November 2001

Using boosting to simplify classification models

V. Wheway
University of Wollongong

Using boosting to simplify classification models

Abstract

Ensemble classification techniques such as bagging, boosting and arcing algorithms have been shown to lead to reduced classification errors on unseen cases and seem immune to the problem of overfitting. Several explanations for the reduction in generalisation error have been presented, with recent authors defining and applying diagnostics such as "edge" and "margin". These measures provide insight into the behaviour of ensemble classifiers, but can they be exploited further? In this paper, a four-stage classification procedure in introduced, which is based on an extension of edge and margin analysis. This new procedure allows inverse sub-contexts and difficult border regions to be detected using properties of the edge distribution. It is widely known that ensemble classifiers 'balance' the margin as the number of iterations increases. However, by exploiting this balancing property and flagging observations whose edges (and margins) are not 'balanced', data sets can often be partitioned into sub contexts and the classification can be made more robust as confounding within a data set is removed. In the majority of cases, the sub-contexts detected are inverse to each other or, quite possibly, the smaller sub-context contains mis-labelled observations. The majority of classification techniques have not been adapted to detect contexts within a data set, and the generalisation error reported in studies to date is based on the entire data set and can be improved by partitioning the data set in question. The aim of this study is to move towards interpretability, and it is shown that, by training on a sub-set of the original training data, we gain simplicity of models and reduced generalisation error.

Keywords

data mining, error analysis, artificial intelligence, modelling, pattern classification

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Wheway, V, Using Boosting to Simplify Classification Models, Proceedings 2001 IEEE International Conference on Data Mining, 26 November-2 December 2001, 558-565.

Using Boosting to Simplify Classification Models

Virginia Wheway
School of Mathematics and Applied Statistics,
School of Information Technology and Computer Science
University of Wollongong

NSW 2522 Australia
vlw04@uow.edu.au

Abstract

Ensemble classification techniques such as bagging, boosting *and* arcing *algorithms have been shown to lead to reduced classification error on unseen cases and seem immune to the problem of overfitting. Several explanations for the reduction in generalisation error have been presented, with authors more recently defining and applying diagnostics such as* edge *and* margin *[4,9,10]. These measures provide insight into the behaviour of ensemble classifiers but can they be exploited further?*
In this paper a four stage classification procedure in introduced, which is based on an extension of edge and margin analysis. This new procedure allows inverse sub-contexts and difficult border regions to be detected using properties of the edge distribution. It is widely known that ensemble classifiers 'balance' the margin as the number of iterations increases. However, by exploiting this balancing property and flagging observations whose edges (and margins) are not 'balanced', datasets can often be partitioned into subcontexts and classification made more robust as confounding within a dataset is removed. In the majority of cases, the subcontexts detected are inverse to each other, or quite possibly, the smaller subcontext contains mislabelled observations. The majority of classification techniques have not been adapted to detect contexts within a dataset and the generalisation error reported in studies to date is based on the entire dataset and can be improved by partitioning the dataset in question. The aim of this study is to move towards interpretability, and it is shown that by training on a subset of the original training data we gain simplicity of models and reduced generalisation error.

1 Introduction

The ability to classify unseen observations efficiently is a desirable property of many data mining algorithms. Numerous barriers may be present when an optimal classification model is being sought. In particular unavoidable dataset symptoms such as noisy data, outliers and 'fuzzy' boundaries inhibit a model's performance. Another such inhibitor is the presence of more than one context within a dataset. i.e. different sections of the dataset being classified according to significantly different models. If undetected, this results in increased generalisation error and a more complex model prone to overfitting. The undetected presence of such subcontexts has previously been attributed to noise and deterioration in generalisation error. If such situations can be detected and either removed or accounted for in the final modelling stage, significant gains may be made in both model simplification and generalisation ability.
This paper is concerned with the classifi-

0-7695-1119-8/01 $17.00 © 2001 IEEE

cation problem, whereby a learner is presented with a training set comprising of a series of n labelled training examples of the form $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, with $y_i \in (1, \ldots, k)$. The learner's task is to use these training examples to produce an hypothesis, $h(\mathbf{x})$, which is an estimate of the unknown relationship $y = f(\mathbf{x})$. This 'hypothesis' then allows future prediction of y_i given new input values of $\mathbf{x}$. A classifier built by combining individual $h(\mathbf{x})$'s to form a single classifier is known as an ensemble. Whilst there are many ensemble building methods in existence, this discussion focusses on the method of boosting which is based on a weighted subsampling of the training examples.

