Principled Constructive Induction

Pankaj Mehra[1]
Coordinated Science Lab.,
University of Illinois,
Urbana, IL 61801.

Larry A. Rendell
Beckman Institute,
University of Illinois,
Urbana, IL 61801.

Benjamin W. Wah[1]
Coordinated Science Lab.,
University of Illinois,
Urbana, IL 61801.

Abstract

A framework for the construction of new features for hard classification tasks is discussed. The approach brings together ideas from the fields of machine learning, computational geometry, and pattern recognition. Two heuristics for evaluation of newly-constructed features are proposed, and their statistical significance verified. Finally, it is shown how the proposed framework can be used to combine techniques for selection of representative examples with techniques for construction of new features, in order to solve difficult problems in learning from examples.

1. Introduction.

The *problem of new terms*, also known as the constructive induction problem, has long been considered a source of difficulty in machine learning (Dietterich, 1982). Simple classifiers using only the primitive features of description have limited learning capabilities. For example:

(i) Single-layered neural networks can realize only those class dichotomies, where the classes are *linearly separable* in the feature space (Minsky, 1969).

(ii) Selective induction can be used to learn only those concepts whose concept-membership function is *smooth* (Rendell, 1986).

Researchers in different areas have recently addressed these fundamental limitations of simple classifiers. Algorithms such as back-propagation (Rumelhart, 1986) can implement learning in multi-layered networks. They implicitly create weighted combinations of primitive features in the internal (hidden) units of such networks. Constructive induction, on the other hand, explicitly constructs and tests new terms from the primitive features (Dietterich *et al.*, 1982) by applying feature-construction operators. Both of these approaches transform the primitive feature space of the problem into another in which the classes to be discriminated are separable using simple discrimination surfaces.

With a few exceptions (Muggleton, 1988), these techniques provide no justification for the heuristics used, they do not integrate theories of selection and evaluation of features, and they have no obvious trade-offs between performance and accuracy. In this paper, we examine the problem from a geometric perspective in hope of developing heuristic techniques that are amenable to analyses of performance and accuracy. Our focus is on a study of constructive induction in two-class discrimination learning problems.

1.1. Simple Versus Hard Classification Problems.

Given a set of d-dimensional training samples, $E = E^{+} \cup E^{-}$, a simple classification problem is to discover a primitive surface (such as a hyperplane in d-1 dimensions) that separates the positive examples E^{+} from the negative examples E^{-}, when the sample points and the hyperplane are represented in the d-dimensional feature space. In a hard classification problem, such a primitive surface does not exist.

1.2. Constructive Induction.

Constructive induction discovers new features of the training sample. It transforms feature spaces to permit primitive separating surfaces which, in turn, enable the use of simple classification techniques for solving hard classification problems.

[1]This research was supported by the National Science Foundation under Grant MIP-88-10584, and by the National Aeronautics and Space Administration under Grant NCC 2-481.

2. Feature Spaces and Inverted Spaces.

Let $E=E^{+}\cup E^{-}$ be the set of examples. Each $e^{i}\in E$ is a d-dimensional vector, $e^{i}=(e^{i}_{1},e^{i}_{2}..,e^{i}_{d})$. We say that the classification problem is defined in a d-dimensional feature-space $F=(f_{1},f_{2},..,f_{d})$, and that the value of the feature f_{j} for the example e^{i} is e^{i}_{j}. A hyperplane H in F has the form H: $W\cdot F=\theta$, where $W=(w_{1},w_{2},..,w_{d})$ is a d-dimensional vector of weights, and θ is a constant. A hyperplane H is said to separate E^{+} and E^{-} if, for any $e^{+}\in E^{+}$ and $e^{-}\in E^{-}$, $(W\cdot e^{+}-\theta)(W\cdot e^{-}-\theta)<0$.

In *feature spaces*, each feature is represented by a dimension, and each example is represented by a point. So, for instance, the example e^{i} is represented by the point $(e^{i}_{1},e^{i}_{2},..,e^{i}_{d})$ in the feature space F.

Inverted spaces, after Watanabe (1985), can be defined, *one for each class in the feature space*, as follows. Each example of the class is represented by a dimension, and features are represented as points. If $E^{+}=\{p^{1},..,p^{m}\}$, then the feature $f_{i}\in F$ maps to the point $(p^{1}_{i},p^{2}_{i},..,p^{m}_{i})$ in the inverted space for the class E^{+}. Similarly, if $E^{-}=\{n^{1},..,n^{l}\}$, then the feature f_{i} maps to the point $(n^{1}_{i},n^{2}_{i},..,n^{l}_{i})$ in the inverted space for the class E^{-}. Figure 1 shows the construction of inverted spaces for the two-dimensional parity (XOR) problem. Even though the framework is general, our examples are restricted to two-dimensional spaces, primarily because it is easy to illustrate such spaces visually.

