Multi-Task Active Learning

Abhay Harpale

CMU-LTI-12-012 Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213 www.lti.cs.cmu.edu

Thesis Committee:
Prof. Yiming Yang (*Chair*)
Prof. Jaime Carbonell
Prof. Tom Mitchell
Dr. Jian Zhang, Citadel Investment Group

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies

Contents

Abstract

Training data acquisition for enabling supervised learning algorithms is an expensive process. Current Active learning (AL) approaches to limit such costs by selectively acquiring supervision have been studied only in the context of single-tasks. Current state-of-art AL strategies proposed for multi-task scenarios myopically focus on improving each of the concerned tasks in isolation. Leveraging inter-task relationships to identify training instances that are most beneficial for jointly learning a set of tasks is an open challenge. In this regard, we present the first investigation of approaches to combine the strengths of Active learning and Multi-task learning (MTL) to reduce training data acquisition costs for supervised learning applications such as multi-labeled classification, hierarchical classification, adaptive filtering and collaborative filtering. AL and MTL have complementary strengths. AL attempts to minimize supervision costs by soliciting supervision on instances which are deemed most useful for training. On the other hand, MTL attempts to deal with scarce training data by jointly learning several related tasks. Our approach leverages inter-task relationships to identify a reference task and corresponding unlabeled instances that should receive supervision to rapidly improve the overall multi-task system. We develop our ideas in the context of a special class of MTL frameworks that model inter-task relationships through shared Bayesian priors. We develop a novel paradigm called *Circle of Influence* that estimates the potential impact of an unlabeled instance at several levels of MTL structure: model of the reference task, the shared model and the models of other related tasks. We empirically demonstrate the superiority of our approach over conventional state-of-the-art AL approaches in the context of Multi-Task Adaptive Filtering.

We further investigate MTAL in the context of heterogeneous tasks, such as jointly learning classification and regression models, where each kind of task can have different supervision costs. In this regard, we present a novel topic-modeling based approach for jointly learning Collaborative filtering and Genre-classification. In the context of this model, we present Transferable Active Learning, a novel strategy that compares potential candidates for supervision by estimating their impact on the underlying topic model. Our experiments on benchmark datasets demonstrate the superior performance of the proposed strategy over conventional approaches.

We also describe a novel Multi-Task Active Learning strategy for the Hierarchical Classification problem. We leverage the knowledge about parent-child relationships to identify instances for learning that can potentially lead to predictions consistent with the given hierarchical structure. We also leverage the parent-child regularization framework to identify influential nodes. Our Active Learning strategy preferentially selects more instances for such influential nodes with the expectation of achieving cascading effects on their descendants. Our experiments on benchmark hierarchical classification datasets demonstrate the superior performance of our approach over conventional single-task approaches.

Finally, for some multi-task scenarios, the oracle maybe unable to provide supervision for certain instance-task pairs. We present a novel strategy to model the oracle's expertise in providing per-task supervision to avoid requests for supervision that the oracle cannot provide. We demonstrate the effectiveness of our approach in significantly reducing the number of failed Active Learning label requests in the

context of Collaborative Filtering, while still sustaining better performance over conventional state-of-the-art approaches.

1

Introduction

Training data acquisition for supervised machine learning algorithms, such as those for classification, regression, rank-learning, collaborative filtering and adaptive filtering, can be very expensive. Consequently, it is crucial to minimize the amount of supervision required for learning such tasks, without significantly sacrificing generalization performance of the learnt models. There can be two ways to address this challenge: either deal with training data scarcity by cleverly incorporating additional stimuli or be selective about acquiring supervision on data that is most beneficial for learning. Some of the popular approaches that take the first route of dealing with data-scarcity are: Semi-supervised learning approaches such as Co-training [Blum and Mitchell, 1998] or Transductive learning [Joachims, 1999] additionally utilize the (usually) abundant unlabeled data; Multi-task learning [Caruana, 1997; Jian Zhang and Yang, 2005] approaches jointly learn several related tasks and utilize inter-task relationships for added supervision to each of the tasks with scarce data. Active learning [Settles, 2009] approaches take the alternate route of being choosy and solicit supervision for a few selected unlabeled instances that are deemed most beneficial for learning. There is also active research in combining strengths of some of these approaches: semi-supervised + multi-task learning [Ando et al., 2005; Liu et al., 2009], or semi-supervised + active

learning [Muslea et al., 2002; Tur et al., 2005]. However, so far, there has been no thorough investigation of approaches that combine the strengths of Active learning and Multi-task learning. It should be noted that the strengths of Active learning and Multi-task learning are complementary. The former identifies the best instances to learn from and the latter attempts to use available instances most effectively by leveraging inter-task relationships. Consequently, there's a huge opportunity for reducing supervision costs by actively selecting instances that are most useful for learning several tasks simultaneously by leveraging inter-task relationships. Through this thesis, we propose to address this gap.

We motivate the challenges involved in achieving this goal through examples of important machine learning applications where Multi-task learning approaches have demonstrated significant success in outperforming approaches that learn such tasks in isolation.

- Multi-labeled classification (MLC): The MLC [Tsoumakas and Katakis, 2007; Yang and Gopal, 2012] scenario deals with classifying instances that may belong to several categories simultaneously. There are several open challenges in minimizing the supervision required for MLC. How to choose instances that are beneficial for simultaneously improving performance on several categories? To further reduce supervision costs, which categories should get additional training instances so that performance on other categories is improved as well? How can we leverage inter-category relationships to achieve these goals?
- Hierarchical classification (HC): HC [Lewis et al., 2004b] is an extension of MLC in which the categories are arranged in a pre-defined hierarchy. In addition to the challenges mentioned above, how to utilize the structural constraints, such as parent-child and sibling relationships, to select the most useful instances for chosen categories such that overall HC performance is improved? How to jointly model such output constrained tasks for enabling further savings?

- Adaptive Filtering (AF): AF [Robertson and Soboroff, 2002] systems monitor a stream of items to discard items that are irrelevant to a user's information need, e.g. spam or news filtering. Learning is based on user feedback over only the delivered instances, not the discarded ones. Since several users might have similar requirements, how to deliver items with a three-fold objective: satisfy the user's information need, improve future performance for a user based on feedback on delivered items, improve future peformance for other similar users based on feedback received for from one user.
- Collaborative Filtering (CF): CF [Breese et al., 1998] approaches recommend items to users based on the similarity of their interests with other users. Learning is based on item ratings provided by the users. By soliciting ratings for a few chosen items, how to rapidly understand a user's interests as well as simultaneously improve recommendations for other users with similar interests? In doing so, it is crucial to model the user's propensity of rating a chosen item, as the user may not provide ratings for items that she doesn't have experience with.

Most of the current Active learning approaches desgined for these and other multi-task applications attempt to select instances that will improve a single task in isolation [Esuli and Sebastiani, 2009]. Other Active learning approaches that do select instances based on their benefit to several tasks, do so by ignoring the implicit inter-task relationships [Reichart et al., 2008; Singh et al., 2009; Yang et al., 2009a], leading to sub-optimal savings. Most of these Active learning approaches also require *complete* labelings for the selected instance, i.e. supervision about all the tasks for the chosen instance. This might lead to redundancy and wastful supervision if several of the tasks are related and obtaining labels about some of them conveys similar information. A few Active learning approaches enable *partial* labeling by choosing the most useful task to label, but in doing so, they ignore inter-task relationships [Roth and Small, 2006]. When explicit inter-task constraints are available, for example in hierarchical classification,

current Active learning approaches do not consider joint learning of the constrained tasks to enable further savings [Zhang, 2010]. The oracle's expertise to supervise items has been studied as part of a recent work in Proactive Learning [Donmez and Carbonell, 2008a], albeit in the single-task setting.

As a solution to these problems, we propose a Multi-Task Active Learning (MTAL) framework, where we investigate solutions to the following challenges

1. **MTAL:** How to selectively acquire training data that will potentially benefit several tasks simultaneously? In Chapter 3, we describe our framework for a class of Bayesian models that enable Multi-Task learning through shared priors among underlying tasks. We then apply the framework to the practically important problem of Multi-Task Adaptive Filtering and present empirical evidence for its effectiveness on benchmark datasets.

2. **MTAL for Heterogeneous Tasks:** How to selectively acquire training data that will potentially benefit diverse types of tasks, such as classification and regression simultaneously. In Chapter 4, we present our topic-modeling approach for jointly learning heterogeneous tasks and present a surrogate Active Learning strategy, called Transferable Active Learning that estimates the impact of acquiring diverse kinds of supervision on the underlying topic model. We implement our model on the example problem of jointly learning Genre Classification and Collaborative Filtering and apply the novel Active Learning strategy for the cold-start problem, i.e. recommending and classifying new or unrated items.

3. **MTAL for Structured Tasks:** How to selectively acquire training data by leveraging known structured relationships among tasks to simultaneously improve them? In Chapter 5, we present a hierarchically regularized Bayesian framework for hierarchical classification. Then we describe two strategies for Active Learning on the proposed Multi-Task learning model. The first strategy chooses to acquire

supervision over instances that are expected to improve the consistency of the parent-child predictions. The second strategy provides higher preference in acquiring supervision at nodes that are deemed to be more influential over their descendants. Combining these two strategies with the popular uncertainty-based Active learning strategy leads to superior performance over single-task Active learning.

4. **Oracle-sensitive MTAL:** How to model per-task training data accessibility to avoid soliciting supervision for instance-task pairs that are beyond the oracle's area of expertise? In Chapter 6, we present a novel strategy for estimating the propensity of the oracle to provide supervision for certain instance-task pairs. Utilizing this score in conjunction with the popular Bayesian Active Learning approach for Collaborative Filtering, we see remarkable decrease in the amount of failed active queries, while still sustaining improvement over popular baselines on benchmark datasets.

In the rest of this chapter, we present the formal problem description and notation used in this thesis. Later, in Chapter 2, we review the current state of art in Active Learning, Multi-Task Learning and Multi-Task Active Learning, followed by a brief survey of the applications relevant to this thesis, namely, Hierarchical Classification, Adaptive Filtering and Collaborative Filtering. As outlined above, the major contributions of this thesis form the content of Chapters 3 through 6. Finally, we conclude the dissertation with a summary of the major contributions, followed by directions for future work.

2

Background

Multi-Task Active Learning (MTAL), as motivated in Chapter 1, is a novel research area. Most work in Active Learning has focused on selecting training instances that will improve a single task, while most work in Multi-Task learning has focused on learning interdependent task models. To clearly understand MTAL, it is therefore crucial to understand the strengths and limitations of various approaches in the two related domains. Here, we review the relevant work in the areas of Active Learning and Multi-Task Learning. We also provide a general overview of concepts in applications that we will be using to demonstrate the strengths of using MTAL.

2.1 Active Learning

Active Learning (AL) has been an actively researched field for about two decades [Seung et al., 1992] [1]. The primary goal in AL is to minimize the amount of training instances required for learning a task by soliciting labels for only those unlabeled instances which are most beneficial for learning a task. Several approaches have been developed for achieving this goal. *Risk Minimization*-based AL approaches choose instances that

[1] Settles [Settles, 2009] maintains a regularly updated comprehensive survey of important work in AL

maximize the expected reduction in future generalization error of the learnt model if those instances were added to the training set. One of the earliest risk-minimization approaches [Cohn et al., 1995] re-trains the classifier over all possible labelings of each instance in the unlabeled set (individually) and compute the expected risk reduction to select the instance that leads to maximal reduction. Re-training a classifier for each instance in the unlabeled set can be computationally infeasible for real applications. Consequently, [Roy and Mccallum, 2001] proposed a Monte Carlo estimation of the error reduction and suggested an efficient incremental training procedure for Naive Bayes classifier to overcome the re-training costs. Consequently, the later similar approaches are also known as Expected Error Reduction approaches. [Donmez and Carbonell, 2008b] demonstrated the effectiveness of using a sampling estimation of the error reduction to avoid re-training the model for each instance in the unlabeled pool. Risk-minimization approaches to AL have been successfully applied to several other different classifiers such as Gaussian Random Fields [Zhu et al., 2003] and Logistic Regression [Guo and Greiner, 2007]. Although most work in this area has been developed in the context of classification, risk minimization approaches can also be defined for other applications that involve minimizing error in the context of other performance measures such as F1, Precision, Recall, and Mean Absolute Error (MAE).

Other AL approaches do not try to directly minimize the future expected risk. Instead, these approaches are based on the idea that instances meeting certain criteria will probably achieve the goal of rapidly improving the system nevertheless. *Uncertainty-based sampling* approaches try to select instances that the current learnt model is most uncertain about. The uncertainty is usually defined as the inability of the classifier to confidently classify an instance into a class. One of the first uncertainty-based approaches [Lewis and Gale, 1994] was applied in the context of Naive Bayes classifiers to select instances that lie on the decision boundary of the classifer. Similar strategies have been developed in the context of other classifiers such as Support Vector Machines

(SVM) [Tong and Koller, 2000] and Decision Trees [Abe and Mamitsuka, 1998; Saar-tsechansky and Provost, 2004]. In the context of SVMs, it has been shown that selecting instances close to the decision boundary amounts to selecting instances that lead to maximal reduction in the version space [Tong and Koller, 2000], i.e. the space of hypotheses consistent with the available training instances. This is desirable as the search space for the learnt model is significantly reduced thereby ensuring that the learnt model is probably one of the best suitable models for the task. There have also been heterogeneous approaches in which the instances are selected based on uncertainty on one classifier and added as training data to another kind of classifier after obtaining the required labels. For example, [Lewis and Catlett, 1994] select instances using a Naive Bayes classifier and add to the training set of a C4.5 decision tree classifier. Among these, the earliest approach is the query-by-Committee [Freund et al., 1997; Seung et al., 1992] approach, where the uncertainty of an instance is measured as the disagreement among a committee of diverse classifiers. Several approaches have been studied for ensuring diversity of the committee members [Melville and Mooney, 2004; Sarawagi and Bhamidipaty, 2002].

