Generative Models for Ticket Resolution in Expert Networks

Gengxin Miao
ECE Department
University of California
Santa Barbara, CA 93106
miao@ece.ucsb.edu

Louise E. Moser
ECE Department
University of California
Santa Barbara, CA 93106
moser@ece.ucsb.edu

Xifeng Yan
CS Department
University of California
Santa Barbara, CA 93106
xyan@cs.ucsb.edu

Shu Tao
IBM T. J. Watson
19 Skyline Drive
Hawthorne, NY 10532
shutao@us.ibm.com

Yi Chen
CSE Department
Arizona State University
Tempe, AZ 85281
yi@asu.edu

Nikos Anerousis
IBM T. J. Watson
19 Skyline Drive
Hawthorne, NY 10532
nikos@us.ibm.com

ABSTRACT

Ticket resolution is a critical, yet challenging, aspect of the delivery of IT services. A large service provider needs to handle, on a daily basis, thousands of tickets that report various types of problems. Many of those tickets bounce among multiple expert groups before being transferred to the group with the right expertise to solve the problem. Finding a methodology that reduces such bouncing and hence shortens ticket resolution time is a long-standing challenge. In this paper, we present a unified generative model, the Optimized Network Model (ONM), that characterizes the lifecycle of a ticket, using both the content and the routing sequence of the ticket. ONM uses maximum likelihood estimation, to represent how the information contained in a ticket is used by human experts to make ticket routing decisions. Based on ONM, we develop a probabilistic algorithm to generate ticket routing recommendations for new tickets in a network of expert groups. Our algorithm calculates all possible routes to potential resolvers and makes globally optimal recommendations, in contrast to existing classification methods that make static and locally optimal recommendations. Experiments show that our method significantly outperforms existing solutions.

Categories and Subject Descriptors

H.1 [**Models And Principles**]: Miscellaneous

General Terms

Algorithms

Keywords

Generative Model, Ticket Resolution, Expert Network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'10, July 25–28, 2010, Washington, DC, USA.

Copyright 2010 ACM 978-1-4503-0055-110/07 ...$10.00.

1. INTRODUCTION

Problem ticket resolution is critical to the IT services business. A service provider might need to handle, on a daily basis, thousands of tickets that report various types of problems from its customers. The service provider's ability to resolve the tickets in a timely manner determines, to a large extent, its competitive advantage. To manage ticket resolution effectively, human experts are often organized into expert groups, each of which has the expertise to solve certain types of problems. As IT systems become more complex, the types of reported problems become more diverse. Finding an expert group to solve the problem specified in a ticket is a long-standing challenge for IT service providers.

In practice, a typical ticket processing system works as follows. A ticket is initiated by a customer or by internal staff, and is subsequently routed through a network of expert groups for resolution. The ticket is closed when it reaches a *resolver group* that provides the solution to the problem reported in the ticket. Figure 1 shows an interaction network between groups with ticket routing examples. Ticket t_1 starts at group A and ends at group D, while ticket t_2 starts at group G and ends at group C (note that we omit the dispatching step in which a ticket is first assigned to the initial group). The sequences $A \rightarrow B \rightarrow C \rightarrow D$ and $G \rightarrow E \rightarrow C$ are called *ticket routing sequences.*

In a large network of expert groups, being able to quickly route a new ticket to its resolver is essential to reduce labor cost and to improve customer satisfaction. Today, ticket

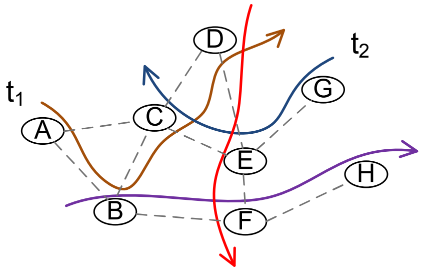

Figure 1: Ticket Routing

routing decisions are often made manually and, thus, can be quite subjective and error-prone. Misinterpretation of the problem, inexperience of human individuals, and lack of communication between groups can lead to routing inefficiency. These difficulties call for computational models that can accurately represent the collaborative relationship between groups in solving different kinds of problems. Such models ought to provide fine-grain information not only to help experts reduce ticket routing errors, but also to help service enterprises better understand group interactions and identify potential performance bottlenecks.

In [17], Shao *et al.* proposed a Markov model-based approach to predict the resolver of a ticket, based on the expert groups that processed the ticket previously. In essence, their approach is a rule-based method, *i.e.*, if group A processed a ticket and did not have a solution, it calculates the likelihood that group B can resolve it. A drawback of that approach is that it is locally optimized and, thus, might not be able to find the best ticket routing sequences. Moreover, it does not consider the contents of the tickets. That is, it uses a "black-box" approach that can neither explain, nor fully leverage, the information related to why group A transfers a ticket to group B, when it cannot solve the problem itself.

In this work, we aim to address these issues by deriving a more comprehensive model that incorporates ticket content. Rather than simply calculating the transfer probability, *i.e.*, $P(B|A)$, between two groups A and B, we build a generative model that captures why tickets are transferred between two groups, *i.e.*, $P(w|A \to B)$, where w is a word in the ticket. In addition, we build a model that captures why a certain ticket can be resolved by a group B, *i.e.*, $P(w|B)$. Finally, we combine the local generative models into a global model, the Optimized Network Model (ONM), which represents the entire ticket resolution process in a network of expert groups.

The Optimized Network Model has three major applications. First, it can be trained using historical ticket data and then used as a recommendation engine to guide the routing of new tickets. Second, it provides a mechanism to analyze the role of expert groups, to assess their expertise level, and to study the expertise awareness among them. Third, it can be used to simulate the ticket routing process, and help analyze the performance of an expert network under various ticket workloads. Due to space constraints, we focus on the first application and demonstrate the superior performance of ONM compared to previous models. We briefly discuss the other two applications, but leave the detailed studies of those applications for future work.

The technical contributions of this work are three-fold:

First, we propose a unified framework, the Optimized Network Model (ONM), to model ticket transfer and resolution in an expert network. We develop solutions to estimate the parameters of ONM, using maximum likelihood estimation. We use a gradient descent method to speed up the parameter learning process.

Second, we develop a novel ticket routing algorithm that analyzes all possible routes in the network, and determines the optimal route for a ticket to its resolver. As our experiments show, this algorithm significantly outperforms existing classification-based algorithms.

