Learning to Form Skill-based Teams of Experts

Radin Hamidi Rad
radin@ryerson.ca
Ryerson University, Canada

Hossein Fani
hfani@uwindsor.ca
University of Windsor, Canada

Mehdi Kargar
kargar@ryerson.ca
Ryerson University, Canada

Jaroslaw Szlichta
jarek@ontariotechu.ca
Ontario Tech University, Canada

Ebrahim Bagheri
bagheri@ryerson.ca
Ryerson University, Canada

ABSTRACT

We focus on the composition of teams of experts that collectively cover a set of required skills based on their historical collaboration network and expertise. Prior works are primarily based on the shortest path between experts on the expert collaboration network, and suffer from three major shortcomings: (1) they are computationally expensive due to the complexity of finding paths on large network structures; (2) they use a small portion of the entire historical collaboration network to reduce the search space; hence, may form sub-optimal teams; and, (3) they fall short in sparse networks where the majority of the experts have only participated in a few teams in the past. Instead of forming a large network of experts, we propose to learn relationships among experts and skills through a variational Bayes neural architecture wherein: *i*) we consider all past team compositions as training instances to predict future teams; *ii*) we bring scalability for large networks of experts due to the neural architecture; and, *iii*) we address sparsity by incorporating uncertainty on the neural network's parameters which yields a richer representation and more accurate team composition. We empirically demonstrate how our proposed model outperforms the state-of-the-art approaches in terms of effectiveness and efficiency based on a large DBLP dataset.

ACM Reference Format:
Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412140

1 INTRODUCTION

There has been increasing interest in the problem of forming a team of experts from an expert network. Forming a team of experts was first introduced in Lappas et al. [7] where the authors proposed optimization functions to measure the communication cost of a team. More recent works on the team formation problem focus on addressing other objectives, e.g., personnel cost and expertise level of the experts, in addition to communication costs, which turns the problem into a multi-objective optimization problem. For instance, Kargar et al. [5] propose to optimize communication cost and personnel cost when the project has a certain budget with known salaries of the team members. Anagnostopoulos et al. [2] introduce the problem of team formation with outsourcing in which tasks arrive in realtime and therefore, neither the number nor the composition of tasks are known *a priori*. The goal of such work is to minimize the overall cost paid for hiring and outsourcing. Rahman et al. [9] also study the problem of forming groups of crowd workers and propose an optimization model for task assignment. However, unlike [2], Rahman et al. [9] take communication costs among crowd workers into account and use the diameter and the sum of distances to measure communication cost.

Nevertheless, existing works fall short with respect to scalability, since the proposed solutions are based on the graph representation of the expert network, in one way or another, and optimization is performed by computationally expensive search methods on the graph. Such methods define collaboration among team members when there is a direct or indirect past collaboration between the experts and optimization happens for a team based on, e.g., the diameter of the subgraph. These methods essentially rely on the computation of shortest path between all pairs of experts within the network. This is computationally prohibitive for large networks. Furthermore, expert networks are often dynamic as collaborations happen in real-time. In such cases, the updates as a result of new collaborations leads to frequent changes in shortest paths, which requires expensive recalculation of indexed shortest paths for all pairs of experts. Although advanced shortest path indexing techniques (such as the one by Akiba et al. [1]) provides fast calculation of proximity functions, the output teams are still sub-optimal and lack accuracy as well as coverage for the given required skills based on the heuristic nature of the solution.

We focus on finding optimal groups of experts, such as *a group of co-authors*, that satisfy two main criteria: (1) maximal coverage for a set of required skills, e.g., {*deep learning*, *computer vision*}, and (2) effective collaboration history among team members, e.g., *past joint publications*. We propose to learn feature representations over a set of teams of experts using a *variational Bayesian neural* architecture[1]. Instead of learning representations over the expert network, we consider each observed team from past collaborations over a set of skills as a training instance. Unlike previous approaches, where the search for the best team is performed over the graph representation of the expert network, we search for variational distributions of experts and skills in the context of a team whose expectation can be

[1]Codebase available at *http://anonymous.4open.science/r/b8584256-b23a-42a1-8e8b-b901826c475c/*

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412140

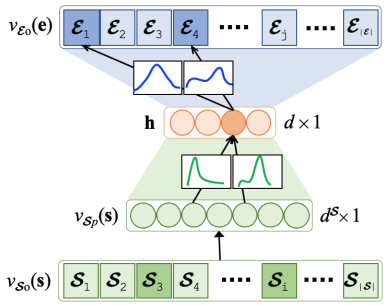

Figure 1: Variational Baysian neural architecture for $\mathcal{S}_p \rightarrow \mathcal{E}_o$.

efficiently approximated with variational posterior to draw probable teams, which are more effective and efficient compared to graph-based methods.