An advantage of inverted spaces is that features become points. Consequently, feature space transformations (the basic operation in constructive induction) can be interpreted geometrically. In inverted spaces, constructive induction is the creation of new points by applying point-to-point, or (set of points)-to-point, transformations. Only some of these transformations are useful: the inclusion of certain additional points in the inverted spaces (feature dimensions in F-space) can *create* separability. In the next few sections, we develop a representation of primitive surfaces and a notion of separability in terms of the inverted-space framework.

3. Linear Separability in Inverted Spaces.

Linear separators are represented geometrically in feature spaces as hyperplanes, and in inverted spaces as *hypercubes*. Figure 2 shows the basic representations of linear separators in both a feature space and the corresponding inverted space for the class $\{e_{1},e_{2}\}$. A hyperplane $W\cdot F=\theta$ divides the feature space F into the half-spaces $W\cdot F<\theta$, and $W\cdot F>\theta$. In order to give a geometric interpretation to the surface defined by $W\cdot F=\theta$ in an inverted space for a set of examples E, the hyperplane equation can be rewritten as

$$\forall e\in E,\ W\cdot F(e)=\theta. \qquad (1)$$

Elements of F (features) map into points in the inverted space. Hence, $W\cdot F$ can be interpreted as a linear combination of the vectors from the origin to the points representing features of F. Thus, as illustrated in Figure 2, if E has k components, equations (1) define a set of k orthogonal lines, oriented parallel to inverted space axes, intersecting at $(\theta,\theta,..\theta)_{k}$. These lines divide the inverted space into 2^{k} orthants. When θ is positive (resp., negative), the half-space $W\cdot f<\theta$ (resp., $W\cdot f>\theta$) of feature space, is represented by the orthant containing the inverted space origin; the other half-space, by the orthant

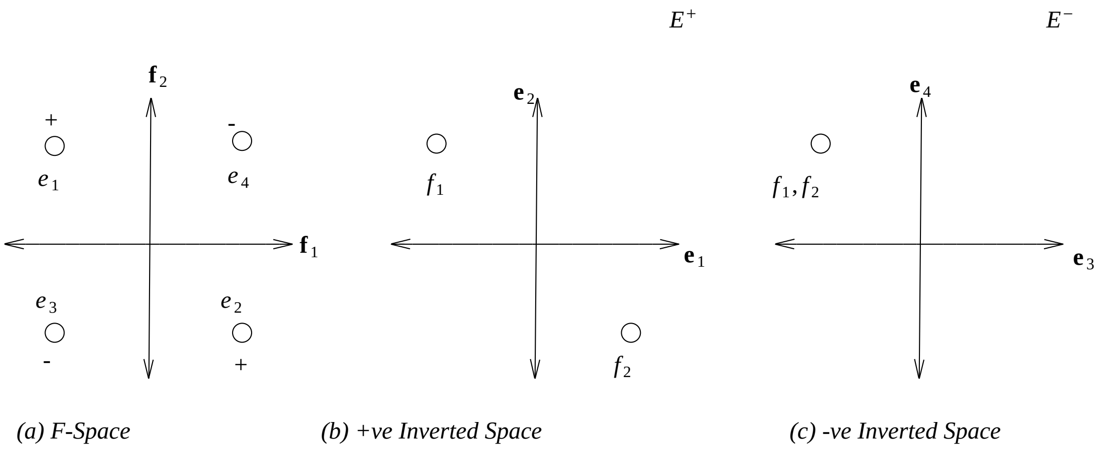


Figure 1. From XOR(f_1,f_2) in (a), the positive examples e_1 and e_2 define the positive inverted space (b), and the negative examples e_3 and e_4 produce the negative inverted space (c).

