Database Normalization as a By-product of Minimum Message Length Inference

David Dowe Nayyar A. Zaidi

Clayton School of IT, Monash University, Melbourne VIC 3800, Australia

December 8, 2010

Our Research Goals

- Database normalization is a central part of database design in which we re-organise the data stored so as to progressively ensure that as few anomalies occur as possible upon insertions, deletions and/or modifications.
- We show here that database normalization follows as a consequence (or special case, or by-product) of the Minimum Message Length (MML) principle of machine learning and inductive inference.

Our Research Goals (Contd)

- There can be many motivations behind a database normalization.
- In this paper, we present a novel information-theoretic perspective of database normalization.
- We consider the structure of the table(s) as a modelling problem for Minimum Message Length (MML).
- MML seeks a model giving the shortest two-part coding of model and data. If we consider **table** structure as a model which encodes data, MML advocates that we should be particularly interested in the variation of the encoding length of model and data as the normalization process re-structures tables for efficient design.

Minimum Message Length

- MML considers any given string S as being a representation in some (unknown) code about the real world.
- It seeks a ([concatenated] two-part) string $I = H : A$ where the first part H specifies (or encodes) a hypothesis about the data S and the second part A is an encoding of the data using the encoded hypothesis.
- If the code or hypothesis is true, the encoding is efficient (like Huffman or arithmetic codes). According to Shannon's theory, the length of the string coding an event E in an optimally efficient code is given by $-\log_2(\mathrm{Prob}(E))$.

Minimum Message Length (Contd)

- The length of A is given by:

$$\#A = -\log_2(f(S|H)) \tag{1}$$

  where $f(S|H)$ is the conditional probability (or statistical likelihood) of data S given the hypothesis H.

- Using an optimal code for specification, the length $\#H$ of the first part of the MML message is given by $-\log_2(h(H))$, where $h(\cdot)$ is the prior probability distribution over the set of possible hypotheses. Using equation (1), the total two-part message length $\#I$ is:

$$\begin{aligned} \#I &= \#H + \#A = -\log_2(h(H)) - \log_2(f(S|H)) \\ &= -\log_2(h(H) \times f(S|H)) \end{aligned} \tag{2}$$

Database Normalization

- The term 1NF describes a tabular data format where the following properties hold. First, all of the key attributes are defined. Second, there are no repeating groups in the table -i.e., in other words, each row/column intersection (or cell) contains one and only one value, not a set of values. Third, all attributes are dependent on the primary key (PK).
- A table is in 2NF if the following conditions hold. First, it is in 1NF. Second, it includes no partial dependencies, that is no attribute is dependent on only a portion of the primary key.
- A table is in 3NF if the following holds. First, it is in 2NF. Second, it contains no transitive dependencies. A transitive dependency exists when there are functional dependencies [1] such that $X \rightarrow Y$, $Y \rightarrow Z$ and X is the primary key attribute.

[1] The attribute B is fully functional dependent on the attribute A if each value of A determines one and only value of B.

Database Normalization Example

Stud-ID	Stud-Name	Stud-Address	Stud-Course	**Unit-No**	Unit-Name	Lect-No	Lect-Name	**Yr-Sem**	Gr
212	Bob Smith	Notting Hill	MIT	FIT2014	Database Design	47	Geoff Yu	2007	I
212	Bob Smith	Notting Hill	MIT	FIT3014	Algorithm Theory	47	Geoff Yu	2007	H
212	Bob Smith	Notting Hill	MIT	EE1007	Circuit Design	47	Geoff Yu	2006	I
213	John News	Caufield	BSc	FIT3014	Algorithm Theory	122	June Matt	2007	H
213	John News	Caufield	BSc	EE1007	Circuit Design	122	June Matt	2007	H
214	Alice Neal	Clayton S	BSc	FIT2014	Database Design	122	June Matt	2007	H
214	Alice Neal	Clayton S	BSc	FIT3014	Algorithm Theory	122	June Matt	2007	I
215	Jill Wong	Caufield	MIT	FIT2014	Database Design	47	Geoff Yu	2007	I
215	Jill Wong	Caufield	MIT	FIT2014	Database Design	47	Geoff Yu	2008	I
216	Ben Ng	Notting Hill	BA	EE1007	Circuit Design	47	June Matt	2007	I
216	Ben Ng	Notting Hill	BA	MT2110	Mathematics-II	47	June Matt	2007	I

Table: **Student-Rec** in 1NF. PK = (**Stud-ID**, **Unit-No**, **Yr-Sem**)

Database Normalization Example (Contd)