To the best of out knowledge, Sapienza et al. [11] is the only work that uses a neural architecture to address the team formation problem. However, this work employs a plain (non-variational) autoencoder neural network, which has been shown to be prone to overfitting and is not able to correctly assess the uncertainty in the training data [3]. This is especially visible, when there is sparsity in the training data where only a few teams have been observed in the past for a given set of skills. Our proposed model shows notable *effectiveness* and *efficiency* improvements when evaluated over the DBLP dataset against the state-of-the-art neural and graph-based methods.

2 TEAM LEARNING

Given a set of required skills to form a team of experts, we employ a variational Bayesian neural network to form the optimal team. We introduce uncertainty on the neural network parameters (weights) and subsequently in the final predictions using variational Bayesian neural architecture, which not only yields a better regularisation via probabilistic weights, but also leads to a richer representation and more accurate predictions [3]. Despite widespread use in other domains, such as language modeling [12], to the best of our knowledge, variational neural networks have not yet been used for team formation. We show that variational neural networks are well-suited and a closer proxy for finding effective teams.

We consider the generative process similar to the deep latent Gaussian model with layers of latent variables [10]. Let $\mathcal{S} = \{\mathcal{S}_i\}$ and $\mathcal{E} = \{\mathcal{E}_j\}$ be the sets of skills and experts, respectively, $(\mathbf{s}, \mathbf{e})$ is a team of experts $\mathbf{e} \subseteq \mathcal{E}; \mathbf{e} \neq \emptyset$, which has been formed with respect to skills $\mathbf{s} \subseteq \mathcal{S}; \mathbf{s} \neq \emptyset$, and $\mathcal{T} = \{(\mathbf{s}, \mathbf{e})\}$ indexes all teams. Our task is to learn $f : \mathcal{P}(\mathcal{S}) \rightarrow \mathcal{P}(\mathcal{E})$, a mapping function of parameters $\boldsymbol{\theta}$ from skills powerset to experts powerset, such that $f(\mathbf{s}; \boldsymbol{\theta}) = \mathbf{e}$.

2.1 Variational Inference

We aim at optimizing the maximum a posteriori of $\boldsymbol{\theta}$ in $f(\cdot, \boldsymbol{\theta})$, i.e., $p(\boldsymbol{\theta}|\mathcal{T})$ where f is a multi-layer variational neural network, $\mathcal{T}$ is the set of teams whose elements $(\mathbf{s}, \mathbf{e})$ consist of an input skill subset $\mathbf{s}$ and a target expert subset $\mathbf{e}$, which are assumed to be drawn independently from a joint distribution $p(\mathbf{s}, \mathbf{e})$, and $\boldsymbol{\theta}$ are real-valued parameters, or weights. By Bayes theorem,

$$p(\boldsymbol{\theta}|\mathcal{T}) \propto p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad \text{where} \quad p(\mathcal{T}|\boldsymbol{\theta}) = \prod_{(\mathbf{s},\mathbf{e})\in\mathcal{T}} p(\mathbf{e}|\mathbf{s}, \boldsymbol{\theta}) \quad (1)$$

and $p(\boldsymbol{\theta})$ is the prior probability of weights. Maximizing $p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ gives the maximum a posteriori estimate of $\boldsymbol{\theta}$. The true prior probability of weights $p(\boldsymbol{\theta})$, however, cannot be calculated analytically or efficiently sampled, and as such, we approximate it by a more tractable distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$ with multivariate diagonal Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. The elements of $\boldsymbol{\sigma}$ are a diagonal covariance matrix which means that weights $\boldsymbol{\theta}$ are assumed to be uncorrelated. Unlike non-variational neural networks, where weights are parameterized with real values (point estimate), herein each weight $\theta_i \in \boldsymbol{\theta}$ is drawn from a Gaussian distribution with a separate mean μ_i and variance σ_i^2 (uncertainty in weights); hence, the number of parameters to learn are doubled when training our Bayesian neural network via variational inference.