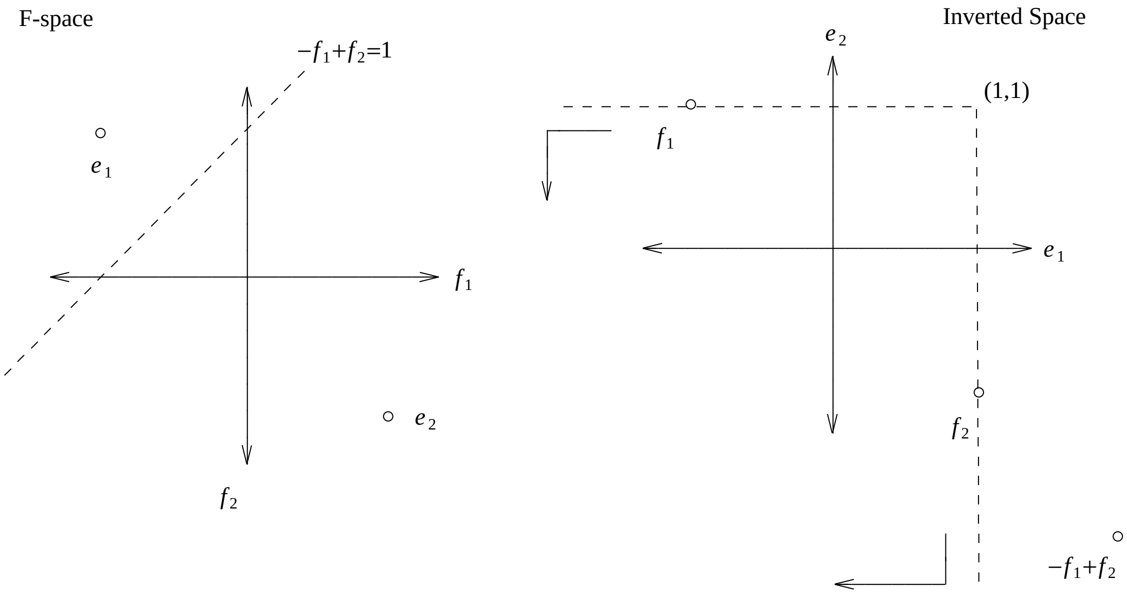


Figure 2. The half-space defined by $W{\cdot}F < \theta$ in F-space, where $W = [-1,1]$, $F = [f_1,f_2]$, and $\theta = 1$, maps into the inside of an inverted-space hypercube with one corner at $(\theta,\theta) = (1,1)$.

opposite the first. In what follows, we assume (w.l.o.g.) that θ is positive. Viewing the aforementioned orthants as infinitely extending hypercubes with one corner at $(\theta,\theta,..\theta)_k$, we call the region satisfying $W{\cdot}f<\theta$ as the *negative* θ-*hypercube*; and, the other, *positive* θ-*hypercube*. In order for a weighted combination of features, $W{\cdot}f$, to separate E^+ and E^-, the point resulting from the linear combination of feature points in the E^+-inverted space must lie inside the positive (resp., negative) hypercube, whenever the corresponding point for the same combination in the E^--inverted space lies inside the negative (resp., positive) hypercube.

Observation. A feature separates two classes E^+ and E^- in the feature space if its E^+-inverted space representation is a point inside the positive (resp., negative) θ-hypercube, and its E^--inverted space representation is a point inside the negative (resp., positive) θ-hypercube.

4. Two Measures of Feature Goodness.

Information-theoretic considerations (Watanabe, 1985) suggest that using a good feature of discrimination provides *compact descriptions of each of the two classes*, and that these *descriptions are maximally distinct*. Geometrically, this constraint can be interpreted to mean that (i) such a feature takes on nearly identical values for all examples of the same class, and (ii) it takes on some different value for all examples of the other class.

We define the **equiparameter line** in the inverted space for a class $E=(e^1,e^2,e^3, .., e^p)$ by the following system of equations:

$$f_i \text{ such that } e_i^1 = e_i^2 =.. = e_i^p$$

so that a feature f_i that lies on this line is maximally uniform within the class. *Deviations of a feature from such uniformity can be approximated by the sine of the angle between the feature vector and the equiparameter line in the inverted space.* Interestingly, this heuristic measure is directly related to the standard deviation of the feature values within the class sample. We state this formally:

Theorem. The standard deviation, σ_f, of the distribution of values of a feature f within a class E of n samples, is related to the angle α, between the equiparameter line and the feature vector for f in the E-inverted space, by the following formula:

$$\sigma_f = \frac{r_f \sin\alpha}{\sqrt{n}},$$

where r_f is the root-mean-square of feature values within the sample.

Proof. Let V_f^e denote the value that the feature f takes for an example $e \in E$. The feature vector, $\underline{f}$, of $f = (V_{e_1}^f, V_{e_2}^f, .., V_{e_n}^f)$. Let $\underline{1}$ denote the vector (1,1,..,1).

