Modeling Career Pathways

Courtney Cochrane and Rohan Thavarajah

December 12, 2016

1 Introduction

In the wake of a recession, recovery is gauged by the state of the unemployment rate. In the wake of the ′Great Recession′, the US economy was characterized by a recovery that was particularly ″jobless″; although output growth had started to rebound, employment had yet to follow suit. Economists sought indicators of labor market health that were less blunt than blanket unemployment. Finding that employment rates were trailing an uptick in job vacancies, Faberman and Mazumder argued that the recovery had been muted by skills mismatch; although there were jobs on offer, workers did not have the skills to fulfill them ([5]).

However, what if the converse is true, and faced with a harsh labor market, individuals take on jobs that do not fully utilize their skillset? Although these workers will now be represented optimistically by the unemployment rate, such concessions are pernicious both from the perspective of wasted human capital and civil happiness. This paper proposes a strategy to measure this mismatch, the degree to which the practicing occupation of individuals in an economy diverge from their preferred role.

We cast our approach in two parts. In part one, we determine an individual's preferred role from their employment history by implementing a suite of classification models. In part two, we migrate to a model that (a) better captures the ″noise″ in occupational data and (b) facilitates conclusions at the economy, rather than the individual, level. By using a hidden Markov model, we distinguish between an ″employment history″ and a ″career path″ which reflect the difference between practicing and preferred roles. We use an Expectation-Maximization algorithm called Baum-Welch to determine the emission and transition matrices that characterize the US economy. We frame our results in terms of labor market noisiness (how frequently preferred and practicing occupations diverge) and rigidity (how often preferred occupation remains constant from one year to the next).

2 Background and Related Work

Through our research we found literature pertaining to predicting and recommending future occupations using classification methods. Particuarly, Paparrizos, Cambazoglu and Gionis discuss their work using supervised machine learning to recommend suitable jobs for people ([10]). Similarly to our work, they use past job transitions, but they also use other factors including job satisfaction and the employee's educational background. Lou, Ren and Zhao consider the problem of predicting the optimal future career path based on a person's employment history. They use Markov Chains and utilize clustering to generate an optimal path to a queried goal position.

The problem we are motivated to solve, however, is at the economy rather than the individual level. Furthermore, although the papers discussed drill down to the level of specific job title, we are interested in distinguishing occupations at the broader level at which they delineate different skillsets. We adopt a hidden Markov model to rein our analysis to the desired scope. Our approach in this respect is most similar to Yamaguchi, who applies a Kalman filter to the dataset we work with to condense detailed occupations into a single indicator of skill growth ([12]). We synthesize information across individuals by applying the Baum-Welch algorithm. To do this, we must apply a version of the algorithm modified to accept multiple observation sequences for which Rabiner serves as our main technical resource ([11]).

3 Problem Specification and Approach

Our aim was two-fold: first, to apply classification techniques to predict an individual's current occupation based on his career in the previous three years, and second, to generate probabilities of transitioning between careers using a Hidden Markov Model approach. For the former approach, we used three different types of classification techniques: Naive Bayes, k-Nearest Neighbors and Random Forest. For the latter approach, we used the Baum-Welch algorithm in order to generate the desired transition probabilities.

3.1 Data Collection and Preprocessing

The dataset we used came from the U.S. Bureau of Labor Statistics's 1997 National Longitudinal Survey of Youth [9]. This dataset contains the responses collected from interviews with $8,984$ people born between 1980 and 1984, on a variety of topics including childhood, employment, health, dating, and substance abuse. For our purposes, we only considered the employment data collected. This file contained the occupation code number for each respondent in the study for each year from 1997 through 2013.

The number of occupation codes used by the Bureau of Labor Statistics surpassed 500; therefore, due to the difficulties using Hidden Markov Models with such large state spaces, we considered job categories instead. We adopted the 25 job categories contained in the occupation code lookup table included with the dataset (see A.1).

We also had to contend with missing data in our dataset. There is much consideration in machine learning literature in regards to handling missing values in data sets. The methods used depend heavily on whether the data is missing completely at random (MCAR), missing at random (MAR), or not missing at random (NMAR) ([1]). The main methods for treating missing values fall into the following categories: discarding samples with missing data, parameter estimation, imputation, and reduced-feature models. Based on the characteristics of our data set and our research into why there was missing data, we decided imputation the most logical approach. Our imputation scheme was as follows: if respondent r was missing career data for year y, we found the most recent valid career group in previous years, old_career, and the most recent valid career group in future years, new_career. If new_career = old_career, then we felt that it was reasonable to assume that respondent r's career had not changed in the interim and imputed the value. If there was no future year data, we still imputed as described, now just based on the most recent previous data. Finally, we omitted missing data that occurred in the years before any valid career data, as we could not justify an imputation strategy in that case.

