A Comparison of Automatic Algorithms for Occupation Coding

BigSurv 2018
Barcelona
October 26, 2018

Malte Schierholz
Measurement of Occupation

Occupations are analyzed within ...

- Economics: individuals' tasks and duties, ...
- Sociology: social status, ...
- Medicine: health risks, ...

How to measure occupations in interviews?
With open-ended questions!
Assign verbatim answers to an official classification

Survey Answers

? Nurse

in a hospital

? Classification

KldB 2010 | Job Title (illustrative)
--- | ---
81302 | Nurse
81313 | Graduated Nurse
81323 | Pediatric Nurse
81102 | Doctor’s Assistant
82102 | Geriatric Nurse

→ Coding is mostly manual work and therefore expensive
→ How to automate it?
Two modes of automation:

- **Computer-assisted coding:** Human choice, see picture
- **Automated coding:** Select top-ranked category

→ How to calculate scores / probabilities?

©CASCOT (https://warwick.ac.uk/fac/soc/ier/software/cascot/)
The Challenge

- Manual occupation coding is hard
  - Coders are trained on the classification and coding conventions
    (Geis and Hoffmeyer-Zlotnik (2000), Paulus and Matthes (2013))
  - Coders draw on information from multiple questions
  - Inter-coder reliabilities range between 61-83%
    (Prigge et al. (2014), Schierholz et al. (2018))

- High-dimensional classification problem
  - Our document-term matrix has 54.880 observations and 14.223 columns (words)
  - 1.291 outcome categories
  - > 15 million parameters in a multinomial logistic regression
Ten Algorithms

Comprehensive comparison of ten algorithms:

1. Coding index (exact)
2. Coding index (\w similars) (Elias et al. (2014))
3. Memory-based Reasoning (Creecy et al. (1992))
4. Adapted Nearest Neighbor (Gweon et al. (2017))
5. Multinomial Logit (R-package glmnet, Friedman et al. (2010))
6. XGBoost (R-package XGBoost, Chen and Guestrin (2016))
7. Coding index + Bayes (fulltext similarity)
8. Coding index + Bayes (substring similarity)
9. Coding index + Bayes (wordwise similarity)
10. Highest probability

Developed for occupation coding

General purpose

Own contribution
## Ten Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Reference data</th>
<th>Text processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Coding index (exact)</td>
<td>Coding Index</td>
<td>String identity</td>
</tr>
<tr>
<td>2. Coding index (\w similars)</td>
<td>Coding Index</td>
<td>String similarity</td>
</tr>
<tr>
<td>4. Adapted Nearest Neighbor</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>5. Multinomial Logit</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>6. XGBoost</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>7. Coding index+Bayes (fulltext similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (fulltext similarity)</td>
</tr>
<tr>
<td>8. Coding index+Bayes (substring similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (substring similarity)</td>
</tr>
<tr>
<td>9. Coding index+Bayes (wordwise similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (wordwise similarity)</td>
</tr>
<tr>
<td>10. Highest probability</td>
<td>Among algorithms 6-9, select the algorithm with highest probability</td>
<td></td>
</tr>
</tbody>
</table>
### First algorithm: Consult a coding index

#### 2010 German Classification

(27,575 job titles)

<table>
<thead>
<tr>
<th>Job title</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Säuglings- und Kinderkrankenschwester/-pfleger</td>
<td>81302</td>
</tr>
<tr>
<td>Säuglings- und Krankenschwester/-pfleger</td>
<td>81302</td>
</tr>
<tr>
<td>Säuglingskrankenschwester/-pfleger</td>
<td>81302</td>
</tr>
<tr>
<td>Säulingsschwester/-pfleger</td>
<td>81302</td>
</tr>
<tr>
<td>Schwestern/Pfleger (Kinderkrankenpflege)</td>
<td>81302</td>
</tr>
<tr>
<td>Schwestern/Pfleger (Krankenpflege)</td>
<td>81302</td>
</tr>
<tr>
<td>Anästhesieschwester/-pfleger</td>
<td>81313</td>
</tr>
<tr>
<td>Dialysafachkraft</td>
<td>81313</td>
</tr>
<tr>
<td>Dialysechwester/-pfleger</td>
<td>81313</td>
</tr>
<tr>
<td>Endoskopieassistent/in</td>
<td>81313</td>
</tr>
<tr>
<td>Endoskopieschwester/-pfleger</td>
<td>81313</td>
</tr>
<tr>
<td>Fachkrankenpfleger/in - Notfallpflege</td>
<td>81313</td>
</tr>
</tbody>
</table>