The cosine of the angle between the equiparameter line and the feature vector can be calculated as follows.

$$\cos\alpha = \underline{f} \cdot \underline{1} \frac{1}{|\underline{f}|\ |\underline{1}|}$$

$$= \sum_{e \in E} V_f^e \frac{1}{(\sum_{e \in E} (V_f^e)^2)^{1/2} \sqrt{n}} \tag{2}$$

The variance of the sample can be calculated as follows.

$$\sigma_f^2 = \frac{n \sum_{e \in E} (V_f^e)^2 - (\sum_{e \in E} V_f^e)^2}{n(n-1)}$$

$$\approx \frac{\sum_{e \in E} (V_f^e)^2}{n} - \frac{(\sum_{e \in E} V_f^e)^2}{n^2}$$

Substituting $r = |\underline{f}|$, and cos α from (2) above,

$$\sigma_f \approx r \sqrt{\frac{1}{n} - \frac{1}{n} \cos^2\alpha}$$

$$= \frac{r \sin\alpha}{\sqrt{n}}$$

□

The second heuristic, dissimilarity between the feature values for the two classes, can be applied by testing a feature for inclusion in the positive θ-hypercube in one of the two inverted spaces, and, the negative θ-hypercube in the other, as already discussed in §3. Alternatively, one can use the sine of the angle in order to ensure that the feature vector lies in a positive orthant in one of the spaces, and in a negative orthant in the other. Thus, *distinctness can be guaranteed by requiring that the following two conditions hold*:

$$|\sin\alpha|, |\sin\beta| < \sin\frac{\pi}{4} = \frac{1}{\sqrt{2}}$$

$$\cos\alpha . \cos\beta < 0$$

where α is the angle between the feature vector and the equiparameter line in the E^+-inverted space, and β is the corresponding angle in the E^--inverted space. Other heuristic tests for hypercube-inclusion are discussed in §5.2.

4.1. Parity (XOR) Revisited.

Figure 3 shows the two inverted spaces for the 2-bit parity problem. The equiparameter lines in each of the inverted spaces are dashed; the boundaries of a (1,1)-hypercube are dotted. Notice that the primitive features score low evaluation on both the proposed heuristics. First, both the features, in both inverted spaces, are oriented perpendicular to the equiparameter line, indicating a maximum deviation of 1. Second, there is no θ such that a positive θ-hypercube includes a feature in one inverted space, and a negative θ-hypercube includes the same feature in the other. Thus, the framework captures both the high variance of the sample within a class, and the linear inseparability between classes. In §6.3, we show that the heuristic measures suggested above are helpful in discovering good features for this problem.

Having introduced the basic notion of inverted spaces, we now consider some pragmatic issues about the framework, such as the amount of information that needs to be retained.

5. Realizing the Inverted Space Framework.

Applying this framework naively can yield complex computations on large matrices. The complexity results from *simultaneous consideration* of a *large number of examples*. To counter this problem, the heuristic approximation methods of §5.1 consider only a subset of training samples that still retains the separability traits of the entire set. To address another

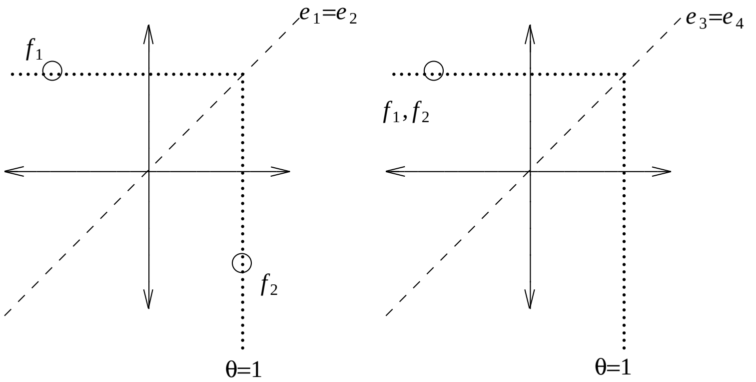


Figure 3. This figure illustrates the linear inseparability of XOR problems in the feature space defined by the features f_1 and f_2. In both the +ve and the -ve inverted spaces shown here, both the feature vectors are oriented perpendicular to the equiparameter line, and there is no θ such that a positive θ-hypercube includes a feature in one of the inverted spaces, and the negative θ-hypercube in the other.

problem, the techniques suggested in §5.2 are amenable to *incremental* learning so that heuristic information is accumulated by examining a few examples at a time.

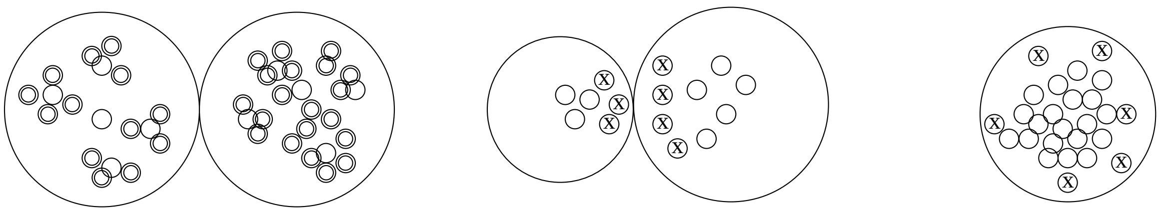
5.1. Using Low-Dimensionality Inverted Spaces.

