Project AI: Domain Based Classifier Selection

Lena Bayeva 6106609
Luís Brandão da Silva 6110169

February 1, 2010

Abstract

In this paper we explore statistical properties of the data belonging to different domains and show how they can be used to provide a practical guidance for finding the best classification method. We investigate what properties of the individual domains can help us generalize about the data belonging to the same or similar domains. We explore the following properties of the data: clusterability (smoothness), class smoothness, linear separability, uncertainty (presence of noise), and feature relevance. We show that these properties are sufficient for distinguishing between artificial and medical data sets and for determining the types of classifiers possibly suitable for these domains. We indicate that additional experiments are needed to make the discussed hypothesis more concrete.
Keywords: domain properties, clusterability, smoothness, classifier selection.

1 Introduction

Understanding what type of classifier model is appropriate for a particular data set is crucial for the success of a learning task. Much research has been conducted to provide estimation of the best classifier for a given data. Among the methods are data visualization, successive application and comparison of classifier performance (trial and error), different forms of meta-learning, i.e. discovering meta-knowledge (induction of knowledge about learning methods performance), stacked generalization, dynamic bias selection, etc. Relating statistical properties of the data to data models and classification algorithms has been explored in a number of studies [3], but few have exploited classification of data domains. Yet, it is common knowledge that some classifiers tend to be more appropriate for specific domains. For example, for *text* classification *Naive Bayes* and *SVM* perform better than *Decision Trees*, for *medical* data *Logistic Regression* does well, and so on.

Our hypothesis is that statistical properties of the data can be associated with the domain. For example, we observe the differences in data sets belonging to *natural* (data gathered from the real world) and *artificial* (generated data like games and functions) domains. Raw natural data are often rich (have many variables and observations), uncertain (have various degrees of noise), clustered (have dense regions), and are best modeled by relatively simple classification models. Data about artificial phenomena

tend to be less noisy, less clustered, and tend to need more complex models. We hope that such domain properties can be captured, measured, evaluated, and used to support decisions that today are made based on common knowledge.

Discovering properties of the domains is a novel approach for selecting and configuring classifiers. For example, by knowing the properties of the artificial domain, we can expect more complex models to be appropriate, in which case we can prune and regularize less. In case of other domains (i.e. medical, text, etc) we expect simpler models to be more general, since unpredicted data are more likely to reflect a real pattern, and the presence of noise is less influential.

Thus we propose and explore domain properties like clusterability, class smoothness, linearity, uncertainty, and feature relevance and point out their potential application for classifier selection. In section two we describe data set properties that we consider to be relevant for domain-based classifier selection. In section three we describe our approach to verification of the hypothesis by application of different classifiers to data sets from two domains: *medical* and *artificial*. In section four a brief overview of the data sets from the two domains is presented. Finally, results, discussion, and conclusions are presented in sections five, six, and seven respectively.

2 Domain Properties

Our hypothesis is that certain statistical properties of the data belonging to different domains can be used to provide a practical guidance for finding the best classification method. In this section we propose properties of the individual domains that can help us generalize about the data belonging to the same or similar domains.

We start by observing that different domains vary in the complexity of the underlying pattern, the proportion of relevant variables, the amount of uncertainty (noise), completeness (i.e. presence/absence of missing information), and so on. Based on these observations we consider the following properties of the data:

- ***Smoothness (or Clusterability)*** One way to depict the structure of the data is by observing that some regions are more dense than others, in other words some instances are closer together and some are farther apart. These dense areas (clusters) can be discovered by the unsupervised clustering methods. We consider data to be *smooth* if it's well *clusterable* - the quality of the clusters is good.

  Clusterability can be measured by first applying unsupervised clustering to the data, and then evaluating cluster quality. The first step involves finding clusters by optimizing some distance measure, commonly Euclidean or Hamming distance, which determines the similarity of two data points and the overall shape of the clusters. The second step is to measure cluster quality. There exist many measures for this (i.e. Davies-Bouldin Validity Index, Silhouette Validation Method, Jaccard Index, and so on). We use Davies-Bouldin Validity Index since it measures how compact the clusters are and how well they are separated.

