2.5. Database Normalization and Information Theory

Contents
2.5.1. Candidate and Super Ke
2.5.2. Functional Dependencies

Let's now take a detour to make a connection between database theory and information theory. Database normalization is the process of structuring a relational database in order to reduce data redundancy and improve data integrity. The key word here is "data redundancy", and that's exactly what entropy measures. Normalization is a tricky topic and I encourage you to read up on it on your own. Here, we will just overview the key ideas in terms of information theory.

Here are some propositions that will help us:

- 1. If $Y = f(X)$, meaning that X "fully" determines the value of Y, then $H(Y|X) = 0$
- 1. And from the previous section, $H(X, Y) = H(Y|X) + H(X)$.

2.5.1. Candidate and Super Keys

A key is a set of attributes within a table whose values can be used to uniquely identify a tuple. For example, in the table below the VIN column uniquely identifies each tuple:

VIN	Make	Model	Year	Color
123342889	Toyota	Corolla	2018	Red
458892199	Toyota	Camry	2015	Blue
298847721		Honda	Fit	2014

What does this mean in terms of information theory? For one, since we know that VIN completely determines the whole tuple (via proposition 1):

$$H(Make, Model, Year, Color|VIN) = 0$$

Then via proposition 2:

$$H(Make, Model, Year, Color, VIN) = H(Make, Model, Year, Color|VIN) + H(VIN) = H(VIN)$$

This leads to an information theoretic definition of a key. Let $(X_j)_{j \in S}$ denote any subset $S \subseteq A$ of columns in a table. S is a superkey if:

$$H[(X_j)_{j \in A}] = H[(X_j)_{j \in S}]$$

In other words, any subset of columns that contains the full information content of the table—meaning the joint entropy of the key columns is equal to the joint entropy of ALL of the columns.

We call this a "super" key because it is not minimal. In the example above, every column combination that contains "VIN" would be a superkey. This leads to the definition of a *candidate key*, which is a minimal superkey:

$$\argmin_{S \in 2^A} |S|$$

$$s.t H[(X_j)_{j \in A}] = H[(X_j)_{j \in S}]$$

Candidate keys are not necessarily unique. For example, we could have a VIN number and a state registration number associated with a car above. Both would fully determine the full table.

2.5.2. Functional Dependencies

Keys fully determine a tuple, sometimes we have determination relationships that do not span all of the columns. These are called functional dependencies. Let S and T be subsets of columns, a functional dependency between S and T implies that the values of S fully determine the values of T—denoted by:

$$S \rightarrow T$$

Again, this is simply a conditional entropy relationship. A functional dependency is any subset of variables S and T such that:

$$H[(X_j)_{j \in T} | (X_j)_{j \in S}] = 0$$

Or equivalently:

$$H[(X_j)_{j \in S \cup T}] = H[(X_j)_{j \in S}]$$