2.2 Objective Function

To estimate the true posterior $p(\boldsymbol{\theta})$ by $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})$, we minimize the Kullback-Leibler divergence between q and p with regard to the Gaussian mean and variance vectors as suggested by Graves [4]:

$$\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})||p(\boldsymbol{\theta}|\mathcal{T})) = \int q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma}) \log[\frac{q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})}{p(\boldsymbol{\theta}|\mathcal{T})}]\mathrm{d}\boldsymbol{\theta} \quad (2)$$

$$= \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})} \log[\frac{q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})}{p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}p(\mathcal{T})] \quad (3)$$

$$= \underbrace{\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})||p(\boldsymbol{\theta})) - \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\sigma})} \log p(\mathcal{T}|\boldsymbol{\theta})}_{\textit{variational free energy}} + \log p(\mathcal{T}) \quad (4)$$

In order to minimize $\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\sigma})||p(\boldsymbol{\theta}|\mathcal{T}))$, we need to minimize the first two terms in Eq. 4, known as variational free energy, given that the log marginal likelihood $\log p(\mathcal{T})$ does not depend on $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

2.3 Model Architecture

We now describe the details of our proposed variational Bayesian neural network. We predict a team of experts $\mathbf{e} \subseteq \mathcal{E}$ for a given skill subset $\mathbf{s} \subseteq \mathcal{S}$ by a mapping function $f(\mathbf{s}; \boldsymbol{\theta})$ using a variational neural network of one dense variational hidden layer $\mathbf{h}$ of size d, without loss of generality to multiple hidden layers, with input layer $v_{\mathcal{S}}(\mathbf{s})$ and output layer $v_{\mathcal{E}}(\mathbf{e})$:

$$\mathbf{h} = \pi_1(\boldsymbol{\theta}_1 v_{\mathcal{S}}(\mathbf{s}) + \mathbf{b}_1) \quad (5)$$

$$v_{\mathcal{E}}(\mathbf{e}) = \pi_2(\boldsymbol{\theta}_2 \mathbf{h} + \mathbf{b}_2) \quad (6)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}_1 \cup \boldsymbol{\theta}_2 \cup \mathbf{b}_1 \cup \mathbf{b}_2 \quad (7)$$

where, $\pi_.$ is a nonlinear activation function, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ whose means and variances are estimated by minimizing variational free energy, $v_{\mathcal{S}}(\mathbf{s})$ is the vector representation of the input skill subset $\mathbf{s}$, and $v_{\mathcal{E}}(\mathbf{e})$ is the vector representation of output expert subset $\mathbf{e}$ given a team $(\mathbf{s}, \mathbf{e}) \in \mathcal{T}$.

To build the vector representations for skill subset $\mathbf{s}$ and expert subset $\mathbf{e}$, we have employed two approaches: occurrence vector representation (o) and pretrained dense vector representation (p). The occurrence vector representation of a given skill subset $\mathbf{s}$ is a boolean vector of size $|\mathcal{S}|$, i.e., $v_{\mathcal{S}_o}(\mathbf{s}) \in \{0, 1\}^{|\mathcal{S}|}$ where $v_{\mathcal{S}_o}(\mathbf{s})[i] = 1$ if $\mathcal{S}_i \in \mathbf{s}$ and 0 otherwise. Likewise, $v_{\mathcal{E}_o}(\mathbf{e}) \in \{0, 1\}^{|\mathcal{E}|}$ where $v_{\mathcal{E}_o}(\mathbf{e})[j] = 1$ if $\mathcal{E}_j \in \mathbf{e}$ and 0 otherwise. In order to obtain pretrained dense vector representations of skill and expert subsets,

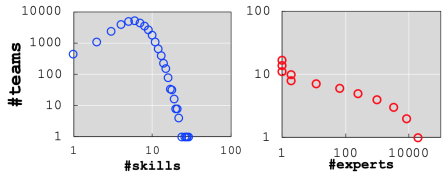

Figure 2: Distribution of teams (articles).