Introduced by Freund and Schapire, boosting is recognised as being one of the most significant recent advances in classification [9]. Since its introduction, boosting has been the subject of many theoretical and empirical studies [3,8,10]. Empirical studies have shown that ensembles grown from repeatedly applying a learning algorithm over different subsamples of the data result in improved generalisation error.

This paper presents the results from an empirical study on boosting and edge analysis. Section 2 introduces the theory behind the study, which is tested in Section 3 on a selection of UCI [1] [1] datasets.

2 Boosting, Margin and Edge

Boosting is an iterative procedure which trains a learner over n weighted observations. Boosting begins with all with all training examples being given equal weight (i.e. $\frac{1}{n}$). At the $m+1$-th iteration, examples which were classified incorrectly at the m-th iteration have their weight increased multiplicatively so that the total weight on incorrect observations is equal to 0.5. Hence, the learning algorithm will be given more opportunity to explore areas of the training set which are more difficult to classify. Hypotheses from these parts of the space make fewer mistakes on these areas and play an important role in prediction when all hypotheses are combined via weighted voting. At each iteration, the weighted error is stored and used in the final voting weight when individual classifiers are combined to form the ensemble. Accuracy of the final hypothesis depends on the accuracy of *all* the hypotheses returned at each iteration and the method exploits hypotheses that predict well in more difficult parts of the instance space. An advantage of boosting is that it does not require any background knowledge of the performance of the underlying weak learning algorithm. Table 1 summarises the details of the boosting algorithm and its weight update methodology.

[1] URL = http://www.ics.uci.edu/~mlearn /MLRepository.html

Table 1. AdaBoost:M1 [9]

AdaBoost:M1

Input: n training instances x_i with labels y_i. Maximum trials, M. Base learner, H.

Initialization: All training instances begin with weight $w_i^0 = 1/n$.

Repeat for M trials:

- Induce classifier, h_m, using weighted training data and H.
- ϵ_m = weighted error for h_m on the training data. If $\epsilon_m > 1/2$, discard h_m and stop boosting. (If $\epsilon_m = 0$, then h_m gets infinite weight.)
- Classifier weight, $\beta_m = \log(\frac{1-\epsilon_m}{\epsilon_m})$
- Re-weight training instances:
  if $h_m(x_i) \neq y_i$ then,
  $w_i^{m+1} = w_i^m/(2\epsilon_m)$
  else, $w_i^{m+1} = w_i^m/2(1-\epsilon_m)$

Unseen instances are classified by voting the ensemble of classifiers h_m with weights β_m.

2.1 Margin and Edge

Recent explanations as to the success of boosting algorithms have their foundations in margin and edge analysis [4,10]. Schapire et al. [10] claim that boosting is successful because it creates a higher margin distribution and hence increases the confidence of correct classification.

Breiman, however, claims the high margin explanation is incomplete and introduces new ensemble techniques which actively improve margin distributions but do not result in improved generalisation error [4,5]. These two measures are defined for the ith training observation at trial m as follows:

> The edge is defined for each observation as the total weight assigned to the incorrect class, with the margin being defined as the total weight assigned to the correct class minus the maximal weight assigned to any incorrect class.

More formally, assume the ensemble comprises a combination of base learners, each of which produce $h_m(\mathbf{x})$ at the m-th iteration. From $h_m(\mathbf{x})$, an error indicator function $I(h_m(\mathbf{x})_i) \neq y_i)$ may be determined. Let c_m represent the vote for the m-th hypothesis with $\sum_m c_m = 1$. Then, for the i-th observation:

$$edge(m,i) = \sum_{j=1}^{m} c_j I(h_j(\mathbf{x}_i) \neq y_i) \quad (1)$$

Whilst the margin is a useful measure due to its interpretability, mathematically it is perhaps not as robust and tractable as the edge. In its pursuit of correctly classifying 'harder' sections of the data space, boosting tends to 'balance' the edge. In other words, the proportion of misclassifications for each observation becomes uniform as the number of iterations increases. This was demonstrated in [5,11]. By utilising this property, a method for detecting noise has been introduced in [11] whereby noisy of difficult observations could be detected via deviations in the overall edge distribution.