Third, we show that, unlike the sequence-only model [17], ONM can explain why tickets are transferred between groups and how intermediate transfer steps can be used in finding the resolver. Hence, it can be used to evaluate the roles and performance of expert groups in a collaborative network.

2. PRELIMINARIES

We use the following notation: $\mathcal{G} = \{g_1, g_2, ..., g_L\}$ is a set of expert groups in a collaborative network; $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ is a set of tickets; and $\mathcal{W} = \{w_1, w_2, ..., w_n\}$ is a set of words that describe the problems in the tickets. A ticket consists of three components: (1) a problem category to which the ticket belongs, *e.g.*, a WINDOWS problem or a DB2 problem, that is identified when the ticket is generated, (2) the ticket content, *i.e.*, a textual description of the problem symptoms, and (3) a routing sequence from the initial group to the final resolver group of the ticket. Although some complex tickets can be associated with multiple problem categories or can involve multiple resolvers, most tickets are associated with one problem category and can be resolved by one expert group. Our model focuses on ticket routing in these common cases.

In the first step of routing, each ticket t is assigned to an initial expert group $g_{init}(t)$. If the initial group cannot solve the problem, it transfers the ticket to another group that it considers the right candidate to solve the problem. After one or more transfer steps, the ticket eventually reaches the resolver group $g_{res}(t)$. The route that the ticket takes in the expert network is denoted $R(t)$. Table 1 shows a ticket example, which is first assigned to group HDBTOIGA, and is finally resolved by group NUS_N_DSCTS.

Table 1: A WINDOWS Ticket Example

ID	Description	Initial group
8805	User received an error R=12 when installing Hyperion. When tried to install again, got success msg, but unable to open the application in Excel	HDBTOIGA

ID	Time	Entry
8805	9/29/2006	... (multi transfer steps) ...
8805	10/2/2006	Ticket 8805 transferred to Group NUS_N_DSCTS
8805	10/2/2006	Resolution: enabled Essbase in Excel

To model the interactions between groups in an expert network, we need to understand how and why the tickets are transferred and resolved. Specifically, we aim to develop a modeling framework that consists of (1) a Resolution Model $M_g(t)$ that captures the probability that group g resolves ticket t, and (2) a Transfer Model $M_{g_i \to g_j}(t)$ that captures the probability that group g_i transfers ticket t to group g_j, if g_i cannot resolve t. Our goal is to develop these two models, and then combine them into a unified network model, that represents the ticket lifecycle in the expert network, as shown in Figure 2.

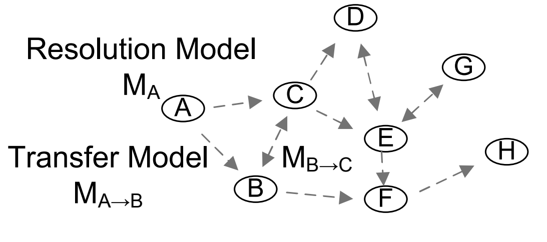

Figure 2: Unified Network Model

3. GENERATIVE MODELS

The ticket contents and routing sequences of the historical tickets provide clues as to how tickets are routed by expert groups. In our expert network, each group has its own special expertise. Thus, if an expert group is capable of resolving one ticket, chances are it can also resolve other tickets with similar problem descriptions. Likewise, similar tickets typically have similar routing paths through the network. In this section, we characterize these properties using generative models.

3.1 Resolution Model (RM)

First, we build a generative model for each expert group using the textual descriptions of the problems the group has solved previously. Given a set $\mathcal{T}_i$ of tickets resolved by group g_i and $\mathcal{W}$ the set of words in the tickets in $\mathcal{T}_i$ we build a resolver profile P_{g_i} defined as the following column vector:

$$P_{g_i} = [P(w_1|g_i), P(w_2|g_i), ..., P(w_n|g_i)]^T \tag{1}$$

Equation (1) represents the word distribution among the tickets resolved by g_i. Here, $P(w_k|g_i)$ is the probability of choosing w_k if we randomly draw a word from the descriptions of all tickets resolved by g_i. Thus, $\sum_{k=1}^{n} P(w_k|g_i) = 1$.

Assuming that different words appear independently in the ticket content, the probability that g_i can resolve a ticket $t \in \mathcal{T}_i$ can be calculated from the resolver profile vector P_{g_i} as follows:

$$P(t|g_i) \propto \prod_{w_k \in t} P(w_k|g_i)^{f(w_k,t)} \tag{2}$$

where w_k is a word contained in the content of ticket t and $f(w_k, t)$ is the frequency of w_k in the content of t.

To find a set of most probable parameters $P(w_k|g_i)$, we use the maximum likelihood method. The likelihood that group g_i resolves all of the tickets in $\mathcal{T}_i$ is:

$$\mathcal{L}(\mathcal{T}_i, g_i) = \prod_{t \in \mathcal{T}_i} P(t|g_i) \tag{3}$$

We maximize the log likelihood:

$$\begin{aligned} P_{g_i} &= \arg\max_{P(\mathcal{W}|g_i)} (\log(\mathcal{L}(\mathcal{T}_i, g_i))) \\ &= \arg\max_{P(\mathcal{W}|g_i)} (\sum_{w_k} n(w_k, \mathcal{T}_i) \log(P(w_k|g_i))) \\ s.t. \quad & \sum_{w_k \in \mathcal{W}} P(w_k|g_i) = 1 \end{aligned}$$

where $n(w_k, \mathcal{T}_i) = \sum_{t \in \mathcal{T}_i} f(w_k, t)$ is the total frequency of the word w_k in the ticket set $\mathcal{T}_i$. Hence, the maximum likelihood solution for the resolver profile vector P_{g_i} is:

$$P(w_k|g_i) = \frac{n(w_k, \mathcal{T}_i)}{\sum_{w_j \in \mathcal{W}} n(w_j, \mathcal{T}_i)} \tag{4}$$

The Resolution Model is a standard multi-class text classifier, which considers only ticket content. In the following discussion, we see that embedded in the ticket routing sequences are the transfer relations between groups, which can be used to improve the accuracy of our model.

3.2 Transfer Model (TM)

As Shao *et al.* [17] pointed out, not only the resolver group, but also the intermediate groups in the ticket routing sequences, contribute to the resolution of a ticket. The reason is that, even if an expert group cannot solve a problem directly, it might have knowledge of which other group is capable of solving it. To capture this effect, we use both the ticket content and the routing sequence to model the transfer behavior between expert groups.