[Mccallum and Nigam, 1998] show that uncertainty-based sampling is not robust against outliers, and there is no guarantee that the trained classifier will lead to better generalization performance. There have also been attempts at using ensemble of classifiers to determine the difficulty of classifying an instance and choosing the most difficult such instances. An alternative is the density-based sampling approach and its variants. By choosing cluster centroids of the unlabeled data, such approaches try to make active selection robust against outliers, and also ensure that enough training instances are available in the most crucial regions of the feature space that are truly representative of the unseen instances. For example, the approach by [Xu et al., 2003] selects cluster centroids of instances near the boundary of the classifier. McCallum and Nigam [Mccallum and Nigam, 1998] take a different approach of Density-weighted

uncertainty sampling and use the available training set to initialize clusters of unlabeled examples, and select uncertain examples in regions of higher density. Donmez and Carbonell [Donmez et al., 2007] demonstrate that uncertainty-based approaches perform particularly poorly when the available training set is tiny and density-based approaches are not effective with larger training set sizes. They instead propose a Dual Strategy for Active Learning (DUAL), a combination stratey which starts with a density-based approach and switches to an uncertainty-based approach after reaching certain criteria.

[Tong and Koller, 2001a] proposed a Bayesian Active Learning approach for parameter learning in probabilistic graphical models. Intuitively, this approach tries to identify instances that when added to the training set will lead the learnt model rapidly towards the true model (parameters). For comparing the learnt model and the expected true model, KL-Divergence between the two models is used as a loss function to be minimized.

These basic AL approaches have several application-specific extensions in diverse domains such as adaptive filtering, collaborative filtering, natural language processing, sensor network deployment, and rank learning. We will be reviewing relevant important extensions in the Section 2.4. Additionally, AL approaches have also been developed for special scenarios such as structured output prediction, for example, sequences and trees [Settles and Craven, 2008]. Another special scenario, active feature labeling, i.e. directly soliciting categories to synthetic documents containing single chosen feature and adding such documents to the training set [Godbole et al., 2004]. Further savings, in terms of data acquisition costs, is due to Active feature-value acquisition, i.e. soliciting feature-values for only the most salient features [Melville et al., 2004]. Additionally, there has been work on active learning with variable annotation costs [Kapoor et al., 2007; Settles et al., 2008], and taking into account the various characteristics of labeling oracles, such as noisiness, expertise, reliability, costs and reluctance [Donmez and Carbonell, 2008a; Donmez et al., 2009].

2.2 Multi-Task Learning

Multi-Task Learning (MTL) has been an active research area in machine learning for more than a decade [Caruana, 1997]. Several learning tasks are related and MTL approaches attempt to leverage such inter-task relationships to enable information sharing among tasks. Such information sharing is particularly useful when each of the individual tasks have limited training data available. Popular MTL approaches have empirically demonstrated significant performance improvements by simultaneously learning related tasks, as compared to learning each in isolation [Jebara, 2004; Jian Zhang and Yang, 2005; Xue et al., 2007; Yu et al., 2005]. Some approaches have also theoretically emphasized the effectiveness of MTL [Ando et al., 2005; Baxter, 2000; Ben-david and Schuller, 2003].

Discovering the *relatedness* of the tasks, and then utilizing it to share information is the key goal in MTL. Most MTL approaches differ in their representation of the task-relationships. The earliest success with MTL was demonstrated in Neural Networks [Caruana, 1997; Silver and Mercer, 2001; Thrun, 1996] where the multiple tasks shared hidden layer nodes. Among the frequentist approaches, the earliest approach applied shrinkage methods to multiple linear regression tasks [Breiman and Friedman, 1997]. A similar recent approach [Ando et al., 2005] additionally harnesses unlabeled data for learning predictive structures for multiple tasks jointly. Seeger et al. [2004] represented task relationships as linear mixtures of Gaussian Processes for multiple response prediction in classification and regression. Some of the popular approaches enable information sharing through shared Bayesian priors among tasks. The intuition here is that related tasks will have somewhat *close* parameters and a common prior or regularizer ensures this. Among the first works in applying shared Bayesian priors to MTL also discovered bounds on the amount of information required to learn a task when it is learnt simultaneously with other tasks. Since tasks have

different levels of relationships among themselves, more flexible alternatives have been proposed that create hierarchies of common priors. Yu et al. [2005] applied a hierarchical Bayesian model to Gaussian Processes (GP) where each task is a linear model. This approach was partly inspired by earlier work on using GP covariance for representing inter-task relationships [Lawrence and Platt, 2004; Minka and Picard, 1997]. A similar approach using Dirichlet Processes was developed for multiple logistic regression classifiers [Xue et al., 2007]. A more generic and comprehensive hierarchical Bayesian approach allows several multi-task scenarios such as independent tasks, noisy tasks, cluster of tasks, tasks with sparse parameters, duplicated tasks and evolving tasks [Zhang et al., 2008]. This approach constructs the task-specific parameters as a linear combination of a shared and local model, with the local model becoming more dominant as more task-specific evidence is observed for a particular task. The different variants are generated by suitably modifying a linear mixing matrix that combines shared and local models. Another set of approaches use shared regularizers to achieve the same effect as shared Bayesian priors among tasks [Argyriou et al., 2008; Jacob and Bach, 2008; Micchelli and Pontil, 2004]. One such regularization approach has been developed for MTL using Kernel methods [Evgeniou et al., 2005]. A newer regularization approach [Agarwal et al., 2010] for enabling task-relatedness is based on the idea of manifolds whereby it is assumed that the parameters of related-tasks are drawn from the same low-dimensional manifold, an idea derived from manifold learning, a non-linear dimensionality reduction technique. Recently, there has also been interest in heterogeneous MTL for simultaneously modeling different kinds of tasks [Yang et al., 2009b].

2.3 Multi-Task Active Learning: A new field

Active learning has been studied in the context of multiple tasks but never when those tasks are learnt jointly using Multi-task learning approaches outlined in Section 2.2. It is

unclear how learning the tasks jointly can benefit Active learning algorithms by inferring and utilizing inter-task relationships. Most of the current Active learning approaches desgined for scenarios such as multi-labeled classification attempt to select instances that will improve a single task in isolation [Esuli and Sebastiani, 2009]. Other Active learning approaches that do select instances based on their benefit to several tasks, do so by ignoring the implicit inter-task relationships [Reichart et al., 2008; Singh et al., 2009; Yang et al., 2009a], leading to sub-optimal savings.

As another related direction, there has been recent work Yang et al. [2012] in understanding the theoretical complexity of Active learning in the Transfer Learning setting. Their work studies the asymptotics of the number of labeled examples that are required to learn a sequence of target tasks, to gauge the benefit of transfer learning in reducing the number of labeled examples required. To explain the label complexity, they use self-verifying Active learning algorithm, which is a single-task Active learning strategy that is not specifically formulated for learning from multiple tasks by explicitly estimating the benefits across several tasks.

Task-selection as an Active learning problem has been studied in recent years. In the context of linguistic annotations, [Reichart et al., 2008] use a simple strategy of selecting tasks in a round-robin fashion and then choosing the best instances to improve the chosen task. The approach by [Roth and Small, 2006] selects the most uncertain task to improve and does not take into account the benefit of adding supervision to a task on other similar or related tasks. When explicit inter-task constraints are available, for example in hierarchical classification, the approach by [Zhang, 2010], scores the instances for every task in isolation and then combines these scores using the inter-task constraints to estimate the overall benefit of labeling an instance. Their approach seems to be the most closest approximation of the Multi-task Active learning setting, however, they do not use joint-learning of tasks for further reduction in supervision costs.

The oracle's expertise to supervise items has been studied as part of a recent work in Proactive Learning [Donmez and Carbonell, 2008a], albeit in the single-task setting.

2.4 Relevant Applications

We plan to demonstrate the effectiveness of our approaches on several important machine learning applications. In this section, we describe some of the application-specific approaches in additional details as groundwork for laying out our ideas in later sections. We also review the current state-of-art in applying AL approaches to these domains.

2.4.1 Hierarchical Classification

Hierarchical Classification (HC) [Lewis et al., 2004b] applications are manifested in fields as diverse as large-scale web-page taxonomies and structural protein classification ontologies. The hierarchical classification scenario is a subset of the multi-class classification problem where the outputs are constrained according to a pre-defined explicit hierarchy. In this case, it is wasteful to create all-pairs of one-vs-one or one-vs-rest classifiers. An instance is now to be classified only among the sibling nodes at a level of the hierarchy, thereafter, among the children of the selected class, descending down to a leaf node in this manner.

The popular strategy [Liu et al., 2005a] for HC uses a battery of hierarchically arranged classifiers such as the Logistic Regression (LR) classifiers in a Hierarchical Divide and Conquer (HDC) setup. An alternative similar approach [Cesa-bianchi and Zaniboni, 2006] uses a battery of Support Vector Machine (SVM) classifiers. For consistency with the later discussion, we will focus on the Bayesian Logistic Regression (BLR) model applied at every node of the hierarchy. Specifically, given a training set $\mathbb{L}_n$ at a node n of the hierarchy, the decision function can be defined as:

$$P(y_n|\mathbf{x}, \mathbb{L}_n) = \int_{\mathbf{w}_n} P(y_n|\mathbf{x}, \mathbf{w}_n) P(\mathbf{w}_n|\mathbb{L}_n) d\mathbf{w}_n \tag{1}$$

In the above Equation, $P(\mathbf{w}_n|\mathbb{L}_n)$ is a shorthand for $P(\mathbf{w}_n|\mathbb{L}_n, \mu, \mathbf{\Sigma})$, as the parameter $\mathbf{w}_n$ is drawn from a prior Gaussian distribution $\mathbf{w}_n \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ for regularization. And,

$$P(y_n = 1|\mathbf{x}, \mathbf{w}_n) = \sigma(\mathbf{w}_n^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}_n^T \mathbf{x}}} \tag{2}$$

After successfully classifying the instance at node n, the instance is passed on to the descendant classifiers (children) $\varphi(n)$ of node n for further fine-grained classification. The training set $\mathbb{L}_{n'}$ for a descendant classifier n' is a subset of the parent node's training set $\mathbb{L}_{n'} \subseteq \mathbb{L}_n$, with only positive instances from the parent set being passed along and the negative instances follow alternative trajectories.

2.4.1.1 Active Learning for Hierarchical Classification

Active Learning has been extensively studied for flat-classification tasks but not for hierarchical classification tasks. As a more relevant work, [Zhang, 2010] demonstrate an approach that utilizes the inter-task output constraints to better identify instances that can lead to improvements in several tasks. Specifically, they compute the *reward* of labeling an instance for each task in isolation and then combine these task-specific scores as an expectation over possible paths that the output can take over the constrained network of outputs. However, they do not jointly learn these tasks and consequently, the savings might be sub-optimal due to lack of information sharing among learnt models for the tasks.

An similar alternative simple approach might be devised by extending the popular uncertainty-based approach by considering the hierarchical constraints. For probabilistic classifiers, the popular uncertainty-based approach uses the classification entropy as an AL scoring function. The higher the entropy, the more confusing the instance is for

the current classifier, and hence the classifier may benefit from obtaining the label for such an instance. In the case of hierarchical classification, the instance will be passed through the hierarchical structure to several classifiers. Thus, we can esimate the *expected* classification entropy of an instance as follows:

$$\mathcal{H}(\mathbf{x}, n) = \left(\sum_{m \in \varphi(n)} p_m \log p_m \right) + \left(\sum_{m \in \varphi(n)} p_m \mathcal{H}(\mathbf{x}, m) \right) \tag{3}$$

where $p_m = P(y = m|\mathbf{x}, \mathbf{w}_n)$. Applied recursively, starting at the root of the hierarchy, Equation 3 will compute the local-classification entropy combined with the expected entropy for each of the descendants, leading to an overall uncertainty score for the instance $\mathbf{x}$.

2.4.2 Adaptive Filtering

Adaptive Filtering (AF) [Robertson and Soboroff, 2002] systems monitor a stream of documents to filter out documents that are relevant to the particular task or user. For example, a stock analyst would like to filter out all the news about stocks in his portfolio, while a technology entrepreneur might be interested in news about latest technology startups. Each of these users can be considered a *task* from the AF system's perspective. AF systems model user interests based on the initial information request (e.g. a query) presented by the user, and the subsequent relevance feedback provided by the user for the results presented to the user. For example, one popular AF approach involves the use of Logistic Regression classifier [Jaakkola and Jordan, 1996] to classify documents into relevant and non-relevant categories for that particular user. The relevance feedback received from the user is used to supplement the training data to retrain the classifier for future predictions.

We briefly describe a popular representative AF approach based on the Logistic Regression (LR) classifier. The LR classifier estimates relevance of a document using

the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ as follows:

$$P(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}} \tag{4}$$

In the above equation, $\mathbf{x}$ is a feature vector representing a document, and $\mathbf{w}$ is the weight vector of regression coefficients. $\mathbf{w}$ is usually fit by maximum likelihood estimation on the available relevant and non-relevant training documents $\mathbb{L}$ for the task.

For our purposes, for consistency among approaches, we will be focusing on the Bayesian Logistic Regression variant. In the Bayesian setting, the parameters $\mathbf{w}$ are usually drawn from a diffuse Gaussian prior distribution $\mathbf{w} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, to ensure regularization. For a fully Bayesian treatment, instead of using a point estimate of the parameters $\mathbf{w}$, one may integrate over the posterior distribution of $\mathbf{w}$ to obtain a Bayesian estimate of $P(y = 1|\mathbf{x}, \mathbb{L})$

$$P(y = 1|\mathbf{x}, \mathbb{L}) = \int_{\mathbf{w}} P(y = 1|\mathbf{x}, \mathbf{w})P(\mathbf{w}|\mathbb{L})d\mathbf{w} \tag{5}$$

In the Adaptive Filtering setting, the performance of a system is usually measured in terms of the *utility* of documents delivered by that system. For example, in the TREC Filtering track, one popular utility metric in the TREC AF community is $T9U = \psi_1 R + \psi_0 N$, where R and N are the number of delivered results the user considered relevant and non-relevant respectively. ψ_1 and ψ_0 are the benefit achieved and loss incurred, by the user due to reading the relevant and non-relevant documents respectively. For TREC-9, $\psi_1 = 2$ and $\psi_0 = -1$.

From the system perspective, the expected utility of delivering a document can be computed as:

$$\mathcal{U}(\mathbf{x}|\mathbb{L}) = \sum_{y \in \{0,1\}} \psi_y P(y|\mathbf{x}, \mathbb{L}) \tag{6}$$

Instances with $\mathcal{U}(x) > t$ are delivered, where t is the dissemination threshold learnt via cross-validation or set to 0 [Zhang et al., 2003].

