Artificial Intelligence 15-381
Machine Learning and Decision Trees

Jaime Carbonell
30-October-2001

Representations for Decision Making

Linear separators

- One-dimensional vs N-dimensional
- Encoded as single-premise rule

Generalization to convex-hull

- Concept bounded by hyper-planes
- Encoded as conjunctive rule

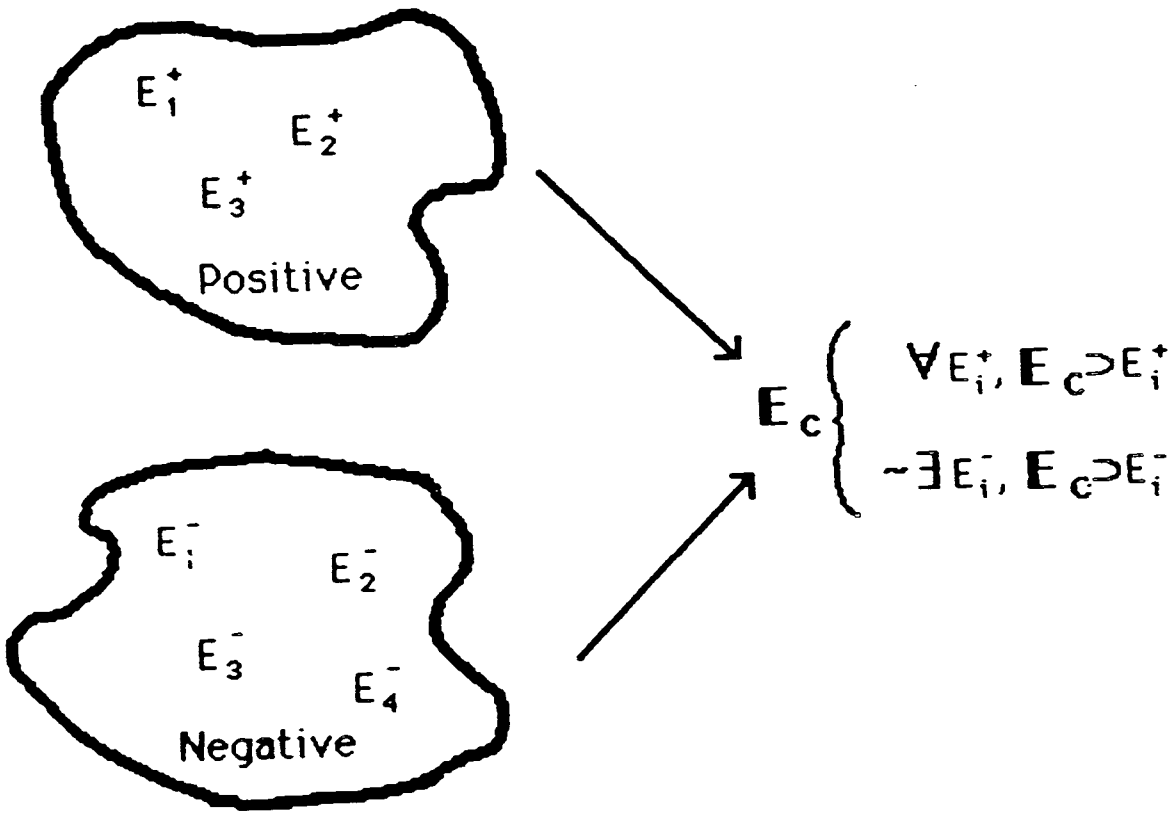
Generalization to multiple convex-hulls

- Disjunctive Concepts
- Encoded as multiple conjunctive rules or as decision trees