Considering an edge $e_{ij} = g_i \rightarrow g_j$ in the expert network, we let $\mathcal{T}_{ij}$ denote the set of tickets that are transferred along the edge e_{ij} and let $\mathcal{W}$ denote the set of words in the tickets in $\mathcal{T}_{ij}$. Using the same technique as described in Section 3.1, we build the transfer profile of an edge between two expert groups as the column vector:

$$P_{e_{ij}} = [P(w_1|e_{ij}), P(w_2|e_{ij}), ..., P(w_n|e_{ij})]^T \tag{5}$$

where $P_{e_{ij}}$ characterizes the word distribution among the tickets routed along edge e_{ij} and $P(w_k|e_{ij})$ is the probability of choosing word w_k if we randomly draw a word from the tickets transferred along edge e_{ij}. Similarly, we derive the maximum likelihood solution for the transfer profile of e_{ij} as follows:

$$P(w_k|e_{ij}) = \frac{n(w_k, \mathcal{T}_{ij})}{\sum_{w_\ell \in \mathcal{W}} n(w_\ell, \mathcal{T}_{ij})} \tag{6}$$

The Transfer Model for the edges can be combined with the Resolution Model for the nodes to form the network model shown in Figure 2. However, the parameters of these models are learned independently and, thus, might not achieve the best modeling accuracy. To address this problem, we study how to optimize the network model by learning these parameters globally.

3.3 Optimized Network Model (ONM)

Both the Resolution Model and the Transfer Model are local models. They are not optimized for end-to-end ticket routing in the expert network. In this section, we present an optimized model that accounts for the profiles of the nodes and edges together in a global setting. Instead of considering only the tickets resolved by a certain expert group or transferred along a certain edge, this model learns its parameters based on the entire set of tickets, using both their contents and their routing sequences. As we will see, this global model outperforms the local models.

3.3.1 Routing Likelihood

When a set $\mathcal{T}_i$ of tickets is routed to a group g_i, some of the tickets will be resolved if g_i has the right expertise, while the rest of the tickets will be transferred to other groups. If g_i resolves a ticket, we assume that g_i transfers the ticket to itself. We let $\mathcal{T}_{ij}$ be the set of tickets that are transferred from group g_i to group g_j. Thus, $\mathcal{T}_i = \bigcup_{j=1}^{L} \mathcal{T}_{ij}$, where $\mathcal{T}_{ii}$ is the set of tickets resolved by group g_i itself, and L is the number of expert groups.

Given a ticket t and the expert group g_i that currently holds the ticket t, the probability that t is transferred from group g_i to group g_j is:

$$\begin{aligned} P(g_j|t, g_i) &= \frac{P(t|e_{ij})P(g_j|g_i)}{Z(t, g_i)} \\ &= \frac{(\prod_{w_k \in t} P(w_k|e_{ij})^{f(w_k,t)})P(g_j|g_i)}{Z(t, g_i)} \end{aligned} \tag{7}$$

where $Z(t, g_i) = \sum_{g_j \in \mathcal{G}} P(t|e_{ij})P(g_j|g_i)$ and $P(g_j|g_i)$ is the prior probability that g_i transfers a ticket to g_j. $P(g_j|g_i)$ can be estimated by $|\mathcal{T}_{ij}|/|\mathcal{T}_i|$. To simplify the notation, we let $P(g_i|t, g_i)$ represent the probability that group g_i is able to resolve ticket t if t is routed to g_i. Hence, $P(w|e_{ii})$ is the resolution model of g_i. Because a ticket description is often succinct with few redundant words, we assume $f(w_k, t) = 1$ if w_k occurs in t and $f(w_k, t) = 0$ otherwise. This assumption significantly simplifies the derivation of the model.

Each historical ticket t has a routing sequence $R(t)$. For example, $R(t) = g_1 \rightarrow g_2 \rightarrow g_3$, with initial group $g_{init}(t) = g_1$ and resolver group $g_{res}(t) = g_3$. We assume that an initial group g_1 is given for each ticket t, *i.e.*, $P(g_1|t) = 1$ and that each expert group makes its transfer decisions independently. In this case, the probability that the routing sequence $g_1 \rightarrow g_2 \rightarrow g_3$ occurs is:

$$\begin{aligned} P(R(t)|t) &= P(g_1|t)P(g_2|t, g_1)P(g_3|t, g_2)P(g_3|t, g_3) \\ &= P(g_2|g_1)P(g_3|g_2)P(g_3|g_3) \\ &\quad \times \frac{P(t|e_{1,2})P(t|e_{2,3})P(t|e_{3,3})}{Z(t, g_1)Z(t, g_2)Z(t, g_3)} \end{aligned}$$

We assume further that the tickets are independent of each other. Thus, the likelihood of observing the routing sequences in a ticket set $\mathcal{T}$ is:

$$\mathcal{L} = \prod_{t \in \mathcal{T}} P(R(t)|t) \tag{8}$$

3.3.2 Parameter Optimization

To find a set of globally optimal parameters $P(w_k|e_{ij})$, we use maximum likelihood estimation to maximize the log likelihood:

$$\begin{aligned} \log \mathcal{L} &= \sum_{t \in \mathcal{T}} \log P(R(t)|t) \qquad (9) \\ &= \sum_{t \in \mathcal{T}} \sum_{e_{ij} \in R(t)} \log \frac{P(t|e_{ij}) \times P(g_j|g_i)}{Z(t, g_i)} \\ &= \sum_{e_{ij} \in \mathcal{E}} \sum_{t \in \mathcal{T}_{ij}} (\log(P(t|e_{ij})) + \log(P(g_j|g_i))) \\ &\quad - \sum_{g_i \in \mathcal{G}} \sum_{t' \in \mathcal{T}_i} \log(Z(t', g_i)) \end{aligned}$$

where $\mathcal{E} = \{e_{ij} | 1 \leq i, j \leq L\}$ and $P(t|e_{ij}) = \prod_{w_k \in t} P(w_k|e_{ij})$. The optimal transfer profile is given by the following constrained optimization problem:

$$\begin{aligned} P(\mathcal{W}|\mathcal{E})^* &= \arg \max_{P(\mathcal{W}|\mathcal{E})} (\log \mathcal{L}) \qquad (10) \\ \text{s.t.} & \sum_{w_k \in \mathcal{W}} P(w_k|e_{ij}) = 1; \\ & P(w_k|e_{ij}) \geq 0 \end{aligned}$$

where $\mathcal{W}$ is the set of words and $\mathcal{E}$ is the set of edges.