For our classification problem we divided up each instance in our dataset into (overlapping) groups of four consecutive years. The first three years of occupation groups were used to predict the final/fourth year's occupation group.

3.2 Naive Bayes

We implemented scikit-learn's Naive Bayes model. We tried different distributions for the likelihood of the features: Gaussian and Multinomial. In our specific problem, the "naive" assumption of independence between pairs of features given the label, is most likely flawed. It's reasonable that there would be some conditional dependencies between your job last year and two years ago, given your current job. However, as McCallum and Nigam conclude, "while this assumption is clearly false in most real-world tasks, Naive Bayes often performs classification very well" ([8]).

3.3 kNN

k-Nearest Neighbors (kNN) is an effective lazy learning algorithm based on the idea that "nearest patterns to a target pattern x, for which we seek the label, deliver useful label information." It is an excellent model when there are low dimensions and many training examples ([6]). We implemented scikit-learn's k-nearest neighbors classifier. This classifier uses a default Euclidean metric to calculate the similarity between instances.

3.4 Random Forest

Random Forest is a popular machine learning algorithm based on two important ideas: Decision trees and bootstrap sampling. Random Forests are composed of many, i.e. a "forest" of, Decision Trees. Decision trees are constructed by finding the "best" attribute in the data set to install at the current node, partitioning the data based on that attribute and continuing recursively until there are either no more features or when each branch is pure in terms of label. The "best" attribute is determined by first calculating the entropy for each attribute. If a given data set (S) only has two different labels, 1 and 0, and a set has 100q% training examples with label 1 and 100p% training examples with label 0, then $Entropy(S) = -q\log_2 q - p\log_2 p$. Next we define Information Gain, as $Gain(S, A) = Entropy(S) - (\sum_{i=0}^{1} P(i)Entropy(S_i))$. At each of the recursive steps, we choose the remaining attribute that has the largest information gain as our decision node ([3]).

Random Forests are composed of a number of Decision Trees. Each tree is trained on a bootstrap sample of the data set (replacement is used). Each tree's decision nodes are picked out of a randomly chosen attribute sample, and then when predicting, either the majority (classification) or mean (regression) of the trees' label predictions is chosen. This is an example of a meta-algorithm called bagging, which operates by "taking in a base learning algorithm and invoking it many times with different training sets" produced from bootstrap samples of the data ([4]). We implemented scikit-learn's Random Forest Classifier.

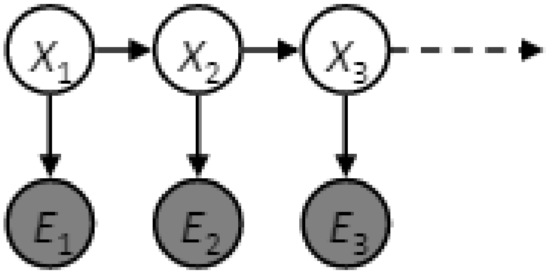

Figure 1: Graphical model describing career paths

3.5 Baum-Welch

Let X_t represent latent preferred occupation and E_t represent practicing occupation at time t. We model an individuals career path using an HMM as shown in Figure 1.

We apply the Baum-Welch algorithm to infer the emission and transition matrices indicative of noise and rigidity underlying the economy. Baum-Welch is an expectation-maximization algorithm which iterates to a solution by alternatively inferring hidden states and then choosing emission and transition probabilities that maximize the likelihood of those states occurring. To synthesize information across individuals, we implement a version of Baum-Welch tailored to multiple sequences of observations.

4 Experiments

4.1 Classification

We divided the test set into two parts:

1. Training/Validation Set: 80% of the data, used to train and test our model for hyperparameter selection. This set is divided up into training and validation sets during the k-fold cross-validation process.

2. Test Set: 20% of the data, used to test the resulting models

We then performed 5-fold cross-validation in order to tune the hyperparameters for our models. For the Naive Bayes Multinomial model we tuned α, the Laplace smoothing parameter. For the Naive Bayes Gaussian model, there were no parameters to tune. For k-nearest neighbors we tuned n, the number of neighbors to consider, and for the Random Forest classifier we tuned *n_estimators*, the number of Decision Trees to use.