Michalski has suggested several approaches to selection of representative samples for a two-class discrimination problem (Michalski, 1975). His ESEL system uses three criteria (as shown in Figure 4):

The *Cluster Centroid (CC) method*. Each class is represented by its centroid and centroids of clusters within the class. In addition, one keeps additional points that are more than twice the cluster standard deviation away from the centroid of every cluster within the class. Sklansky et al. use a similar approach (Faroutan, 1987). Rendell's Probabilistic Learning System (Rendell, 1986) also constructs such representations.

The *Near Miss (NM) method*. Choose from a class those points that are nearest to the opposite class. Similar techniques have been explored by Winston (1975) and MacGregor (1988). The idea is that any separating surface for the two subsets consisting only of near-misses would still separate the two classes.

The *Outstanding Representative (OR) method*. Choose the subset of examples from each class whose convex hull is the same as that for the entire class. This means choosing the extremal points of each class. Any surface that separates the extremal points must, of necessity, separate the two classes. Lambert (Lambert, 1969) has proposed similar techniques.



CC method NM method OR Method

Figure 4. Methods for selecting representative examples

Several researchers have recently noticed improved performance in learning systems that pay special attention to boundary cases (Ahmad, 1988, Kohonen, 1988). On the theoretical side, Cover (1965) has suggested that for two-class discrimination problems using surfaces with *d* degrees of freedom, on the average, 2*d* samples can capture the information of a possibly infinite training set. The implication of Cover's result, and the existence of mechanisms for finding boundary patterns, make the inverted-space framework a viable technique.

5.2. Using Approximate Tests of Separability

In our proof of the theorem, we showed a relationship between the sine of the angle between the feature vector and the equiparameter line, and the standard deviation of the value distribution for a feature. Considering that the only information needed to calculate the latter is the sum of sample values, the sum of their squares, and the number of values seen, suggests incrementally maintaining the information required to calculate the sine of the angle.

The following tests implement a cheaper heuristic approach to testing feature points for inclusion in θ-hypercubes:

Letting $|f|_E$ denote the length of the feature vector for *f* in the inverted space for *E*, and, *l* (resp., *m*) the dimensionality of E^- (resp., E^+),

$$|\underline{f}|_{E^+} < \theta,\ |\underline{f}|_{E^-} > \theta\sqrt{l} \rightarrow \textit{separability},$$

$$|\underline{f}|_{E^+} > \theta\sqrt{m},\ |\underline{f}|_{E^-} < \theta \rightarrow \textit{separability}.$$

These tests do not cover all the cases. However, the advantage over exact tests for inclusion is that only two items of information per feature need to be maintained, and those can be computed incrementally. The heuristic provides a positive answer to a query about inclusion if the feature vector lies within the largest hypersphere that can be inscribed in the hypercube in question, and a negative answer if it lies outside the smallest hypersphere that can circumscribe the hypercube. We do not detail these methods here, primarily because there are several ways of approximating the real measures of goodness, and these are only two of them.

6. Constructive Induction Using Inverted Spaces.

A constructive induction problem is defined by a set of *feature-construction operators* to be applied to the primitive features, or to combinations thereof. Table 1 shows the geometric interpretation of some commonly used constructive induction operators in terms of the inverted space framework. Given a difficult discrimination learning problem, and the description of features in the form of inverted spaces for both the classes, the goal of constructive induction is to create features that (i) lie close to the equiparameter line in both the inverted spaces, and (ii) satisfy the inclusion tests for some θ-hypercube. In the following, we examine how one might apply the inverted-space framework to construct new features.

Knowing which transformations the various construction operators achieve lets one select the particular operator to apply. Additional heuristics, such as using Boolean operators on Boolean features, can provide additional bias for selection. The constructive induction algorithm employed is a search in the space of constructed attributes. The inverted space framework is useful in both the generation and the testing phases of this algorithm.

Table 1

Geometric Interpretation of CI Operators

Constructor	**Inverted Space Operation**
Linear Combination	Interpolation/Extrapolation
Thresholding	Hypercube Membership
AND	Minimizing each Dimension
OR	Maximizing each Dimension
NOT	Reflection about the Origin
Scalar Addition	Translation
Scalar Multiplication	Scaling

6.1. Feature Generation

This employs several heuristics based on the measures of feature goodness discussed in §4. One heuristic uses a linear combinations of d=max(l,m) features, where $l=|E^+|$, and $m=|E^-|$. A (d−1)-dimensional surface passing through these features is made to intersect the equiparameter line in one of the spaces. The intersection yields the exact values of weights to be applied in the linear combination. The new feature is evaluated according to the measures of goodness.