Evaluation Functions

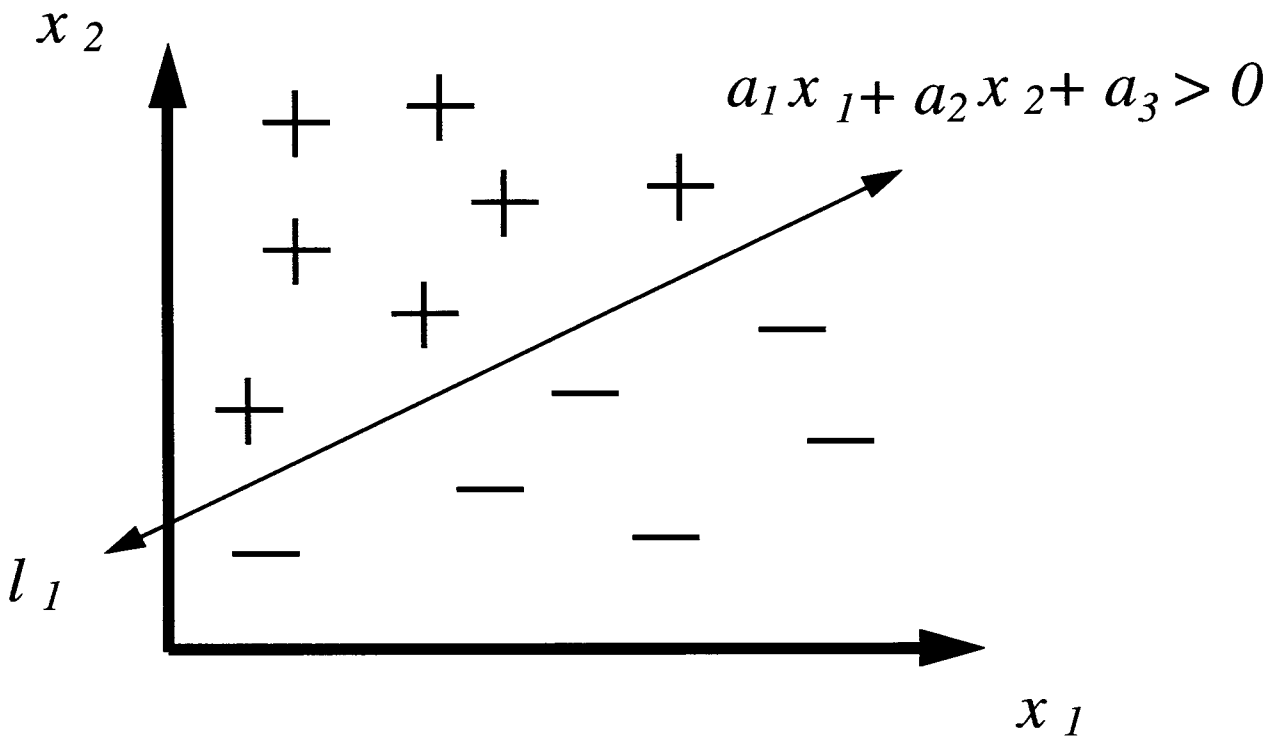
- Linear weighted sums
- Context-sensitive cross-terms

LEARNING FROM EXAMPLES.



- Incremental vs One - Shot
- Positive examples with bounded generalization
  vs positive + negative examples
- Near - miss analysis
  $\delta(E^-_{NS}, E_c) \ll \delta(E^-_{AVE}, E_c)$
- "Best guess" vs "version space" notion
- Generalization vs discrimination
- Internal vs external sample generation
- Descriptors
  - finite valued
  - continuum valued
  - hierarchical
- Conjunctive vs disjunctive generalization

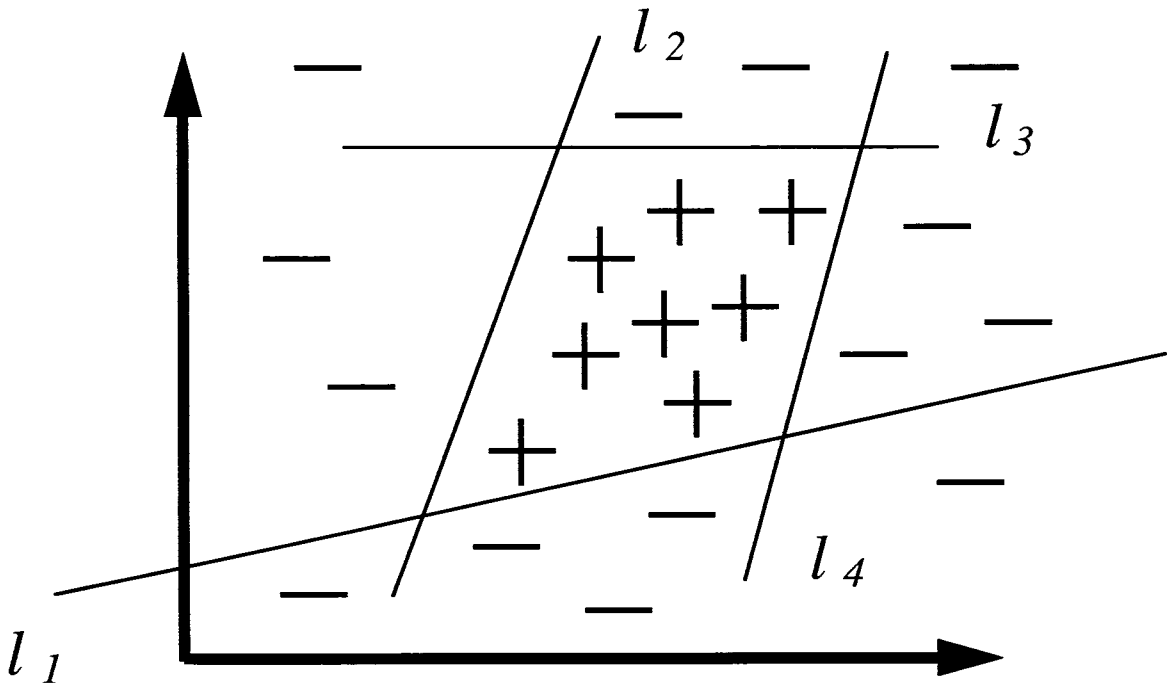
Linear Separators



2D (two attributes)
No Noise

If $l_1 > 0$ Then "YES"

Convex Hull Classifier



2D (two attributes)
No Noise

$$\text{If } \left\{ \begin{array}{l} l_1 > 0 \\ \& \; l_2 > 0 \\ \& \; l_3 > 0 \\ \& \; l_4 > 0 \end{array} \right\} \text{ Then "YES"}$$