This optimization problem is not convex, and it involves many free dimensions (the degree of freedom is $(|\mathcal{W}| - 1) \times |\mathcal{G}|^2$). It cannot be solved efficiently with existing tools.

Thus, we seek solutions that are near-optimal but easier to calculate. Our approach is to update the parameters $P(w_k|e_{ij})$ iteratively to improve the likelihood. Specifically, we use the steepest descent method to maximize the lower bound of the log likelihood. By Jensen's inequality, we have

$$Z(t, g_i) \leq \prod_{w_k \in t} \sum_{g_\ell \in \mathcal{G}} P(g_\ell|g_i)P(w_k|e_{i\ell}) \tag{11}$$

Combining Equation (9) and Equation (11), we have:

$$\begin{aligned} \log \mathcal{L} \quad \geq \quad \lfloor \log \mathcal{L} \rfloor &= \sum_{e_{ij}} \sum_{t \in \mathcal{T}_{ij}} (\log(P(t|e_{ij})) + \log(P(g_j|g_i))) \\ &\quad - \sum_{g_i \in \mathcal{G}} \sum_{t' \in \mathcal{T}_i} \sum_{w_k \in t'} \log(\sum_{g_\ell \in \mathcal{G}} (P(g_\ell|g_i) \times P(w_k|e_{i\ell}))) \end{aligned}$$

The gradient is given by:

$$\begin{aligned} \nabla \lfloor \log(\mathcal{L}) \rfloor &= \frac{\partial \lfloor \log \mathcal{L} \rfloor}{\partial P(w_k|e_{ij})} \\ &= \frac{\sum_{t \in \mathcal{T}_{ij}} n(w_k, t)}{P(w_k|e_{ij})} \\ &\quad - \frac{P(g_j|g_i) \times \sum_{t' \in \mathcal{T}_i} n(w_k, t')}{\sum_{g_\ell \in \mathcal{G}} P(g_\ell|g_i) \times P(w_k|e_{i\ell})} \end{aligned}$$

Using the values of $P(w_k|e_{ij})$ calculated in Equation (6) as the starting point, we iteratively improve the solution along the gradient. To satisfy the constraints, we calculate the projection of the gradient in the hyperplane defined by $\sum_{w_k \in \mathcal{W}} P(w_k|e_{ij}) = 1$ to ensure that the solution stays in the feasible region. The profiles of the edges in the network are updated one at a time, until they converge. Although the gradient-based method might produce a local optimum solution, it estimates the model parameters all together from a global perspective and provides a better estimation than the TM locally-optimized solution.

4. TICKET ROUTING

We now study the application of the generative models introduced in Section 3 to ticket routing.

Given a new ticket t and its initial group $g_{init}(t)$, a routing algorithm uses a model $\mathcal{M}$ to predict the resolver group $g_{res}(t)$. If the predicted group is not the right resolver, the algorithm keeps on predicting, until the resolver group is found. The performance of a routing algorithm can be evaluated in terms of the number of expert groups it tried until reaching the resolver. Specifically, we let the predicted routing sequence for ticket t_i be $R(t_i)$ and let $|R(t_i)|$ be the number of groups tried for ticket t_i. For a set of testing tickets $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$, we evaluate the performance of a routing algorithm using the Mean Number of Steps To Resolve (MSTR) [17] given by:

$$S = \frac{\sum_{i=1}^{m} |R(t_i)|}{m} \tag{12}$$

The ticket routing problem is related to the multi-class classification problem in that we are seeking a resolver (class label) for each ticket. Different from a classification problem, our goal here is not to maximize the classification precision, but to minimize the expected number of steps before the algorithm reaches the right resolver.

Nevertheless, we can adapt a multi-class classifier to fit our problem. We assume that a classifier C predicts group g as the resolver of ticket t, with probability $P(g|t)$. A simple approach is to rank the potential resolver groups in descending order of $P(g|t)$ and then transfer the ticket t to them one by one, until the right resolver is found. In this approach, the ranking of groups does not change, even if the current prediction is incorrect. We take the Resolution Model as an example, and as the baseline method, for building a classifier. Then, we develop two dynamic ranking methods, using the Transfer Model and the Optimized Network Model, to achieve better performance.

4.1 Ranked Resolver

The Ranked Resolver algorithm is designed exclusively for the Resolution Model (RM). Expert groups are ranked based on the probability that they can resolve the ticket according to the ticket content.

Given a new ticket t, the probability that expert group g_i can resolve the ticket is:

$$\begin{aligned} P(g_i|t) &= \frac{P(g_i)P(t|g_i)}{P(t)} \qquad (13) \\ &\propto P(g_i)\prod_{w_k \in t} P(w_k|g_i)^{f(w_k,t)} \end{aligned}$$

Here, $P(g_i)$ is the prior probability of group g_i being a resolver group, which is estimated by $|\mathcal{T}_i|/|\mathcal{T}|$, where $\mathcal{T}_i$ is the set of tickets resolved by g_i and $\mathcal{T}$ is the ticket training set.

A routing algorithm for this model is to try different candidate resolver groups in descending order of $P(g_i|t)$. The algorithm works fine unless the new ticket t contains a word that has not appeared in the training ticket set $\mathcal{T}$. In that case, $P(g_i|t)$ is zero for all i. To avoid this problem, we introduce a smoothing factor λ to calculate the probability, *i.e.*,

$$P(w|g_i)^* = \lambda \times P(w|g_i) + (1-\lambda)/|\mathcal{W}| \qquad (14)$$

Using the smoothed value $P(w|g_i)^*$ guarantees a positive value of $P(g_i|t)$ for all i.

4.2 Greedy Transfer

The Greedy Transfer algorithm makes one step transfer predictions and selects the most probable resolver as the next step.

