Unsupervised Learning

For an *unsupervised learning* problem, we do not focus on prediction of any particular thing, but rather try to find interesting aspects of the data.

Finding associations: We try to find combinations of values for variables that occur especially often.

Example: An unusually large number of people buy both vodka and caviar.

Clustering: We try to find groups of cases that are similar.

Example: Clusters of patients with similar symptoms are called "diseases".

Dimensionality reduction: We try to map the variables to a smaller number of variables that retain much of the important information. (PCA is an example.)

Example: The "inflation rate" captures much of the information present in the price increases for many commodities.

Probability density estimation: We try to learn the probability distribution of all the variables. This can be used for solving many problems.

These formulations are related. Many of them involve *latent variables*, which identify clusters or correspond to low-dimensional representations of the data.

Pros and Cons of Unsupervised Learning

Let's compare the "unsupervised" task of estimating $P(x_1, x_2, \ldots, x_n)$ with the "supervised" task of estimating $P(x_1 \,|\, x_2, \ldots, x_n)$.

From $P(x_1, x_2, \ldots, x_n)$ we can get $P(x_1 \,|\, x_2, \ldots, x_n)$, and also $P(x_n \,|\, x_1, \ldots, x_{n-1})$ and many other useful things.

But estimating $P(x_1, x_2, \ldots, x_n)$ may be much more difficult than estimating $P(x_1 \,|\, x_2, \ldots, x_n)$. Perhaps the effort to do unsupervised learning isn't useful if we're only interested in $P(x_1 \,|\, x_2, \ldots, x_n)$.

But even if we're interested only in predicting x_1, we might sometimes need things like $P(x_1 \,|\, x_2)$, if only x_2 is observed.

And we may have much more data for unsupervised learning than for supervised learning.

Consider a baby learning to recognize objects. They get enormous amounts of visual input. Only occasionally does someone tell them, "that's a cup". Most of what babies learn is unsupervised. They likely know that there are cups before they know the word "cup".

Association Rules

A popular "data mining" procedure is to find *association rules* from a large data base of information.

The data consists of the values for p variables in N cases, where p may be big (eg, 1000) and N may be very big (eg, 1,000,000). There may be many missing data values.

It is convenient to recode all variables as binary — eg, "education", with values of "<HS", "HS", "UG", "GS", would be coded as four binary variables. Exactly one of these variables will be 1, the others 0, unless "education" is missing, in which case they are all 0.

We'll call these binary variables (derived form all the original variables) Z_k, for $k = 1, \ldots, K$.

Our first task is now to find all subsets, $\mathcal{K}$, of $\{1, \ldots, K\}$ that occur with probability greater than some threshold t:

$$\Pr\left(Z_k = 1 \text{ for all } k \in \mathcal{K} \right) \quad > \quad t$$

The event above can also be written as $\prod_{k \in \mathcal{K}} Z_k = 1$.

Finding Association Rules

We estimate the probabilities by frequency in the data base, so we search for all "item sets" $\mathcal{K}$ (subsets of $\{1, \ldots, K\}$) for which the "support" satisifies

$$T(\mathcal{K}) \;=\; \frac{1}{N} \sum_{i=1}^{N} \prod_{k \in \mathcal{K}} z_{ik} \;>\; t$$

This can be done efficiently provided that t is big enough that only a small fraction of the possible subsets meet this criterion.

Crucial observation: If $\mathcal{L} \subseteq \mathcal{K}$ then $T(\mathcal{L}) \geq T(\mathcal{K})$.

The algorithm finds all $\mathcal{K}$ such that $T(\mathcal{K}) > t$ in order of increasing size of $\mathcal{K}$.

First, we can easily find all such singleton $\mathcal{K}$. (Just look at $(1/N) \sum_{i=1}^{N} z_{ik}$.)

To find $\mathcal{K}$ of size $m > 1$, we need look only at $\mathcal{K}$ for which all subsets of size $m-1$ have $T(\mathcal{K}) > t$. This is a relatively small fraction of all possible $\mathcal{K}$.

We can find all the association of size up to m with support greater than t in only m passes over the data.

From Item Sets to Association Rules

Once we have found item sets, $\mathcal{K}$, with support greater than t, we can use them to find "association rules".

If $\mathcal{K} = A \cup B$, we produce the possible association rule $A \Rightarrow B$. This is interpreted as saying that when A occurs, B also tends to occur.

The support, $T(A \Rightarrow B)$, of this rule is defined to be $T(\mathcal{K})$, and can be interpreted as an estimate of $\Pr(A \text{ and } B)$.

For each such possible rule, we can also compute the "confidence",

$$C(A \Rightarrow B) \;=\; \frac{T(A \Rightarrow B)}{T(A)}$$

This is an estimate of $\Pr(B \,|\, A)$.

We can also compute the "lift",

$$L(A \Rightarrow B) \;=\; \frac{C(A \Rightarrow B)}{T(B)}$$

This is an estimate of $\Pr(A \text{ and } B) \,/\, \Pr(A)\Pr(B)$. If it's close to 1, then the rule doesn't really tell us anything.

Example of Market Basket Analysis

The book has an example (section 14.2.3) of association rules found from a marketing data base. The data has anwsers to questionaires filled out by 9409 shoppers. Here are some association rules found:

Association Rule 1: support 25%, confidence 99.7%, lift 1.03.

Number in household = 1
Number of children = 0

$$\Downarrow$$

Language in home = English

Association Rule 2: support 13.4%, confidence 80.8%, lift 2.13.

Language in home = English
Householder status = own
Occupation = professional/managerial

$$\Downarrow$$

Income $\geq$ \$40,000

Do these seem like reasonable rules?