Approximate Concepts

Types of noise in data

- Classification noise
- Boundary noise
- Systematic errors

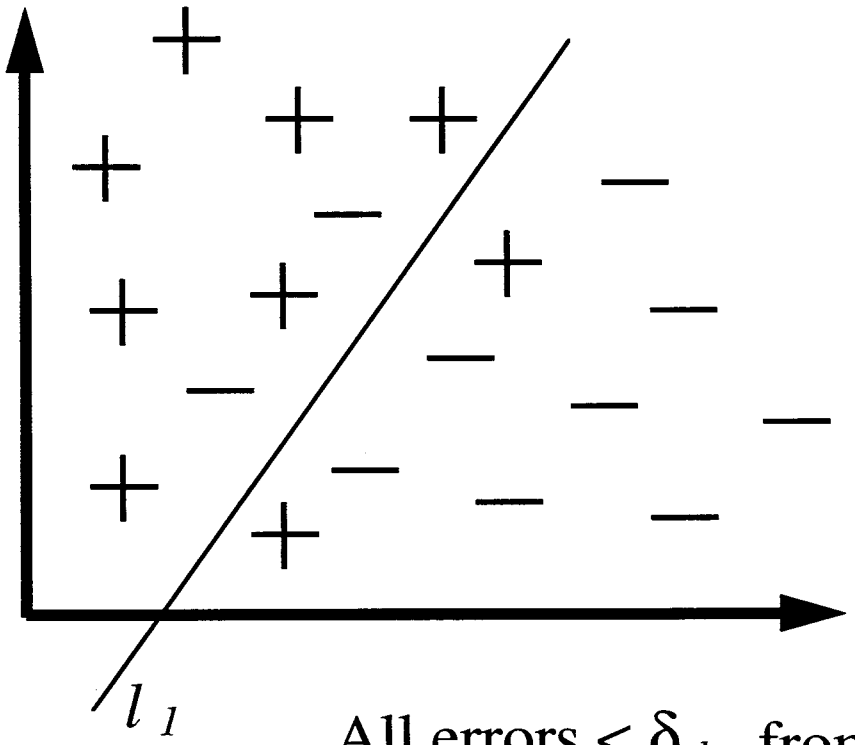
Error Functions

- Equi-weight (L0 norm)
- Distance-weighted (L1 norm)
- Least Squares (L2 norm)
- Zero-tolerance (L-infinity norm)

Trade-offs

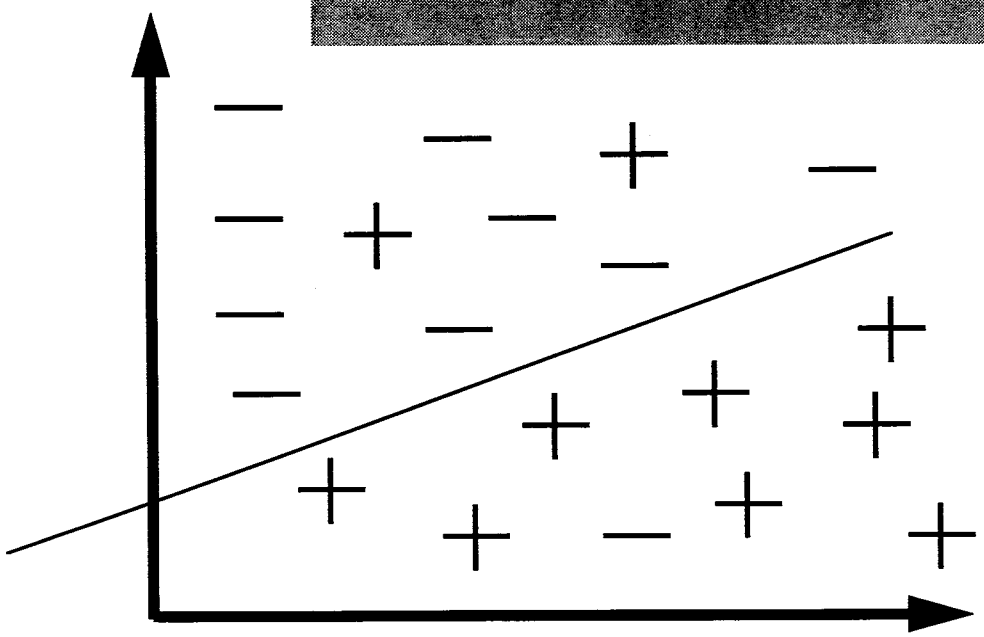
- Concept simplicity vs accuracy
- Overtraining on data (more later)
- Convergence time vs decision-rule form