When a new ticket t first enters the expert network, it is assigned to an initial group g_{init}. Instead of calculating which group is likely to solve the problem, we determine the group to which the ticket should be transferred, because tickets should be transferred to the group that can solve the problem or the group that knows which group can solve the problem. The probability that a ticket t is routed through the edge $e_{init,j} = g_{init} \rightarrow g_j$, where $g_j \in \mathcal{G} \setminus \{g_{init}\}$, is:

$$\begin{aligned} P(g_j|t, g_{init}) &= \frac{P(g_j|g_{init})P(t|e_{init,j})}{\sum_{g_l \in \mathcal{G}} P(g_l|g_{init})P(t|e_{init,l})} \qquad (15) \\ &= \frac{P(g_j|g_{init})\prod_{w_k \in t} P(w_k|e_{init,j})^{f(w_k,t)}}{\sum_{g_l \in \mathcal{G}} P(g_l|g_{init})\prod_{w_k \in t} P(w_k|e_{init,l})^{f(w_k,t)}} \end{aligned}$$

Note that smoothing is applied as described in Equation (14).

The expert group $g^* = \arg\max_{g_j \in \mathcal{G}} P(g_j|t, g_{init})$ is selected to be the next expert group to handle ticket t. If g^* is the resolver, the algorithm terminates. If not, the algorithm gathers the information of all previously visited expert groups to make the next step routing decision. If a ticket t has gone through the expert groups in $R(t)$ and has not yet been solved, the rank of the remaining expert groups in $\mathcal{G} \setminus R(t)$ is:

$$Rank(g_j) \propto \max_{g_i \in R(t)} P(g_j|t, g_i) \qquad (16)$$

and the ticket is routed to the group with the highest rank. The rank of g_j is determined by the maximum probability of $P(g_j|t, g_i)$ for all the groups g_i that have been tried in the route. The ranked order of the candidate resolvers might change during routing.

4.3 Holistic Routing

Th Holistic Routing algorithm recognizes the most probable resolver that can be reached within K transfer steps, and selects the next group from a global perspective. In the experiments, we set K equal to 3. Instead of predicting only one step as do the Ranked Resolver and Greedy Transfer algorithms, the Holistic Routing algorithm calculates the probability that a candidate group can be reached and can solve the ticket in multiple steps.

For a new ticket t, the one step transition probability $P(g_j|t, g_i)$ between two expert groups g_i and g_j is calculated using Equation (15). Thus, we perform a breadth-first search to calculate the probability that a ticket t is transferred by g_i to g_j in exactly K steps. This probability can be estimated iteratively, using the following equations:

$$\begin{aligned} P(g_j, 1|t, g_i) &= \begin{cases} P(g_j|t, g_i) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \\ P(g_j, K|t, g_i) &= \sum_{g_k \in \mathcal{G}; k \neq j} P(g_k, K-1|t, g_i)P(g_j|t, g_k) \\ &\qquad \text{if } K > 1. \end{aligned}$$

If $g_l = g_{init}$ the initial group for ticket t, the above equation can be written as:

$$P(g_j, K|t, g_l) = vM^K \qquad (17)$$

where v is the unit vector whose lth component is 1 and other components are 0. The one step transfer probability matrix M is a $|\mathcal{G}| \times |\mathcal{G}|$ matrix, where an entry of M is the one step transition probability between the expert groups g_i and g_j given by:

$$M(i,j) = \begin{cases} P(g_j|t, g_i) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

The probability that g_j can resolve the ticket t in K or fewer steps starting from the initial group g_{init} (which is used to rank the candidate resolver groups) is:

$$Rank(g_j|g_{init}) \equiv \sum_{k=1}^{K} P(g_j, k|t, g_{init}) \times P(g_j|t, g_j) \qquad (18)$$

where $P(g_j|t, g_j)$ is the probability that g_j resolves t if t reaches g_j (see Equation (7)). Starting with g_{init}, we route t to the group $g^* = \arg\max_{g_j \in \mathcal{G}; j \neq init} Rank(g_j|g_{init})$.

Theoretically, we can derive the rank in closed form for an infinite number of transfer steps. In practice, M^K decays quickly as K increases, due to the probability of solving the ticket at each step. A small value of K suffices to rank the expert groups.

Given the predicted expert group g_k, if ticket t remains unresolved and needs to be transferred, the posterior probability of g_k being the resolver for t is zero and the one step transfer matrix M needs to be updated accordingly. Thus, if g_k is not the resolver, the elements in the kth row of M are updated by:

$$M(k,j) = \frac{P(g_j|t, g_k)}{\sum_{i, i \neq k} P(g_i|t, g_k)} \quad \text{for } j \neq k$$

Once M is updated, the algorithm reranks the groups according to Equation (18) for each visited group in $R(t)$. That is, $Rank(g_j) \propto max_{g_i \in R(t)} Rank(g_j|g_i)$. The group with the highest rank is selected as the next possible resolver.

For a given new ticket, the Holistic Routing algorithm is equivalent to enumerating all of the possible routes from the initial group to any candidate group. For each route $r = \{g_1, g_2, \dots, g_m\}$ for a ticket t, we calculate the probability of the route as:

$$P(r|t) = P(g_m|t, g_m) \prod_{1 \le j \le m-1} P(g_{j+1}|t, g_j)$$

The probability that group g_j resolves ticket t is:

$$Rank(g_j) \equiv \sum_r P(r|t) \text{ for all r ending at } g_j$$

Figure 3 shows an example where a ticket t enters the expert network at group A. We enumerate all of the routes that start at A and end at D to calculate how likely D resolves the ticket. Note that loops in the routes are allowed in the calculation in Equation (17). It is also possible to calculate the resolution probability without loops. However, because the intermediate groups for each route must be remembered, the calculation might take a long time.

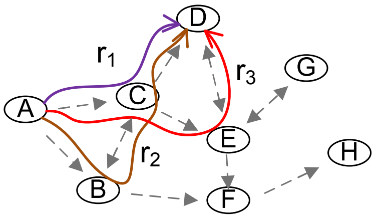

Figure 3: Holistic Routing

5. EXPERIMENTAL RESULTS

To validate the effectiveness of our models and the corresponding routing algorithms,[1] we use real-world ticket data. The evaluation is based on problem tickets collected from IBM's problem ticketing system throughout 2006. When a ticket enters the system, the help desk assigns a category indicating a problem category for the ticket. For each problem category, a number of expert groups (ranging from 50 to 1,000) are involved in resolving the tickets.