Instead of detecting single, unique outliers, the notion of deviation from the main edge distribution may be used to detect clusters of observations behaving differently to the rest of the data. By isolating these observations and attempting to determine their collective structure, we may gain insight into areas of the input space which should be segregated or at the least, treated with caution.

Section 3 extends this notion and describes a process for detecting clusters of noise within a dataset.

3 A Four Stage Classification Process

3.1 Stage 1: Detect Obvious Outliers

In the noise study discussed in [11], plots of $edge(10,i)$ versus observation number are drawn as the first step in detecting noise or anomolies in the data. Identified via a significant deviation from a 'balanced' edge distribution, such observations can be examined and if justified, removed. If observations are removed from the original dataset, boosting trials must be re-run as changes in the dataset results in adjustment of the edge values for all other observations.

3.2 Stage 2: Detect Clusters via Edge Diagnostic Plots

At the second stage of this process, obvious outliers have been removed and the edge values for each observation ($edge(m,i)$, $m = 1 \ldots 10$) stored at the completion of each iteration.

The best edge measure for the task of detecting clusters of observations with unique properties is unclear. Measures are sought which will capture and differentiate behaviour deviant from the main 'core' of the data.

Observations which are consistently classified correctly will have a low average edge and a low variance of the edge. Observations which are persistently misclassified will also have a low edge variance but high average edge. Groups of difficult observations which collectively behave in the same manner but differ in structure from the majority of observations will have a high edge variance as boosting will alternate between correct and incorrect classification. Such clusters of observations will fall in a common location when the mean edge is plotted against the variance of the edge.

It is therefore proposed to calculate the mean and variance of the sequence of edge

values for each observation. Depending on the succession of correct or incorrect classifications for each observation, these edge statistics will vary greatly between observations. Observations which can be grouped together will have similar mean and variance of their edge values.
More formally, the edge measures for $m = 10$ boosting trials are calculated as:

- $\hat{E}[edge(10,i)] = \frac{1}{10}\sum_{m=1}^{10} edge(m,i)$
- $\hat{Var}[edge(10,i)]$
  $= \frac{1}{10}\sum_{m=1}^{10}(edge(m,i) - \hat{E}[edge(m,i)])^2$

The next step in the classification process encompasses the plotting of $\hat{Var}[edge(10,i)]$ versus $\hat{E}[edge(10,i)]$. Any clusters or subcontexts within the dataset become apparent when this diagnostic plot is drawn. Signature behaviour is noted on all datasets tested but detail is restricted to the *colic* dataset only.

3.3 Stage 3 - Define a new Classification Problem to Differentiate Between Clusters

Analysis proceeds by separating the dataset into the clusters as appear on the variance versus mean edge plot. Cluster number is used as target class variable in a new classification problem. This may illuminate differences between the two clusters and improve classification accuracy as confounding on contexts (clusters) will be removed.
This process goes beyond simply selecting observations which were initially misclassified or by choosing observations with a high proportion of misclassifications. It is highlighting observations which have a certain variance and mean structure, symptomatic of flipping between correct and incorrect classification as boosting proceeds.

3.4 Stage 4 - Retrain the Classifier on a Subset of the Original Data

Once clusters have been identified, training new classifiers on each of the partitioned datasets will result in more generalisable models over these partitions. The difficulty arises for unseen observations when the correct cluster is unknown.
If the clusters identified in Stage 2 are able to be classified using only the predictor variables, it would be simple to use the resulting classifier to determine which cluster a new unseen observation would fall into. However, if the classification procedure fails to discriminate between clusters without using the original target variable, the optimal way to proceed is to assume all new observations are best modelled according to the classifier trained on the larger partition. This notion is shown to lead to reduced classification error as demonstrated in Section 5.