Boundary Noise



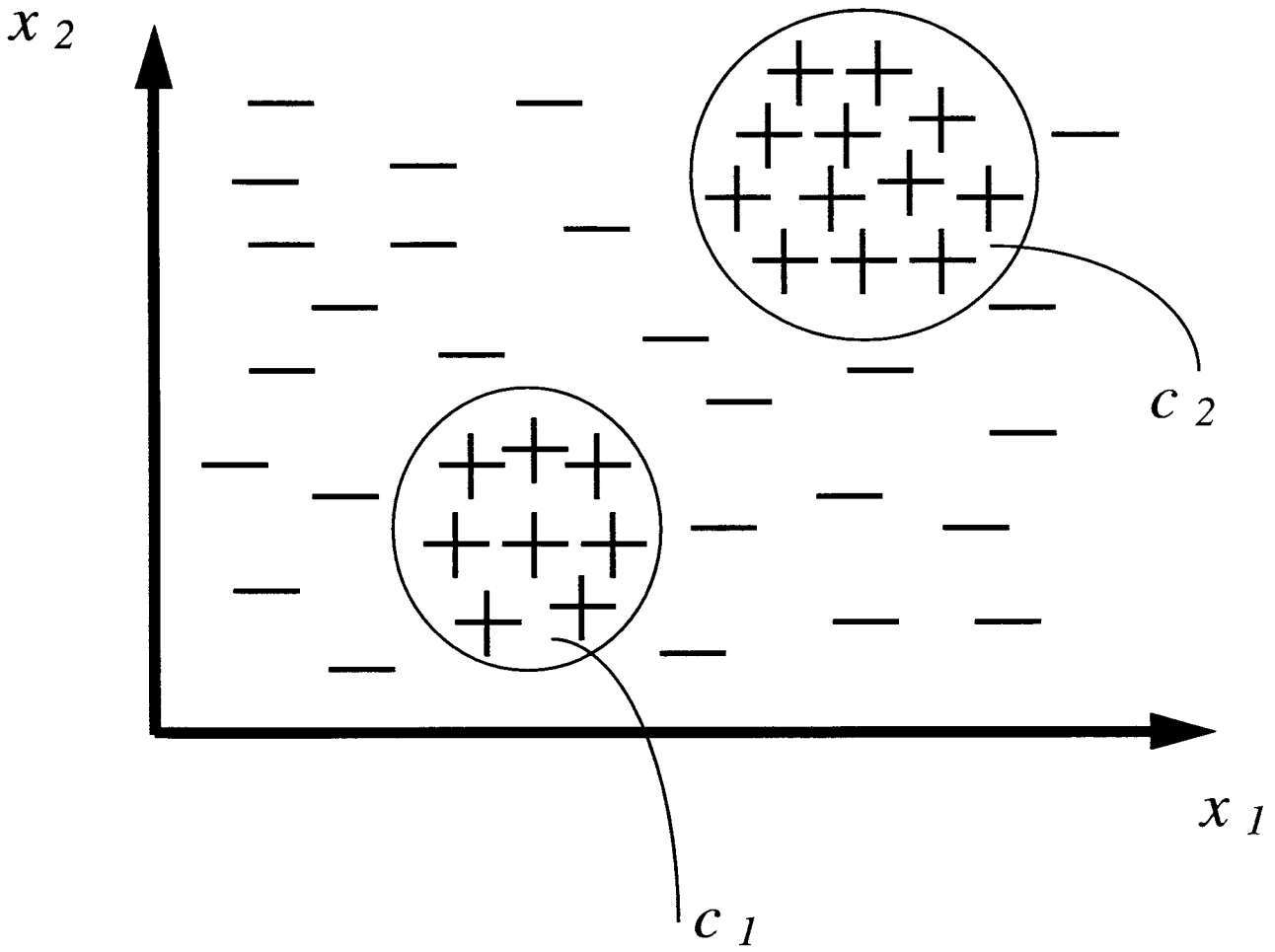
All errors < δ_{l1} from boundary

Random Noise



No constraint on loction of errors

Disjunctive Concepts



For instance "most profitable cutomer"

is c_1 : pays interest always on rotating balance

<u>or</u> c_2 : transacts > $5000/month

MACHINE LEARNING TECHNIQUES

WHAT THEY ARE

- Targeted induction of patterns from data
- Or, knowledge-compilation (speed-up learning)
- Numeric data, symbolic data, or (typically) both
- Produce decision rules, trees, etc.
- When succeed -- produce idiot-savant systems

APPLICABILITY CRITERIA:

- Large-enough volumes of training data (typically 10^3 to 10^6 records)
- Well-defined objective function (e.g. what constitutes fraud, for detection)
- Absence of near-optimal human expertise (else, expert-system approach is often better)
- Absence of efficient pure-mathematical methods* (such as linear regression, which are easier)
- Induction tools, such as C4.5 or NNets
- Person with skills in induction methods, tools (and preferably statistics and expert systems)

Machine Learning:
INDUCTIVE Techniques

OBJECTIVE:

Find, categorize, and exploit regularities in large volumes of potentially noisy numerical and symbolic data.

TECHNIQUES:

- *Concept Formation*
  (Version spaces, star, ILP, ...)
- *Decision-Tree Induction*
  (ID3, C4.5, CART, ...)
- *Neural Networks*
  (Backprop, Recurrent, Hopfield, ...)
- *Analogical Generalization*
  (NNeighbor, CBR, Derivational Analogy, ...)
- *Numerical Optimization*
  (Statistical, Reinforcement Learning, ...)

Machine Learning: Selected Successful Applications

Credit Card Fraud Recovery

- Decision Tree Induction: DB Mining
- Going operational in major bank.
- [Alt Tech: NNets, rules]

Autonomous Land Vehicle

- Knowledge-guided NNet
- Drove across USA 98.2% autonomously

SPHINX and JANUS

- HMMs, NNets, Lang Mod's
- Best performance (ARPA, Verbmobil)

Industrial AUTONs

- Reinforcement Learning, opt.
- Production efficiency for continuous manufacturing applications

Decision Tree Induction

What are Decision Trees?