For each model we followed the same protocol: for each of the 5-folds, we trained our model on the training set, then recorded our model's accuracy on the validation set. These five accuracy values were averaged to give the model's score for the given hypermarameter setting. Then, the hyperparameters that maximized the accuracy of the model was chosen. Finally, the chosen model's accuracy on the held-out test set was evaluated.

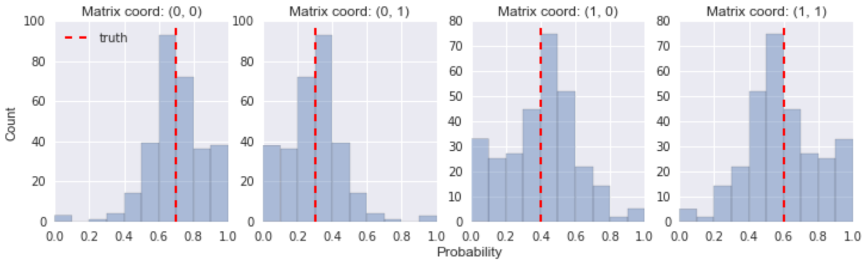

Figure 2: Transition Probabilities

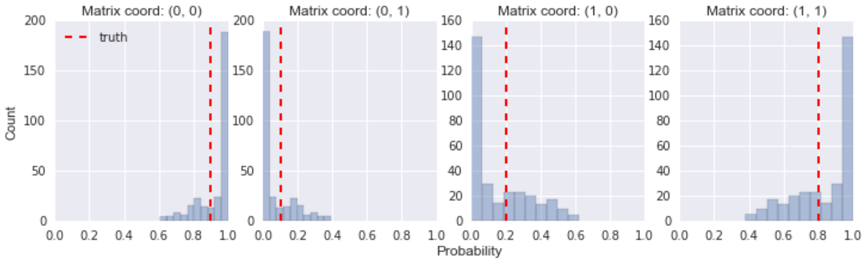

Figure 3: Emission Probabilities

4.2 Baum-Welch

We begin by implementing the standard conception of Baum-Welch described in Bilmes ([2]) ourselves. To evaluate our implementation, we simulate a sequence of 100 observations using arbitrary emission, transition and start probabilities:

$$start\_probabilities = \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}$$

$$transition\_probabilities = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

$$emission\_probabilities = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

Occluding these probabilities, we then apply our implementation of Baum-Welch to determine if we are able to recover them using the observation sequence alone. Figure 2 and 3 plot our estimates of the transition and emission probabilities after repeating this process 300 times.

The majority of our estimates, particularly for the transition probabilities, are concentrated around the truth. However, probabilities along the diagonal of the matrix are skewed towards one, and off-diagonals towards zero. This is a vestige of the starting probabilities with which we prime Baum-Welch. We formulate our starting guesses as the identity matrix with small noise terms added to all cells to ensure that all probabilities are initially non-zero. Should Baum-Welch converge to a local maximum, it is unsurprising that it should find one that strikes a middle-ground between the truth and identity.

We argue that the identity matrix is appropriate as a starting guess as occupations tend to be relatively sticky from one year to the next. When we move to real data, we chose starting matrices by setting diagonal entries equal to one third and then add small noise terms to the remaining cells so that each row sums to one.

This approach is appropriate for a single individual's career path. However, we wish to infer the emission and transition probabilities that characterize the entire economy. The graphical model of our data is a bundle of HMM chains wherein each chain describes an individual's career path. In addition, each chain is independent of every other one but they all subscribe to the same initial, transition and emission probabilities. We migrate from our implementation of Baum-Welch to the hmmlearn package which modifies Baum-Welch for multiple observation sequences as described in Rabiner ([11]).

5 Results

5.0.1 Classification

Through 5-fold cross validation we determined that the hyperparameters for the optimal Multinomial Naive Bayes model was $\alpha = 1.0$. With this hyperparameter, the model had an accuracy of 0.20528 on the test set. The Gaussian Naive Bayes Model, which did not take any hyperparameters, had an accuracy of 0.3395 on the test set.

Through 5-fold cross validation we determined that the optimal number of neighbors for the k-nearest neighbors algorithm was 9. This model had an accuracy of 0.73162 on the test set. Due to this model performing very similarly to the Random Forest classifier on the validation set (and beating the Naive Bayes models by a large margin), we decided to apply bagging to the k-nearest neighbors model. We again ran 5-fold cross validation, now with 9 neighbors, but a variable value for $n\_estimators$. We found that the optimal number of estimators was 70. This model had an accuracy of 0.73956 on the test set.