Source: KldB 2010

#### 2018 US Classification

(6,539 job titles)

<table>
<thead>
<tr>
<th>Job Title</th>
<th>Code</th>
<th>Category Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospice Registered Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Obstetrical Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Oncology Registered Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>PACU Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Pediatric Registered Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Post-Anesthesia Care Unit Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Psychiatric Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>RN</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Triage Registered Nurse</td>
<td>29-1141</td>
<td>Registered Nurses</td>
</tr>
<tr>
<td>Certified Registered Nurse Anesthetist</td>
<td>29-1151</td>
<td>Nurse Anesthetists</td>
</tr>
<tr>
<td>Certified Registered Nurse Anesthetist (CRNA)</td>
<td>29-1151</td>
<td>Nurse Anesthetists</td>
</tr>
<tr>
<td>DNAP</td>
<td>29-1151</td>
<td>Nurse Anesthetists</td>
</tr>
<tr>
<td>Doctor of Nurse Anesthesia Practice</td>
<td>29-1151</td>
<td>Nurse Anesthetists</td>
</tr>
<tr>
<td>Certified Nurse Midwife</td>
<td>29-1161</td>
<td>Nurse Midwives</td>
</tr>
<tr>
<td>Certified Nurse Midwife (CNM)</td>
<td>29-1161</td>
<td>Nurse Midwives</td>
</tr>
<tr>
<td>Acute Care Nurse Practitioner</td>
<td>29-1171</td>
<td>Nurse Practitioners</td>
</tr>
<tr>
<td>Adult Nurse Practitioner</td>
<td>29-1171</td>
<td>Nurse Practitioners</td>
</tr>
<tr>
<td>Cardiology Nurse Practitioner</td>
<td>29-1171</td>
<td>Nurse Practitioners</td>
</tr>
</tbody>
</table>

Source: 2018 SOC
Second algorithm:

Consult a coding index (based on string similarity)


©CASCOT (https://warwick.ac.uk/fac/soc/ier/software/cascot/)
## Ten Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Reference data</th>
<th>Text processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Coding index (exact)</td>
<td>Coding Index</td>
<td>String identity</td>
</tr>
<tr>
<td>2. Coding index (\w similars)</td>
<td>Coding Index</td>
<td>String similarity</td>
</tr>
<tr>
<td>4. Adapted Nearest Neighbor</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>5. Multinomial Logit</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>6. XGBoost</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>7. Coding index+Bayes (fulltext similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (fulltext similarity)</td>
</tr>
<tr>
<td>8. Coding index+Bayes (substring similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (substring similarity)</td>
</tr>
<tr>
<td>9. Coding index+Bayes (wordwise similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (wordwise similarity)</td>
</tr>
<tr>
<td>10. Highest probability</td>
<td>Among algorithms 6-9, select the algorithm with highest probability</td>
<td></td>
</tr>
</tbody>
</table>
Documents:

\[d_1 = \text{Nurse in a hospital} \rightarrow d_2 = \text{Hospital manager} \rightarrow \ldots\]

Document-term matrix:

\[
\begin{array}{cccccc}
\ldots & a & \text{hospital} & \text{in} & \text{manager} & \text{nurse} & \ldots \\
\ldots & 1 & 1 & 1 & 0 & 1 & \ldots \\
\ldots & 0 & 1 & 0 & 1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]

Prediction techniques:

Based on vector similarity (k-NN-like):

3. Memory-based Reasoning (Creecy et al. (1992))

4. Adapted Nearest-Neighbor (Gweon et al. (2017))

Based on loss minimization:

