Pre-publication draft of paper accepted for publication in the Proceedings of the Fourth Data Mining Conference (DM IV) 2003 pp 265-275 Publisher: WIT Press

# Convex Hulls as an Hypothesis Language Bias

D.A. Newlands, G.I.Webb
Deakin University, Victoria 3217, Australia
Monash University, Victoria 3800, Australia

June 5, 2003

**Abstract**

Classification learning is dominated by systems which induce large numbers of small axis-orthogonal decision surfaces which biases such systems towards particular hypothesis types. However, there is reason believe that many domains have underlying concepts which do not involve axis orthogonal surfaces. Further, the multiplicity of small decision regions mitigates against any holistic appreciation of the theories produced by these systems, notwithstanding the fact that many of the small regions are individually comprehensible. We propose the use of less strongly biased hypothesis languages which might be expected to model concepts using a number of structures close to the number of actual structures in the domain. An instantiation of such a language, a convex hull based classifier, CH1, has been implemented to investigate modeling concepts as a small number of large geometric structures in n-dimensional space. A comparison of the number of regions induced is made against other well-known systems on a representative selection of largely or wholly continuous valued machine learning tasks. The convex hull system is shown to produce a number of induced regions about an order of magnitude less than well-known systems and very close to the number of actual concepts. This representation, as convex hulls, allows the possibility of extraction of higher level mathematical descriptions of the induced concepts, using the techniques of computational geometry.

## 1 Introduction

Classification learning has been dominated by the induction of axis-orthogonal decision surfaces in the form of rule-based systems, decision trees, inductive logic programming and decision graphs. While the induction of alternate forms of decision surface has received some attention, in the context of non-axis orthogonal decision trees, statistical clustering algorithms, instance based learning and regression techniques [15, 17, 1, 31, 19, 8, 7, 28, 12], this issue has received

little attention in the context of decision rules. In learning systems that divide the concept space using axis parallel decision surface, the learned concept can only be expressed in terms of a collection of hyperrectangles. The lack of congruence between the hypothesis language and the underlying concepts causes the generation of a multiplicity of small and inappropriately shaped regions, the sum of which gives some degree of approximation to the underlying concepts.

In a domain containing a small number of concepts, it is debatable whether a representation that involves tens or hundreds of small regions contributes to human comprehensibility. Certainly each small area may be individually explicable but holistic comprehension may be impossible. Also the hyperrectangular structure imposed on the domain may be a subset or superset, depending on the vagaries of the sampling process, of the volume for which the interpretation is true. Thus, the underlying hypothesis language may cause the exclusion of volumes that are explicable and the inclusion of volumes that are not.

One might, on philosophical grounds, assert that the universe changes slowly and regularly as one traverses it and so underlying, natural, concepts will tend to exhibit some regularity and smoothness. One obvious approach to modelling concepts is to assert that there is underlying symmetry in the universe and that an hypothesis language which treats all dimensions symmetrically is attractive. However, the scales of the metrics which we, as observers, impose on the external universe might vary between dimensions so that the induced concept is not symmetric in the set of imposed metrics. This work proposes the induction of concepts represented by convex hulls and investigates their congruence to actual underlying concepts and their utility as classifiers. It is expected that having an hypothesis language which is congruent with the underlying actual concepts should lead naturally to a concise representation of those concepts. It is expected that, although individual rules or groups of decision surfaces may be moderately sized, composite objects, the collection of rules describing the domain, will be both simple and small. Indeed, the closeness of the number of induced concepts and the number of underlying concepts is seen as a measure of the appropriateness of the hypothesis language for that domain. It might be considered that it offers insight into the underlying actuality of that part of the universe.

In some contexts, it will be desirable that the rules developed by computer systems be comprehensible by humans. Typically, machine learning systems produce many rules per class and, although each rule may be individually comprehensible, holistic appreciation of concepts modeled may be impossible due to the fragmentary representation. We contend that comprehending the set of induced rules is quite different from comprehending the actual underlying domain and that claims of human comprehensibility of the domain via the rule sets may be quite unjustified. Each concept, constructed as a large, convex polytope in this work, is expected to correspond closely to a single underlying concept of the domain. Although the structure of such concepts is not directly comprehensible, the form of the concepts gives access to work on extracting mathematical

descriptions via the techniques of computational geometry including diameters of polytopes, intersections and equations for the surfaces of polytopes [22]. An important characteristic of systems developed in this work will be the small number of the regions representing a concept.