Through 5-fold cross validation we determined that the optimal number of decision trees in the Random Forest Model was 250. This model produced an accuracy of 0.74100 on the test set. Using this model, we were also able to investigate which features were most important, through the built in ”feature_importances_” attribute. This attribute produces a normalized vector containing a feature importance value for each feature. We found that the Random Forest model had learned to assign a lot of importance (52%) to the most recent career (the third feature), and then the importance declined with time. The first feature had a feature importance of 0.198, and the second feature had a feature importance of 0.278. Therefore the model was learning the common sense notion that your career last year is the best predictor of your current career.

See Table 1 for a comparison of results between our models. Ultimately, k-Nearest Neighbors and Random Forest outperformed the Naive Bayes models significantly. However, the Gaussian Model did perform better than the Multinomial one. While bagging improved the k-nearest neighbors accuracy, it did so by only a small amount. The Random Forest model performed the best, but the differences between the accuracy of the k-Nearest Neighbors, k-Nearest Neighbors Bagged and Random Forest models are not statistically significant.

5.0.2 Baum-Welch

The emission and transition matrices, produced by the Baum-Welch algorithm, associated with the 1997 and 1979 cohorts are pictured below. Cell (X_1, E_1) in the emission matrix reflects the probability that an individual whose preferred latent occupation is X_1 will practice occupation

Table 1: Comparison of Test Set Accuracy Between Models

	Accuracy
Multinomial Naive Bayes	0.2053
Gaussian Naive Bayes	0.3395
k-Nearest Neighbors	0.7316
k-Nearest Neighbors Bagged	0.7396
Random Forest	0.7410

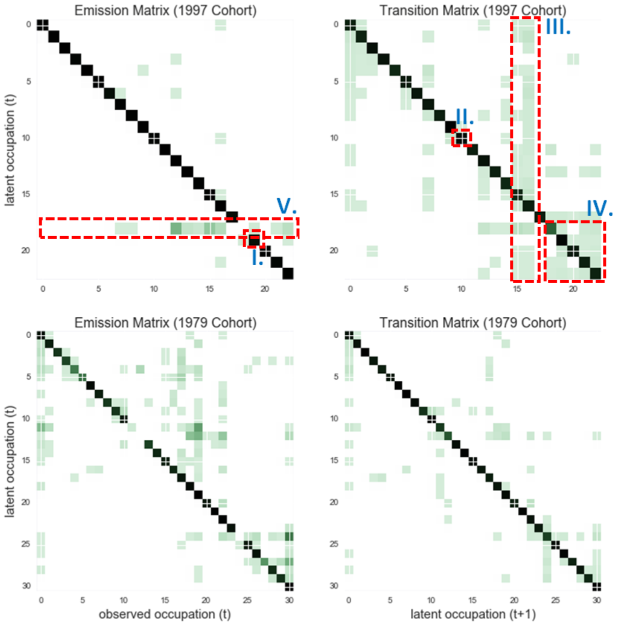

Figure 4: Occupational emission and transition matrices by cohort

E_1. Cell (X_1, X_2) in the transition matrix reflects the probability that an individual whose latent occupation is X_1 at time t, will transition to X_2 at time $t + 1$.

There are two kinds of interpretation that fall out of such matrices. One can draw conclusions at the occupational level:

I. As a sanity check, more noisy occupations (such as Construction), have less mass on the diagonal of the emission matrix.

II. More rigid occupations (such as Healthcare) have more mass on the diagonal of the transition matrix.

III. The swathes of green down columns 15 and 16 of the transition matrix correspond to Sales and Office and Administrative Support. This suggests that these occupations are accessible to individuals of a variety of occupational backgrounds pointing to low barriers-to-entry.

IV. The cluster of green cells at the bottom right of the transition matrix suggests a large degree of mobility between Construction, Installation, Maintenance and Repair, Production and Transport and Material Moving.

Note that we are alerted to a preprocessing step that would further refine our results. The 18th row in the emission matrix [V.] corresponds to Fishing and Hunting, Forest and Logging. The distribution of probabilities along the row is most likely the result of us only having 62 instances for training. A similar argument can be made for rows 11, 12 and 24 of the emission matrix for the 1979 cohort. A refinement would be to either merge these occupations with their closest analog, or drop the handful of individuals that practice them altogether.