- ***Class Smoothness*** In addition to unsupervised clusterability, we often want to predict the class or category to which instances belong to by means of supervised classification.

  Class Smoothness of a distribution, can be defined as a property of the data, such that if instances are close together in some attributes, they are also close together in other attributes.

  In the context of supervised classification this means that instances that are similar or close together (i.e. belong to the same cluster) are more likely to have the same class than the instances that are farther apart (i.e. belong to different clusters).

  Class smoothness can be estimated by a combination two measures: the *number of clusters* and the *class error*.

  Class error is determined by finding the dominant class in each cluster and considering instance count belonging to another class as an error. The total class error is an average over all clusters:

$$classError = \frac{\sum_{Clusters} clusterError}{dataSetSize}$$

$$clusterError = \#ClusterPoints - \#DominantClassPoints$$

  We presume that some domains are more smooth then others and evaluating smoothness can thus help us determine a learning bias. For example, in a smooth domain *1-nearest-neighbour* is likely to give good results because its bias aligns with the domain.

- *Linear separability* Linear separability is another informative property of the domain. If the data points are linearly separable - can be separated by a hyperplane - then a linear classifier, i.e. Linear Regression or Perceptron, is appropriate. Measuring this property requires running one of the linear classification algorithms. However, it provides additional isights into the structure of the data and values of other properties (i.e. clusterability).

- *Uncertainty (presence of noise)* Knowing how noisy the data is can help identify and configure an appropriate learner. A learner that is robust to noise corresponds to a simpler, more general model, that is less sensitive to outliers.

  For example, linear models like *Logistic Regression* tend to be robust to uncertainty, and are frequently applied for classification of medical domains which tend to be quite noisy. Different techniques can be applied to make existing classifiers robust to noise. For example, to avoid overfitting decision trees can be pruned, early stopping techniques can be applied for perceptron training, weight decay for neural networks, or ridge regression for regression.

  Automatic estimation of uncertainty is not trivial. The best estimate can be obtained from the error of the best classifier given that a solid evaluation procedure

is used. Sometimes the amount of noise is known aprior, i.e. specified by an expert. For example, *artificial* domain typically contains no noise or a small amount of noise (often introduced on purpose).

- *Feature relevance* Some features are more relevant for classification than others. For example, not all features in the *medical* domain are relevant, while almost all features in the *artificial* domain are important (i.e. all attributes are needed for as successful outcome of a game). This is in part due to the fact that artificial domains are modeled knowing apriori which features are relevant, while in medical domains utility of the features is not always explicitly known.

  Feature relevance can be spotted with the help of Decision Trees, where some attributes are more discriminative with respect to the class, while others are less so or not necessary at all. For that reason, pruned trees generalize better, as unimportant and noisy attributes are eliminated.

In the next sections we attempt to show that the properties discussed above are sufficient for distinguishing between *artificial* and *medical* data sets and determining the types of classifiers possibly suitable for these domains. We do that by applying classifiers that construct distinctly different models for the data sets from the two domains and relating the learner performance with data properties.

3 Approach

We perform a series of experiments on the data sets from the two domains *artificial* and *medical* using several learning algorithms described below. We run the algorithms using WEKA tool (version 3) [5] with default settings and 10-fold cross-validation enabled. We have selected classification algorithms that construct different classes of models and exhibit different generalization performance on unseen data depending on the simple bias - a prior probability over hypotheses.

For example, if the data is linearly separable a simple *Linear Regression* or a *one-layer Perceptron* model will be sufficient. A *Decision Tree* model can be more complex for the same data if it is distributed along the diagonal, for example. In this case vertical and horizontal split of a *Decision Tree* doesn't capture the structure of the data well, and thus doesn't generalize well on new examples, see Figure 1. A *Linear Discriminant* will do much better in this case.