4 Demonstration on the UCI *colic* dataset

The 4-stage edge based procedure is now demonstrated on the *colic* dataset. (For more detail on this and other UCI datasets tested in this study, refer to Table 3.) The *colic* dataset requires binary classification pertaining to whether or not a horse had surgery (response =*yes/no*). A total of 23 input predictors are used in the classification process. Decreased generalisation performance has been observed on the *colic* dataset when boosting is applied. The four stage classification process suggested in this paper sheds some light on this phenomenon.
After detecting and removing an obvious outlier using the methodology from [11], a scatter plot of $\hat{Var}[edge(10,i)]$ versus $\hat{E}[edge(10,i)]$ is drawn and shown as Figure 1. Figure 1 shows 2 more possible outliers in observations *133* and *212*. After examining these observations there is no obvious reason as to why they should be considered outliers so they are retained in the *colic* dataset (in fact, it is seen in subsequent analysis that these observations are correctly classified by a model which is the inverse of the main model).

4.1 Stages 2 and 3 : Formulation of a new Classification Problem

The $\hat{Var}[edge(10,i)]$ versus $\hat{E}[edge(10,i)]$ plot is drawn for the *colic* data in Figure 1. Two clusters are ev-

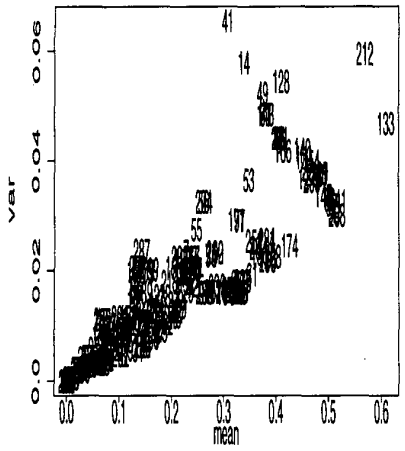

Figure 1. Plot of $\hat{Var}[edge(10,i)]$ vs. $\hat{E}[edge(10,i)]$: *colic* data.

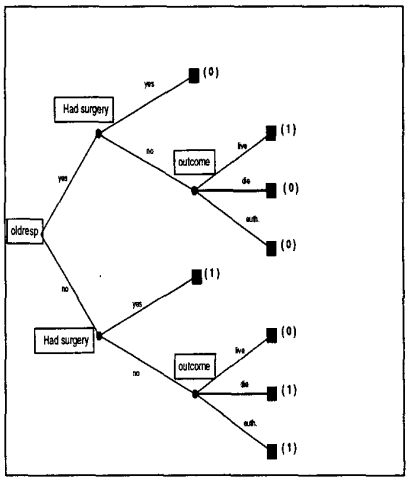

Figure 2. Decision tree for $edgeflag = 0/1$ response : *colic* data.

ident: one cluster being those observations exhibiting a positive relationship between $\hat{Var}[edge(10,i)]$ and $\hat{E}[edge(10,i)]$ and the others a negative one. Closer inspection reveals that all observations which were initially misclassified at iteration 1 fall into smaller 'negative' cluster, along with a small number of observations which were initally classified correctly.

A new dataset named *colicflag* is created for which a new response variable is appended to the *colic* dataset. This new response, $edgeflag$ is assigned a value of 1 if an observation lies in the upper right hand cluster of Figure 1 and 0 otherwise. A total of 14.1% of the observations were flagged with $edgeflag = 1$. This creates a new classification problem with $edgeflag = 0/1$ as the new response variable. If $c4.5$ or another classification algorithm can find repeatable structure in differentiating $edgeflag = 0$ from $edgeflag = 1$ using the available 23 predictor variables, then it may be concluded that such structure forms boundary regions of uncertainty, or differentiates between two contexts within the dataset. Because an aim of this study is interpretability, single trees are fitted as opposed to boosted ensembles, since boosted models become un-interpretable very quickly.

Several unsuccessful attempts to classify $edgeflag$ using only the initial 23 predictor variables were made. Only when the original response is included in the predictors did a concise, highly accurate split result. The original *(yes/no)* response was returned as the initial split on the decision tree, indicating different mechanisms occurring for different values of the initial response, given identical input contributions.