For each problem category, we partition the data set into the training data set and the testing data set. Using the training data set, first we build the generative models introduced in Section 3. Then, we evaluate the effectiveness of the routing algorithms by calculating the number of routing steps (*i.e.*, MSTR) for the testing tickets. In particular, we compare our generative models with the Variable-Order Markov Model (VMS) proposed in [17]. Our experiments demonstrate:

- **Model Effectiveness:** The *Optimized Network Model* significantly outperforms the other models.
- **Routing Effectiveness:** Among the Ranked Resolver, Greedy Transfer and Holistic Routing algorithms, Holistic Routing achieves the best performance.
- **Robustness:** With respect to the size of the training data set, the time variability of the tickets, and the different problem categories, our solution that combines *ONM* and *Holistic Routing* consistently achieves good performance.

We obtained our experimental results using an Intel Core2 Duo 2.4GHz CPU with 4GB memory.

5.1 Data Sets

We present the results obtained from tickets in three major problem categories: AIX (operating system), WINDOWS (operating system), and ADSM (storage management), as shown in Table 2. Tickets in these three categories have quite different characteristics. The problem descriptions for WINDOWS and ADSM tickets tend to be more diverse and, hence, more challenging for our models.

Table 2: Ticket Resolution Data Sets

Category	# of tickets	# of words	# of groups
AIX	18,426	16,065	847
WINDOWS	16,441	8,521	638
ADSM	3,563	1,815	301

These three data sets involve approximately 300 to 850 expert groups. For a new ticket, finding a resolver group among so many candidates can be challenging.

Table 3: Resolution Steps Distribution

Steps	Percentage
2	68%
3	25%
4	6%
>=5	1%

Table 3 shows the distribution of resolution steps for tickets in the WINDOWS category. We are more interested in solving tickets with long resolution sequences, because these tickets received most of the complaints.

5.2 Model Effectiveness

First, we compare the effectiveness of the three generative models, Resolution Model (RM), Transfer Model (TM), and Optimized Network Model (ONM) developed in Section 3, against the Variable-Order Markov Model (VMS) introduced in [17]. VMS considers only ticket routing sequences in the training data.

Each of the above models has its corresponding routing algorithm. VMS uses the conditional transfer probability learned from routing sequences to predict the resolver group. For RM, we use the Ranked Resolver algorithm. For TM and ONM, we can use either the Greedy Transfer algorithm or the Holistic Routing algorithm. In these experiments, we use the Holistic Routing algorithm to evaluate both models. For comparison, we also include the result of ONM using the Greedy Transfer algorithm. More details for the comparison between the Greedy Transfer algorithm and the Holistic Routing algorithm are shown in Section 5.3.

[1] The source code is available at http://www.uweb.ucsb.edu/~miao/resources.html.

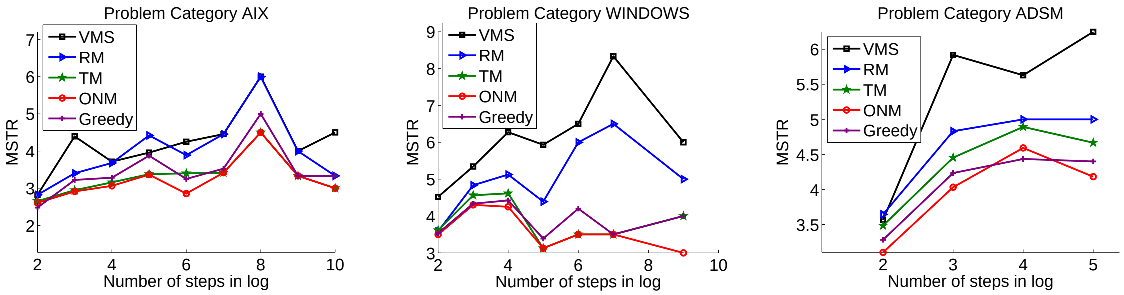

Figure 4: Prediction Accuracy of Different Models

Because a routing algorithm might generate an extremely long routing sequence to resolve one ticket (considering that we have more than 300 expert groups in each problem category), we apply a cut-off value of 10. That is, if an algorithm cannot resolve a ticket within 10 transfer steps, it is regarded as unresolvable. Using this cut-off value, we define the *resolution rate* of a ticket routing algorithm to be the proportion of tickets that are resolvable within 10 steps.

We randomly divide the tickets in each problem category into two subsets: the training data set and the testing data set, where the former contains 75% of the tickets, and the latter contains 25% of the tickets. The four models are trained based on the training set, and the performance of the algorithms is compared.

Figure 4 compares the prediction accuracy of the four models. The x-axis represents the number of expert groups involved in the testing data set, where the routing decisions are made by a human. The y-axis represents the resulting MSTR when the testing tickets are routed automatically using a model. Obviously, smaller MSTR means better prediction accuracy. As shown in the figure, TM and ONM (which combine the ticket contents and the routing sequences) result in better prediction accuracy than either the sequence-only VMS model or the content-only RM. Moreover, ONM achieves better performance than TM, which indicates that the globally optimized model is more accurate in predicting a ticket resolver than the locally optimized model.

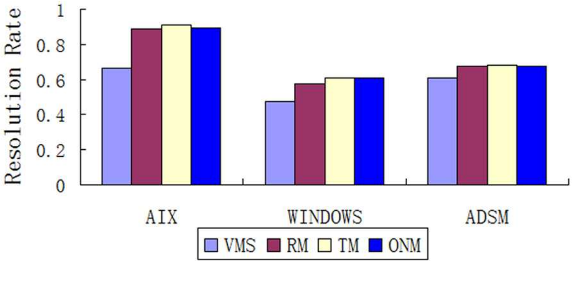

Figure 5: Resolution Rate

Combining together the ticket contents and the routing sequences not only boosts prediction accuracy, but also increases the resolution rate of the routing algorithm. Figure 5 shows that TM and ONM can resolve more tickets than either VMS or RM.

For RM and TM, the training time is mainly spent on counting word frequencies on transfer edges and at resolvers. For all three data sets, the time is less than 5 minutes. For ONM, the transfer profiles are updated one at a time and the optimization process repeats for multiple rounds until the transfer profiles converge. The training process takes less than 3 hours for all three data sets.

5.3 Routing Effectiveness

Using the same experimental setup as in Section 5.2, we compare the effectiveness of the Greedy Transfer and Holistic Routing algorithms.

Both of these algorithms can be executed on the TM and ONM generative models. We consider all four combinations: TM+Greedy, TM+Holistic, ONM+Greedy, and ONM+Holistic.