we adapt paragraph vectors by Le and Mikolov [8]. Specifically, we consider each team $(\mathbf{s}, \cdot) \in \mathcal{T}$ as a document and skills $\mathcal{S}_i \in \mathbf{s}$ as the document words. Using distributed memory model, we map skill subsets into a real-valued embedding space of $d_{\mathcal{S}}$ dimensions, i.e., $v_{\mathcal{S}_p}(\mathbf{s}) \in \mathcal{R}^{d_{\mathcal{S}}}$. Likewise for $v_{\mathcal{E}_p}(\mathbf{e})$, considering each team $(\cdot, \mathbf{e}) \in \mathcal{T}$ as a document and experts $\mathcal{E}_j \in \mathbf{e}$ as the document words, we map expert subsets into a real-valued vectors of size $d_{\mathcal{E}}$, i.e., $v_{\mathcal{E}_p}(\mathbf{e}) \in \mathcal{R}^{d_{\mathcal{E}}}$. In Figure 1, we show our proposed model when pretrained dense vector representation for skill subset ($v_{\mathcal{S}_p}(\mathbf{s})$) and occurrence vector representation for expert subset ($v_{\mathcal{E}_o}(\mathbf{e})$) have been used for input and output layers, respectively.

As seen in Figure 1, given a skill subset $\mathbf{s}$, we are able to form a team of experts ($\mathbf{s}$, $\mathbf{e}$=?) by entering the pretrained vector representation of the skill subset $\mathbf{s}$ in the input layer. Each parameter (weight) $\theta_i \in \boldsymbol{\theta}$ in the neural network is then drawn from its estimated Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i^2)$. In the output layer, the probability of membership for all experts $\mathcal{E}_j \in \mathcal{E}$ is finally calculated based on Eq. 5 and Eq. 6 and the team members $\mathbf{e} \subseteq \mathcal{E}$ are those experts who have the top-k highest probabilities.

3 EXPERIMENTS

3.1 Setup

3.1.1 Dataset. As suggested in [7], we choose DBLP as the benchmark. We consider each author to represent an expert and the authors of each publication to form a team. We form the skill set $\mathcal{S}$ from the set of keywords extracted from the title of the publications as suggested by [7, 14]. Having applied preprocessings such as stemming and stop word elimination, the top-2,000 {1,2,3}-gram keywords with the highest tf-idf scores form the skill set. In summary, $|\mathcal{T}| = 33,002$ teams over $|\mathcal{S}| = 2,000$ skills (keywords) and $|\mathcal{E}| = 2,470$ experts (authors) are included in our gold standard dataset whose overall distribution by the number of skills and experts is shown in Fig. 2.

3.1.2 Evaluation Strategy. To evaluate the effectiveness of our model for identifying expert teams, we performed 10-fold cross validation where publications of each author were randomly distributed across different folds. Having trained the mapping function $f : \mathcal{P}(\mathcal{S}) \rightarrow \mathcal{P}(\mathcal{E})$ on teams in the training folds, we compare the predicted team $f(\mathbf{s}; \boldsymbol{\theta}) = \mathbf{e}'$ with the observed team $\mathbf{e}$ in $(\mathbf{s}, \mathbf{e})$ from the test fold. We report the average performance using *i*) ranking metrics: mean average precision (map), mean reciprocal rank (mrr), normalized discounted cumulative gain (ndcg), and recall for the top-k experts of highest probabilities in the ranked list of predictions. Further, we measure the scalability of our proposed model compared to the baselines. We report the execution time of the methods when using same computing power with 12 cpu cores, 64gb memory, and a gpu unit of 3,584 cores and 11gb memory.

3.1.3 Baselines. We evaluate our model vs. the following baselines:

Graph-based. `Zihayat et al.` [14]'s work is among the latest based on graph search techniques. Therein, each team is represented as a spanning subtree in the expert network and optimization is performed by minimizing the sum of distances among team members. We also select the pioneering work by `Lappas et al.` [7] where teams are subtrees with the minimum diameter (largest shortest path between any two nodes).

Collaborative filtering. The problem of team formation can be viewed as a recommendation task where team members are recommended by a given set of required skills. We adopt recurrent recommender network (`rrn`) by Wu et al. [13], which is a neural collaborative filtering approach as well as `svd++` [6], a well-known matrix factorization method for recommendation. We performed grid search over bin and factor sizes in {10, 20, ..., 100} to select the best settings and other hyperparameters were set to default.

Neural-based. The only other neural-based method for the problem of team formation is by `Sapienza et al.` [11]; an autoencoder of one hidden layer of size 1024 to learn the adjacency matrix representing the experts directed network. The `rrn` [13] method can also be considered as a baseline in this category.