Yet another heuristic limits the application of thresholding only to features close to the equiparameter line. Constructors such as AND and NOT are applied only to Boolean features. Sometimes, several features map into the same point in one of the inverted spaces, but to different points in the other. In such cases, one can apply operators such as AND, OR, difference, and equality, in order to construct new features.

In general, this phase employs substantial domain-knowledge. Operators specific to an application domain can still be interpreted in the inverted spaces, and this knowledge of transformations can be used (abductively) for feature generation.

6.2. Feature Evaluation

Features are evaluated on the basis of the goodness criteria introduced in §4. Different constructive induction algorithms employ different search mechanisms, but all can use this measure of goodness to serve as a heuristic estimate. Usually, the cheaper version of the inclusion/exclusion test will suffice. The more expensive test involving all the dimensions of the inverted space may be limited only to promising features (those that lie close to the equiparameter line). Other techniques, such as dynamic bias management, may be employed to focus the search.

In the following, we illustrate the effect of such constructions on the XOR problem. Recall from §4.1 that this problem *requires* constructive induction.

6.3. Parity Revisited

Figure 5 shows the inverted spaces for the parity problem. It also shows the map of a new feature, $f_1 \Lambda f_2$. This new feature is recommended by the feature-generation heuristic for the AND operator. The linear combination of f_1 and f_2 yields $0.5f_1+0.5f_2$ upon intersection with the equiparameter line in the inverted space. Still, no single feature scores highly on the evaluation criteria. Further construction suggests taking the difference of the two constructed features, thus yielding the final constructed feature $0.5f_1+0.5f_2-f_1 \Lambda f_2$ which separates the two classes.

6.4. Constructive Induction as Merging Peaks

It has been suggested that hard problems in concept-learning require the formation of disjuncts. This is supported by recent results in computational learning theory (Kearns, 1987). Rendell (1988) has suggested that membership functions of

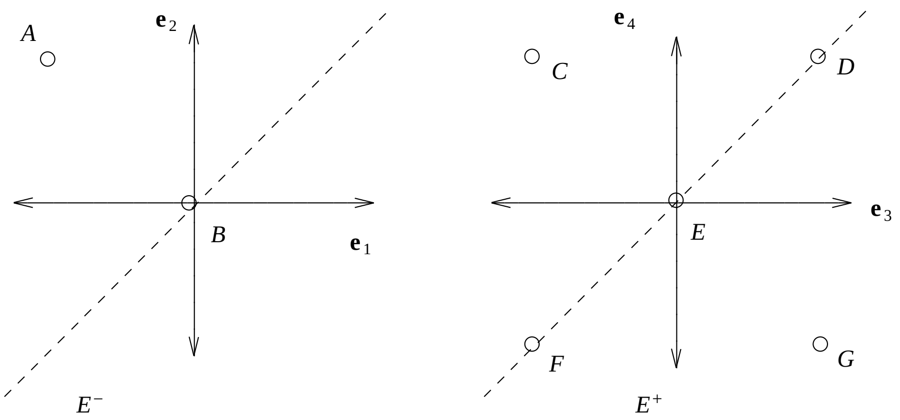


Figure 5. The negative inverted space is shown on the left and the positive inverted space on the right. f_1 maps to points A and C, f_2 maps to points A and G, $f_1 \Lambda f_2$ maps to A and F, $0.5f_1+0.5f_2$ maps to A and E, and $0.5f_1+0.5f_2-f_1 \Lambda f_2$ maps into B and D, respectively. Notice that both B and D lie on their respective equiparameter lines.