2.4.2.1 Active Learning for Adaptive Filtering

[Zhang et al., 2003] devised a strategy called *exploration and exploitation* to perform Active Adaptive Filtering. The *exploitation* component is a passive AF strategy that delivers an item if $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathbb{L}) > 0$. The *exploration* component is an Active Learning component based on a metric called *Utility Divergence*. Utility Divergence computes the difference between the expected utility of a hypothetical *true* model Θ^* and the model based on expected feedback $\Theta|(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ and instances are scored based on their potential to reduce the gap between utility of the true model and utility of the current model.

2.4.3 Collaborative Filtering

Collaborative Filtering(CF) [Breese et al., 1998; Hofmann, 2003; Hofmann and Puzicha, 1999; Si and Jin, 2003] has now become a popular and important technique for finding user-relevant information, e.g. movies, books and music, based on similarity of interests between groups of users. CF approaches identify interest groups of users and recommend movies based on memberships of users to different interest groups. Current CF approaches can be divided into two major categories, model-based and memory-based (also known as instance-based or lazy learning). Memory-based techniques are similar to K-nearest neighbor algorithms as learning is delayed till the prediction phase. The model-based approaches make the intuitive assumption that users/items can be grouped based on their interests, and all model-based techniques apply different techniques to discover these latent or hidden *interest*-groups. Some of the popular model-based approaches are Aspect Model [Hofmann, 2003; Hofmann and Puzicha, 1999], Flexible Mixture Model [Si and Jin, 2003], and the Multiple Cause Vector Quantization model [Boutilier et al., 2003] and most recently, matrix factorization [Salakhutdinov and Mnih, 2008a] approaches. Collaborative filtering has also been demonstrated as an intermediate step for other application, for example for generating personalized search

results in information retrieval [Harpale et al., 2010].

Aspect Model [Hofmann, 2003; Hofmann and Puzicha, 1999] is a probabilistic latent semantic model in which users are considered to be a mixture of multiple interests or aspects. Thus each user $m \in \mathbb{M}$ has a probabilistic membership in each of the aspects, $z \in \mathbb{Z}$. Furthermore, users in the same interest groups have same movie-rating patterns, thus, users and items (e.g. movies $x \in \mathbb{X}$) are independent from each other given the latent class variable, z. Each item x can be rated on a finite scale (e.g. $\mathbb{Y} = \{1, 2, \ldots, 5\}$), with the rating $y \in \mathbb{Y}$. Thus the probability of each tuple in the dataset can be computed as follows:

$$P(y|x, m) = \sum_{z \in Z} p(y|x, z)P(z|m) \tag{7}$$

Equation 7 consists of two parts, $p(y|x, z)$ and $P(z|m)$. It can be observed that the first term $p(y|x, z)$ does not depend on the user and represents the group-specific model. The *global-model* consists of such group-specific models. The second term $P(z|m)$ is the user-personalization term. The *user-model* θ_m consists of such user-personalization terms i.e. $\theta_m = \{\theta_{m_z} : \theta_{m_z} = P(z|m), \forall z \in \mathbb{Z}\}$.

The **Flexible Mixture Model** (FMM) Si and Jin [2003], is a modified version of the Aspect Model in which there are two layers of latent aspects $\{z_u, z_m\}$, one grouping the users with similar interests (z_m) and one grouping the items with similar patrons (z_x). The probability of a tuple (r, x, m), is factored similar to the Aspect Model:

$$P(r, x, m) = \sum_{z_m, z_x} P(z_m)P(z_x)P(u|z_m)P(m|z_x)P(r|z_m, z_x) \tag{8}$$

Both the Aspect Model and the Flexible Mixture Model are probabilistic clustering models and are fit to the data using Expectation Maximization (EM) [Dempster et al., 1977; Hofmann, 2003].

Users generally follow different distributions to rate the items. For example, some users provide very extreme ratings while some users provide moderate ratings. For learning models which generalize over such diverse users, the ratings are usually normalized so that ratings for each user have zero mean and unit variance [Hofmann, 2003].

2.4.3.1 Active Learning for Collaborative Filtering

When new users join a system, although the system already has a strong global-model, the user-model for new users is very weak, since the system has very few ratings available from such new users. Thus, the goal of active learning in CF is to obtain ratings for more items from the new user. Instead of randomly selecting movies for rating from the user, active learning algorithms minimize the number of such queries required to learn a stronger user-model.

One of the popular techniques of active learning is to solicit a rating for an item which will minimize the expected entropy of the user model. In the context of Collaborative Filtering, this leads to the following equation for a user m, where θ_{m_z} denotes the user-group mixing probabilities $P(z|m)$, and $\theta_{m_z|x,y}$ denotes the model posterior after retraining the user-model based on a newly obtained rating y for movie x from the user i.e. $P(z|x,m,y)$:

$$x_m^* = \arg\min_{x \in \mathbb{X}} - \left\langle \sum_{z \in Z} \theta_{m_z|x,y} \log \theta_{x_z|x,y} \right\rangle_{P(y|x,m)} \tag{9}$$

Equation 9 denotes the expected entropy of the user-model after being trained over additional information of rating movie x with rating y. Since the exact rating y is not known for the unrated movies, the expected value of rating based on the current model $P(y|x,m)$ is used, as shown in the equation. Minimizing entropy leads to pure interest groups, in which each user has strict adherence to just one interest group. [Jin and Si, 2004] demonstrate that minimizing entropy is counterproductive for active learning in

the collaborative filtering domain, since in reality, users can be a mixture of multiple interests.

As a remedy to this problem, [Jin and Si, 2004] propose a Bayesian Selection (BS) approach similar to the active learning approach towards parameter estimation in Bayesian networks [Tong and Koller, 2001b]. This approach identifies item x, such that the updated model $\theta_{m|x,r}$ will be accelerated towards the true user model θ_m^{true}.

$$x_m^* = \arg\max_{x \in \mathbb{X}} \left\langle \sum_{z \in Z} \theta_{m_z}^{true} \log \frac{\theta_{m_z|x,y}}{\theta_{m_z}^{true}} \right\rangle_{P(y|x,m)} \tag{10}$$

Equation 10 is a negated KL-Divergence equation which is maximized when the estimated distribution is equal to the true distribution being modeled.

Since the *true* user model is unknown beforehand, it is estimated as the expectation over the posterior distribution of the user model. Similar to equation 9, the rating y is unknown for unrated movies and the expected value is used instead. [Jin and Si, 2004] provides a computationally efficient way of approximating the posterior distribution.

3

Multi Task Active Learning for Homogeneous Tasks

In this chapter, we lay out the fundamental concepts of our approach to Multi-Task Active Learning (MTAL). We define a general approach to estimating system-wide (over several tasks) benefit of labeling an instance.

A central theme in Multi-Task Learning (MTL) frameworks is the creation of shared structure/parameters/regularizers to model inter-task relationships and enable information sharing among related tasks. A suitable Active Learning (AL) strategy must work with such generalized MTL frameworks to estimate the benefit of labeling an instance on the reference task as well as other tasks. In this chapter, we present one such general strategy in the context of a simple hierarchical Bayesian MTL model shown in Figure 1. Consider a set of tasks $\mathbb{M} = \{1, \ldots, M\}$, each with task-specific parameters $\mathbf{w}_m$. The inter-task relationships are enabled by drawing the task-specific parameters from shared priors Θ. To be able to *infer* the inter-task relationships, Θ is not fixed, but rather drawn from a distribution governed by fixed hyperpriors Φ. The particular distributions or nature of these priors and hyperpriors is irrelevant to the ensuing discussion and will not be elaborated here. By filling in such details and with custom modifications, this

model can be suitably adapted to some of the popular MTL frameworks [Xue et al., 2007; Zhang et al., 2008].

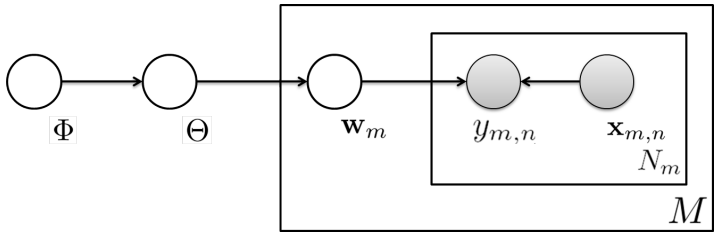


Figure 1: Graphical model representation of a simple generic hierarchical Bayesian Multi-task learning approach (several irrelevant details have been left out to maintain the generic nature)

For such a general Bayesian system, the predictive model for task $m \in \mathbb{M}$ can be summarized as follows:

$$p(y_m|\mathbf{x}_m, \Phi, \mathbb{L}) = \int_{\Theta} \int_{\mathbf{w}_m} P(y_m|\mathbf{w}_m, \mathbf{x}_m)\ P(\mathbf{w}_m|\Theta)\ P(\Theta|\Phi, \mathbb{L})\ d\mathbf{w}_m\ d\Theta \tag{11}$$

To evaluate the quality of the predictive model, we can define the performance measure $\Psi_m(\mathbb{L})$ as the generalized cumulative performance of the task on all possible instances from the input distribution of that task.

$$\Psi_m(\mathbb{L}) = \int_{\mathbf{x}_m} \psi_m(p(y_m|\mathbf{x}_m, \Phi, \mathbb{L}))\ p(\mathbf{x}_m)\, d\mathbf{x}_m \tag{12}$$

where $\psi_m(p(y_m|\mathbf{x}_m, \Phi, \mathbb{L}))$ denotes the task's performance on an instance $\mathbf{x}_m$ for the predictive model $p(y_m|\mathbf{x}_m, \Phi, \mathbb{L})$.

Based on this chosen model, we now describe a general Multi-Task Active Learning (MTAL) strategy in the following sections. Although our treatment is presented in the context of this model, at the conceptual level it should be adaptable to other popular MTL frameworks, even non-Bayesian models through corresponding mechanisms for representing inter-task relationships. We will demonstrate a concrete example later in

Chapter 3.5.

3.1 Local Active Learning

A typical single-task focused AL strategy will select an instance $\mathbf{x}_m^* \in \mathbb{U}_m$ to label for a reference task $m \in \mathbb{M}$ such that the chosen instance provides the maximal performance improvement in that task, as described earlier, which we reproduce here for clarity.

$$\mathbf{x}_m^* = \underset{\mathbf{x}_m \in \mathbb{U}_m}{\arg\max} \langle \Psi_m(\mathbb{L} \cup (\mathbf{x}_m, y_m)) \rangle_{y_m} - \Psi_m(\mathbb{L}) \tag{13}$$

We can ignore the second component of the above equation $\Psi_m(\mathbb{L})$ as it is equal for all new instances and just focus on maximal future performance instead. Also, since the true label y_m is unknown during active selection, the first component of Equation 13 is computed as an expectation over values of y_m, as indicated by the angled brackets. Computing the future peformance requires the estimation of the posterior predictive distribution after addition of the potential instance.

$$p(y'_m|\mathbf{x}'_m, \Phi, \mathbb{L} \cup (\mathbf{x}_m, y_m)) = \int_\Theta \int_{\mathbf{w}_m} P(y'_m|\mathbf{w}_m, \mathbf{x}'_m)\ P(\mathbf{w}_m|\Theta)\ P(\Theta|\Phi, \mathbb{L} \cup (\mathbf{x}_m, y_m))\ d\mathbf{w}_m\ d\Theta \tag{14}$$

While such an approach seems reasonable, it leads to several questions about the impact of the newly added instance into the model. In the Equation 14, we are expecting the shared parameters Θ to be influenced if the instance $(\mathbf{x}_m, y_m)$ were added to the training set of task m. But, in making the choice of this instance, we are only focused on the performance measure of the task m. Is this really reasonable? The global parameters Θ influence the performance of other tasks in the system. What if this strategy of narrowly focusing on the performance improvement of the task m leads to degraded performance of other tasks in the system?

3.2 Global Active Learning

An alternative is to select an instance that will be evaluated solely on its benefit to the local task model $\mathbf{w}_m$, thereby not allowing the instance to influence the global parameters Θ, and avoiding the adverse impact it can have on the performance of other tasks. Simply put, we can compute the posterior only at the *local* task-level by re-writing Equation 14 as follows:

$$p(y'_m|\mathbf{x}'_m, \Phi, \mathbb{L}\cup(\mathbf{x}_m, y_m)) = \int_{\Theta}\int_{\mathbf{w}_m} P(y'_m|\mathbf{w}_m, \mathbf{x}'_m)\ P(\mathbf{w}_m|\Theta, \mathbf{x}_m, y_m)\ P(\Theta|\Phi, \mathbb{L})\ d\mathbf{w}_m\ d\Theta \tag{15}$$

3.3 Benevolent Active Learning

By not allowing the training set from one task to influence the other tasks in the system, this defeats the purpose of multi-task learning. We propose an alternative strategy that allows posterior updates to Θ, but instead of restricting performance evaluation to a particular reference task, we compute the cumulative performance over all tasks in the system influenced by Θ to avoid negative impact on other tasks.

$$\mathbf{x}^*_m = \underset{\mathbf{x}_m\in\mathbb{U}_m}{\arg\max}\ \Psi_{\mathbb{M}}(\mathbb{L}\cup(\mathbf{x}_m, y_m)) = \underset{\mathbf{x}_m\in\mathbb{U}_m}{\arg\max} \sum_{m'\in\mathbb{M}} \Psi_{m'}(\mathbb{L}\cup(\mathbf{x}_m, y_m))) \tag{16}$$

Here, for the posterior predictive model, we use Equation 14 which considers the influence of the new instance on shared parameters Θ, and thereby allows us to update the model parameters for other tasks for estimating the benefit.

3.4 Discussion

So which of these strategies is better? Is there really one good strategy? In Bayesian approaches with deeper shared parameter hierarchies, there might be even more options in choosing the right granularity of instance influence. In our work on Multi-Task Active Learning, we present this flexible choice of **Circle of Influence (CoI)** of the instance. Simply put, CoI is the set of parameters that are expected to benefit from the supervision obtained on an instance. One may choose to improve only the parameters of the reference task for the narrowest CoI, or performance improvement over all tasks which we consider the widest CoI. In MTL scenarios with groupings of related task, the CoI may be restricted at the group-level, or at different group levels in the case of hierarchical groupings.