5. Multinomial logistic regression (R-package glmnet, Friedman et al. (2010))

6. Gradient boosting with decision trees (R-package XGBoost, Chen and Guestrin (2016))
Documents:

\[ d_1 = \text{Nurse in a hospital} \rightarrow \]
\[ d_2 = \text{Hospital manager} \rightarrow \]

\[ \vdots \]

Document-term matrix:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>hospital</th>
<th>in</th>
<th>manager</th>
<th>nurse</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Prediction techniques:

Based on vector similarity (\(k\)-NN-like):

3. Memory-based Reasoning
   (Creecy et al. (1992))

4. Adapted Nearest-Neighbor
   (Gweon et al. (2017))

Based on loss minimization:

5. Multinomial logistic regression
   (R-package glmnet, Friedman et al. (2010))

6. Gradient boosting with decision trees
   (R-package XGBoost, Chen and Guestrin (2016))
Documents:

\[ d_1 = \text{Nurse in a hospital} \rightarrow \]
\[ d_2 = \text{Hospital manager} \rightarrow \]

\[ \vdots \]

Document-term matrix:

\[
\begin{array}{cccccc}
\vdots & a & \text{hospital} & \text{in} & \text{manager} & \text{nurse} & \vdots \\
\vdots & 1 & 1 & 1 & 0 & 1 & \vdots \\
\vdots & 0 & 1 & 0 & 1 & 0 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]

New problem:

Algorithms 3-6 fail if important words (especially job titles) are not in the training data

- Infrequent jobs (e.g., ornithologist, horseshoer)
- Misspellings

Algorithm 2 already solved the issue. How to combine the different approaches?
## Ten Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Reference data</th>
<th>Text processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Coding index (exact)</td>
<td>Coding Index</td>
<td>String identity</td>
</tr>
<tr>
<td>2. Coding index (\w similars)</td>
<td>Coding Index</td>
<td>String similarity</td>
</tr>
<tr>
<td>4. Adapted Nearest Neighbor</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>5. Multinomial Logit</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>6. XGBoost</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>7. Coding index+Bayes (fulltext similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (fulltext similarity)</td>
</tr>
<tr>
<td>8. Coding index+Bayes (substring similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (substring similarity)</td>
</tr>
<tr>
<td>9. Coding index+Bayes (wordwise similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (wordwise similarity)</td>
</tr>
<tr>
<td>10. Highest probability</td>
<td>Among algorithms 6-9, select the algorithm with highest probability</td>
<td></td>
</tr>
</tbody>
</table>
A novel algorithm relies on the coding index, but uses past data as well.

Key idea:

- **Training**: Estimate for each entry in the coding index a posterior predictive distribution (Hierarchical Bayesian model)
  \[ P(\text{code}_j | \text{index entry}) \]

- **Prediction**: Search for a similar index entry and output its distribution as estimated

Three ways of calculating string similarity (Algorithms 7-9):

7. **Fulltext similarity**: Similar if the verbal answer differs by at most one character
8. **Substring similarity**: Similar if an entry from the coding index is a substring of the verbal answer
9. **Wordwise similarity**: Similar if one word in the verbal answer differs by at most one character
A novel algorithm relies on the coding index, but uses past data as well.

**Key idea:**

- **Training:** Estimate for each entry in the coding index a posterior predictive distribution (Hierarchical Bayesian model)
  \[ P(code_j | \text{index entry}) \]
- **Prediction:** Search for a similar index entry and output its distribution as estimated

Three ways of calculating string similarity (Algorithms 7-9):