Figure 6 shows that, for each generative model, the Holistic Routing algorithm consistently outperforms the Greedy Transfer algorithm. These results validate our hypothesis that, even if an expert group is not the resolver for a problem ticket, it might have appropriate knowledge of which group can resolve the ticket. Therefore, besides the information about which groups resolve which tickets, the intermediate transfer groups can be instrumental in routing tickets to the right resolver, which is why the Holistic Routing algorithm has better performance.

The computational time for both routing algorithms to make a routing decision is less than 1 second, which is negligible compared to the time spent by the selected expert group to read and handle the ticket.

5.4 Robustness

For our generative models and routing algorithms to be useful in practice, they must apply to different problem categories and training samples. To confirm this, we divided the data in different ways with respect to the size of the training data set, the time variability of the tickets, and the different problem categories, as presented in Table 4. For each training set, we rebuilt the models and applied the routing algorithms to measure the resulting MSTR for the corresponding testing set. Given the previous analysis, we focus on ONM and Holistic Routing.

Table 4: Data Sets for Robustness

Training Set	Testing Set
Jan 1 - Mar 31, 2006	Apr 1 - Apr 30, 2006
Jan 1 - Apr 30, 2006	May 1 - May 31, 2006
Jan 1 - May 31, 2006	Jun 1 - Jun 30, 2006

As shown in Figure 7, with larger training data sets, the resulting MSTR tends to become smaller. Despite the vari-

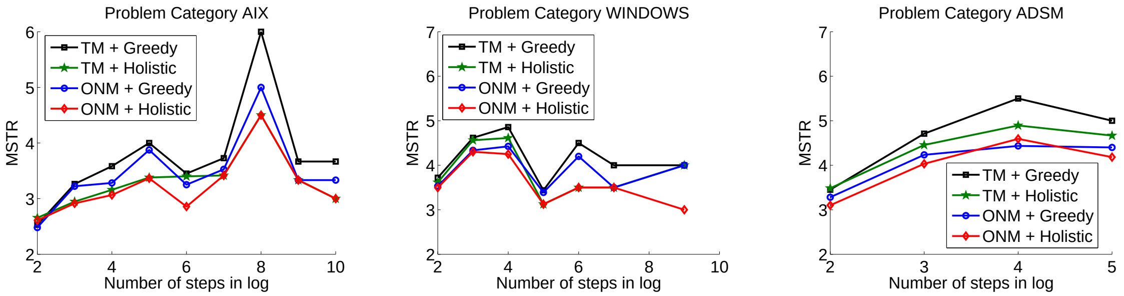

Figure 6: Routing Efficiency: Greedy Transfer vs. Holistic Routing

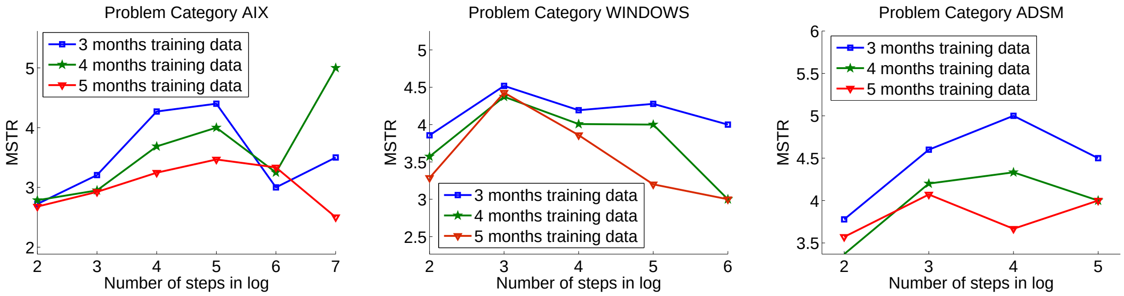

Figure 7: Robustness of ONM and Holistic Routing with Variable Training Data

ations in the size of the training set, our approach yields consistent performance. The problem descriptions in these ticket data sets are typically short and sparse. The results demonstrate that generative modeling is particularly effective for this type of data.

6. DISCUSSION

We have focused on using the proposed model to make effective ticket routing decisions. However, the model has other significant applications, namely, expertise assessment in an expert network and ticket routing simulation for performance analysis and workforce/resource optimization. We briefly discuss these applications below.

6.1 Expertise Assessment

In essence, our model represents the interactions between experts in an enterprise collaborative network. By analyzing ticket transfer activities at the edges of the network, we can identify different roles of individual expert groups, *i.e.*, whether a group is more effective as a ticket resolver or a ticket transferrer. We can also analyze the expertise awareness between groups.

For instance, Figure 8 shows the most prominent words derived from ONM in the context of tickets transferred from group A to group B (List 1), as well as those resolved by group B itself (List 2). List 1 is related to system boot failures (bluescreen, freeze), while List 2 is related to data loading issues in hard drives. The mismatch between the two lists, indicates that either A is not well aware of B's expertise, or A thinks that B can better identify the resolvers for tickets described by words in List 1. Further analysis is needed to understand these interactions and implications. Our model can facilitate such analysis.

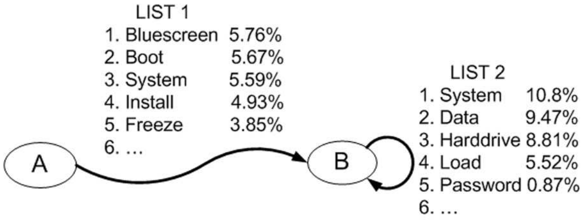

Figure 8: Expertise Awareness Example

6.2 Ticket Routing Simulation

Our model can be used to simulate the routing of a given set of tickets. The simulation can help an enterprise analyze its existing ticket routing process to identify performance bottlenecks and optimize workforce/resources. Moreover, the simulation can be used to assess the "criticality" of expert groups, *e.g.*, whether the routing performance is improved or degraded, if a group is removed from the network. Such a knockout experiment is infeasible in practice, but can be conducted by simulation.

7. RELATED WORK

Ticket routing can be considered an extension of the text classification problem, which has been extensively studied in the literature [4, 5, 9, 11, 15, 19, 20, 24]. For instance, Yang and Liu [19] studied the robustness of different text categorization methods. Calado *et al.* [5], Lu and Getoor [11], and Sen *et al.* [15] proposed methods to combine content and link information for document classification.