Proposed model (vBnn). Our proposed model is a variational Baysian neural network (`vBnn`) with the hidden layer of size $d = 100$, relu and sigmoid are the activation functions for the hidden layer (π_1) and the output layer π_2, respectively. The top-k experts with highest probabilities would form the predicted team given the input skills. Optimization was performed using Adam and learning rate $\eta = 0.001$. Regarding the vector representations for input skills and output experts, i.e., $v_{\mathcal{S}}(\mathbf{s})$ and $v_{\mathcal{E}}(\mathbf{e})$, we have studied both occurrence vector representation (o) and pretrained dense vector representation (p) (see §2.3), which results in four different variations. We denote $\mathcal{S}_o$, $\mathcal{S}_p$, $\mathcal{E}_o$, and $\mathcal{E}_p$ when referring to them.

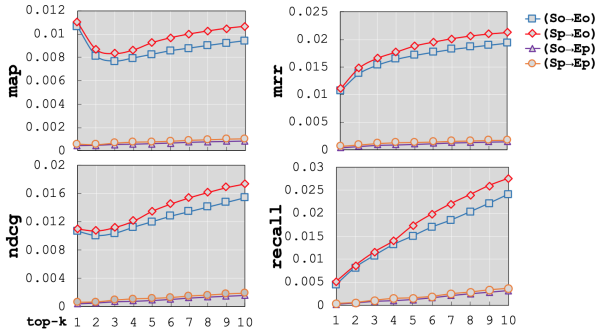

Figure 3: The effect of vector representation.

3.2 Results

Effectiveness. We analyze the effect of vector representation on our proposed `vBnn` and report the performance of the four variations in Fig. 3. The overall trend indicates that the performance of the proposed model for all its variations increases with the number of predicted experts to form a team in terms of all ranking metrics. Specifically, the $(\mathcal{S}_p \rightarrow \mathcal{E}_o)$ variation is the best variation where skills are represented by dense pretrained vectors in the input layer, i.e., $\mathcal{S}_p$, and experts are represented as occurrence vector to form a team in the output layer, i.e., $\mathcal{E}_o$. While the same trend is observed up to top-100 experts, we only report up to top-10, which is the maximum number of experts in a team in our dataset.

Next, we compare the best variation of our proposed model against the baselines at their best settings in Fig. 4. `vBnn` consistently outperforms all the other baselines in terms of all ranking metrics.

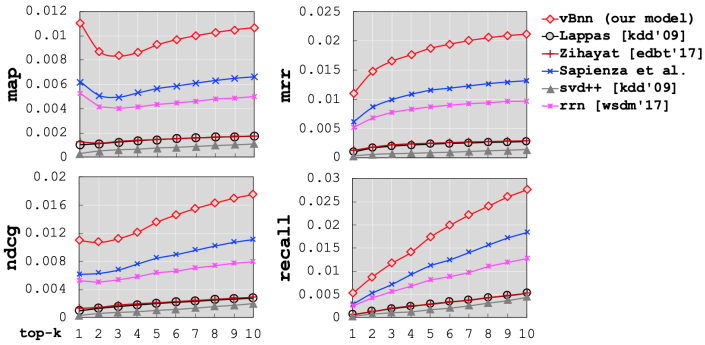

Figure 4: The performance of our model vs. baselines.

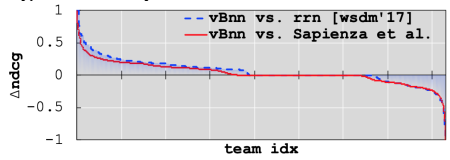

Figure 5: Pairwise help/hurt of vBnn vs. other neural baselines.

Amongst baselines, neural-based methods, i.e., `Sapienza et al.` and `rrn` are the runner-up and graph-based methods are generally far less effective. We attribute the poor accuracy of graph-based methods, including their state of the art, i.e., `Zihayat et al.`, to the fact that they only consider limited subgraphs of the expert network due to the high computational expense of finding all-pairs shortest paths on the expert network. On the other hand, the neural-based methods, including our model, take all past collaboration history into account. vBnn further introduces uncertainty in its weights to avoid overfitting to sparse training set where for majority of experts there are very few teams; an impact that is overlooked in other neural-based method.

To explore how the uncertainty of neurons affected our proposed model, we visualized the average ndcg improvement/degrading for predicting top-{1:10} experts in each team in Fig. 5. vBnn is the most successful in skewing the diagram to the left; showing consistent improvement for a higher number of teams of experts compared to `Sapienza et al.` and `rrn`.