One additional concern would be that of inductive bias caused due to selective sampling of examples. Especially in the case of Global and Local Active learning, choosing instances that are beneficial for one tasks can lead to a biased update of the common parts of the model, i.e. the shared prior over the tasks. This update might be detrimental to the overall performance of the system. Benevolent Active Learning might not suffer from such bias, since each chosen example is pre-screened for its beneficial impact on the other tasks of the system. Consequently, if it is necessary to utilize Local or Global Active Learning, it is also crucial to downplay the impact of such selectively chosen instances on the central components of the system. One crude remedy could be that of ensuring balance in the number of instances sampled per-task, so that each task is equally represented when updating the common components of the model.

Our recent work in applying AL strategies with varying CoI to Multi-Task Adaptive Filtering [Harpale and Yang, 2010] shows that there is no single CoI that always outperforms the rest and the choice is dependent on the various stages of learning. Instead, we find that a combination strategy using multiple CoI scores performs the

best. We describe this work in detail in Section 3.5.

3.5 Case Study: Active Multi-Task Adaptive Filtering

Adaptive Filtering (AF) Robertson and Soboroff [2002] systems monitor a stream of documents to filter out documents that are relevant to the particular task or user. For example, a stock analyst would like to filter out all the news about stocks in his portfolio, while a technology entrepreneur might be interested in news about latest technology startups. Each of these users can be considered a *task* from the AF system's perspective. AF systems model user interests based on the initial information request (e.g. a query) presented by the user, and the subsequent relevance feedback provided by the user for the results presented to the user. For example, one popular AF approach involves the use of Logistic Regression classifier Jaakkola and Jordan [1996] to classify documents into relevant and non-relevant categories for that particular user. The relevance feedback received from the user is used to supplement the training data to retrain the classifier for future predictions.

Most research in AF systems has focused on learning each task independently of other tasks. In this work, we will refer to such approaches as Single-Task AF (STAF) approaches. In the initial stages of learning, when the feedback from each user is limited, these approaches suffer from data sparsity, leading to weaker models, and consequently poorer performance. Multi-task learning methods have shown significant success in mitigating this per-task data sparsity problem by sharing information across multiple tasks. For example, to learn the interest-model of a particular stock analyst, it could be useful to identify common important features for portfolio tracking based on the feedback received from other stock analysts. Irrespective of their portfolios, all stock analysts are usually interested in common news regarding corporate announcements, balance sheets, government regulations, and day-to-day stock market indicators pertinent to their stocks.

In this work, we will refer to such approaches, which leverage information from multiple tasks, as Multi-Task Adaptive Filtering (MTAF) approaches.

The future performance of an AF system relies on the feedback received on delivered items. If the system myopically focuses on delivering only (perceived) *relevant* documents (better immediate performance), the feedback received on these documents may not lead to the best learnt task models (in the future), thereby limiting the usefulness of such feedback for future predictions. In this work, we present an Active Learning (AL) framework for MTAF that additionally takes into account the perceived benefit of feedback on items before making a delivery. In the MTAF setting, the AL system has a three-fold objective: 1) Provide relevant documents, 2) Feedback provided on a delivered document should maximally improve the task-specific performance in the future, 3) Feedback provided on a delivered document, should maximally improve the overall system (multiple tasks) in the future. The current AL approaches focuses only on the second objective, thereby narrowly focusing on task-specific performance improvements. In this work, for satisfying the goal 1, we chose the Multi-Task Logistic Regression with Dirichlet Process priors Blei and Jordan [2005]; Xue et al. [2007] for facilitating information sharing across tasks. For goals 2 and 3, we propose a novel AL framework based on a new scoring function called *Utility Gain*. This scoring function is inspired by the popular empirical risk minimization approaches in Active Learning Melville and Provost [2005]. However, the current empirical risk minimization approaches only focus on minimizing the risk of one task, while we develop variants that selectively focus on one task, global model, or all tasks, as required during the various phases of learning. Consequently, our framework chooses instances that might lead to maximal (expected) gain in performance of the system (task-specific or global) for future predictions.

Table 1: Important notation used in this chapter

Symbol	Description
M	Total number of tasks
$m \in \{1, \ldots, M\}$	A task
N_m	Total number of instances for task m
d	Dimensionality of feature space
$\mathbf{x}_{m,n} \in \mathcal{R}^d$	The n'th data instance for task m
$y_{m,n} \in \{0, 1\}$	The label of instance $\mathbf{x}_{m,n}$
$\mathbf{w}_m \in \mathcal{R}^d$	LR parameters for task m
$k \in \{1, \ldots, \infty\}$	A group or cluster
$\mathbf{w}^*_k \in \mathcal{R}^d$	LR parameters of the k'th group
$\phi_{m,k} \in [0, 1]$	Group k mixing proportion for task m

3.5.1 Our Approach: Multi-Task Adaptive Filtering

Owing to the success of Logistic Regression (LR) classifiers in Adaptive Filtering, we choose a multi-task approach based on LR classifiers. Our chosen approach is depicted graphically in Figure 2. Table 1 lists the notation used in this work. This approach is based on the Mutli-Task classification approach developed by Xue et al. [2007]. A similar approach, consisting of Support Vector Machines for individual tasks, has been used for content-enhanced collaborative filtering by Yu et al. [2004]. We summarize the generative process of the model in Algorithm 1. The approach clusters/groups related tasks by drawing the parameters for the related tasks from a mixture of Gaussians (as evident from the line 12 of the Algorithm 1). Intuitively, related tasks will share information by getting grouped into the same cluster(s). For example, features that are indicators of interests of a football fanatic (e.g. teams, scoreboards, win/loss decisions) may be inferred from the relevance judgments available from other sports enthusiasts. The grouping of related tasks ensures that unrelated tasks (e.g. stock-portfolio tracking vs sports-news filtering) do not contaminate, as such contamination might lead to poorer understanding of the individual tasks.

As the optimal number of groups is unknown apriori, a Dirichlet Process model is inferred over the tasks to discover the optimal number of groups pertinent to the tasks. The

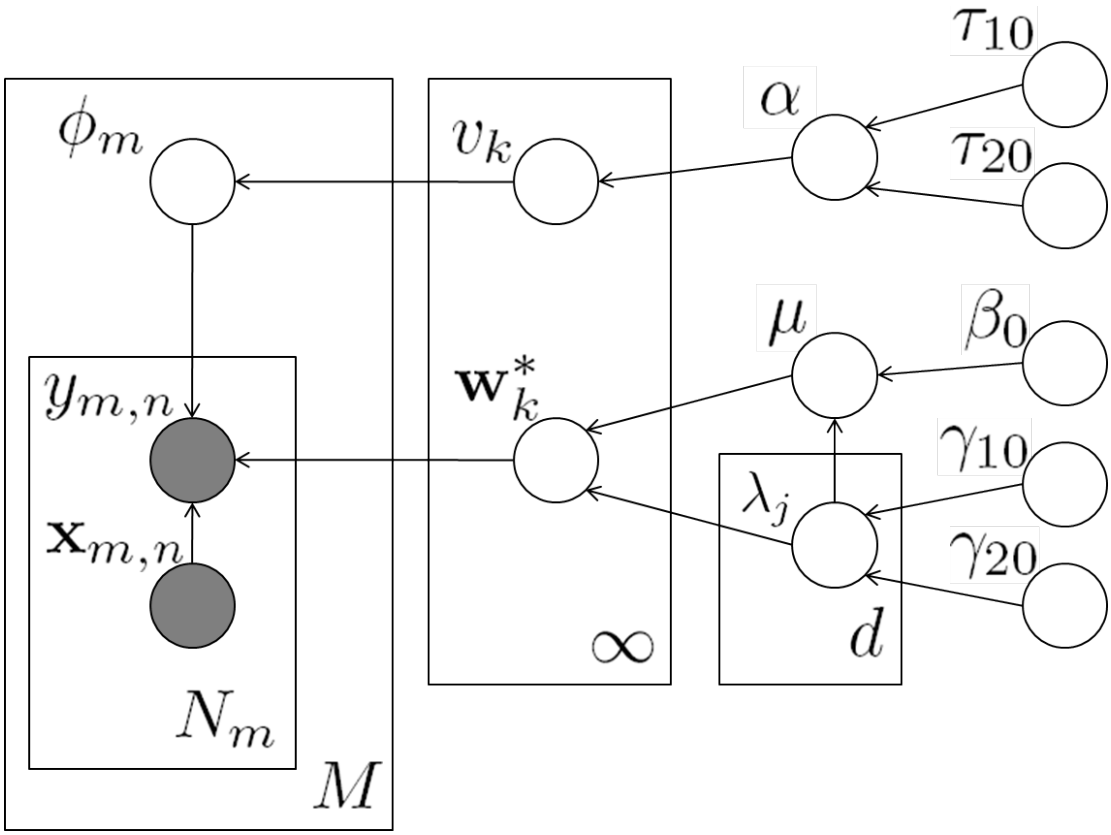


Figure 2: Graphical model representation of multi-task adaptive filtering based on Dirichlet Processes.

Dirichlet Process is essentially a mixture model with potentially *infininte* components. The actual number of components participating in the formation of the tasks is based on the parameter α, also known as the innovation parameter. Larger values of α lead to more participating components and vice-versa. The optimal value of α (and other parameters) can be inferred from the data using variational inference Xue et al. [2007].

With the knowledge of the inferred parameters for a task m, the decision function for a new instance $\mathbf{x}_{m,\bullet}$ follows the distribution:

$$P(y_{m,\bullet} = 1|\mathbf{x}_{m,\bullet}, \mathbf{w}_m) = \sum_{k=1}^{K} \phi_{m,k} \sigma(\mathbf{w}_k^{*T} \mathbf{x}_{m,\bullet}) \tag{17}$$

Algorithm 1 MTAF Generative Process

Fixed diffuse hyperpriors: $\tau_{10}, \tau_{20}, \beta_0, \gamma_{10}, \gamma_{20}$
Draw $\lambda_j \sim Gamma(\gamma_{10}, \gamma_{20}), \forall j = 1, \ldots, d$
Let Λ be a diagonal matrix with elements $\lambda_j, \forall j$
Draw $\mu \sim \mathcal{N}(0, (\beta_0 \Lambda)^{-1})$
Let $\Sigma = \Lambda^{-1}$
Draw $\alpha \sim Gamma(\tau_{10}, \tau_{20})$
Draw $v_k \sim Beta(1, \alpha), \forall k = 1, \ldots, \infty$
Draw $\mathbf{w}_k^* \sim \mathcal{N}(\mu, \Sigma), \forall k = 1, \ldots, \infty$
Let $\pi_k = v_k \prod_{i=1}^{k-1}(1 - v_i), \forall k$
for $m = 1$ **to** M **do**
    Draw $\phi_m \sim Multinomial(1; \pi_1, \ldots, \pi_\infty)$
    Let $\mathbf{w}_m = \sum_{k=1}^{\infty}(\mathbf{w}_k^*)\phi_{m,k}$
    Draw $y_{m,n} \sim Binomial(1, \sigma(\mathbf{w}_m^T \mathbf{x}_{m,n})), \forall n = 1, \ldots, N_m$
end for

Again, like in Equation 5, for a fully Bayesian treatment, we integrate over the posterior distribution of $\mathbf{w}_k^*$, instead of using a point estimate:

$$P(y_{m,\bullet} = 1 | \mathbf{x}_{m,\bullet}, \mathbf{w}_m) = \sum_{k=1}^{K} \phi_{m,k} \int \sigma(\mathbf{w}_k^{*T} \mathbf{x}_{m,\bullet}) P(\mathbf{w}_k^* | \mu, \Sigma, D) d\mathbf{w}_k^* \tag{18}$$

The above integral does not have an analytical solution. Xue et al. [2007] suggest the use of an approximate form of the integral based on MacKay [1992]. Combining Equation 18 and Equation 6, one can compute the expected utility $\mathcal{U}(\mathbf{x}_{m,\bullet})$ of delivering the document $\mathbf{x}_{m,\bullet}$. A passive MTAF approach will deliver the document if $\mathcal{U}(\mathbf{x}_{m,\bullet}) > t$, where t is a dissemination threshold, usually set to zero, or maybe learned via cross-validation. We call this approach *passive*, because in choosing to deliver $\mathbf{x}_{m,\bullet}$, the system did not foresee the benefit of getting the feedback on that document. Consequently, the user effort in providing feedback on this item may go wasted as it may not result in better results (for the user) in the future. We remedy this situation in the next section by proposing our Active Learning framework for the MTAF approach.

3.5.2 Our Approach: Active Learning for Adaptive Filtering

In a classification setting, an AL approach typically selects instances that (if labeled) are expected to improve the classification performance the most. Along similar lines, in the MTAF setting, an AL approach could estimate the perceived benefit of delivering an item in three different ways. Firstly, does the delivered item $\mathbf{x}_{m,\bullet}$ lead to improved performance on that task m in the future (based on feedback received on $\mathbf{x}_{m,\bullet}$). Secondly, will the feedback on delivered item $\mathbf{x}_{m,\bullet}$ lead to improvements in the global model (e.g. α, μ, Σ)? Finally, does feedback on the delivered item $\mathbf{x}_{m,\bullet}$ lead to improvements in other tasks in the system? We propose an Active Learning solution for each of these objectives in the following sections.