- Disjunctive concept classifiers
- Binary or N-ary class membership
- Hyper-rectangle approximators (usually)

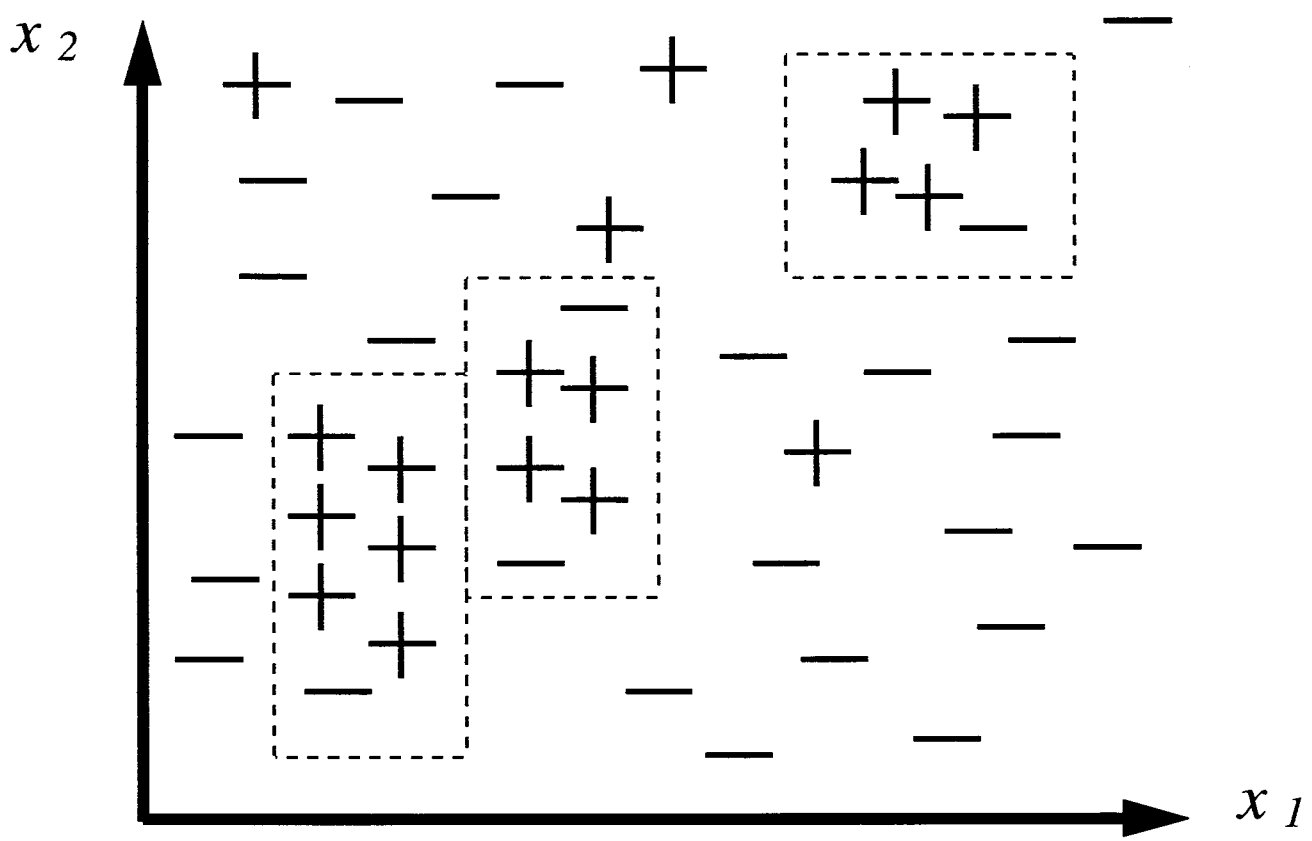
What Decision Trees do well

- Can weed out irrelevant attributes
- Noise-tolerant
- Human readable
- Capable of over-training compensation (pruning)