If the data is less separable (due to presence of noise or the underlying non-linear structure) non-linear learners like *Logistic Regression* and *multi-layer Perceptron* make for better models. Simple classifiers like *Decision Stump* and *1-Nearest-Neighbor (1NN)* can be used to verify a better performance on "simpler", more smooth data.

A brief overview of the selected algorithms:

Linear Regression - A learner provides a linear approximation of the target function and intends to measure closeness to simple linear separation. [2]

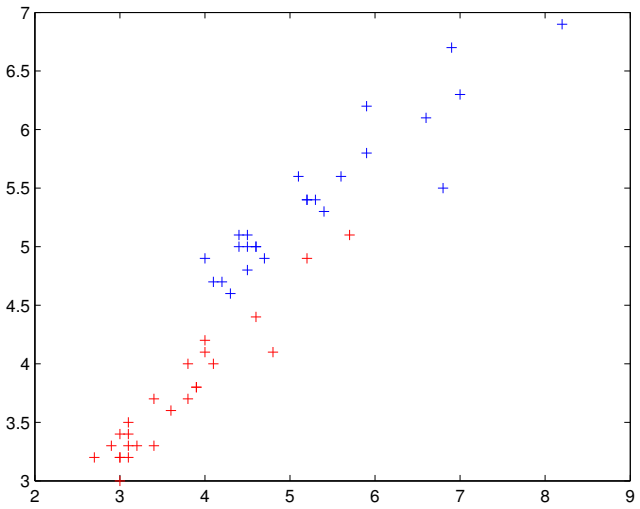

Figure 1: Diagonal Distribution of data

Simple Vote Perceptron - Similar to linear regression it constructs a linear model for a given feature vector. It determines parameters by minimizing an error function on a training set.

Logistic Regression - A slightly less linear model used for binomial regression.

Multilayer Perceptron - A non-linear model that finds parameters (weights) that allow it to depict and separate non-linear patterns in data.

Decision Tree (Pruned and Unpruned) - Decision tree model is constructed using hierarchy of branches, where the root node is the most discriminating attribute (based on some measure, i.e. information gain), and the leaves are final predictions (i.e. class values). Unpruned decision trees tend to overfit for noisy data, but at the same time can capture more complex data. Pruned decision trees eliminate unnecessary branches that cause overfitting and thus are more robust on noisy domains.

Decision Stump - A single decision node (based on one attribute) is used to classify test examples. This learner also aims to establish closeness to linear separability.

1-Nearest Neighbor - The test set is classified based on the classification of the closest training example and measures the closeness of instances belonging to the same class. [2]

Along with algorithm performance, we have measured cluster properties that can help us determine data smoothness. We ran unsupervised EM (Expectation Maximization)

algorithm on the data sets and recorded the *number of clusters* and the *class error*. Given that all our data sets are binary, a large number of clusters can indicate more complex and less clusterable data. Large class error is another sign of poor cluster quality.

For the experiments we use datasets described in more details in the section below. We have chosen to compare two domains *medical* and *artificial* as we consider them to have distinctly different properties. These domains differ in smoothness - we expect *medical* domain to be more "natural", more smooth, hence clusterability of this domain should be better. *Artificial* domain is less smooth and thus less clusterable.

The domains differ in the nature of uncertainty - medical data is more noisy compared to artificial. The linear separability of the data is also different: we expect medical domain to be more linear; at the same time linear separability is complicated by noise. Artificial data is less noisy, but by nature less linearly separable. Clusterability is dependent upon linearity, since more linearly separable data tends to form distinct clusters.

In medical domain we expect a certain amount of irrelevant variables, while in artificial data sets almost all attributes should be relevant, i.e. all attributes defining a game are needed for correct classification. Thus, the structure of the medical domain is expected to be less complex compared to the artificial and better depicted by simpler classification models.