This may seem intuitive as observations which are difficult to classify will naturally be predicted oppositely the main 'core' of data. However, in this and other UCI datasets tested we see that each cluster has equivalent input ($\mathbf{x}_i$) values but reversed y_i (class) values. Figure 2 shows the resulting decision tree for classification of the $edgeflag$ variable. Note how the branches beyond the *Had Surgery* split are inverse to each other. Not only are the subtrees different on the initial split variable but are the exact inverse at greater depths in the decision tree.

Having a two stage model requiring prior knowledge of the original class variable will be futile in predicting unseen cases but leads us to hypothesise inverse models being present within a dataset.

4.2 Stage 4 - Retrain the Classifier on a Majority Subset of the *colic* Dataset

Using the *edgeflag* variable defined in Section 4.1, the *colic* dataset is split into two distinct datasets. Although we require the original target variable to do this, an important property is demonstrated. As will be shown, this results in a simple highly generalisable model fitting the majority of the data.
The partitioned datasets are labelled *horsec0* and *horsec1* respectively, with 0 and 1 representing the value of *edgeflag* determined in Section 4.1. The number of observations in each dataset is 262 and 37 respectively. The initial binary classification problem of *had surgery=yes/no* using 23 input predictors may then be re-estimated for each sub-dataset (*horsec0, horsec1).* For interpretability, a *c*4.5 decision tree using default values was used as the classification method. The results are encouraging with training error for the *horsec0* dataset being only 0.38% (as opposed to 14.99% on the entire *colic* dataset). Refer to Table 2 for detail of generalisation performance on the *colic* dataset. The figures presented in the final column refer to the average generalisation error estimated via 10-fold cross-validation , with bracketted figures giving the standard error of this estimate. The large standard errors evident for the *horsec1* dataset are based on test sets of 3-4 observations and may not be representative.

Table 2. Cross-validation results for *colic* dataset.

Dataset	Class	Method	Gen error
colic	(y/n)	*c*4.5 single	14.99(8.76)
colic	(y/n)	*c*4.5 10 boost	21.01(7.21)
colicflag	(0/1)	*c*4.5 single	0.34(1.08)
horsec0	(y/n)	*c*4.5 single	1.53(2.68)
horsec0	(y/n)	*c*4.5 10 boost	0.38(1.20)
horsec1	(y/n)	*c*4.5 single	3.33(10.25)
horsec1	(y/n)	*c*4.5 10 boost	3.33(10.25)

The decision tree is fitted on a reduced set of training data and the generalisation error on this partition will not be representative of the generalisation error for the entire dataset. Section 6 outlines logic for a new generalisation error estimate which encompasses error from both partitions of the dataset.

5 Generalisation Error Calculations

An empirical study is undertaken on a selection of UCI[2] [1] datasets to test the process outlined in Section 3. For all datasets, the term 'boosting' refers to the AdaBoost.M1 algorithm presented in Section 2. Unless otherwise described for a particular dataset, 10 boosting iterations were applied using *c*4.5 with default options as the base learner.
Table 3 gives a description of the UCI datasets used in this study.

Table 3. Summary of UCI datasets used in this study - as per [9])

Dataset	Cases	Num Classes	Cont. Attr.	Disc. Attr.
colic	368	2	10	12
credit	690	2	7	13
heart-h	294	2	8	5
heart-c	303	2	8	5
h'titis	155	2	6	13

Because it is proposed to use the simpler model trained on the majority partition for future prediction, generalisation error must be re-defined. Assume the proportion of observations falling into the main cluster is given as p. $(0 < p < 1)$. Let e_1 represent the estimated generalisation error for the simple classifier trained on this cluster. 10 fold crossvalidation is used as the process for estimating e_1. 10 Now, since the remainder of the data is modelled in an inverse fashion to the main cluster, the generalisation errror for this section of the data will at worst be estimated at $1 - e_1$. Denoting the new overall generalisation error

[2]URL = http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 4. New estimates of generalisation error for a selection of UCI datasets.

dataset	% data with $edgeflag = 1$	e_{init}	$e_{cluster}$
colic	14.13	21.01	14.40
credit	9.3	18.24	12.17
heart-h	12.03	22.61	15.99
heart-c	9.9	18.41	16.64
hepatitis	7.1	16.00	14.98

estimate as $e_{cluster}$, we have:

$$e_{cluster} \approx (p*e_1)+(1-p)*(1-e_1) \quad (2)$$

Calculations of this generalisation error after applying the four stage classification process for a selection of UCI datasets are presented in Table 4. Note: e_{init} refers to the generalisation error as estimated on a 10 iteration boosted classifier trained on the entire dataset.