What Decision Trees do NOT do

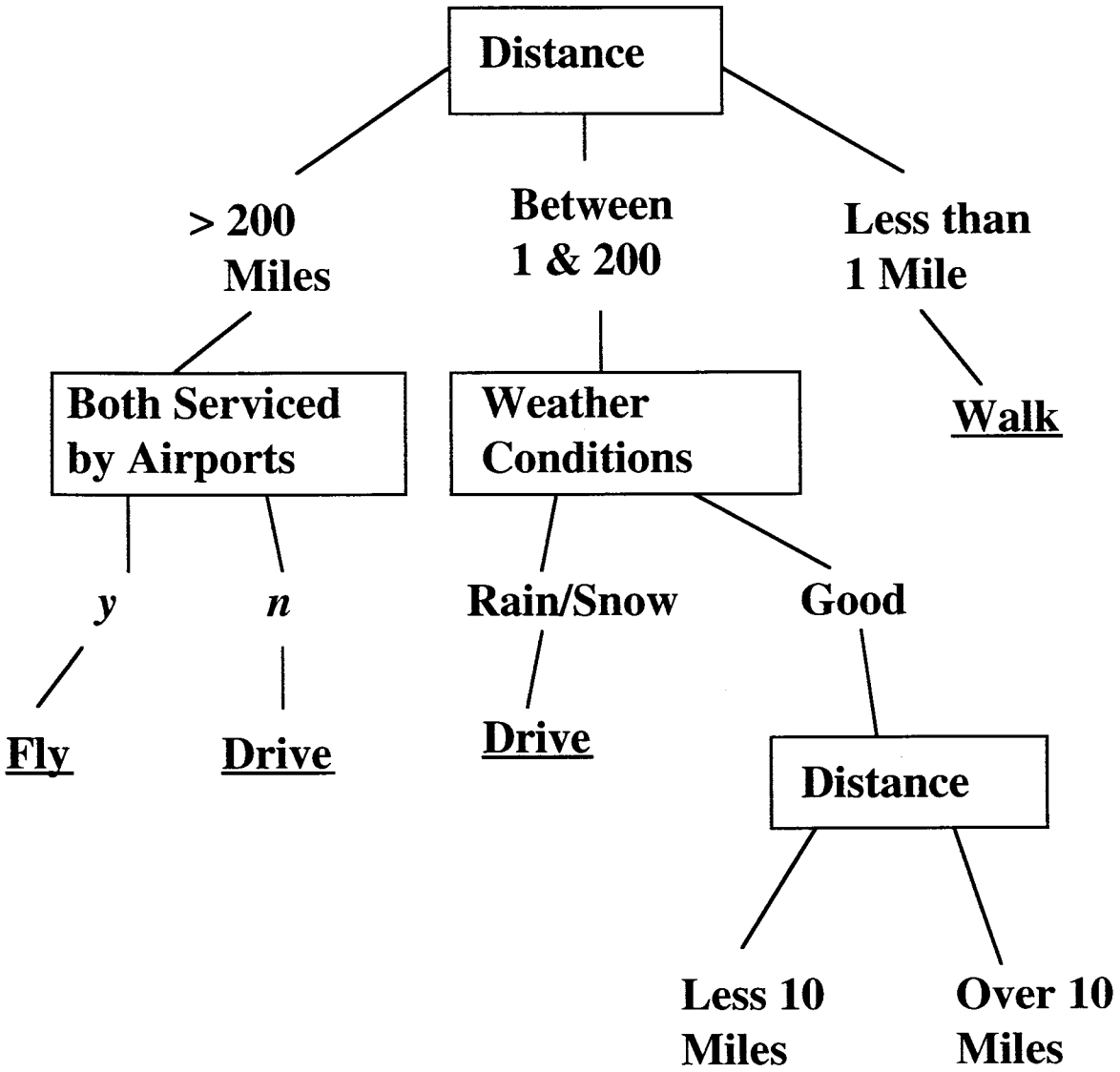
- Derived attributes (up to knowledge engineer)
- Find patterns in uncategorized training data (a.k.a. "unsupervised learning" *a la* clustering)

Rectangular Approximations of Disjuncture Decision Surfaces



This is what Decision Trees
typically compute (but can be generalized)

Decision Tree to Select Mode of Transportation



Note that outcomes (e.g. "Drive") and tests (e.g. "Distance") can recurr.

Information Theory Basics

Let S = collection of classified examples, such as credit-card applints classified as to whether FUSA will or will not accept them

P_+ = Proportion of S accepted
P_- = Proportion of S rejected

Define Entropy (s) = $- P_+ \log_2 P_+ - P_1 \log_2 P_-$

Entropy is a measure of the uncertainty in S.
For example, if $p_+ = 1$ & $P_- = 0$, Entropy = 0
or if $P_+ = 0$ & $P_- = 1$, Entrpoy = 0
Because there is no uncertainty in S.

However, if $P_+ = P_- = .5$, Entropy = 1
Because one bit of information is required for each element of S to eliminate its class-member uncertainty.

Information Theory, cont'd.