Stud-ID	Stud-Name	Stud-Address	Stud-Course	Lect-No	Lect-Name
212	Bob Smith	Notting Hill	MIT	47	Geoff Yu
213	John News	Caufield	BSc	122	June Matt
214	Alice Neal	Clayton S	BSc	47	Geoff Yu
215	Jill Wong	Caufield	MIT	47	Geoff Yu
216	Ben Ng	Notting Hill	BA	122	June Matt

Table: **Student** in 2NF. PK = **Stud-ID**

Unit-No	Unit-Name
FIT2014	Database Design
FIT3014	Algorithm Theory
EE1007	Circuit Design
MT2110	Mathematics-II

Table: **Unit** in 2NF and 3NF, PK = **Unit-No**

Stud-ID	**Unit-No**	**Yr-Sem**	Grade
212	FIT2014	2007	D
212	FIT3014	2007	HD
212	EE1007	2006	P
213	FIT3014	2007	HD
213	EE1007	2007	HD
214	FIT2014	2007	HD
214	FIT3014	2007	D
215	FIT2014	2007	D
215	FIT2014	2008	D
216	EE1007	2007	P
216	MT2110	2007	D

Table: **Stu-Unit-Rec** in 2NF and 3NF. PK = (**Stud-ID**, **Unit-No**, **Yr-Sem**)

Database Normalization Example (Contd)

Stud-ID	Stud-Name	Stud-Address	Stud-Course	Lect-No
212	Bob Smith	Notting Hill	MIT	47
213	John News	Caufield	BSc	122
214	Alice Neal	Clayton S	BSc	47
215	Jill Wong	Caufield	MIT	47
216	Ben Ng	Notting Hill	BA	122

Table: **Student** in 3NF. PK = **Stud-ID**

Lect-ID	Lect-Name
47	Geoff Yu
122	June Matt

Table: **Lecturer** in 3NF, PK = **Lect-No**

MML Interpretation of Normalization

- Our simple example of the normalization process from has resulted in four distinct tables - namely, Student, Lecturer, Unit, and Stu-Unit-Rec.
- Normalization is nothing but judicious re-structuring of information via tables.
- we can write the first-part message length (encoding the model) as:

$$\#H \quad = \quad |<T>| + |<A>| + \sum_{t=1}^{T} AP_t \qquad (3)$$

where T is the number of tables, A is the number of attributes. AP_t denotes the encoding length of table t's attributes and its primary key.

$$AP_t \quad = \quad \log_2(\mathrm{A}) + \log_2 \binom{A}{a_t} + \log_2(\mathrm{a}_t) + \log_2 \binom{a_t}{p_t} \qquad (4)$$

MML Interpretation of Normalization (Contd)

$$AP_t = \log_2(\mathrm{A}) + \log_2 \binom{A}{a_t} + \log_2(\mathrm{a}_t) + \log_2 \binom{a_t}{p_t} \quad (5)$$

- where a_t is the number of attributes in the t^{th} table, p_t denotes the number of attributes in the primary key. (We know that $1 \leq a_t \leq A$, so $\log_2(A)$ is the cost of encoding a_t, and $\log_2 \binom{A}{a_t}$ is the cost of saying which particular a_t attributes are in the t^{th} table. Similarly, since $1 \leq p_t \leq a_t$, $\log_2 a_t$ is the cost of encoding p_t, and $\log_2 \binom{a_t}{p_t}$ is the cost of saying which particular p_t attributes are in the primary key of the t^{th} table.)
- Note that this is only one way of specifying the model. We have taken only the number of tables, attributes in each table and attributes constituting the PK in each table into account in specifying a model. Other models could be used.

MML Interpretation of Normalization (Contd)

- The number of rows in the 1NF form of the table is an important variable. We have denoted it by L in the preceding equations. $L = 11$ in table 1 and depends on how many students are taking how many courses in each semester.
- We will later show that there is not a huge need for normalization if each student is taking only one unit, as 2NF will encode the same (amount of) information as 1NF.
- As more students take more courses, the need for normalization arises.