Given our assumptions about properties of the two domains: *medical* - more smooth in class values, linearly separable, noisy, with some irrelevant features, and *artificial* - less smooth in class values, less linearly separable (more complex), less noisy, with few or no irrelevant features, we expect the following results:

- Medical domain should have fewer clusters with smaller class error compared to artificial domain - indication of smoothness.
- *Logistic* and *Perceptron* algorithms should perform better than *Decision Trees* for medical domain. This is because the data is more linear and more noisy, and *Decision Trees* tend to generalize less well compared to the other two algorithms.
- *Decision Stump* algorithm should perform better on medical data, since individual attributes should be better separated for medical domain.
- *Decision Trees*, on the other hand, should perform better on artificial domain as they are capable of capturing a complex structure and are less robust to noise.
- *1NN* should be more linear for linearly separable data and less linear for more complex or noisy data. Medical domain is noisy and artificial domain is more complex. We therefore, expect comparable results for the two domains.

We can also make predictions about what classifier will perform better on each of the domains given their properties. We expect linear classifiers, i.e. Linear Regression and Voted Perceptron, to build similar models that should do well on medical domain. Logistic Regression and Multilayer Perceptron are less linear and should perform better

on artificial domain. Unpruned Decision Tree is more sensitive to noise and will generalize poorly on the medical data sets, while performance of the pruned Decision Tree should improve. For artificial domain no major difference between pruned and unpruned Decision Trees should be observed as it is less noisy.

4 Data sets

The datasets selected for the experiment have been obtained from the UCI repository [1]. Given that the experiments are conducted using the WEKA tool we use the datasets available in WEKA format at the WEKA dataset collection website [4].

Table 1 summarizes the properties of the selected data sets. All of them are binary and the dimensionality of the *kr-vs-kp* and *poker* was reduced to enable the experiments in WEKA (convergence of certain algorithms). The poker data set was additionally transformed to have binary class values - the original version included 9 classes. This transformation was required to run *linear regression* and *voted perceptron* algorithms over this dataset.

Table 1: Properties of the data sets

DataSet	Type	Instances	Attributes	Discrete att.	Continuous att.	Missing values (%)
arrhythmia	Medical	452	279	73	206	21
breast-cancer	Medical	286	9	9	0	0.3
colic	Medical	368	22	15	7	30
diabetes	Medical	768	8	0	8	0
heart-staglog	Medical	270	13	0	13	0
hepatitis	Medical	155	19	19	0	5.59
liver-disorders	Medical	345	6	6	0	0
lymph	Medical	142	18	15	3	0
balance	Artificial	576	4	4	0	0
kr-vs-kp	Artificial	594	36	36	0	0
poker	Artificial	4002	10	0	10	0
tick-tac-toe	Artificial	958	9	9	0	0
autos	Artificial	205	25	11	14	1.17
bridges.data.version1	Artificial	107	12	8	4	6
led1	Artificial	2005	7	0	7	0
page-blocks	Artificial	5473	10	0	10	0

5 Results

In this section we present experiments illustrating that it is possible to select an appropriate learner for a domain on the basis of the properties of that domain.

Using the data sets described above and the WEKA toolkit we run Linear Regression (LinR), Logistic Regression (LR), Voted Perceptron (VP), Multilayer Perceptron (MP), Unpruned and Pruned Decision Trees (UDT and PDT) and the 1-Nearest-Neighbor (1NN) algorithms. Additionally, we run the unsupervised EM clustering algorithm to get an estimate of the number of clusters for each dataset (EM C.) and measure how the cluster are related with the target class (EM E.). We evaluate the quality of the clusters with the Davies-Bouldin Validity Index (DC).

Table 2 summarizes de results of the experiments by presenting the error rates per algorithm and the number of clusters estimated by the EM algorithm.