## 2 Choice of Implementations of Convex Hull Forming Algorithms

Several algorithms for the construction and specification of convex hulls have been published [2, 3, 5, 9, 10, 11, 16, 21, 20, 24]. Typically, but not necessarily, a convex hull is specified by a set of oriented hyperplanes. The orientation is specified by the components of an outward pointing vector of length one, perpendicular to the hyperplane and the position by the perpendicular distance of the hyperplane from the origin. A point is said to be *beneath* a plane if it is coincident with the plane or on the correct (internal to the polytope) side and *beyond* the plane otherwise. The time complexity of forming convex hulls of N points in $\Re^d$ has been shown [26, 13] to be $O(N^{\lfloor (d+1)/2 \rfloor})$ For a point to be within a convex hull, it must be beneath every hyperplane. As soon as a point is beyond any hyperplane, it is known not to be within the convex hull. Since the convex hull constraint will provide some degree of smoothing to the edges of concepts, there is some expectation of avoiding problems of overfitting[29] and oversearching[23] naturally.

There are a number of implementations available but the choice for this work is constrained by

1. the need for the implementation to accept input as attribute vectors. Specifically, algorithms that work in dual space [22] are not easily usable as it is desirable to use well-known data sets (from UCI Repository) without transformation.

2. the algorithm should use a floating point representation of values. Conversion to integers may be done automatically for the training set as it is processed by the convex hull constructing software. However, it has to be done externally for the test points since they are not processed by the hull constructing software. The use of integral values also reduces the sensitivity of the classifier at points over the decision surfaces that do not have integer coordinates.

3. no algorithm that uses rotation of the axis system (especially if the rotation is random) can be acceptable as attributes are not interchangeable.

4. some forms of internal scaling of data values may not be acceptable because they cannot be repeated outwith the software package (particularly scaling dependent on the volume of the initial simplex).

5. the algorithm needs to function in spaces of high dimensionality and for most algorithms 5D is very high since the number of facets and ridges becomes very large. For example, Klee [14] estimates the number of facets, F(d,N), of a d-polytope with N vertices could be as large as

$$F(d,N) = \begin{cases} \frac{2N}{d} \begin{pmatrix} N - \frac{d}{2} - 1 \\ \frac{d}{2} - 1 \end{pmatrix} & for\ d\ even \\ 2 \begin{pmatrix} N - \lfloor \frac{d}{2} \rfloor - 1 \\ \lfloor \frac{d}{2} \rfloor \end{pmatrix} & for\ d\ odd \end{cases}$$

However, the expected number of facets for random points is proportional to $\log^{d-1} n$ [6].

6. the algorithm should output a facet list with components of the unit normal and the distance from the origin to facilitate later tests for inclusion of points in the hull by the concept learning software.

The qhull software [27], which is an implementation of the Grünbaum Beneath-Beyond Theorem [11], was chosen for the construction of convex hulls and will be called from the classifier software written for this work. This choice was made because it provides straightforward use of data sets from the UCI repository [18] without transformation, output in an immediately useful form; control over the size of the facet list, easy access to testing new points for inclusion via inner products, no rotation of the axis system containing the points,and scaling can be done simply to both training and test data sets if necessary.

# 3 CH1 Algorithm

In the system developed, the antecedent of each rule is, in principle, represented by a single convex hull projected onto the instance space. The consequent of each rule is a class. There may be more than one rule per class. These rules are held in a decision list into which an initial default rule, for the most populous class, is inserted. Subsequent rules, particularly exceptions to the current rule, are prepended to the decision list in the expectation that this strategy will shorten the list [30]. The main loop continues until there are no misclassified points or the rule just constructed does not reduce the number of misclassified points

The system should be tested on purely numeric domains but most data sets will have some categorical attributes so an extension to the structure will be made to be able to process domains which are largely, or wholly numeric. As categorical attributes cannot take part in hulls, a separate rule component is formed for them and so, if we consider examples E, with attributes $1 \ldots k$ being categorical and attributes $k+1 \ldots n$ being numeric, a Rule, R, of a class is

- a vector of attribute value sets, $< A_1, \ldots, A_k >$ where $A_j = \bigcup_{i=1}^{e} E_{ij}$ and $E_{ij}$ is the value of the $j-th$ categorical attribute of the $i-th$ example and
- a set $\{P_j\}$ where each $P_j$ is a hyperplane over the numeric attributes.

