Capita Selecta Artificial Intelligence
Statistical Relational Learning

Thijs Dieltjens Wout Vekemans

December 2015

1 Probabilistic Databases

1.1 Tables

The database contains musicians, bands, albums and songs. Each musician plays in one or more bands, each band plays some songs, and each song is on one or more albums. The *plays_in* relation can be seen in table 1, the *song* relation is in table 2 and the albums are in table 3.

1.2 Queries

The first query is "Which are the songs that are played by Dave Grohl".

SQL

```
SELECT s.title
FROM song s, plays_in p, album a
WHERE p.artist="Dave Grohl" and a.title = s.album and a.band = p.band
```

First Order Logic
Q(t) = ∃ a,b,i,y,j (plays_in("Dave Grohl",b)∧ album(i,y,b,a)∧ song(j,t,a))

ProbLog

```
result(Artist,Title) :-
  plays_in(Artist,Band),
  song(_,Title,Album),
  Album(_,_,Band,Album).

query(result("Dave Grohl",_)).
```

The second query is Boolean : "What is the probability that Dave Grohl participated on the album *Nevermind*?"

First Order Logic

Q = ∃ b,i,y, (plays_in("Dave Grohl",b)∧ album(i,y,b,Nevermind)

ProbLog

```
album_artist(Artist, Album) :-
        plays_in(Artist, Band),
        album(_,_,Band, Album).

query(album_artist("Dave Grohl", "Nevermind")).
```

Manual Inference

$$
\begin{aligned}
P &= P(plays\_in(\text{'Dave Grohl'}, Band)) * P(album(\_, \_, Band, \text{'Nevermind'})) \\
&= P(plays\_in(\text{'Dave Grohl'}, \text{'}Foo\ Fighters'')) * P(album(\_, \_, \text{'}Foo\ Fighters', \text{'Nevermind'}) \\
&\quad + P(plays\_in(\text{'Dave Grohl'}, \text{'Nirvana'}) * P(album(\_, \_, \text{'Nirvana'}, \text{'Nevermind'})) \\
&= 0.7 * 0 + 0.3 * 0.97 \\
&= 0.291
\end{aligned} \tag{1}
$$

2 Beverage Preferences

Buy predicate The buy predicate looks as follows:

```
0.1::buy(Person, Drink) :-
preference(Person, Drink, like_little).

0.5::buy(Person, Drink) :-
preference(Person, Drink, like_much).

0.9::buy(Person, Drink) :-
preference(Person, Drink, like_verymuch).
```

Subjective Information The *preference_soft* relation looks like this:

```
0.4::preference_soft(Person, Drink, like_no);
0.5::preference_soft(Person, Drink, like_little);
0.1::preference_soft(Person, Drink, like_much) :-
  preference(Person, Drink, like_no).

0.5::preference_soft(Person, Drink, like_no);
0.4::preference_soft(Person, Drink, like_much);
0.1::preference_soft(Person, Drink, like_verymuch) :-
  preference(Person, Drink, like_little).
```

Plays_in

Artist	Band	P
Dave Grohl	Foo Fighters	0.7
Dave Grohl	Nirvana	0.3
Kurt Cobain	Nirvana	0.88
James Hetfield	Metallica	0.8

Table 1: Plays_in relation

Song

Id	Title	Album	P
1	Learn To Fly	There Is Nothing Left To Lose	0.68
2	Learn To Fly	Greatest Hits	0.28
3	Lithium	Nevermind	0.7
4	Fade To Black	Ride The Lightning	0.69

Table 2: Song table

Album

Id	Release	Band	Title	P
1	1984	Metallica	Ride The Lightning	0.95
2	1991	Nirvana	Nevermind	0.97
3	1999	Foo Fighters	There Is Nothing Left To Lose	0.92
4	2009	Foo Fighters	Greatest Hits	0.24

Table 3: Album table

```
0.1::preference_soft(Person, Drink, like_much);
0.1::preference_soft(Person, Drink, like_no);
0.4::preference_soft(Person, Drink, like_little);
0.4::preference_soft(Person, Drink, like_verymuch) :-
  preference(Person, Drink, like_much).

0.5::preference_soft(Person, Drink, like_verymuch);
0.1::preference_soft(Person, Drink, like_little);
0.4::preference_soft(Person, Drink, like_much) :-
  preference(Person, Drink, like_verymuch).
```

We made the assumption that the sum of all probabilities for each like-level should be 1, so for example P(like_no = like_no) = 0.4. In file `beverages.pl`, the `buy2` relation uses this soft preference

The probability of John buying whiskey using the normal preferences is 0.1, since he likes it a little. When using the soft preference, there is a probability of 0.5 that he does not buy it (P(like_no = like_little) = 0.5), a probability of 0.4 that he buys it with a probability of 0.5 (P(like_little = like_much) = 0.4) and a probability of 0.1 that he buys whiskey with a probability of 0.9 (P(like_little = like_verymuch) = 0.1). When these probabilities are multiplied and added, the result is 0.29. It can be seen that this is not the same as when using the hard preference.

Beverage Ontology The preference using the distance between drinks can be found in file `beverages.pl`, as `preference_with_distance`. We encoded the ontology as tree structures with a name, a parent and a list of children. Each of these children is a tree on its own. To calculate the distance between two drinks, the list of parents for eacht drink is calculated. With these two lists, the closest common ancestor is found, and the distances of the two drinks to this ancestor are added.

Using this, we found that John is most likely to buy ale (P = 0.32). Intuitively, one would think this as well, since he likes lager much. Paul is the least likely to buy ale (P = 0.18), since he doesn't like whiskey, which is also alcoholic (and therefor quite close to ale), but he likes coffee very much, but the distance between coffee and ale is quite long, so it has almost no influence.

Additional background information can be added in several ways. For example the alcoholic level of the different beverages could be added to the system, and people would be more likely to buy drinks with an alcoholic level close to what they like. Another example is the adding of a recommender system, where people with similar preferences are analysed. For example, when person A likes whiskey, and person B, C and D like whiskey and beer, person A will probably like beer.

3 Document Classification

The code for our noisy-or model can be found in `noisy-or.pl`, while the naive bayes can be found in `naive-bayes.pl`. The results of training these systems are in respectively `noisy_trained.pl` and `bayes_trained.pl`.