3.5.2.1 Local Active Learning

In this section, we discuss an AL approach that scores instances based on the perceived future benefit of delivering these instances to the current task m. Consequently, we first define our notion of *future benefit* $\mathcal{LUG}_m(\mathbf{x}_{m,\bullet})$ of delivering an instance to the task m.

$$\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \\ \mathcal{LSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) - \mathcal{LSP}_m(\mathcal{D}) \tag{19}$$

In the above equation, $\mathcal{LSP}_m(\mathcal{D}')$ denotes the expected system performance on task m when trained on any data $\mathcal{D}'$. Intuitively, the benefit of delivering an instance $\mathbf{x}_{m,\bullet}$ is estimated as the *improvement* in the task-specific performance (for task m), if the feedback of that instance $(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ was added to the training set $\mathcal{D}$. In Equation 19, the system doesn't know the true label $y_{m,\bullet}$ for the instance $\mathbf{x}_{m,\bullet}$. As a result, we

rephrase the equation as an expectation over the possible labels.

$$\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}) = \sum_{y\in\{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D)\mathcal{LSP}_m(\mathcal{D}\cup(\mathbf{x}_{m,\bullet}, y)) - \mathcal{LSP}_m(\mathcal{D}) \tag{20}$$

For the case of Adaptive Filtering, we define the system performance in terms of the expected utility of all potential instances the system will monitor. Consequently, $\mathcal{LUG}_m$ is the *localized* utility gain of task m.

$$\mathcal{LSP}_m(\mathcal{D}') = \int_{\Omega_{\mathcal{LU}_m}(\mathbf{x}_{m,\circ})} p(\mathbf{x}_{m,\circ})\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}')d\mathbf{x}_{m,\circ} \tag{21}$$

In Equation, 21, there are two important points to note. Firstly, the integral sums over $p(\mathbf{x}_{m,\circ})$, i.e. the task-specific population. It should be noted that each task in the system may have it's own sample population from which documents are being monitored/filtered. For example, a stock analyst may subscribe to news feeds from business magazines, while a sports enthusiast may subscribe to news feeds from sports magazines, leading to different $p(\mathbf{x}_{m,\circ})$ for these two tasks. (Ofcourse, the sports enthusiast is not interested in all sports news being served, but only those that match her interests. It is the goal of the AF system to identify those interests and filter the relevant news from the subscribed feed. Hence the system performance is evaluated over the sample population of the subscribed feed.). In this Equation, the domain of the integral is defined as $\Omega_{\mathcal{LU}_m}(\mathbf{x}_{m,\circ}) = \{\mathbf{x}_{m,\circ} : \mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}') > 0\}$. This means that the future system performance will only be calculated based on the instances that will be delivered; undelivered instances will be ignored from the system performance calculation due to the nature of the $T9U$ utility function.

The second subtle point in Equation 21, is the localized nature of the utility function $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}')$. For computing the expected future system performance $\mathcal{S}_m(\mathcal{D}\cup(\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$, we compute the utitlity $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}\cup(\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$ based only on the

updated *local* model $\mathbf{w}_m$ of the task m. This means, we do not consider the effect of updating the global parameters α, μ, Σ after observing the new training instance $(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$. If $\mathbf{w}_m$ is the current (trained on $\mathcal{D}$) local model, we can use the Bayes' rule to obtain the posterior distribution $P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet})$. We use the variational approximation method suggested by Jaakkola and Jordan [1996] to obtain the posterior. With the knowledge of the posterior distribution $P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet})$, the localized expected utility can then be defined as:

$$\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) = \int_{\mathbf{w}_m} \sum_{y \in \{0,1\}} \psi_y P(y|\mathbf{x}_{m,\circ}, \mathbf{w}_m) P(\mathbf{w}_m|\mathbf{x}_{m,\bullet}, y_{m,\bullet}) d\mathbf{w}_m \tag{22}$$

In the Adaptive Filtering setting, the system will then choose to deliver items that provide immediate utility $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D})$, or are potentially beneficial in improving the system in future iterations, as estimated by expected future utility $\mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D})$. We express the joint objective as a linear combination:

$$\mathcal{LAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_L \mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \tag{23}$$

Here, δ_L is the weightage given to the AL component of the system. We discuss the nature of this weight in Section 3.5.2.5. Owing to the localized focus of this approach on the perceived improvement of the current task, we call this approach Local Active Learning (LAL). We expect LAL to perform well if the global model, consisting of α, μ, Σ, is already strong, and thus does not need to be updated. Thus, LAL is expected to be effective on new tasks that are added to an already well-performing MTAF system. We deal with the case of the improving the global model in the next section.

3.5.2.2 Global Active Learning

As mentioned in the previous section, LAL does not foresee the updates/improvements in the global model consisting of $\Theta = \{\alpha, \mu, \Sigma\}$. Global AL (GAL) fixes this by updating

the global model for computing the expected utility gain in Equation 20. It should be noted that, just like LAL, the goal of GAL is still to improve the future performance of task m (the task to which the system plans to deliver instance $\mathbf{x}_{m,\bullet}$), albeit based on improvements in the global model. Consequently, we define the estimation of the *global* utility function $\mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$. This is a modification of localized utility $\mathcal{LU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet}))$ in equation 22.

$$\begin{aligned}
&\mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y_{m,\bullet})) \\
&= \int_{\Theta} \sum_{y \in \{0,1\}} \psi_y P(y|\mathbf{x}_{m,\circ}, \Theta) \\
&\qquad P(\Theta|\mathbf{x}_{m,\bullet}, y_{m,\bullet}) d\Theta \\
&= \int_{\Theta} \sum_{y \in \{0,1\}} \int_{\mathbf{w}_m} \psi_y P(y|\mathbf{x}_{m,\circ}, \mathbf{w}_m) \\
&\qquad P(\mathbf{w}_m|\Theta)(P(\Theta|\mathbf{x}_{m,\bullet}, y_{m,\bullet}) d\mathbf{w}_m d\Theta
\end{aligned} \tag{24}$$

It is important to note that the posterior model global $P(\Theta|(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ has been trained on additional available feedback from the task m only. This is because, we are computing the expected global future utility if the instance is delivered to task m, that is if the feedback is received from task m. Replacing local utility $\mathcal{LU}$ in Equation 21 with expected future global utility $\mathcal{GU}$, we get the *globalized* system performance $\mathcal{GSP}_m$ of task m.

$$\mathcal{GSP}_m(\mathcal{D}') = \int_{\Omega_{\mathcal{GU}_m}(\mathbf{x}_{m,\circ})} p(\mathbf{x}_{m,\circ}) \mathcal{GU}_m(\mathbf{x}_{m,\circ}|\mathcal{D}') d\mathbf{x}_{m,\circ} \tag{25}$$

Substituting the globalized system performance $\mathcal{GSP}_m$ from Equation 25 into Equation 20, we get the globalized utility gain

$$\begin{aligned}
\mathcal{GUG}_m(\mathbf{x}_{m,\bullet}) = &\sum_{y \in \{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D) \mathcal{GSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y)) \\
&- \mathcal{GSP}_m(\mathcal{D})
\end{aligned} \tag{26}$$

Finally, substituting the globalized utility gain $\mathcal{GUG}_m$ into Equation 27, we get the scoring function for the Global Active Learning GAL approach.

$$\mathcal{GAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_G \mathcal{GUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \tag{27}$$

3.5.2.3 Benevolent Active Learning

So far, LAL and GAL have focused on improved performance of the task m to which the system plans to deliver the instance $\mathbf{x}_{m,\bullet}$. But it is not clear if the feedback on that instance will lead to improvements in other tasks. To achieve this goal, we devise another AL score, called *Benevolent* AL, as the system tries to deliver instances to task m that might lead to improvements in other tasks. In this context, we revise the definition of $\mathcal{GSP}_m$ to $\mathcal{BSP}_m$, which sums over the performance of each task in the system, based on the feedback received on the instance $\mathbf{x}_{m,\bullet}$ delivered to task m.

$$\mathcal{BSP}_m(\mathcal{D}') = \sum_{m'=1}^{M} \int_{\Omega_{\mathcal{GU}_m}(\mathbf{x}_{m',\circ})} p(\mathbf{x}_{m',\circ}) \mathcal{GU}_m(\mathbf{x}_{m',\circ}|\mathcal{D}') d\mathbf{x}_{m',\circ} \tag{28}$$

It is important to note that inside the summation for each task m', the instances $\mathbf{x}_{m',\circ}$ are being sampled from the distribution $p(\mathbf{x}_{m',\circ})$ of incoming documents for that particular task m'. This stems from the fact that the distribution of incoming instances $P(\mathbf{x}_m)$ may be different for different tasks, as discussed earlier. Based on $\mathcal{BSP}_m$, for clarity and completeness, we provide the Equations of the corresponding utility gain $\mathcal{BUG}_m$ and AL scores $\mathcal{BAL}_m$.

$$\mathcal{BUG}_m(\mathbf{x}_{m,\bullet}) = \sum_{y \in \{0,1\}} P(y|\mathbf{x}_{m,\bullet}, D) \mathcal{BSP}_m(\mathcal{D} \cup (\mathbf{x}_{m,\bullet}, y)) - \mathcal{BSP}_m(\mathcal{D}) \tag{29}$$

$$\mathcal{BAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_B \mathcal{BUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \tag{30}$$

3.5.2.4 Analysis of LAL, GAL, and BAL

In Table 2, we summarize the major ideas from proposed AL approaches. Each approach focuses on improvements in different parameters of the model, thereby leading to different consequences. The LAL approach focuses on improving its own task. Such approach can be expected to perform well when the global model is already strong, and less susceptible to drastic change based on feedback on a new instance. The GAL approach restimates the global model, but in the process computes the future utility only based on the current task. This selfish strategy of modifying the global model without foreseeing the effect on other tasks in the system can be detrimental to the overall performance of the model, making this approach undesirable. The BAL approach updates the global model, but at the same time studies the effect of the update on the utility gain of other tasks in the system, thereby ensuring that the global model doesn't get biased towards a particular task. Consequently, we expect BAL to perform superior to other methods in the initial stages of AF when global model is weak.

Table 2: A comparative summary of AL approaches. The first column lists the AL score to decide delivery of instance $\mathbf{x}_{m,\bullet}$ to task m. The second column lists the parameters that will be (potentially) improved (for better future utility) if the system retrains on the feedback received on the instance delivered based on the corresponding AL score

Delivery criteria	Potentially improved parameters based on feedback
$\mathcal{LAL}_m$	$\mathbf{w}_m$
$\mathcal{GAL}_m$	$\mathbf{w}_m, \alpha, \mu, \Sigma$
$\mathcal{BAL}_m$	$\alpha, \mu, \Sigma, \mathbf{w}_{m'}, \forall m' \in \{1, \ldots, M\}$

3.5.2.5 Combined AL for MTAF

Based on the discussion in the previous section, each of the above methods can be combined to come up with a meta-AL system for AF that tries to satisfy multiple

objectives at different phases of learning. We use weighted linear combination of the individual utility gain scores to come up with the Meta AL score

$$\mathcal{MAL}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) = \mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_L \mathcal{LUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_G \mathcal{GUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) + \delta_B \mathcal{BUG}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) \quad (31)$$

The weight parameters δ_L, δ_G and δ_B should vary with the quality of the model, local as well as global. A stronger global model means lower δ_G. Similarly, δ_B should decrease so that the tasks can then focus on improving locally, thereby maintaining a higher value of δ_L. We test these hypotheses empirically in Section 3.5.3.4

3.5.3 Experiments

3.5.3.1 Datasets

We chose to use 2 datasets that are popular in the TREC filtering community, namely, RCV1 (84 categories, 810,000 documents) Lewis et al. [2004b] and 20 Newsgroups (20 categories, 18,846 documents) Joachims [1997]. For both datasets, we modeled each category as a task. For experiments on both datasets, we start AF with only one known relevant document per task. It should be noted that RCV1 is a multi-labeled dataset, meaning each document belongs to multiple classes/tasks, thereby indicating some level of overlap/relatedness among tasks (e.g. tasks sharing the same documents). 20 Newsgroups is not multi-labeled and each document belongs to only one class. However, it is known that the categories can be grouped into 6 groups based on subject matter, namely, comp.*, rec.*, sci.*, talk.*, misc.* and others. Thus, the *relatedness* of tasks is in the feature-space, as documents are not shared among tasks. For example, the feature *computer* is a strong indicator of the tasks in comp.* group. We chose this dataset to see if the MTAF approaches are able to discover these hidden groups of related categories, even if the documents are not shared among tasks.

3.5.3.2 Methods

Primarily, we are interested in comparing the four Active Learning approaches LAL, GAL, BAL and MAL in the MTAF setting to see which approach rapidly improves utility, and how they perform in the various phases of filtering. We also wish to compare these 4 approaches to the passive version that decides to deliver an instance without taking any future utility gain into account. (i.e. delivery criteria is $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) > 0$).

Zhang et al. [2003] devised a strategy called *exploration and exploitation* to perform Active Adaptive Filtering, and our AL framework is inspired by their work. The *exploitation* component is a passive AF strategy that delivers an item if $\mathcal{U}_m(\mathbf{x}_{m,\bullet}|\mathcal{D}) > 0$. The *exploration* component is an Active Learning component based on a metric called *Utility Divergence*, similar to the *Utility Gain* score developed in this work. Utility Divergence, however, doesn't compute the actual gain in utility of the system based on feedback on the instance. It instead computes the difference between the expected utility of a hypothetical *true* model Θ^* and the model based on expected feedback $\Theta|(\mathbf{x}_{m,\bullet}, y_{m,\bullet})$ and instances are scored based on their potential to reduce the gap between utility of the true model and utility of the current model. In our results, we will refer to these approaches as STAF-passive (only exploitation), STAF-active-UD (exploitation and exploration using Utility Divergence score).

3.5.3.3 A note on implementation

The various methods were implemented in MATLAB. For integrating over the posterior distribution of a variable (e.g. w_k^*), we used the Metropolis Hastings algorithm. Specifically, we draw several (S) samples from the posterior of the corresponding distribution of the variable, and averaged over the various outputs to obtain a probabilistic integral, similar to the approach described in Zhang et al. [2003]. For example, to implement Equation 6, we sample S samples from the posterior distribution

$P(\mathbf{w}|D)$ and the integration was implemented as:

$$\mathcal{U}(\mathbf{x}|\mathcal{D}) = \frac{1}{S}\sum_{s=1}^{S}\sum_{y\in\{0,1\}}\psi_y P(y|\mathbf{x},\mathbf{w}^{(s)}) \tag{32}$$

In this specific case, the posterior distribution was derived by using the Laplace approximation method Xue et al. [2007]. Other posterior distributions for the MTAF case were based on the derivations available in Xue et al. [2007]. For integrating over $P(\mathbf{x})$, the samples were drawn from the data stream observed so far, consisting of delivered and undelivered items, to get a non-parametric estimate of $P(\mathbf{x})$.