Ticket routing is also related to the multi-class classification problem [14]. Compared to multi-class classification, ticket routing has distinct properties. First, ticket routing involves multiple predictions if the current prediction is not correct, which leads to different evaluation criteria. Second, ticket routing takes place in a network, which is also different from the traditional classification problem. Third, instead of relying on a single classifier, ticket routing requires leveraging the interactions between multiple local classifiers to find a globally optimized solution.

Belkin *et al.* [4] and Zhou *et al.* [24] introduced text classification using graph-based methods. Collective classification, such as loopy belief propagation [22], mean field relaxation labeling [21], interactive classification [12] and stacked models [10], are popular techniques for classifying nodes in a partially labeled graph. The problems studied in these methods are quite different from our problem, as we assume one resolver in the network for a given ticket, and the classification needs to be repeatedly applied until the resolver is found.

Generative models and maximum likelihood estimation are standard approaches. Generative models seek the joint probability distribution over the observed data. Classification decisions are typically made based on conditional probabilities formed using Bayesian rules. One example is the Naive Bayes classifier [8, 23], which assumes conditional independence between variables. Another example is the Gaussian Mixture Model [13], which estimates the probability distribution using a convex combination of several Gaussian distributions. These models are good for analyzing sparse data. We chose the generative model because the transition probabilities in the ticket resolution sequences can be seamlessly embedded in the probabilistic framework. Our contribution is the combination of multiple local generative models to yield a globally optimized solution.

Besides the generative models, discriminative models, such as the Support Vector Machine (SVM), have been shown to be effective for text classification [9]. One can potentially build a support vector classifier for each resolver and each transfer relationship. However, they are locally optimized for individual resolvers and transfer relationships; once trained, the SVM classifiers remain stationary. In our approach, the resolver predictions can be dynamically adjusted if previous predictions are incorrect.

The ticket routing problem is also related to the expert finding problem, *i.e.*, given a keyword query, find the most knowledgeable persons regarding that query. The expert finding algorithms proposed by Balog *et al.* [3] and Fang and Zhai [7] use a language model to calculate the probability of an expert candidate to generate the query terms. Serdyukov *et al.* [16] enhanced those models by allowing the candidates' expertise to be propagated within networks, *e.g.*, via email. Deng *et al.* [6] explored the links in documents such as those listed in DBLP [1]. Expert recommendation systems also use text categorization techniques to characterize bugs [2] and documents [18]. Because most expert finding algorithms are content-based, they share the same weakness of the Resolver Model (RM) given in Section 3.1.

Our study has demonstrated that better routing performance can be achieved by combining ticket contents and routing sequences together. Nevertheless, considering existing sophisticated text classification methods and language models, it is an open research problem to investigate how to embed these models in a collaborative network and learn their parameters in a holistic way for ticket processing, a challenging issue in the IT service industry.

8. CONCLUSIONS

We have presented generative models that characterize ticket routing in a network of expert groups, using both ticket content and routing sequences. These models capture the capability of expert groups either in resolving the tickets or in transferring the tickets along a path to a resolver. The Resolution Model considers only ticket resolvers and builds a resolution profile for each expert group. The Transfer Model considers ticket routing sequences and establishes a locally optimized profile for each edge that represents possible ticket transfers between two groups. The Optimized Network Model (ONM) considers the end-to-end ticket routing sequence and provides a globally optimized solution in the collaborative network. For ONM, we present a numerical method to approximate the optimal solution which, in general, is difficult to compute.

Our generative models can be used to make routing predictions for a new ticket and minimize the number of transfer steps before it reaches a resolver. For the generative models, we presented three routing algorithms to predict the next expert group to which to route a ticket, given its content and routing history. Experimental results show that the proposed algorithms can achieve better performance than existing ticket resolution methods.

9. ACKNOWLEDGMENTS

This research was sponsored in part by the U.S. National Science Foundation under grants IIS-0905084/0915438, IIS-0917228 and by the Army Research Laboratory under cooperative agreement W911NF-09-2-0053 (NS-CTA). The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

10. REFERENCES

[1] DBLP: http://www.informatik.uni-trier.de/∼ley/db/.

[2] J. Anvik, L. Hiew, and G. C. Murphy. Who should fix this bug? In *ICSE'06*, pages 361–370, 2006.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'06*, pages 43–50, 2006.

[4] M. Belkin, P. Niyogi, V. Sindhwani, and P. Bartlett. Manifold regularization: A geometric framework for learning from examples. 2004.

[5] P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, and M. A. Gonçalves. Combining link-based and content-based methods for Web document classification. In *CIKM'03*, pages 394–401, 2003.

[6] H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on DBLP bibliography data. In *ICDM'08*, pages 163–172, 2008.

[7] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR'07*, pages 418–430, 2007.

[8] A. Jamain and D. J. Hand. The naive Bayes mystery: A classification detective story. *Pattern Recognition Letters*, 26(11):1752–1760, 2005.

[9] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML'98*, pages 137–142, 1998.

[10] Z. Kou and W. Cohen. Stacked graphical models for efficient inference in Markov random fields. In *SDM'07*, 2007.

[11] Q. Lu and L. Getoor. Link-based text classification. In *IJCAI Workshop on Text Mining and Link Analysis*, 2003.

[12] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Statistical Relational Learning*, pages 42–49, 2000.

[13] H. H. Permuter, J. M. Francos, and I. Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[14] J. Platt, N. Cristianini, and J. Shawe-taylor. Large margin DAGs for multiclass classification. In *NIPS'00*, pages 547–553, 2000.

[15] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3), 2008.

[16] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *CIKM'08*, pages 1133–1142, 2008.

[17] Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis. Efficient ticket routing by resolution sequence mining. In *SIGKDD'08*, pages 605–613, 2008.

[18] X. Song, B. L. Tseng, C.-Y. Lin, and M.-T. Sun. Expertisenet: Relational and evolutionary expert modeling. In *User Modeling*, pages 99–108, 2005.

[19] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR'99*, pages 42–49, 1999.

[20] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML'97*, 1997.

[21] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. on Information Theory*, 51(7):2282–2312, 2005.

[22] J. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS'00*, pages 689–695, 2000.

[23] H. Zhang. The optimality of naive Bayes. In *FLAIRS Conference*, 2004.

[24] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. S. Olkopf. Learning with local and global consistency. In *NIPS'04*, pages 321–328, 2004.