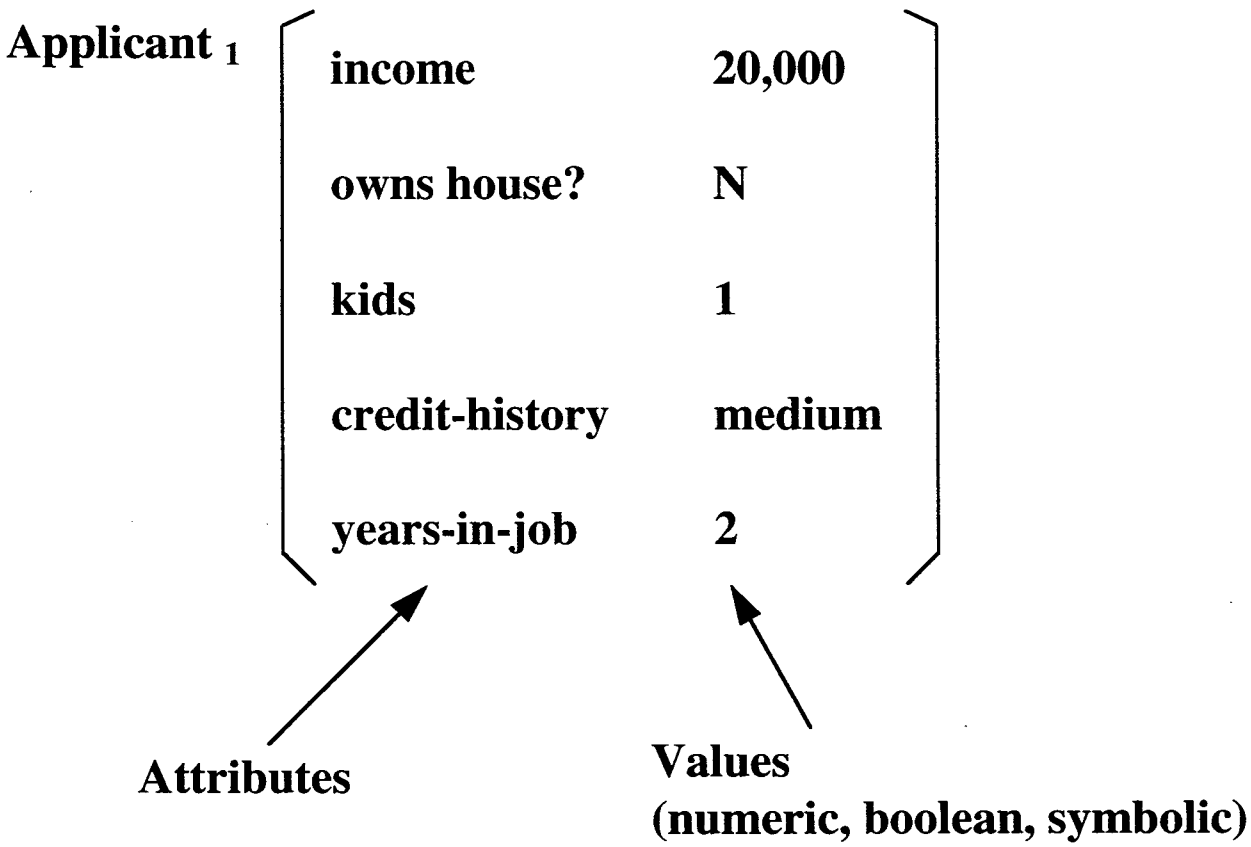
If there are *n* classes (rather than just "good" and "bad" credit risks), then, in general:

$$\text{Entropy}(s) = \sum_{i=1}^{n} - p_i \log_2 p_i$$

Goal of a classifier is to minimize entropy of a collection of examples, i.e. to predict their class with maximal accuracy (= minimal uncertainty.)