7. **Fulltext similarity:** Similar if the verbal answer differs by at most one character
8. **Substring similarity:** Similar if an entry from the coding index is a substring of the verbal answer
9. **Wordwise similarity:** Similar if one word in the verbal answer differs by at most one character
## Ten Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Reference data</th>
<th>Text processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Coding index (exact)</td>
<td>Coding Index</td>
<td>String identity</td>
</tr>
<tr>
<td>2. Coding index (\w similars)</td>
<td>Coding Index</td>
<td>String similarity</td>
</tr>
<tr>
<td>4. Adapted Nearest Neighbor</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>5. Multinominal Logit</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>6. XGBoost</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>7. Coding index+Bayes (fulltext similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (fulltext similarity)</td>
</tr>
<tr>
<td>8. Coding index+Bayes (substring similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (substring similarity)</td>
</tr>
<tr>
<td>9. Coding index+Bayes (wordwise similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (wordwise similarity)</td>
</tr>
<tr>
<td>10. Highest probability</td>
<td>Among algorithms 6-9, select the algorithm with highest probability</td>
<td></td>
</tr>
<tr>
<td>Algorithm</td>
<td>Reference data</td>
<td>Text processing</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>1. Coding index (exact)</td>
<td>Coding Index</td>
<td>String identity</td>
</tr>
<tr>
<td>2. Coding index (\w similars)</td>
<td>Coding Index</td>
<td>String similarity</td>
</tr>
<tr>
<td>4. Adapted Nearest Neighbor</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>5. Multinominal Logit</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>6. XGBoost</td>
<td>Training data</td>
<td>Document-term matrix</td>
</tr>
<tr>
<td>7. Coding index+Bayes (fulltext similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (fulltext similarity)</td>
</tr>
<tr>
<td>8. Coding index+Bayes (substring similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (substring similarity)</td>
</tr>
<tr>
<td>9. Coding index+Bayes (wordwise similarity)</td>
<td>Coding Index &amp; Training</td>
<td>String similarity (wordwise similarity)</td>
</tr>
<tr>
<td>10. Highest probability</td>
<td>Among algorithms 6-9, select the algorithm with highest probability</td>
<td></td>
</tr>
</tbody>
</table>
Application in survey interviews envisioned:

- Only texts from the first open-ended question are available (unlike current coding situations)
- Let the computer decide what to ask next
  - Reduce respondent burden
  - Increase data quality
Automated & Computer-assisted Coding

1. What is the occupational activity you do in your main job?

Nursing

Next

2. Automated Coding
(=Computer selects the top-ranked category)

81302 (Nurse)

Computer-assisted Coding
(=Human chooses from a list of suggestions)
Data

2 German surveys, coded into the latest German occupational classification

- **Dataset 1**: 55,944 texts from *BIBB/BAuA Employment Survey 2012*  
  (Hartmann et al. (2012), Hall et al. (2015))  
  - Split in training and test data

- **Dataset 2**: 1,064 texts from *Selectivity effects in address handling*  
  (Schierholz et al. (2018))  
  - Test data only

Coders had access to more occupational variables in the second study - higher coding quality?
1. Performance of Automated Coding
(=Computer selects the top-ranked category)

Metrics used:
- Production rate: Proportion of answers that are coded automatically
- Agreement rate: proportion of coded answers that are in agreement with human coder

Trade-Off: Production rate ↔ Agreement rate

2. Performance of Computer-assisted Coding
(=Human coder chooses from a list of suggestions)
Automated Coding – Results

Prediction Algorithm

1. Coding index (exact matches)
2. Coding index (w similars)
3. Memory–based Reasoning
4. Adapted Nearest Neighbor
5. Multinomial Logit
6. XGBoost
7. Coding index + Bayes (fulltext sim.)
8. Coding index + Bayes (substring sim.)
9. Coding index + Bayes (wordwise sim.)
10. Highest Probability

$N_{test} = 1.064$ (Dataset 1)
Automated Coding – Results

![Graph showing the comparison of automatic algorithms for occupation coding. The x-axis represents the production rate, and the y-axis represents the agreement rate. The graph compares different prediction algorithms, including Coding index (exact matches), Coding index (w similars), Memory-based Reasoning, Adapted Nearest Neighbor, Multinomial Logit, XGBoost, Coding index + Bayes (fulltext sim.), Coding index + Bayes (substring sim.), Coding index + Bayes (wordwise sim.), and Highest Probability. The test statistic $N_{test} = 1.064$ (Dataset 1).]
Automated Coding – Results