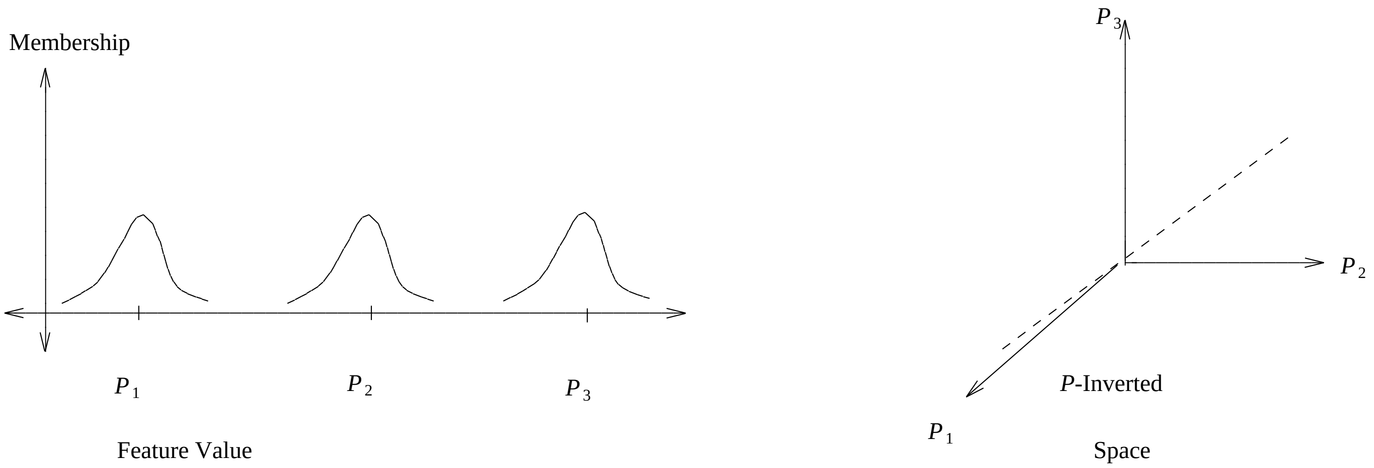


Figure 6. Constructive induction as merging of peaks.

Representative examples from the concept function in feature space (left) are used to construct a low-dimensionality inverted space (right). Features that lie on the equiparameter line (shown dashed) in the inverted space can be seen as merging peaks, provided these features are constructed by applying continuous, compact transformations to the original features.

hard concepts have multiple peaks in the primitive feature space. Simple surfaces cannot be used to discriminate between classes whose membership functions have multiple peaks. Rendell (1988) advocates transforming the feature spaces so that the various peaks *merge*. In the transformed space, the membership function is smooth, and simple surfaces can discriminate.

Let F be the set of primitive features of a disjunctive problem, and let P=(p^1,p^2,..,p^m) be the centroids of the peaks of the membership function. (See Figure 6) Using the reduced-dimension inverted spaces discussed in §5.1, one can construct the P-inverted space, and map into it the features of F. If a feature receives a high evaluation in the P-inverted space, it must lie close to the equiparameter line. In other words, it takes on nearly identical values for the examples corresponding to peaks in the membership function. Now, if such a feature is constructed using a nonlinear, continuous mapping, it can be shown to merge the peaks of the membership function in the feature space. Similarly, simultaneous construction in the inverted space for negative examples, and the application of the hypercube-inclusion test, can result in the construction of features that transform complex learning problems into simple ones.

6.5. Feature Construction in Inverted Spaces.

The inverted space framework is particularly suitable for constructive induction because the inverted spaces remain fixed during construction. This is in sharp contrast to the traditional feature space representation used to study pattern recognition operations, where new features contribute new dimensions. The two major ideas that distinguish our work from that of Watanabe (1969) are the concept of the equiparameter line, and the angle that a feature vector makes with this line. In our recent research, we have started exploring the inverted space representation of value-coded features (aka. overlapping localized receptive fields, coarse-coded distributed representations). The concepts of angle and distance in the feature space relate, respectively, to generalization and discrimination — two conflicting goals in the design of classifiers.

Besides the obvious application of constructed features in classification, one of the authors has also explored the application of constructed variables in numeric optimization problems (Lowrie and Wah, 1988). There is no reason to believe that inverted space analysis should be restricted to classification problems. In the future, we plan to explore inverted space representations of variables in optimization problems.

7. Conclusions and Future Work.

We have presented a new framework for representing classes, examples, and features. It is applied to the constructive induction problem. Principles for evaluating newly-constructed features are developed, and are shown to guide the construction process. The exact measure for *compactness* is shown to have a direct relationship with a statistical measure of the spread of a distribution. Several inexpensive heuristics for feature evaluation arise from relaxation of measurement accuracy.

Our analysis covers a variety of algorithms and heuristics for construction and evaluation of features. In particular, it synthesizes techniques for selection of examples with those for selection and construction of features. By posing the dual of the classification learning problem, it suggests new techniques similar to the primal-dual algorithm (Papadimitriou, 1982) for linear programming. For example, one might use geometric techniques such as hull-finding and proximity analysis (Preparata, 1985) in order to discover representative examples in the feature space, while constructing good features in the inverted space of these examples, repeating the process on the reduced feature space. The inverted-space representation, therefore, provides a rich basis for such integration of geometric and statistical techniques.

The framework is important as an analytical tool. It allows complex nonlinear operations on features. We have demonstrated its ability to represent a variety of construction and classification operations. In future, we would like to extend the analysis to include other problems in classification. One such problem is redundancy-elimination. Inverted space permits the use of geometric tests, such as collinearity, and coplanarity, for detection of dependencies between features. We plan to study the statistical properties of geometric operations, and the geometric interpretation of statistical operations, in order to arrive at a synthesis of techniques for problems like dimensionality reduction, clustering, and multi-class discrimination (Duda, 1973).