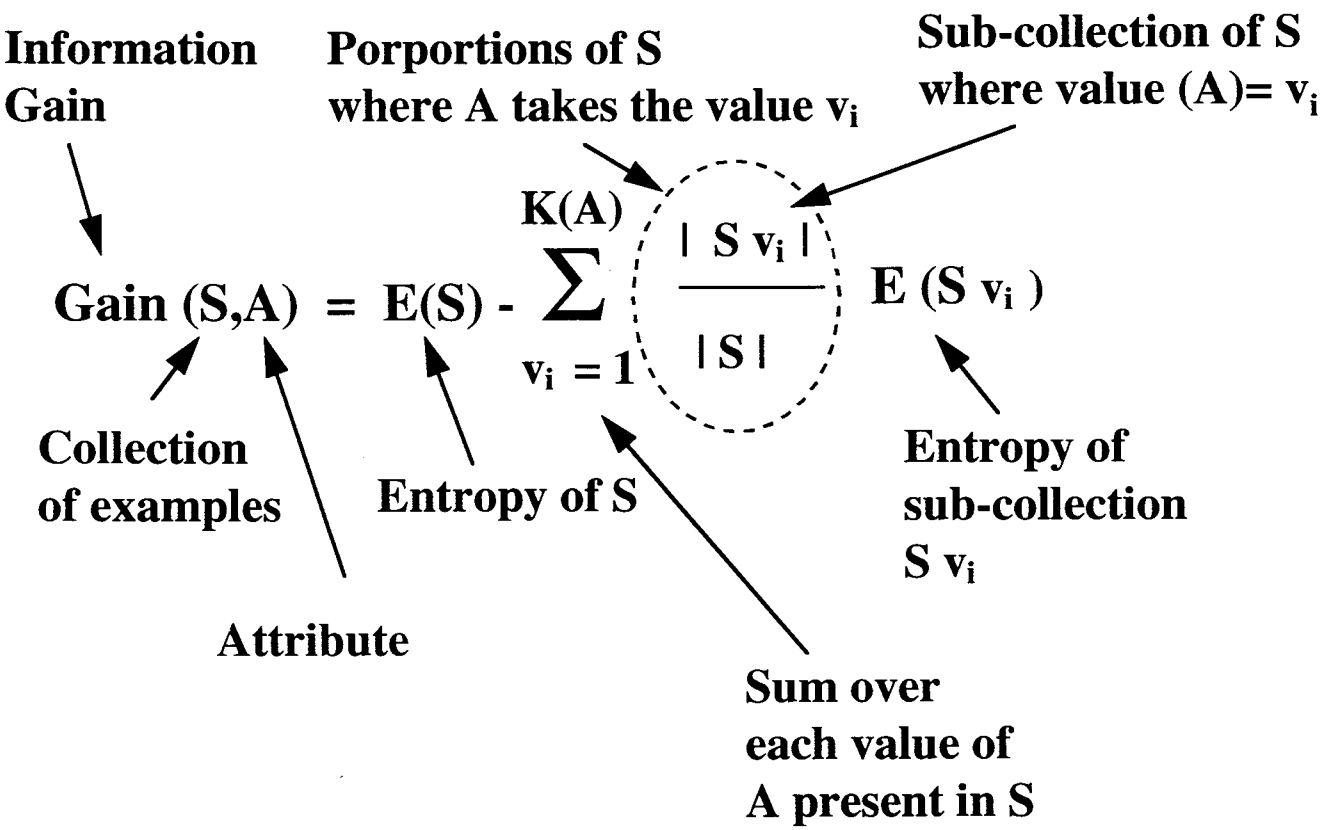
Information Gain Metric

Let each example ε S be represented as a set of atributes and values, e.g.



Question: Given a set {Applicant$_i$}, which attribute best predicts creditworthiness? (i.e., which reduces entropy the most?)

Information Gain, cont'd.



Goal Choose A among $\{A_i\}$ such that Gain (S,A) is maximized, for collection S.

Simplified D-Tree Induction Algorithm

1. E(S) < MinEntropy
   output largest-class-in-S & Halt

2. Let A = Argmax Gain (A_j , S)
   $A_j \varepsilon$ **A**

3. Call D-Tree recursively on each value of A in S
   D-Tree ($S_{Aj = Vi}$)
   & D-Tree ($S_{A = V2}$)
   . . .
4. Assemble results of 3 creating a new node spitting on values of A & return D-Tree

Mutual Information

Useful in:

- A feature space: $f_1, f_2, ..., f_n$
- Training instances: n-vectors of feature values
- Find predictive correlation among features

Intuitive definition

The degree to which features f_i and f_j mutually predict each other -- measured as number of bits of additional information in f_i, given f_j)

Naive formulation [Fano, Church]

$$MI(f_i, f_j) = \log_2 \frac{P(f_i, f_j)}{P(f_i)P(f_j)}$$

Since: $P(f_i, f_j) = P(f_i)P(f_j|f_i)$

$$MI = \log_2 \frac{P(f_i)P(f_j|f_i)}{P(f_i)P(f_j)} = \log_2 \frac{P(f_j|f_i)}{P(f_j)}$$

Mutual Information [Cont.]

Complete formulation [Michie, ...]

$$MI(f_i, f_j) = \sum_{Val(f_i, f_j)} P(f_i, f_j) \log_2 \frac{P(f_i, f_j)}{P(f_i)P(f_j)}$$

Relation to Information Gain

- Let C be just one feature in vector: f_j
- Generalize: use all-but-jth feature to predict jth-feature
- Generalize to use m-of-n features to predict remaining features

Data for Inducing Creditworthiness in New Card Applications

Acct. numb.	Income in K/yr	Job Now?	Delinq accts	Max cycles	Owns home?	Credit years	Final disp.
1001	25	Y	1	1	N	2	Y
1002	60	Y	3	2	Y	5	N
1003	?	N	0	0	N	2	N
1004	52	Y	1	2	N	9	Y
1005	75	Y	1	6	Y	3	Y
1006	29	Y	2	1	Y	1	N
1007	48	Y	6	4	Y	8	N
1008	80	Y	0	0	Y	0	Y
1009	31	Y	1	1	N	1	Y
1011	45	Y	?	0	?	7	Y
1012	59	?	2	4	N	2	N
1013	10	N	1	1	N	3	N
1014	51	Y	1	3	Y	1	Y
1015	65	N	1	2	N	8	Y
1016	20	N	0	0	N	0	N
1017	55	Y	2	3	N	2	N
1018	40	N	0	0	Y	1	Y
1019	80	Y	1	1	Y	0	Y
1021	18	Y	0	0	N	4	Y
1022	53	Y	3	2	Y	5	N
1023	0	N	1	1	Y	3	N
1024	90	N	1	3	Y	1	Y
1025	51	Y	1	2	N	7	Y
1026	20	N	4	1	N	1	N
1027	32	Y	2	2	N	2	N
1028	40	Y	1	1	Y	1	Y
1029	31	Y	0	0	N	1	Y
1031	45	Y	2	1	Y	4	Y
1032	90	?	3	4	?	?	N
1033	30	N	2	1	Y	2	N
1034	88	Y	1	2	Y	5	Y
1035	65	Y	1	4	N	5	Y
1036	12	N	1	1	N	1	N
1037	28	Y	3	3	Y	2	N
1038	66	?	0	0	?	?	Y
1039	50	Y	2	1	Y	1	Y
1041	?	Y	0	0	Y	8	Y
1042	51	N	3	4	Y	2	N
1043	20	N	0	0	N	2	N
1044	80	Y	1	3	Y	7	Y
1045	51	Y	1	2	N	4	Y
1046	22	?	?	?	N	0	N
1047	39	Y	3	2	?	4	N
1048	70	Y	0	0	?	1	Y
1049	40	Y	1	1	Y	1	Y