Interactive Knowledge Graph Completion with Naive Bayes

Anaïs Siebers & Felix Hamann

18. April 2021

Zusammenfassung

Construction knowledge bases (KB's) is a time consuming process. To assist a knowledge engineer with discovering new facts about domain entities, our goal is to achieve a scenario like the famous 10-questions-game. An oracle asks for relations or relation-tail combinations which should be as accurate as possible. We propose an approach using Naive Bayes to asses the possibilities and their probability.

1 Definitions

1.1 Symbols

$P(a)$	probability of a
X	set of all feature vectors
X_i	a specific feature vector
C	set of all classes
C_k	a specific class
$\hat{y}$	highest score
$P(C_k)$	prior for class k
R	set of all relations $r_1, \dots, r_n$
r_i	a specific relation
L	set of all tails $l_1, \dots, l_m$
l_i	a specific tail
$P(X_i \vert C_k)$	posterior or class conditional density
T	set of all tuples: $T \subset (R \cup L)$
t_i	a specific tuple
X_{ex}	set of all existing relations and tuples: $X_{ex} \subseteq (R \cup T)$
X_{not}	set of all not-existing relations and tuples: $X_{not} \subseteq (R \cup T)$

For Xex and X_{not} it can also be said that they do not contain an element of the other set: $X_{ex} \cap X_{not}$ is always $\emptyset$.

The following section deals with Naive Bayes and how Naive Bayes can be applied to the 20-question-game scenario.

2 Naive Bayes

Naive Bayes uses a stochastic approach based on Bayes' theorem:

$$P(C_k \mid X) = \frac{P(C_k \mid X) * P(X)}{P(C_k)} \quad (1)$$

This equation expresses the probability

for a class C_k under the condition that X occurred.

The equation is simplified and modified for classification problems. The approach is called ***Naive Bayes*** since all conditional probabilities are multiplied conditionally independently. For example: Given some news articles to be classified in categories such as *sport, economy, computer science, art, society, /dots*, it is obvious that the interdependence between the words (features) used in the article is of meaning. The probability fo the word **'artificial'** under the condition of the article category *computer science* is high but even higher if it is also considered that the word **'intelligence'** appeared. Because calculating the conditional probability for all features is inefficient, the interdependence of words in Naive Bayes is neglected. This leads to a more efficient and sufficient execution time: from 2^n to $2 * n$.

The score of the most probable class C_k is calculated by $\hat{y}$. Therefore the score for every class in C is calculated:

$$\hat{y} = max\{P(C_k \mid X) \forall k \in 1, \ldots, k\} \quad (2)$$

$$= max\{P(C_k) \cdot \prod_{i=1}^{n} P(X_i \mid C_k)\} \quad (3)$$

All features $x_1, \ldots, x_n$ are multiplied assuming independence under the condition of the current class C_k resulting in a score. The class C_k with the highest score $\hat{y}$ is the most probable class. $P(C_k)$ is called ***prior*** and describes the marginal probability for the class k. It is multiplied with the product of all class conditional probabilities (***posterior***) $P(X_i \mid C_k) \forall i \in 0, \ldots, n$. X_i represents a single feature, respectively X is a set of features.

2.1 Transfer of the approach

Naive Bayes calculates the probability that a certain class C_k is most probable under the condition of values $x_1, \ldots, x_n$. Comparing the probabilities / score calculated for $C_1, \ldots, C_k$, the class with the highest score is most likely to be the correct class for the new value:

$$k = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; P(C_k) \cdot \prod_{i=1}^{n} P(x_i \mid C_k) \quad (4)$$

Transferred to the problem that relations of an unknown entity need to be predicted, it has to be determined what is represented by C_k and which values are included in the calculation $x_1, \ldots, x_n$. Since the novel entity is added as head, the information searched is the context of the head. The evaluated classes $C_k \in \{r_1, \ldots, r_n, t_1, \ldots, t_m\}$ compromise the relations ($R = \{r_1, \ldots, r_n\}$) and the relation-tail combinations ($T = \{t_1, \ldots, t_m\}$). The co-occurrence of relations and relation-tail combinations is the only information known about the class, the features are the relations and relation-tail combinations (excluding the searched class): $x \in (R \cup T) \setminus C_k$.

As there are multiple iterations, information for a class is successively acquired and needs to be evaluated in the next calculation. There are two possible approaches of including the information provided by the posterior. A question in this scenario has to possible answers: yes and no - depending on the fact if the relation (or relation-tail combination) exists for the novel entity.

If the feature was confirmed by the expert, the class conditional probability is used. But if the feature was proposed and marked false by the expert, the probability $P(x_i|\text{not}C_k)$ is used.

Thus there is a need of two sets: X_{ex} and X_{not}. X_{ex} contains the correctly predicted features where X_{not} contains the features which the algorithm proposed but which are incorrect.

To include this information in the equation described above there are some necessary adaptations:

1. The candidate classes must exclude the existing and not-existing features: $C \in (R \cup T) \setminus (X_{ex} \cup X_{not})$. One case requires special attention. If a class in X_{not} is a relation, all classes which are tuples containing the relation also need to be excluded from C.

2. When the probability is calculated it needs to be determined whether the fact is correct or not. If it is correct, it can simply be added to X_{ex}. If it is incorrect, it is added to X_{not}. Relations and relation-tail combinations are treated differently: If a **relation** is incorrect, all tuples containing the relation have to be added to X_{not} as well. **Relation-tail combinations** can simply be added to X_{not}. There are no other effects.

3. The class conditional density calculation needs to be adapted as well. In comparison to before only confirmed facts and facts which have a conditional probability are considered while determining the probability (before every value was used). This means that the first iteration is only based on the prior of C_k. The following iterations also take the confirmed/denied features into account. Two cases can be distinguished: If the feature is **in** X_{ex}, the probability is used. If it is **in** X_{not}, the probability $P(x_i|\text{not}C_k)$ is used.

3 Evaluation of questions

There are two main goals for this project:

- The amount of questions should be minimised.
- As many correct facts as possible should be suggested.

To optimise towards these goals, an evaluation of the result is needed.