To classify a new example $X$, with attribute vector $< x_1 \ldots x_n >$ as being covered by Rule, $R$,

$$is\_covered(X, R) = \bigwedge_{j=1}^{k} (x_j \in A_j) \wedge \bigwedge_{h \in \{P_j\}} is\_beneath(X, h)$$

where is_beneath() is true iff the point defined by the numeric attributes of $X$ are beneath the hyperplane $h$. While for domains with both categorical and numeric attributes both the convex and categorical hulls will exist, we will refer to the categorical and numeric hull pair as 'the hull' for ease of expression. If a set of points has zero thickness in one or more dimensions, then the quickhull software exits with an error indicating the hull is degenerate and the convex hull is replaced with an hyperrectangle which contains a maximum and minimum value pair for each attribute. This seems to be an infrequent occurence and, for the purposes here, it will be assumed that these are treated transparently by the CH1 software.

## 3.1 Time Complexity of classification using CH1

Consider the classification of $n$ points in $d$ dimensional space where there are $c$ actual concepts. It is a design expectation that there will be approximately $c$ hulls. It is also the case that a point that is outside any given hull will be *beneath* approximately half the facets and *beyond* the other half of the facets of the given hull. Firstly, the time for considering the coverage of categorical values is negligible compared to coverage of continuous attributes and will be ignored. Thus, it is only necessary to consider a point as being covered when it is beneath every facet for a given hull. Since the hulls are in a decision list and the facets are unordered, they will both be inspected sequentially but when the current point is found to be beyond the current facet, the inspection of facets of the current hull can be abandoned. Assuming the data points are approximately evenly divided among hulls, a hull will have $\log^{d-1} \frac{n}{c}$ facets. Typically, we would expect to test half the hulls before finding one that does not cover the current point. Thus testing a single point has time complexity $O(c * \log^{d-1} \frac{n}{c})$.

## 3.2 Implementation of CH1

The algorithm was implemented in C and interfaced to the quickhull software [4]. When a convex hull has been created, it is stored in the calling program's rule

list for use when classifying test points. Each rule contains a set of categorical attribute values and a list of convex hull facets. Each facet in the list contains

- the signed offset of the hyperplane from the origin (the sign specifies which side is *beneath* and which is *beyond*.
- a list of the components of a unit outward pointing normal to the hyperplane.
- the distance above the plane that is considered to be *beneath* the plane. This is usually a small number reflecting the rounding errors in the calculations or a representation of the thickness of a hull (used by Quickhull when needing an approximation to a hull).

Thus, in an N-dimensional domain, each facet is represented by $N + 2$ floating point numbers.

# 4 Experimental Evaluation

The implementation was evaluated using 22 data sets from the UCI Repository [18]. Since the implementation of CH1 is principally using convex hulls, domains with few or no continuous values will not be used in the evaluation. Domains that are wholly continuous are of the most interest but, since categorical attributes can be handled, domains with a small number of categorical attributes and many continuous ones can be used. Each experiment involved shuffling the data set and paritioning 80% into the training set and 20% into the test set. The experiment was run 100 times, using matched data sets, on each domain using CH1 and C4.5 and the average results are shown in Table 1. Because of prohibitive run-times, only subsets of the available data were used for some domains (marked with *).

# 5 Analysis of Results

For CH1, the number of structures is the number of convex hulls that are constructed and for C4.5 the number of structures is the number of hyperrectangular regions that are identified. The number of actual concepts in each domain (which is unknown to CH1) and the number of concepts induced by CH1 and C4.5, averaged over 100 runs, are shown in Table 1. The average number of concepts for satimage and shuttle1 are lower than one might expect because the sample used contains effectively only 5 classes for satimage rather than 6 and, for shuttle1, contains about 3 classes rather than 7. The smallness of the number of concepts induced by CH1 can be seen in comparison with C4.5. The number of hulls induced by CH1 is always very close to the number of actual concepts in each domain. In only 4 cases, out of 22, does CH1 produce more