Stud-ID m_1	Stud-Name m_2	Stud-Address m_3	Stud-Course m_4	**Unit-No** m_5	Unit-Name m_6	Lect-No m_7	Lect-Name m_8	**Yr-Sem** m_9	Grade m_{10}
5	5	5	5	4	4	2	2	3	3

Table: Number of unique instances for each attribute in table 1, 1NF of our initial example

MML Interpretation of Normalization (Contd)

$$
\begin{aligned}
I_{1NF} &= \#H_{1NF} + \#A_{1NF} \\
&= \#H_{1NF} + L \times (\log_2 m_1 + \log_2 m_2 + \log_2 m_3 + \cdots + \log_2 m_{10})
\end{aligned}
$$

$$
\begin{aligned}
I_{3NF} &= \#H_{3NF} + \#A_{3NF} \\
&= \#H_{3NF} + m_1 \times (\log_2 m_1 + \log_2 m_2 + \log_2 m_3 + \log_2 m_4 + \log_2 m \\
&\quad + m_7 \times (\log_2 m_7 + \log_2 m_8) \\
&\quad + m_5 \times (\log_2 m_5 + \log_2 m_6) \\
&\quad + L \times (\log_2 m_1 + \log_2 m_5 + \log_2 m_9 + \log_2 m_{10})
\end{aligned}
$$

MML Interpretation of Normalization (Contd)

	$\#H$ (first part's length)	$\#A$ (second part's length)	total message length
1NF	10.22	203.03	213.25
2NF	36.45	154.89	191.34
3NF	46.26	153.84	200.10

Table: Code length (bits) of model and data for different NFs on small example

	$\#H$ (first part's length)	$\#A$ (second part's length)	total message length
1NF	10.22	14210	14220
2NF	36.45	8150	8186
3NF	46.26	7876	7922

Table: Encoding length (in bits) of model and data for different NFs, Number of Students (m_1) = 100, Number of Units (m_5) = 30, Number of Lecturers (m_7) = 15, $L = 300$

MML Interpretation of Normalization (Contd)

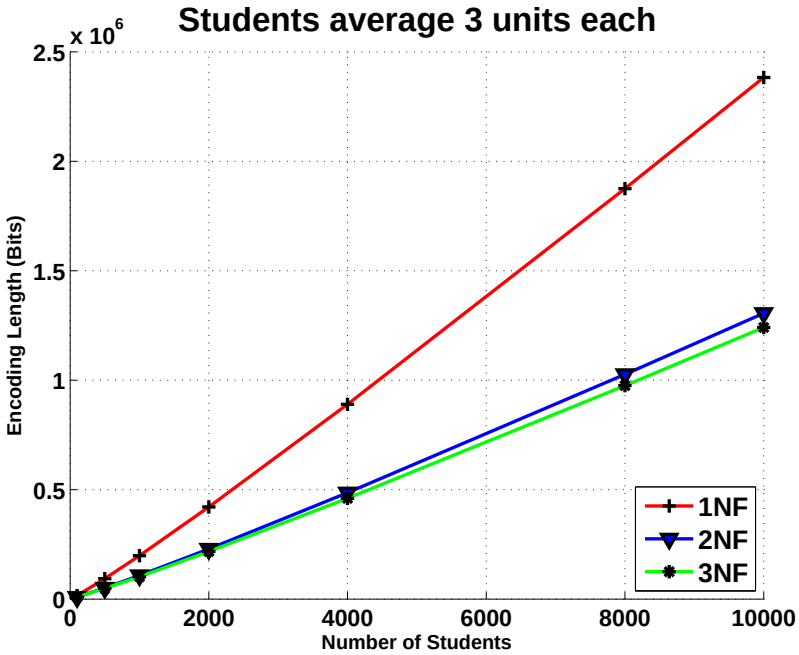

Figure: Variation in total message length (I) by varying number of students (m_1) and L for different NFs. The number of Units (m_5) is set to 30 and the number of Lecturers (m_7) is set to 15. $L = 3m_1$

Conclusion

- We have presented database normalization as a consequence of MML inference.
- With an example, we demonstrated a typical normalization procedure and analyzed the process using the MML framework. We found that with higher NFs, the model is likely to become more complicated, but the data encoding length is decreased. If there is a relationship or dependency in the data (according to database normalisation principles), then - given sufficient data - MML will find this. This suggests that normalization is - in some sense - simply following MML.

Conclusion (contd)

- Though we have limited ourselves here to 1^{st}, 2^{nd} and 3^{rd} normal forms (NFs), applying MML can also be shown to lead to higher NFs such as Boyce-Codd Normal Form (BCNF), 4NF and 5NF. Indeed, recalling the notion of MML Bayesian network, normalizing and breaking down tables into new tables can be thought of as a (MML) Bayesian net analysis - using the fact that (in some sense) databases could be said to have no noise. And, in similar manner, (the notion of) attribute inheritance (where different types of employee - such as pilot and engineer - have their own specific attributes as well as inheriting common employee attributes) can also be inferred using MML.

Questions