Efficiency. We report the empirical efficiency of our model in comparison with other baselines for the inference phase, i.e., how efficient an already trained model is at the time of predicting experts for an unseen team given a set of input skills. Fig. 6 shows the models' inference elapsed time for the increasing number of input skills and output experts. Overall, the execution time for the graph-based methods is consistently slower than other methods. Also, contrary to the graph-based methods whose execution time increases with the number of input skills, neural-based models maintain steady execution time in all cases, as the graph-based methods start a new partial/full search over the entire experts network to find the optimized subgraph as the final output team. To the contrary, neural models have already learned the optimized distributions of teams over skills and experts in the training phase; thus, in the inference phase, their execution time depends merely on the complexity of their architecture (i.e., the size and number of layers), which is computationally much less taxing compared to the size and order of the experts network.

In Fig. 7, we investigate which neural model achieves reasonable inference accuracy more quickly (with less training time). In terms

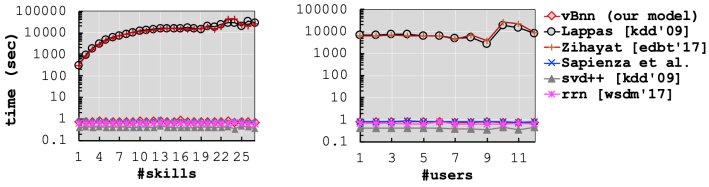

Figure 6: The inference time.

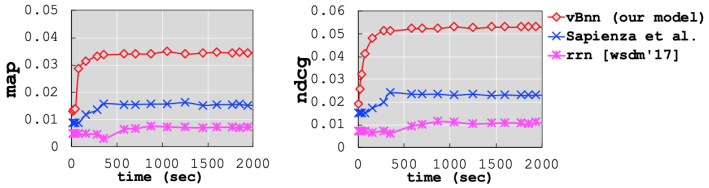

Figure 7: The inference performance vs. training time.

of average map and ndcg for the top-{1:10} predicted experts in a team for a set of input skills, vBnn outperforms other neural-based competitors with much less training time which shows that our method is more robust to over-fitting and can quickly learn from sparse data.

4 CONCLUDING REMARKS

We focused on the problem of team formation within expert networks by satisfying two main constraints, namely maximal coverage of a set of required skills and existing collaboration history between the members of the team. We proposed a variational Bayesian neural network architecture for group formation that is more effective than prior state of the art. We compared the performance of the various variations of our proposed model against each other as well as several categories of baselines and discussed how our method is able to outperform them. As future work, we plan to generalize our approach to support for role-based membership of experts in a team given a set of input skills and associated roles.

REFERENCES

[1] T. Akiba, Y. Iwata, and Y. Yoshida. 2013. Fast Exact Shortest-Path Distance Queries on Large Networks by Pruned Landmark Labeling. In *SIGMOD*. 349–360.
[2] A. Anagnostopoulos, C. Castillo, A. Fazzone, S. Leonardi, and E. Terzi. 2018. Algorithms for Hiring and Outsourcing in the Online Labor Market. In *KDD*. 1109–1118.
[3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. 2015. Weight Uncertainty in Neural Networks. (2015). http://arxiv.org/abs/1505.05424
[4] Alex Graves. 2011. Practical Variational Inference for Neural Networks. In *NIPS*. 2348–2356.
[5] M. Kargar, M. Zihayat, and A. An. 2013. Finding Affordable and Collaborative Teams from a Network of Experts. In *SDM*. 587–595.
[6] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *KDD*.
[7] T. Lappas, L. Liu, and E. Terzi. 2009. Finding a Team of Experts in Social Networks. In *KDD*. 467–476.
[8] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML Workshops*, Vol. 32. 1188–1196.
[9] H. Rahman, S. Roy, S. Thirumuruganathan, S. Amer-Yahia, and Das G. 2019. Optimized Group Formation for Solving Collaborative Tasks. *VLDBJ* 28, 1 (2019), 1–23.
[10] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*. 1278–1286.
[11] A. Sapienza, P. Goyal, and E. Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers in Big Data* 2, 1 (2019), 1–13.
[12] J. Su, S. Wu, D. Xiong, Y. Lu, X. Han, and B. Zhang. 2018. Variational Recurrent Neural Machine Translation. In *AAAI*. 5488–5495.
[13] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. 495–503.
[14] M. Zihayat, A. An, L. Golab, M. Kargar, and J. Szlichta. 2017. Authority-based Team Discovery in Social Networks. In *EDBT '17*. 498–501.