Alternatively, one can draw conclusions at the economy level. The proportion of total probability mass **off** the diagonal of the emission matrix is indicative of noise. Likewise, the proportion of total probability mass **on** the diagonal of the transition matrix is indicative of rigidity. We find that Youth in 1997 tend to have less noisy career paths; they are less likely than youth in 1979 to temporarily take on a role that diverges from their latent career choice. However, examining the proportion of probability mass along the diagonal of the transition matrix, we find that youth in 1997 also tend to have less rigid career paths; they are more likely than Youth in 1979 to permanently change careers. Such matrices may be of interest to recession economists (e.g. is a cohort that experiences a severe recession more likely to be malleable with respect to their career choice as has been the case for Youth in 1997). They may also be of interest to job-seekers (e.g. which occupation has lowest barriers-to-entry).

6 Discussion

Classification techniques turned out to have more of a challenge predicting career trajectory than we hypothesized. Even with a drastically reduced number of labels (because we converted to 25 occupation groups versus the 500 specific occupation types), our best model only had an accuracy of approximately 75%. As discussed above, the Random Forest model did learn to give the most importance to the most recent career data known, which validates our intuition about career paths. This may indicate a weakness in the way occupations were grouped by the U.S. Bureau of Labor Statistics. While we do not know which algorithm they used to group together different occupations, because individuals rarely completely change their occupation type (a Protective Service employee will probably not be a Computer and Mathematical employee next year), we suspect that better occupation grouping could be produced. Lou, Ren and Zhao ([7]) encountered this same problem of their models being weak due to poor clustering (although they performed their own clustering), so the pre-processing clustering (perhaps with the k-means algorithm) would be an interesting problem to consider in future work.

Regarding Baum-Welch we arrive at two sets of conclusions. At the occupation-level we are able to separately map how individuals in different occupations behave with respect to temporary versus permanent career transitions. At the economy level, we are able to look at a cohort of workers and comment on the rigidity and noise that characterize the prevailing labor market. There are two refinements that would improve interpretability: (1) as discussed above, pre-treat those occupations which are rare in our dataset in advance of analysis (2) between the 1979 and 1997 cohorts, the Bureau of Labor Statistics switched classification systems for occupation codes. Generating an occupation grouping system which could be applied to both cohorts would facilitate comparison between labor market conditions in specific occupations across cohorts.

A Appendix

A.1 Occupation Groupings

1. Management
2. Business and Financial Operations
3. Computer and Mathematical
4. Architecture and Engineering
5. Life, Physical and Social Sciences
6. Community and Social Service
7. Legal
8. Education, Training and Library
9. Arts, Design, Entertainment, Sports and Media
10. Healthcare Practitioner and Technical
11. Healthcare Support
12. Protective Service
13. Food Preparation and Serving Related
14. Building and Grounds Cleaning and Maintenance
15. Personal Care and Service
16. Sales
17. Office and Administrative Support
18. Agriculture Workers
19. Fishing and Hunting, Forest and Logging
20. Construction and Extraction
21. Installation, Maintenance, and Repair
22. Production
23. Transportation and Material Moving
24. Military
25. Unemployed

A.2 System Description

All our code is contained in the github repo at this link: $https : //github.com/rohanthavarajah/cs182\_modeling\_ca$

The code for the data cleaning and classification parts of the project are contained in ”Data Cleaning and Classification.ipynb.” The file shows the implementation of the different classification methods as well as the accuracy scores that we reported.

Our implementation of Baum-Welch and simulations can be found in ”Part I. Baum-Welch from Scratch - experimental.ipynb”. ”Part II. Baum-Welch Applied - main results.ipynb” applies the ”hmmlearn” package to real occupational data and yields the emission and transition matrices of interest.

A.3 Group Makeup

Courtney handled the data cleaning and imputation and worked on trying to find suitable Baum-Welch packages. Rohan wrote our own implementation of the Baum-Welch algorithm and applied the hmmlearn package to our data (as well as NLTK and tensorflow based packages which produced unsatisfactory results). Courtney implemented the classification algorithms. We both wrote and edited the paper.

References

[1] Gustavo EAPA Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

[2] Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. 1998.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[5] Jason Faberman and Bhash Mazumber. Is there a skills mismatch in the labor market? 2012.

[6] Oliver Kramer. *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013.

[7] Yu Lou, Ran Ren, and Yiyang Zhao. A machine learning approach for future career planning.

[8] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[9] U.S. Bureau of Labor Statistics. National longitudinal survey of youth 1997.

[10] Ioannis Paparrizos, B. Barla Cambazoglu, and Aristides Gionis. Machine learned job recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys ’11, pages 325–328, New York, NY, USA, 2011. ACM.

[11] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. 1988.

[12] Shintaro Yamaguchi. Tasks and heterogeneous human capital. 2011.