3.5.3.4 Experimental Results and Discussion

First we compare the MTAF approaches (MTAF-Passive, LAL, GAL, BAL, and MAL) to test the hypotheses we made in Section 3.5.2.4. Figure 3 shows the trends (in terms of T9U utility) on the RCV1 dataset. AL approaches are typically most effective in the initial phases of filtering and so the Figure 3 shows performance upto the filtering of the first 5000 instances on the RCV1 dataset. The results validate our hypothesis that the LAL approach performs quite poorly in the initial stages of filtering, when the global model itself is quite weak. It can also be observed that BAL approach, that tries to improve the overall utility of all the tasks in the system performs the best initially. This is because improved utility of all tasks reflects in an stronger global model. LAL improves at a faster pace once the learnt global model is stronger (due to learning from more feedback). The GAL approach, that tries to improve the global model, without foreseeing improvements (or degradation) in other task performs inferior to the LAL approach. We believe that the GAL approach sometimes makes wrong decisions about instance selection, by selecting those instances that will lead to improvement in one task, but mostly degrade other tasks (because the utility gain in tasks is not taken into account). We found that setting $\delta_G = 0$ in Equation 31, i.e. ignoring the $\mathcal{GUG}_m$ component corresponding to GAL, in the MAL approach led to better performance of

MAL. Consequently, the MAL approach, which combines the strengths of the LAL and the BAL approaches, outperforms all the other approaches.

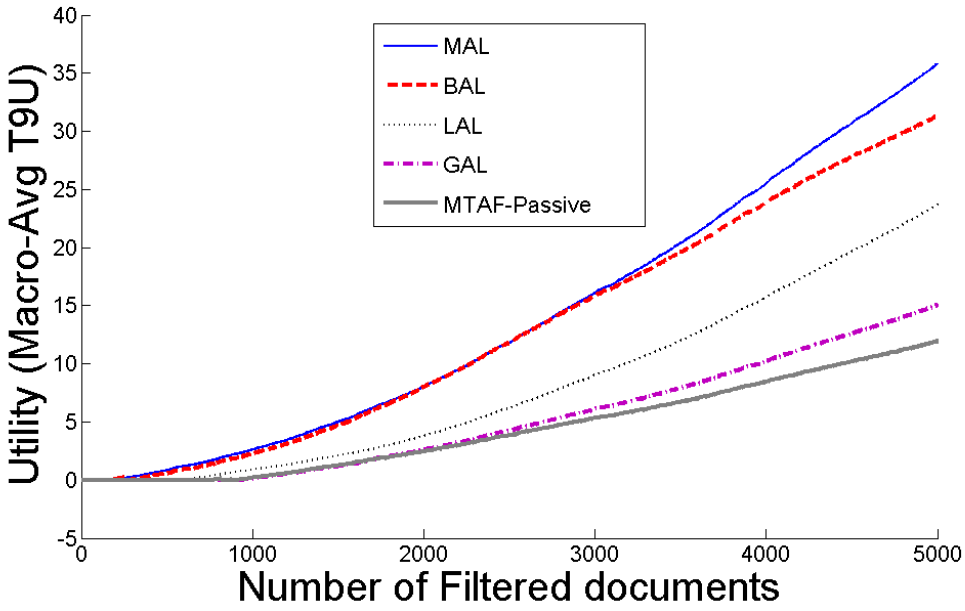


Figure 3: Comparison of the MTAF AL approaches on the RCV1 dataset (similar trends for 20 Newsgroups)

Next, we compare representative MTAF approaches (MTAF-passive and MTAF-MAL) to the STAF approaches to study the benefit of using a multi-tasking setup. We also compare the AL scoring functions: our *Utility Gain* (UG) criteria and the *Utility Divergence* (UD) criteria proposed in Zhang et al. [2003]. We call these variants of the STAF-active approach STAF-active-UG and STAF-active-UD respectively. In Figure 4, we observe that the MTAF approaches, active as well as passive, outperform their STAF counterparts. It can be observed that our best performing approach, MAL (T9U = 423), has a performance improvement of more than 20 percent over that of STAF-active-UD (T9U=348). A paired t-test shows strong statistically significant evidence (p-value = 0.0002) for the superiority of our approach over the current state-of-art STAF-active-UD. We also observe that the starting utility of the MTAF approaches is higher, due

to information sharing between tasks, to overcome the per-task data sparsity problem in the STAF approaches. Regarding the AL score, we observe that the performance of UG and UD is quite similar for STAF-active. Empirically, it seems that the goal of UD (reduce the gap between the utility of a hypothetical true model and the learnt model) is quite similar to the goal of UG (increase the future utility of the learnt model the most).

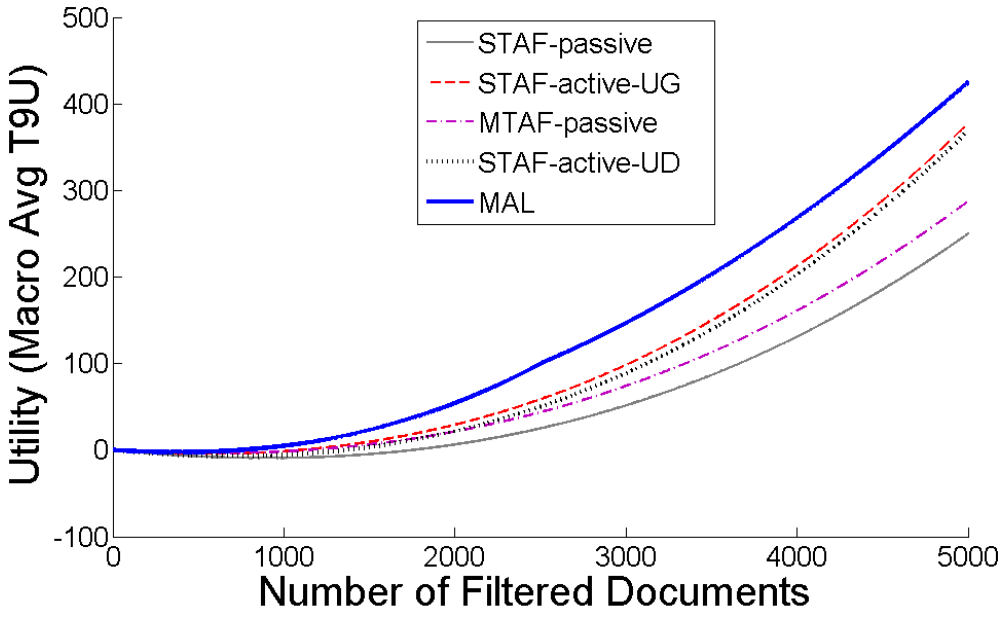


Figure 4: Comparison of MTAF and STAF approaches on the 20 Newsgroups dataset (similar trends for RCV1)

3.6 Summary

In this work, we have explored various Active Learning approaches to the Multi-Task Adaptive Filtering. To score the benefit of delivering an instance, we developed a new metric called *Utility Gain* that estimates the improvement in system performance (in terms of utility) if the system is re-trained on the feedback received for the

delivered instance. In the MTAF setting, we compare the effect of selfish (local) AL approaches (that focus on improvements in one task) to a benevolent AL approach (that evaluates the benefit of labeling an instance across many tasks). Our empirical analysis demonstrates the superior performance of the benevolent approach in the initial phase of filtering, when the global model of the MTAF system is weak, while a more rapid improvement in the performance of the selfish approaches in later stages, when the global model is strong (i.e. global model may not benefit much from AL). We also demonstrate that a combined approach, called meta-AL, that combines the strengths of local and benevolent AL approaches, is superior to the individual approaches.

There are several areas for exploration in the future. The sampling approaches presented in this work can be infeasible in a large-scale adaptive filtering system with millions of tasks. In such an environment, it is necessary to first segregate the tasks so that each of the smaller AF systems can handle their respective tasks. It is also necessary to derive analytic solutions to the sampling strategies described here to come up with closed-form/quicker expected future utility evaluation schemes. Another challenge is the problem of spam. In a practical MTAF system, how does the MTAF system protect its genuine users/tasks from other malicious users. In a benevolent AL approach the feedback from the malicious users is potentially harmful to the system, and consequently to the genuine users. So how does a Benevolent AL approach safeguard againt malicious use?

4

Transferable Active Learning for Heterogeneous Tasks

Does the MTAL strategy described in Chapter 3 perform equally well in the *heterogenous* setting, that is, when the tasks being learnt jointly are of diverse kinds [2]. Diverse tasks require diverse supervision and may involve different supervision costs. Then, the key question to answer is: How can we formulate the learning cost and benefit of AL across heterogeneous tasks and different types of training instances. These questions have not been studied in the MTAL setting and answering these questions is the focus of this Chapter.

We provide a concrete example to put the concept into perspective. Commercial sites such as Amazon or Netflix typically learn models to perform two kinds of tasks: Firstly a classification of items, for example, movies and books, into genres such as `Drama`, `Comedy`, and `Horror`. This classification is typically useful as a browsing aid for their users to surf their vast inventory of products. Secondly, such sites also offer item-recommendation services that learn user-preferences through previous purchase/rating

[2] In our work, heterogeneity arises in diverse kinds of outputs. For example classification, regression, ranking are diverse kinds of tasks

history. These heterogeneous tasks require different kinds of supervision: ratings for item-recommendation and category labels for classification. From a Multi-Task Learning perspective, are there benefits to jointly learn these seemingly different tasks? Intuitively, it seems that there could be significant benefits: Knowledge about the genre of a movie can help to recommend it to users who are primarily interested in that genre. Also, ratings for a movie can pin-point the interest group for that movie, which can then be used to predict the genre for the movie based on the genres that are typically liked by that interest group. From an AL perspective, how can we acquire supervision that improves either or both kinds of tasks? For example, can we acquire genre labels that improve the performance on the recommendation task? Alternatively, which items should be rated so that the performance of the genre-classification task is improved? Also, the cost of acquiring each kind of supervision is different. In such a cost-sensitive setting, how can AL choose to acquire the cheaper supervision with the goal of simultaneously improving both kinds of tasks?

An additional challenge arises out of scalability. The MTAL strategy described in Chapter 3 can be generically applied to most Multi-Task scenarios, as long as the impact of adding labeled instances to one task on other tasks can be estimated. Certain multi-task scenarios however feature lots of tasks making this estimation computationally intractable, particularly in the case of Benevolent Active Learning. In such scenarios, it is important to devise alternative strategies that can potentially benefit several tasks simultaneously, without explicitly having to compute the impact of each of the tasks. This is true of the aforementioned collaborative filtering scenario. Explicitly analyzing the impact of acquiring a genre label for an item on the recommendations to millions of users of a recommendation service is computationally intractable.

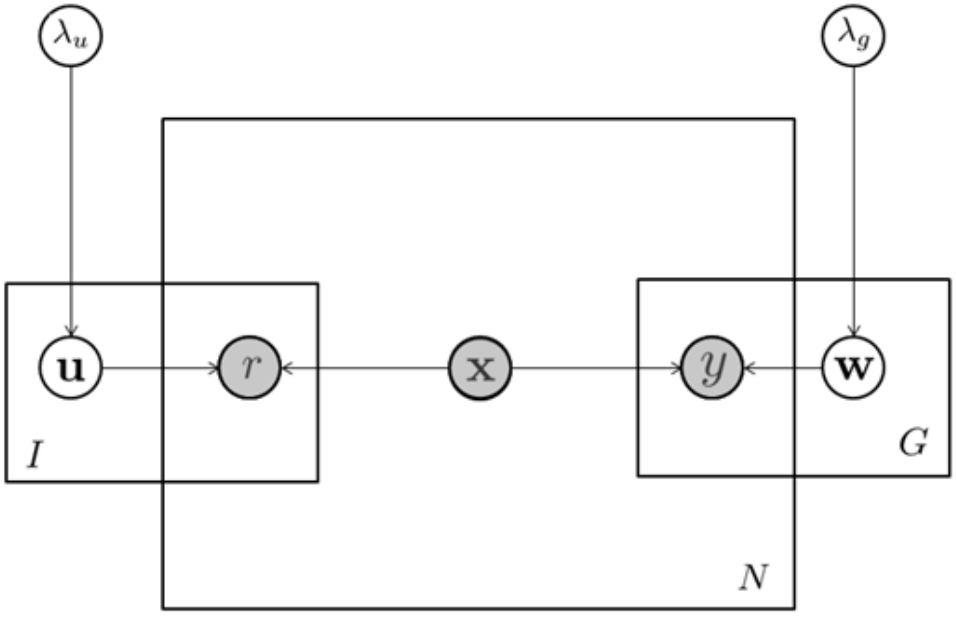


Figure 5: A simple model for learning heterogeneous tasks

4.1 Modelling heterogeneous tasks

Figure 5 shows a simple graphical model representing the joint learning for heterogeneous tasks. It should be noted that both the groups of tasks share a common input $\mathbf{x}$ for each of the items N. As a simple example, r could be regression outputs $r \in \mathbb{R}$, and $\mathbf{u}_i, \forall i \in I$ are the parameters of the regression tasks I. The y could be categories for classification tasks, for example $y \in \{0, 1\}$ and $\mathbf{w}$ would be the weights of the Logistic Regression classifiers. Classification and Regression are chosen here for simplicity, but the model is amenable to other kinds of tasks such as ranking by introducing appropriate complexity.

This simplified model doesn't allow information flow between the two kinds of tasks. The classification tasks can benefit from each through a shared prior λ_g but may not benefit from the regression tasks as the shared input is always known. Similarly, the regression tasks can benefit from each other through a shared prior λ_u, but may not

benefit from classification tasks.

This suggests a simple modification of the model using a topic-model approach depicted in Figure 6. In this model, inspired by topic-modeling, introduces a set of latent topics between the observables. These latent factors θ are drawn per-item and for each item the factors form a probability simplex $\sum \theta_n = 1$. Consequently, any observable for an item, r, y, or $\mathbf{x}$, might impact the latent factors and lead to improvements in regression and classification tasks by enabling information flow between these two kinds of tasks.