A Comparison of Automatic Algorithms for Occupation Coding, Malte Schierholz

\[ N_{\text{train}} = 54.880 \text{ (Dataset 1)} \]
\[ N_{\text{test}} = 1.064 \text{ (Dataset 1)} \]
Automated Coding – Results

A Comparison of Automatic Algorithms for Occupation Coding, Malte Schierholz

Prediction Algorithm

1. Coding index (exact matches)
2. Coding index (w similars)
3. Memory-based Reasoning
4. Adapted Nearest Neighbor
5. Multinomial Logit
6. XGBoost
7. Coding index + Bayes (fulltext sim.)
8. Coding index + Bayes (substring sim.)
9. Coding index + Bayes (wordwise sim.)
10. Highest Probability

Test Data

\[ N_{\text{train}} = 54.880 \text{ (Dataset 1)} \]
\[ N_{\text{test}} = 1.064 \text{ (Dataset 1)} \]
\[ N_{\text{test}} = 1.064 \text{ (Dataset 2)} \]
1. Performance of Automated Coding
(=Computer selects the top-ranked category)

2. Performance of Computer-assisted Coding
(=Human coder chooses from a list of suggestions)

Agreement rate is calculated in a different way:
- For automated coding: highest-ranked category in agreement with human coder?
  \( k = 1 \) until now
- For computer-assisted coding: human-coded category among top \( k \) suggestions?
  \( k = 5 \) in the following
Computer-Assisted Coding – Results

A Comparison of Automatic Algorithms for Occupation Coding, Malte Schierholz

**Prediction Algorithm**

1. Coding index (exact matches)
2. Coding index (w similars)
3. Memory-based Reasoning
4. Adapted Nearest Neighbor
5. Multinomial Logit
6. XGBoost
7. Coding index + Bayes (fulltext sim.)
8. Coding index + Bayes (substring sim.)
9. Coding index + Bayes (wordwise sim.)
10. Highest Probability

**Test Data**

- $N_{train} = 54.880$ (Dataset 1)
- $N_{test} = 1.064$ (Dataset 1)

**Graph**

- X-axis: Production Rate
- Y-axis: Agreement Rate (among top 5)
A Comparison of Automatic Algorithms for Occupation Coding, Malte Schierholz

Computer-Assisted Coding – Results

**Prediction Algorithm**

1. Coding index (exact matches)
2. Coding index (w similars)
3. Memory-based Reasoning
4. Adapted Nearest Neighbor
5. Multinomial Logit
6. XGBoost
7. Coding index + Bayes (fulltext sim.)
8. Coding index + Bayes (substring sim.)
9. Coding index + Bayes (wordwise sim.)
10. Highest Probability

**Test Data**

<table>
<thead>
<tr>
<th>Data</th>
<th>N_{train}</th>
<th>N_{test}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>54.880 (Dataset 1)</td>
<td>1.064 (Dataset 1)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.064 (Dataset 2)</td>
</tr>
</tbody>
</table>
Performance
- Algorithm 2 (Coding index w similars) is often used in practice but outperformed by other algorithms (e.g. XGBoost).
- Combining the coding index and previously coded data is very promising (algorithms 7-10)
  - Best algorithm is problem-dependent

Generalization
- Poor generalization from one dataset to different test data

Application
- Agreement rates are often insufficient for fully automated coding
- Manual verification will usually be necessary, making computer-assisted coding most promising
  - Current plan: Suggest plausible answer options to respondents during the interview
Thank you!

Related publications:

The methods presented in this talk are also implemented in an R-Package:
https://github.com/malsch/occupationCoding

Background material:

Malte Schierholz
Malte.Schierholz@iab.de
References I


URL: http://doi.acm.org/10.1145/2939672.2939785


URL: https://doi.org/10.7803/501.12.1.4.10
URL: https://metadaten.bibb.de/download/684


URL: https://link.springer.com/article/10.1007/BF03374441


URL: https://www150.statcan.gc.ca/n1/pub/12-001-x/1988002/article/14586-eng.pdf