| Domain | No. of Concepts | No. Hulls CH1 | No. Regions C4.5 | Accuracy CH1 | Accuracy C4.5 |
|---|---|---|---|---|---|
| balance-scale | 3 | 5.6 | 108.2 | 83.26[2] | 76.67 |
| bcwo | 2 | 2.8 | 33.2 | 99.06 | 99.42[1] |
| bupa | 2 | 3.7 | 95.8 | 58.77 | 63.49[1] |
| cleveland | 2 | 3.7 | 66.8 | 55.54 | 87.73[1] |
| echocardiogram | 2 | 2.6 | 13.5 | 70.81[2] | 69.11 |
| german | 2 | 7.3 | 76.4 | 50.71 | 60.66[1] |
| glass | 7 | 13.0 | 49.4 | 55.21 | 66.18[1] |
| heart | 2 | 3.7 | 11.4 | 67.71 | 80.29[1] |
| hepatitis | 2 | 2.0 | 4.4 | 40.51 | 41.57[1] |
| horse-colic | 2 | 2.3 | 9.6 | 62.01 | 77.03[1] |
| hungarian | 2 | 3.6 | 55.2 | 62.48 | 88.55[1] |
| ionosphere | 2 | 5.0 | 29.4 | 90.83[2] | 87.71 |
| iris | 3 | 3 | 8.6 | 69.85 | 94.74[1] |
| new-thyroid | 3 | 3.0 | 14.5 | 75.78 | 91.83[1] |
| page-blocks | 5 | 7.6 | 14.8 | 57.47 | 87.20[1] |
| pid | 2 | 5.1 | 39.0 | 80.30 | 82.34[1] |
| satimage * | 6 | 5.8 | 19.4 | 72.36 | 91.33[1] |
| segment | 7 | 22.0 | 63.0 | 91.94 | 94.41[1] |
| shuttle1 * | 7 | 4.8 | 9.0 | 66.32 | 93.84[1] |
| sonar | 2 | 2.5 | 31.6 | 56.64 | 74.12[1] |
| soybean-large | 19 | 19.7 | 70.2 | 81.01 | 90.68[1] |
| wine | 3 | 3.0 | 39.5 | 84.04 | 87.01[1] |

Table 1: Comparison of CH1 and C4.5

than twice as many hulls as concepts whereas C4.5 produces less than four times as many regions as underlying concepts only 5 out of 22 times. Using a sign test, the number of hulls produced by CH1 is clearly superior to C4.5 at p=0.01. It is clear that the primary objective of having a system which induces a similar number of structures to the underlying actuality has been achieved. Thus, it is clear that the less strong bias of convex hulls is more appropriate in some way than the stronger bias of C4.5.

The comparison of average accuracy over the matched datasets does not favour the convex hulls and this poorer predictive accuracy weakens but does not invalidate the claims for the appropriateness of the bias of convex hulls. It is tempting to infer that most of the datasets in the UCI Repository were submitted after experiments with hyperrectange-based classifiers and that there is an implicit bias in the data set as a result. This is plausible but there are some other difficulties which need clarification before establishing such a claim. Despite the claims of good performance for new convex hull generating packages, it is a fact that they struggle with dimensionality of 5 or greater and exhibit very long run times. To cope with these difficulties, the algorithm makes approximations, merges facets and manipulates a thickness for each facet of the convex hull. It is suspected that these affect the final hulls being constructed.

# 6 Conclusions

It has been shown that the use of convex hulls for induction is practicable by the implementation of such a system. The resultant classifier had a much less strong hypothesis language bias than axis orthogonal systems and realised the possibility of representing concepts as a few convex hulls, rather than a multiplicity of small inappropriately shaped regions. The number of hulls generated was close to the actual number of underlying concepts and never much greater, especially in comparison to the typical multiplicity of regions generated by C4.5. A classifier that induces one large structure per concept rather than many small structures is philosophically appealing in its economy of representation. The possibility of using the tools of computational geometry to extract higher level mathematical descriptions of concepts was noted although it has not been demonstrated. Considering the correspondence between the number of induced concepts and the number of underlying concepts, it can be claimed that convex hulls provide a good hypothesis language bias and, also, that this suggests the underlying geometry of these domains is of a similar structure.

The accuracy results have been disappointing and the cause of this needs investigation to understand if convex hulls are to be generally useful. Given that there is also a processing time problem with convex hull generating software packages, it may be necessary to use less direct techniques to construct the hulls in future work.