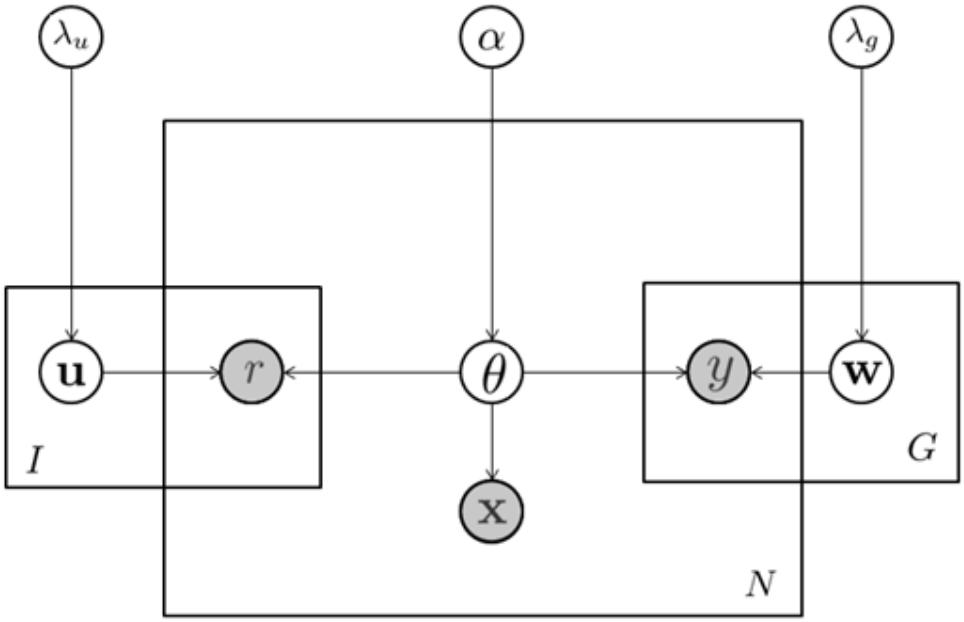


Figure 6: A generic joint model for learning heterogeneous tasks

We provide more concrete instantiation of this model in the context of jointly learning genre-classification and Collaborative Filtering tasks in later in Chapter 4.3.

4.2 Transferable active learning

Let $\mathbb{T} = \{t_1, \ldots, t_T\}$ denote the set of tasks. Let $\mathbb{S}_t$ denote the potential candidates for acquiring supervision on the task t. Let ζ_s denote the cost of acquiring supervision s. Finally, let the perceived benefit of acquiring supervision s be $\Delta_t(s)$ for a task t. The benefit could be estimated in terms of improvement in accuracy (for classification) or, in terms of reduction in the mean-absolute error (for recommendations). TAL can then choose to acquire supervision s^* using the criteria below:

$$s^* = \underset{s \in \bigcup_{t \in \mathbb{T}} \mathbb{S}_t}{\arg\max} \left\langle \frac{\sum_{t \in \mathbb{T}} \Delta_t(s)}{\zeta_s} \right\rangle_s \tag{33}$$

In this equation, the angled brackets $\langle \rangle_s$ denote the expectation over the possible values of s, since the true supervision is unknown. Intuitively, Equation 33 compares the supervision candidates with respect to their benefit/utility per unit-cost and prefers to acquire supervision that is most cost-effective. A straightforward implementation of this strategy will require estimating the posterior model for each possible value of each supervision candidate to estimate the benefit on each of the tasks collectively. Such a naive approach will be computationally expensive and will pose additional challenges. One such challenge is the estimation of the true per-task benefit, which can only be estimated through cross-validation like approaches, adding to further computational woes. Instead, we could exploit the commonality among tasks to derive a *surrogate* task. The surrogate task may be chosen based on two criteria: Firstly, improvements in the surrogate task should be indicative of improvements in the original tasks. Secondly, improvements in the surrogate task should be easily measurable. Subject to these conditions, surrogate tasks can be expected to achieve the TAL goal in an efficient manner.

$$s^* = \underset{s \in \bigcup_{t \in \mathbb{T}} \mathbb{S}_t}{\arg\max} \left\langle \frac{\Delta_{\text{surrogate}}(s)}{\zeta_s} \right\rangle_s \tag{34}$$

4.3 Case Study: Jointly learning Genre Classification and Collaborative Filtering

In this work, we analyze Transferable Active Learning, in the context of simultaneously learning to perform genre classification and collaborative filtering. To leverage cross-task information sharing, we develop a novel joint model for learning genre-classification (GC) and collaborative filtering (CF) tasks. Specifically, we enhance the Collaborative Topic Regression (CTR) Wang and Blei [2011] model with a component for simultaneously learning GC, based on regularized Logistic Regression classifiers. Equipped with this model, we present improved performance of the *genre-enhanced Collaborative Filtering* and *ratings-enhanced genre-classification* over their conventional vanilla counterparts. Later, we showcase *Transferable Active Learning* (TAL), a novel AL strategy for minimizing supervision costs in the case of heterogeneous tasks. The goal of Transferable Active Learning (TAL) is to seek information about one kind of task, that is beneficial for learning another kind of task. In the context of our chosen application, i.e. genre-enhanced Collaborative Filtering, TAL could choose to obtain genre-information, with the goal of improving the CF performance. Alternatively, TAL could also choose to acquire ratings for improving performance of the genre-classification task. TAL could be an effective strategy to minimize supervision costs for a particular kind of task if the supervision for other task could be acquired more readily. Finally, the choice to acquire a particular kind of supervision could be based on the inherent cost of obtaining genre-labels versus the cost of obtaining ratings from a user, with the goal of improving the overall model (both kinds of tasks). This is in contrast to conventional AL approaches that seek supervision for the task that is to be improved. We avoid

the explicit and exhaustive computation of benefit of acquiring supervision on one task over other tasks Harpale and Yang [2010] by instead measuring the estimated impact of the supervision on a surrogate task that can be measured more efficiently. We evaluate our ideas on the challenge of Active Learning for cold-start recommendation and classification of new items for which no ratings and genre information is available apriori. This is in contrast to existing work in Active Learning for CF which has focused on the user-focused perspective: acquiring ratings with the goal of improving recommendations for a new user. As the surrogate task, we seek to acquire supervision to minimize the entropy of the topical distribution of the item in the joint model.

Recent work in Collaborative Filtering has demonstrated significant improvements in the performance of item recommendation using the item-genre information as an additional feature in the model Yang et al. [2008]. However, such models typically assume that the genre is always available as a truth value. In reality though, genre prediction is a task in itself. To minimize the overall cost of learning such diverse tasks, it is crucial to ensure that supervision obtained on one kind of task (classification) is also useful for improving the other kind of task (item-recommendation). It is particularly important for this example as supervision might be cheaper for the classification task using in-house domain experts while supervision might be expensive or harder to get for the item-recommendation task as that depends on the users who might get annoyed with active queries. In this work, we develop a novel joint learning model for genre-classification and item recommendation and employ a novel *Transferable* AL strategy that attempts to minimize supervision costs by soliciting classification/rating for items with the goal of improving both the genre-classification and item-recommendation tasks.

The rest of the chapter is organized as follows. In Section 4.3.1, we describe our novel joint-learning approach and in Section 4.3.3, we present empirical evidence of the success of the joint model over learning each task in isolation. In Section 4.3.2, we describe our new Transferable AL strategy for leveraging the joint model to acquire alternate

supervision for improving a kind of task. We also present the empirical comparison of our approach to conventional active and passive learning strategies to showcase the savings in supervision costs made possible by our approach. Finally, in Section 4.4, we conclude with a discussion on open challenges for further exploration.

4.3.1 Our Approach: Genre Classification and Collaborative Topic Regression: G+CTR

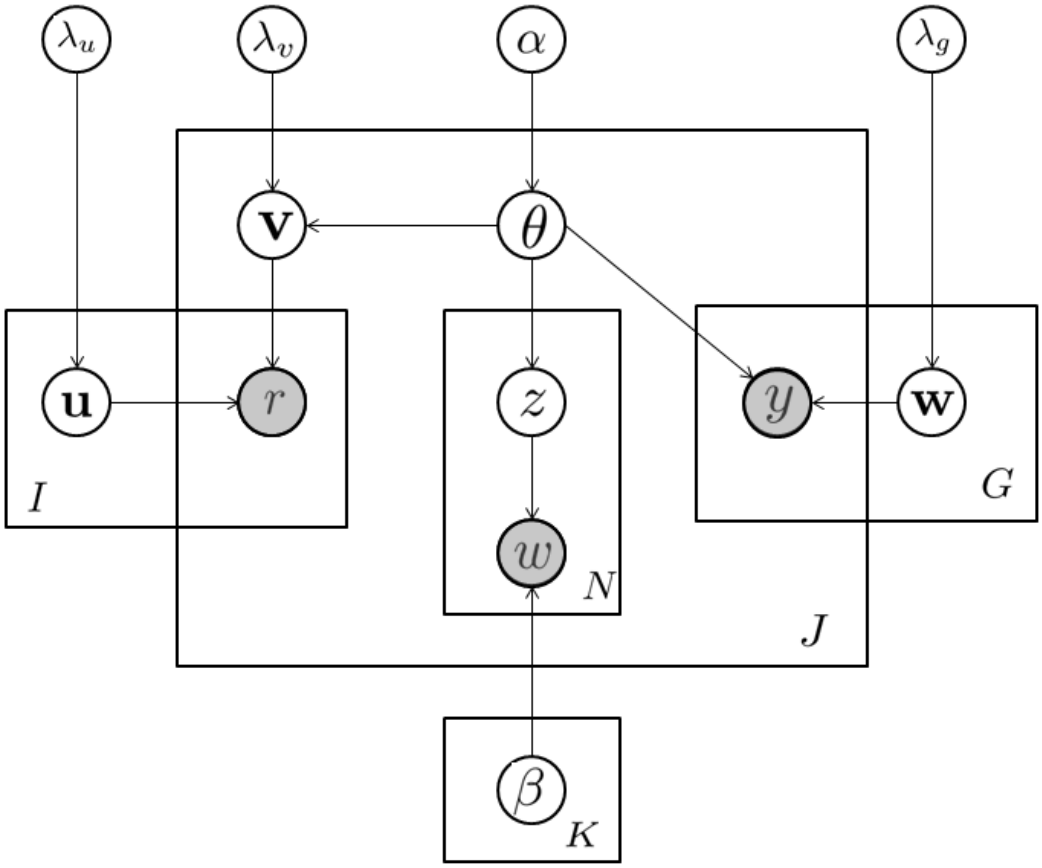


Figure 7: G+CTR model

Our joint learning model builds upon the successful Collaborative Topic Regression (CTR) Wang and Blei [2011] model by supplementing it with additional structure for the genre-classification task. CTR is a topic-modelling approach that simultaneously

utilizes rating-based and content-based information of an item, with the ultimate goal of improving CF performance. The rating-based component is based on the Probabilistic Matrix Factorization Hu et al. [2008]; Salakhutdinov and Mnih [2008b] approaches that have been demonstrated to be extremely successful for CF. The content-based component is based on the popular Latent Dirichlet Allocation (LDA) Blei et al. [2003] approach. For our enhancements, we have included regularized Logistic Regression classifiers for modeling item-genres. We call our model *Genre and Collaborative Topic Regression* (G+CTR). The model is depicted in Figure 7.

The generative process of the G+CTR model is as follows ($\mathbb{I}_K$ is a K-dimensional identity matrix):

1. For each user $i \in \{1, \ldots, I\}$, draw a user interest vector $\mathbf{u}_i \sim \mathcal{N}(0, \lambda_u^{-1}\mathbb{I}_K)$
2. For each genre $g \in \{1, \ldots, G\}$, draw a weight vector $\mathbf{w}_g \sim \mathcal{N}(0, \lambda_g^{-1}\mathbb{I}_K)$
3. For each item $j \in \{1, \ldots, J\}$
    a) Draw topic proportions $\theta_j \sim \text{Dirichlet}(\alpha)$
    b) Draw item latent offset $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1}\mathbb{I}_K)$
    c) Set the item latent vector to $\mathbf{v}_j = \theta_j + \epsilon_j$
    d) For each word w_{jn}, $n \in \{1, \ldots, N\}$
        i. Draw topic assignment $z_{jn} \sim \text{Mult}(\theta_j)$
        ii. Draw word $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$
4. For each user-item pair (i, j), draw the rating $r_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, c_{ij}^{-1})$, where c_{ij} is the precision parameter for the rating r_{ij}.
5. For each genre-item pair (g, j), set the genres as: $\mathbf{y}_{jg} = \sigma(\mathbf{w}_g \cdot \theta_j) \geq 0.5$, where $\sigma(\mathbf{w}_g \cdot \theta_j) = (1 + \exp(-\mathbf{w}_g \cdot \theta_j))^{-1}$

In this generative process, steps 1 and 4 depict the generative process for a Probabilistic Matrix Factorization model Hu et al. [2008]; Salakhutdinov and Mnih [2008b]. Steps 2 and 5 depict the generative process for a Bayesian Logistic Regression Jaakkola and

Jordan [1996] classifier. Finally, step 3 describes the generative process of a Latent Dirichlet Allocation (LDA) Blei et al. [2003] model.

Learning The inference follows along the same lines as for CTR, with appropriate modifications to take the genre-based supervision into account. Specifically, we maximize the complete log-likelihood of the joint model to learn the parameters (constants have been ignored below, and $\alpha = 1$)

$$\begin{aligned}
\mathcal{L} = & \sum_j \sum_n \log \left(\sum_k \theta_{jk} \beta_{k,w_{jn}} \right) \\
& - \sum_{i,j} \frac{c_{ij}}{2} (r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 \\
& + \sum_{j,g} \log \left(\sigma(\mathbf{w}_g \theta_j)^{y_{jg}} (1 - \sigma(\mathbf{w}_g \theta_j)^{(1-y_{jg})}) \right) \\
& - \frac{\lambda_u}{2} \sum_i \mathbf{u}_i^T \mathbf{u}_i \\
& - \frac{\lambda_v}{2} \sum_j (\mathbf{v}_j - \theta_j)^T (\mathbf{v}_j - \theta) \\
& - \frac{\lambda_g}{2} \sum_g \mathbf{w}_g^T \mathbf{w}_g
\end{aligned} \tag{35}$$

We adopt the EM-like approach of Wang and Blei [2011]. The derivation for $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_I\}$, $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$, and $\beta_{1:K}$ are unchanged from their CTR and LDA counterparts as they are independent of the classification component given θ and the reader is referred to Blei et al. [2003]; Wang and Blei [2011] for details.