Acknowledgements.

Darrell Hougen suggested the proof technique for interpreting the method of §4 statistically. The authors wish to thank Mark Gooley, Subutai Ahmad, and the anonymous referees for their helpful comments. David Aha of Univ. of California, Irvine, and Giulia Pagallo of University of California, Santa Cruz, pointed out a few oversights in our IJCAI paper, prompting this revision.

References

[Ahmad,1988] S. Ahmad, ‘‘Scaling and Generalization in Neural Networks: a Case Study,’’ in *Proc. 1988 Connectionist Models Summer School*, ed., D. Touretzky, G. E. Hinton, T. J. Sejnowski. Palo Alto, CA: Morgan Kaufmann, 1988.

[Cover, 1965] T. M. Cover, ‘‘Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition,’’ *IEEE Trans. on Elec. Computers*, vol. 14, pp. 326-334, June 1965.

[Dietterich *et al.*, 1982] T. G. Dietterich, B. London, K. Clarkson, and G. Dromey, ‘‘Learning and Inductive Inference,’’ in *The Handbook of Artificial Intelligence*, ed., P. R. Cohen and E. A. Feigenbaum. Kaufmann, 1982.

[Duda *et al.*, 1973] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[Faroutan *et al.*, 1987] I. Faroutan and J. Sklansky, ‘‘Feature Selection for Automatic Classification of Non-Gaussian Data,’’ *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-17, 1987.

[Kearns *et al.*, 1987] M. Kearns, M. Li, L. Pitt, and L. G. Valiant, ‘‘Recent Results on Boolean Concept Learning,’’ *Proc. Fourth Int’l. Workshop on Machine Learning*, Morgan Kaufmann, 1987.

[Kohonen *et al.*, 1988] T. Kohonen, G. Barna, and R. Chrisley, ‘‘Statistical Pattern Recognition with Neural Networks: Benchmarking Studies,’’ *Proc. Second Int’l. Conf. Neural Networks*, pp. I-61-I-68, IEEE, 1988.

[Lambert, 1969] P. F. Lambert, ‘‘Designing Pattern Categorizers with Extremal Paradigm Information,’’ in *Methodologies of Pattern Recognition*, ed., S. Watanabe. Academic Press, 1969.

[Lowrie and Wah, 1988] M. Lowrie and B. W. Wah, ‘‘Learning Heuristic Functions for Numeric optimization Problems,’’ *Proc. COMPSAC 88*, pp. 443-450, IEEE, 1988.

[MacGregor, 1988] J. N. MacGregor, ‘‘The Effects of Order on Learning Classifications by Example: Heuristics for Finding the Optimal Order,’’ *Artificial Intelligence*, vol. 34, pp. 361-370, North-Holland, 1988.

[Michalski, 1975] R. S. Michalski, ‘‘On the Selection of Representative Samples from Large Relational Tables for Inductive Inference,’’ *U. Illinois (Chicago circle) tech. report*, No. M.D.C 1.1.9, July, 1975.

[Minsky *et al.*, 1969] M. Minsky and S. Papert, ‘‘Perceptrons: An Introduction to Computational Geometry.’’ Cambridge, Massachusetts: MIT Press, 1969.

[Muggleton *et al.***, 1988]** S. Muggleton and W. Buntine, ''Constructive Induction in First-order Logic,'' *Proc. Workshop on Change of Representation and Bias*, 1988.

[Papadimitriou *et al.***, 1982]** C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[Preparata *et al.***, 1985]** F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[Rendell, 1986] L.A. Rendell, ''A Framework for Induction and a Study of Selective Induction,'' *Machine Learning*, vol. 1, Kluwer academic Press, June 1986.

[Rendell, 1988] L. A. Rendell, ''Learning Hard Concepts,'' *Proc. European Workshop in Learning (EWSL-88)*, 1988.

[Rumelhart *et al.***, 1986]** D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning internal Representations By Error Propagation,'' in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds., D. E. Rumelhart, J. L McClelland and the PDP Research Group, vol. 1, pp. 318-362, Cambridge, MA: MIT Press, 1986.

[Watanabe, 1969] S. Watanabe, ''Object-Predicate Reciprocity and its Application to Pattern Recognition,'' *Proc. Information Processing 68*, pp. 1608-1613, Amsterdam: North-Holland, 1969.

[Watanabe, 1985] S. Watanabe, *Pattern Recognition: Human and Mechanical*, Wiley Interscience, 1985.

[Winston, 1975] P. H. Winston, ''Learning Structural Descriptions from Examples,'' in *The Psychology of Computer Vision*, ed., P. H. Winston, New York, NY: McGraw-Hill, 1975.