem. Inf. Model. 47(1), 219–227 (2007)
5. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in Neural Information Processing Systems, pp. 24–30. MIT Press, Cambridge (1996)

6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
7. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of American Statistical Association 32, 675–701 (1937)
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
9. Joachims, T.: Transductive inference for text classification using support vector machines. In: International Conference on Machine Learning (ICML), pp. 200–209. Bled, Slowenien (1999)
10. Johansson, U., Niklasson, L.: Evolving decision trees using oracle guides. In: CIDM, pp. 238–244. IEEE, Los Alamitos (2009)
11. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: International Conference on Machine Learning, pp. 121–129 (1994)
12. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. Advances in Neural Information Processing Systems 2, 231–238 (1995)
13. Nemenyi, P.B.: Distribution-free multiple comparisons. PhD-thesis. Princeton University (1963)
14. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
15. Sutherland, J.J., O'Brien, L.A., Weaver, D.F.: A comparison of methods for modeling quantitative structure-activity relationships. J. Med. Chem. 47(22), 5541–5554 (2004)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)