Table 2: Error rates for each algorithm

DataSet	LinR	LR	VP	MP	U. DT	P. DT	DS	1NN	EM Error	EM C.	DC
arrhythmia	66.49	16.6	14.53	10.03	34.73	33.19	44.03	47	38	5	3.08
breast-cancer	45.61	31.47	29.37	34.965	34.62	30.77	31.12	34	24	3	1.49
colic.arff	37.53	18.75	38.04	19.02	15.76	14.13	18.48	18.75	26	5	2.22
diabetes	40.37	22.4	33.46	24.87	26.17	26.3	28.52	29.81	15	8	1.54
heart-staglog	36.5	17.04	34.44	20.74	25.56	20.74	26.67	24.81	11	6	1.47
hepatitis.arff	36.85	17.42	21.29	20	21.94	21.94	23.23	19.35	10	3	1.43
liver-disorders	47.28	31.88	35.94	32.17	31.59	31.59	40	37.1	42	4	1.4
lymph	34.76	20.42	21.13	13.3803	24.65	24.65	21.13	18	24	3	1.55
Avg (medical)	43.17	22	28.53	21.9	26.88	25.41	29.15	28.6	23.75	4.63	1.77
Stddev (medical)	10.43	6.27	8.52	8.55	6.54	6.44	8.97	10.36	11.77	1.77	0.59
balance	28.58	6.59	5.9	2.43	11.46	12.67	36.8	9.2	0	4	1.11
kr-vs-kp	25.15	5.56	8.25	1.85	2.02	2.02	18.35	9.25	6	13	1.83
poker	50.1	49.08	50.05	46.85	47.55	47.43	51.8	46.65	50	1	2.35
tick-tac-toe	38.11	1.67	14.3	3.03	14.19	14.72	30.06	18.37	0	20	1.57
autos	42.64	29.27	26.45	10.74	18.54	20.49	55.12	23.9	37	9	1.6
bridges	71.57	40.95	22.8	17.54	36.19	31.42	43.81	40	42	5	1.55
led1	23.83	10.37	10.62	11.37	10.37	10.37	10.02	11.62	10	4	1.06
page-blocks	16.44	3.53	2.96	1.43	3.14	3.01	6.87	4.13	8	8	1
Avg (artificial)	37.05	18.38	17.67	11.91	17.93	17.77	31.6	20.39	19.13	8	1.51
Stddev (artificial)	17.76	18.66	15.38	15.28	16.02	15.26	18.49	15.5	20.38	6.09	0.45

Given the results, we now review our hypothesis about classifier performance using the assumptions about properties of the two domains discussed in section three.

First we note that the assumption that medical data is more linearly separable compared to artificial data was not confirmed with the help of *Linear Regression* and *Voted Perceptron*. The error rate for the medical domain is on average higher than than for the artificial domain, but at the same time the standard deviation is lower. Thus, it is not possible to conclude if there is a difference between the two domains without expanding the experiments to a larger collection of data.

The fact that *Logistic* and *Perceptron* are better than *Decision Trees* for medical domain indicate that the data is more linear and more uncertain, thus and *Decision Trees* fail to generalize, while non-linear models depicted the structure of the data and handled noise more gracefully.

The results for the *Decision Trees* confirmed our expectations: pruned decision trees perform worse on the medical domain, showing overfitting of the model due to noise. We can also confirm this by observing that pruning improves performance of the decision tree on the medical data more so than on artificial. In the case of artificial data, no significant difference is observed between the two decision tree models. This indicates less noise and fewer unnecessary attributes compared to the medical domain.

The *Decision Stump* performs better on the medical data, showing that individual attributes are more separable. This can also indicate that some of the variables aren't as important in medical domain, and separation on more discriminative attributes generalizes better.

1NN showed on average better performance on artificial domain. A possible explanation can be that we did not take into account a stronger presence of noise in natural domains, as well as their sparseness compared to artificial domains. To demonstrate smoothness, therefore, additional experiments are needed that take these parameters into account (i.e. using appropriate weighting techniques, or comparing domains with equal amounts of noise).

Class error and Davies-Bouldin Validity Index are on average higher for medical domain, but standard deviation is much higher for artificial data. This can be explained

by the presence of uncertainty in the medical data on one hand, and smaller cluster sizes that result in a smaller average error for the artificial domain on the other. In the cases where the cluster size is small for the artificial domain, the class error is also large (i.e. poker). A better measure can be proposed to capture these aspects of *class smoothness*.