For the *colic, credit* and *hearth-h* datasets, this simplification of models results in an impressive reduction in generalisation error. For the *heart-c* and *hepatitis* datasets, reduction of error occurs but is less dramatic. In any case, this process certainly doesn't deteriorate generalisation error, and for all datasets we have gained model simplicity.

6 Discussion

In many cases, the $\hat{Var}[edge(10,i)]$ versus $\hat{E}[edge(10,i)]$ procedure identifies clusters no differently than simply selecting observations which were initially misclassified by a single decision tree on the entire dataset. Although discouraging, this phenomenon brings the following points to light:

- The usual practice of applying boosting to the entire dataset gives difficult observations an opportunity to be modelled correctly by the classifier. It was seen for the datasets tested, that even after 20 boosting iterations, the boosted ensemble was no closer to making consistent predictions on these groups of observations. The variance of the edge also remains large as boosting flips between correct and incorrect classification on these observations.
- Closer inspection revealed these observations to occupy a similar region in predictor space to that of the 'core' data but the class labels are reversed. This could be attributed to mislabelling of the output or absence of a relevant predictor variable.
- In all datasets tested, the initial decision tree trained on the entire dataset was significantly more complex in its attempt to accommodate for the observations falling into the smaller cluster. A very simple robust tree with low generalisation error results when $c4.5$ is trained on the main cluster only. Using the tree trained on a subset of the data does not significantly deteriorate generalisation performance, as demonstrated in Table 4 for a selection of UCI datasets.
- Boosting is often successful within the 'core' data cluster, even when unsuccessful on the entire dataset.

7 Conclusion

This study has proposed a new multistage classification process which not only improves generalisation error but also results in easily interpretable classification models. In all datasets tested it was seen that a simple, highly accurate model built on the majority of the data was preferable to a more complex model trained on the entire dataset. The more complex model fitted to the entire dataset was a result of the need to accommodate noise and difficult observations. This additional detail was seen to cloud the underlying core classification model. It was also proposed that the UCI datasets tested contain sections of mislabelled data, or are lacking the measurement of a key predictor variable.
The clusters identified from the edge diagnostics are usually unable to be discriminated without the use of the initial class variable. However, if the simple tree trained on the majority cluster is applied

to the entire dataset, the overall misclassification rate is reduced, sometimes substantially.

Further work will concentrate on automatic cluster detection and testing other possible edge measures across a varying number of boosting iterations and datasets.

References

[1] Blake, C.L. & Merz, C.J. (1988). UCI Repository of Machine Learning Databases. url=http://www.ics.uci.edu/~mlearn /MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences.

[2] Breiman, L. (1996a). Bagging predictors. *Machine Learning, 26(2)*, 123–140.

[3] Breiman, L. (1996b). *Bias, Variance and Arcing Classifiers* (Technical Report 460). Statistics Department, University of California, Berkeley.

[4] Breiman, L. (1997). *Arcing the edge* (Technical Report 486). Statistics Department, University of California, Berkeley.

[5] Breiman, L. (1999). *Random Forests - Random Features* (Technical Report 567). Statistics Department, University of California, Berkeley.

[6] Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 148–156). Morgan Kaufmann.

[7] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalisation to on-line learning and an application to boosting. *Journal of Computer and System Sciences , 55(1)*, 119–139.

[8] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

[9] Quinlan, J. R. (1996). Bagging, boosting and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. (pp. 725–730). Menlo Park California, American Association for Artificial Intelligence.

[10] Schapire, R. E. , Freund, Y., Bartlett, P. & Lee, W. S. (1998). Boosting the margin:a new explanation for the effectiveness of voting methods. *Annals of Statistics, 26(5)*, 1651–1686.

[11] Wheway, V. L. (2000). Using Boosting to Detect Noisy Data *To appear in Lecture Notes in Computer Science 2112:AIApps 2000* Springer