One possible such approach would be to use evolutionary programming tech-

niques to place the hyperplanes, directly, avoiding the long compute times, approximations within the hull software and artificialities such as thickness of facets. Other approaches, for instance DIPOL92 [25], use regression and do not position the hyperplanes in the same way that CH1 does and it would be interesting to explore the differences. A demonstration of actual progress to the extraction of an higher level mathematical description of the concepts modeled by the hulls would also be desirable.

# References


[1] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] W. Altherr. An algorithm for enumerating the vertices of a convex polyhedron. *Computing*, 15:181–193, 1975.

[3] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Computational Geometry*, 8:295–313, 1992.

[4] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. Submitted to ACM Trans. Mathematical Software, May 1995.

[5] J.L. Bentley, M.G. Faust, and F.P. Preparata. Approximation algorithms for convex hulls. *Comms. of the ACM*, 25(1):64–68, 1982.

[6] J.L. Bentley, H.T. Kung, M. Schkolnick, and C.D. Thompson. On the average number of maxima in a set of vectors. *J. ACM*, 25:536–543, 1987.

[7] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth Int. Group, Belmont, California, 1984.

[8] C.E. Brodley and P.E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.

[9] Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Verlag, 1987.

[10] I.Z. Emiris, J.F. Canny, and R. Seidel. An efficient approach to removing geometric degeneracies. In *Proceedings of the 8th Annual ACM Symposium on Computational Geometry*, pages 74–82, 1992.

[11] B. Grünbaum. Measures of symmetry for convex sets. In *Proc. 7th Symposium in Pure Mathematics of the AMS*, pages 233–270, 1961.

[12] D. Heath, S. Kasif, and S. Salzberg. Learning oblique decision trees. In *Proc. 13th IJCAI*, pages 1002–1007. Morgan Kaufmann, 1993.

[13] M. Kallay. Convex hulls in higher dimensions. Technical report, Dept. Math., University of Oklahoma, Norman, Oklahoma, 1981.

[14] V. Klee. Convex polytopes and linear programming. In *Proc. IBM Sci. Comput. Symp: Combinatorial Problems*, pages 123–158, 1966.

[15] C. Matheus and L.A. Rendell. Constructive induction on decision trees. In *Proceedings of IJCAI*, pages 645–650, 1989.

[16] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Cambridge University Press, Cambridge, England, 1971.

[17] Michalski, R.S., Mozetic, I., Hong, and N. J. Lavrac. The multi-purpose incremental learning system aq15 and its testing and application to three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045. Morgan Kaufman, 1986.

[18] P.M. Murphy and D.W. Aha. The uci repository of machine learning databases, http://www.ics.uci.edu/ mlearn/mlrepository.html.

[19] S.K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[20] F.P. Preparata. An optimal real-time algorithm for planar convex hulls. *Comms. of ACM*, 22(7):402–405, 1979.

[21] F.P. Preparata and S.J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Comms. of ACM*, 20(2):87–93, 1977.

[22] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Texts and Monographs in Computer Science. Springer-Verlag, New York, 1985.

[23] J.R. Quinlan and R.M. Cameron-Jones. Oversearching and layered search in empirical learning. *IJCAI*, pages 1019–1025, 1995.

[24] S. Schuierer, G.J.E. Rawlins, and D. Wood. A generalisation of staircase visibility.

[25] B. Schulmeister and Wysotzki. The piecewise linear classifier dipol92. In *Proc. ECML94*, pages 411–414, 1994.

[26] R. Seidel. A convex hull algorithm optimal for points in even dimensions. Master's thesis, U. of B.C., Canada, 1981.

[27] Software Development Group, Geometry Center, 1300 South Second Street, Suite 500, Minneapolis, MN 55454, USA. *Qhickhull Software Manual*.

[28] P.E. Utgoff and C.E. Brodley. Linear machine decision trees. Technical report, U. Mass. at Amherst, 1991.

[29] C.J.C.H. Watkins. Combining cross-validation and search. In *Progress in Machine Learning;Proc.of EWSL 87*. Sigma Press, Wilmslow, 1987.

[30] G.I. Webb. Recent progress in learning decision lists by prepending inferred rules. In *Second Singapore International Conference on Intelligent Systems*, pages B280–B285, 1994.

[31] Simon Yip and Geoffrey I. Webb. Discriminant attribute finding in classification learning. In A. Adams and L. Sterling, editors, *AI'92 – Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence*, pages 374–379, Hobart, 1992. World Scientific.