Finally, we relate the results with our predictions about classifier performance on the two domains. Contrary to our expectations, Linear Regression and Voted Perceptron exhibit on average better performance on *artificial* domain. As expected, Logistic Regression and Multilayer Perceptron learners show better results on *artificial* domain. Pruned Decision Trees perform better for both domains, and as expected for *medical* domain we observe more improvement compared to unpruned Decision Trees.

6 Conclusion and Discussion

In this project we analyze how statistical properties of different domains can be explored to select the most appropriate machine learning methods for each domain. Our analysis provides some evidence of a bias associated with the properties of a domain, but more experiments are needed to support this claim. Some of the experiments we have conducted were in favor of the assumptions we have made about properties of the domains. For example, *Decision Stump*, *Decision Trees*, *Logistic Regression*, *Multilayer Perceptron*, and cluster evaluation methods confirmed our hypothesis about *medical* and *artificial* domains. Others, like *1NN* and *linear regression* gave ambiguous results that require further investigation. This indicates that more experiments are necessary to confirm our assumptions.

We believe this research can be validated and expanded with additional experiments and improvements that we now describe. Firstly, existing data sets can be extended for the two domains to make sure the results are consistent. Secondly, additional domains can be added to the experiments, i.e. text. Both *artificial* and *medical* domains were chosen assuming that their properties are different. The results of our experiments point out the existence of some differences between the two domains namely class smoothness, uncertainty, complexity, etc. The inclusion of a *text* domain can strengthen our results as it has distinct properties from both medical and artificial domains: features defined as a bag-of-words, sparseness of data, presence of noise, and many irrelevant features. Finally, the selected domain properties and their corresponding measures can be revised. For example, linear separability and clusterability of the data can be combined into a single clusterability or smoothness measure. Cluster sizes need to be taken into account with the proposed cluster quality measures. Feature relevance can be evaluated using Pruned Decision Trees - irrelevant attributes are likely to be eliminated. Noise levels can be measured with the help of a classifier - based on the error on test data. In our experiments we didn't quantify the amount of uncertainty or noise in the data, but used prior knowledge (i.e. data set descriptions) as an estimate.

Even though more research is needed to obtain more solid results that support our hypothesis, we can already see the usefulness of this approach from the practical point of view. Given the properties of the domain we can make predictions about what learn-

ers can give better results, i.e. multilayer Perceptron works best on *artificial* data. Conversely, we can potentially predict properties and the domain of the data based on performance of a set of learners on that data. This is another useful result that can be explored. For example, if a non-linear model does much better than a linear model, as can be seen with multilayer Perceptron and Logistic Regression for *artificial* domain, then a model is probably more complex. Comparing performance of pruned and unpruned Decision Trees can give some indication of presence of irrelevant attributes or noise. Thus, properties can be combined to predict a domain.

We believe that this research is a good step in a direction of domain classification approach to the selection of learning methods. It is mainly relevant to the development of machine learning applications and can instigate a shift from the usual paradigm, where an expert makes an algorithm selection based on implicit knowledge, to a more theoretically solid approach.

References

[1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007. URL `http://archive.ics.uci.edu/ml/`. [Online; accessed 31-January-2010].

[2] Hilan Bensusan, Christophe Giraud-Carrier, and Bernhard Pfahringer. What works well tells us what works better. In *Proceedings of ICML'2000 workshop on What Works Well Where*, pages 1–8. ICML'2000, June 2000. URL `http://www.cs.bris.ac.uk/Publications/Papers/1000469.pdf`. [Online; accessed 31-January-2010].

[3] Austrian Research Institute For Artificial Intelligence. Metal publications lists. URL `http://www.ofai.at/research/impml/metal/metal-publications.html`. [Online; accessed 31-January-2010].

[4] Weka Machine Learning Project. Weka - dataset collection, . URL `http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html`. [Online; accessed 31-January-2010].

[5] Weka Machine Learning Project. Weka, . URL `http://www.cs.waikato.ac.nz/ml/weka`. [Online; accessed 31-January-2010].