Knowing θ_j, we can optimize for $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_G\}$ of the classification component, independent of U, V. Specifically, we provide a Stochastic Gradient Descent (SGD) update rule for learning $\mathbf{w}_{gk}$, with learning rate η, which can be easily derived by setting the derivative of the elements containing $\mathbf{w}_{gk}$ in the log-likelihood to zero.

$$\mathbf{w}_{gk} \leftarrow \mathbf{w}_{gk} + \eta\left(\theta_{jk}(y_{jg} - \sigma(\mathbf{w}_g\theta_j)) - \lambda_g\mathbf{w}_{gk}\right)$$

Given U, V, and W, it is not straightforward to learn θ, due to the complex interaction with V and W. To overcome this situation, we first separate the terms containing θ_j from the likelihood in Equation 35 and apply the Alternating Direction Method of Multipliers (ADMM) Boyd et al. [2011] approach to solve the resulting optimization problem. Specifically, we introduce a vector of dummy variables γ and an additional constraint $\theta_j = \gamma$ to construct an optimization problem $\mathcal{L}(\theta_j, \gamma)$ with split functions, as shown below

$$\begin{aligned}
&\text{minimize} \quad \mathcal{L}(\theta_j) + \mathcal{L}(\gamma) \\
&\text{subject to} \quad \theta_j = \gamma \\
&\mathcal{L}(\theta_j) = \frac{\lambda_v}{2}(v_j - \theta_j)^T(v_j - \theta_j) - \sum_n \log\left(\sum_k \theta_{jk}\beta_{k,w_{jn}}\right) \\
&\mathcal{L}(\gamma) = -\sum_g \log\left(\sigma(\mathbf{w}_g\gamma)^{y_{jg}}(1 - \sigma(\mathbf{w}_g\gamma)^{(1-y_{jg})})\right)
\end{aligned} \tag{36}$$

The objective function is now separable in θ_j and γ, and can be minimized by first forming an augmented Lagrangian Bertsekas [1996] $\mathcal{L}_\rho(\theta_j, \gamma, \xi)$, where ξ is a vector of Lagrange multipliers for the equality constraints, and then applying the following iterative update rules (to be run till convergence).

$$\theta_j^{(t+1)} := \arg\min_{\theta_j} \mathcal{L}(\theta_j, \gamma^{(t)}, \xi^{(t)}) \tag{37}$$

$$\gamma^{(t+1)} := \arg\min_{\gamma} \mathcal{L}(\theta_j^{(t)}, \gamma, \xi^{(t)}) \tag{38}$$

$$\xi^{(t+1)} := \xi^{(t)} + \rho(\theta_j^{(t)} - \gamma^{(t)}) \tag{39}$$

Equation 37 is the optimization problem from CTR (for solving θ_j) and Equation 38 is the usual Logistic Regression problem (in terms of γ) and both can be solved independently using the corresponding previously known techniques.

Prediction We have two predictive tasks: Item-recommendation (CF) and Genre-classification. These two tasks can be performed after inferring the appropriate parameters as:

$$r_{ij} = \mathbf{u}_i^T \mathbf{v}_j \tag{40}$$

$$y_{gj} = \sigma(\mathbf{w}_g, \theta_j) \geq 0.5 \tag{41}$$

It can be observed that G+CTR opens up several new possibilities. Ratings can now be predicted for out-of-matrix items, if the content or genre can be obtained for such items. Similarly, genre can be predicted for an item, even in the absence of its content, if ratings are available on that item to learn its topical representation. Such situations commonly appear in the multimedia classification setting where content-based classification is infeasible with current state-of-art.

4.3.2 Our Approach: Transferable Active Learning for G+CTR

For the G+CTR model, TAL is expected to be useful in the cold-start scenario: a new item arrives, and is devoid of any rating and genre information. For this scenario, the structure of the G+CTR model suggests a potential surrogate task: topic discovery for an item. Inferring the best θ for an item is a task in itself, albeit hidden. In the absence of any ratings, content or genre information for an item, the item has an equal probability of belonging to each of the topics. Consequently, the predictive tasks on that item, classification and recommendation, are ineffective for lack of topic attribution. Every bit of new information acquired about an item, including content, ratings or genres, makes it possible to assign it to more relevant topics and disassociate it from the irrelevant ones.

This latter scenario has a lesser *topical entropy*, enabling more informed predictions for classification and recommendation. Thus, topical entropy minimization can be used as the chosen surrogate measure of effectiveness for a topic-modelling based approach to jointly learn diverse tasks. Mathematically, if $\mathcal{H}(\theta_j|s)$ denotes the entropy [3] of the topical distribution after acquiring supervision s, then we can re-write the formulation in Equation 42 to choose a supervision for an item j can be described as follows:

$$s^* = \underset{s \in \bigcup_{t \in \mathbb{T}} \mathbb{S}_t}{\arg\min} \left\langle \frac{\mathcal{H}(\theta_j|s)}{\zeta_s} \right\rangle_s \tag{42}$$

For the G+CTR model, $\mathbb{T}$ consists of two kinds of tasks, classification and recommendation. The candidates for supervision (s) include all users from whom we do not have any ratings for the item, as well as all the genres for whom we do not have membership information for that item.

It should also be observed that minimizing the topical entropy of an item is akin to minimizing the uncertainty of the surrogate task on that item. Uncertainty minimization strategies are popular in the AL literature. TAL, instead of minimizing the uncertainty over the final tasks, minimizes the uncertainty of the surrogate.

4.3.3 Experiments

4.3.3.1 Dataset

We use the publicly available MovieLens [4] dataset. The dataset consists of 2114 users, 10110 movies (items) and 20 genres. Total user-item-rating triples are 855599 suggesting a very sparse rating pattern (only 4 percent of all possible user-item pairs have been rated). On an average, each movie has been rated by approximately 85 users, with the maximum number of users that have rated a movie being 1670 (for the movie: `The Matrix`, a popular movie) and many movies with only a single rater. Each user has

[3] $\mathcal{H}(x) = -\sum p_x \log p_x$
[4] http://www.grouplens.org/node/462

provided ratings for at least 20 movies, with an average of 404 movies per user. On an average each movie belongs to 2 genres, with maximum of 8 genres and minimum 1.

4.3.3.2 Genre-enhanced Collaborative Filtering

For the first experiment, in Figure 8, we compare the performance on the CF task in the presence/absence of genre-information for each item. For the in-matrix setting, the results have been averaged over 5 randomized 80/20 percent training/testing split (over ratings) for each item to evaluate the performance of the CF task on that item. We present all the known genre-based information for the *with-Genre* setting and hide all of it in *without-Genre* setting. For the out-of-matrix (cold-start) setting, for each of the splits from the in-matrix setting, we randomly chose 20 percent of the items and hid all of their ratings reserved for training, thereby simulating new incoming items that no user has rated. For each item, the evaluation set remains the same as that in the in-matrix setting. Our evaluation is item-focused: the Mean Absolute Error (MAE) reported here is the average of the per-item MAE [5] and is defined in Equation 43, where $\mathbb{J}$ denotes the test set of items and $|\mathbb{J}|$ the cardinality of that set. $\mathbb{U}_j$ denotes the set of users who have rated an item j, and $r, \hat{r}$ denote the true and predicted rating for a given user-item pair.

$$\texttt{MAE} = \frac{1}{|\mathbb{J}|} \sum_{j \in \mathbb{J}} \left(\frac{1}{|\mathbb{U}_j|} \sum_{u \in \mathbb{U}_j} |r_{j,u} - \hat{r}_{j,u}| \right) \tag{43}$$

We can clearly observe significant improvements in the performance of CF when the model is supplemented with the genre-information. The performance improvement is particularly more in the case of out-of-matrix predictions (for new items that do not have any ratings) due to the additional guidance offered by genres to arrive at the topical distribution of items. This replicates the success of the previous studies that

[5] Other empirical evaluations have reported user-focused MAE, i.e. averaged over the MAE of each user, and are not directly comparable

have looked at the use of genres for improving CF task Yang et al. [2008], but with a more interpretable approach in the form of topic-modelling.

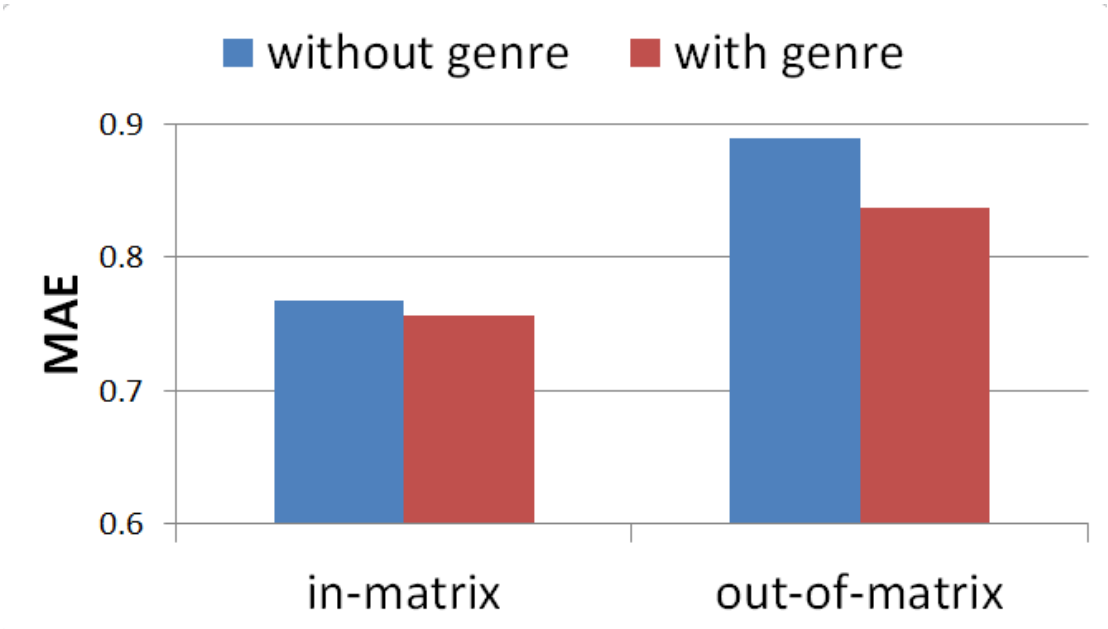


Figure 8: CF performance comparison

4.3.3.3 Rating-enhanced Genre Classification

As an alternate setting, we estimate the benefit of knowing rating information for the genre-classification task. Again we use a 80/20 percent training/testing split (over items) to evaluate the performance on the classification task. We present all known ratings about an item in the *with-CF* setting, and hide it all in *without-CF* setting. Thus, in the case of *without-CF*, the classification is only guided by the item's content [6], a standard situation in text classification. Maintaining the item-focused nature of this work, the Macro-Average F1 scores have been averaged over items by averaging over per-item F1 scores, as described in Equation 44, where $\mathtt{F}_j$ denotes the F1 score over item j.

[6] In our experiments, content means the plot description of the movie

$$\mathtt{F} = \frac{1}{|\mathbb{J}|} \sum_{j \in \mathbb{J}} \mathtt{F}_j \tag{44}$$

Again, in Figure 9, we observe significant improvement in genre-classification performance with the availability of rating information.

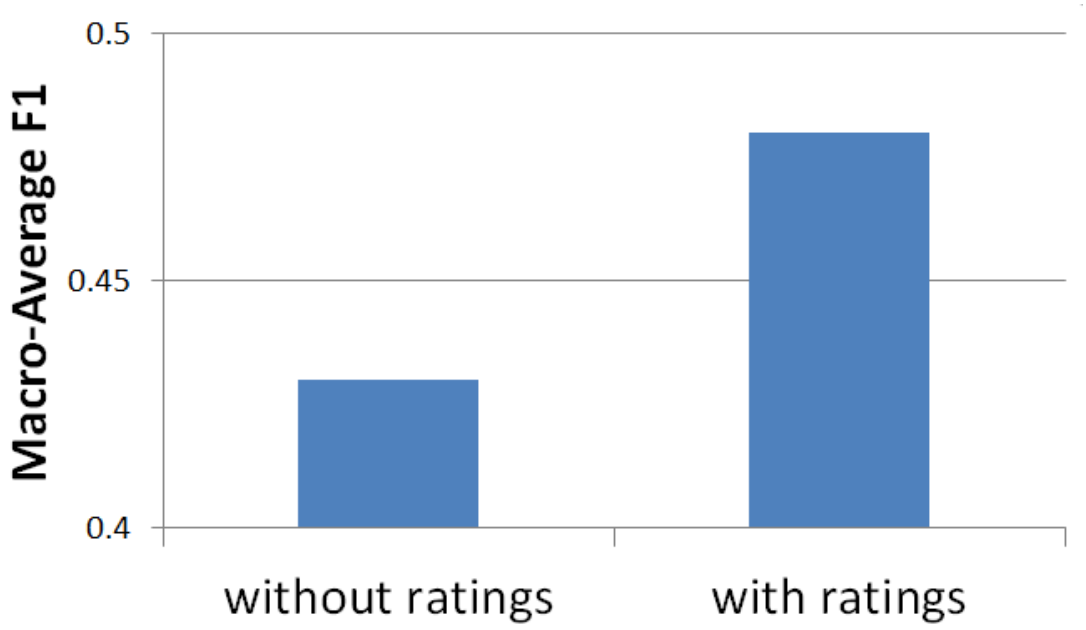


Figure 9: Performance on the Genre-classification task

4.3.3.4 Experiments:Transferable Active Learning

For these experiments, we first segregate the items into old and new items. All rating and genre information is known about the old items, and new items are the cold-start candidates. For the cold-start items, we segregate the known ratings and genres into two sets, active and test set, using a 80/20 percent split. The *active set* forms the candidates available for supervision. For real applications, this set is unknown apriori, but approaches such as ProActive Learning Donmez and Carbonell [2008a] make it possible to discover this set. Exploration of this added complexity in the context of

TAL is left for future work and will not be examined here. In every active iteration, the TAL criteria in Equation 42 is applied to acquire the most cost-effective supervision from the active set. Upon including the acquired supervision into the training set of the item, the required model parameters are re-estimated to perform further predictions, for both the recommendation and classification task. After each iteration, the predictions are evaluated using the held-out test set. It should be noted that the TAL criteria requires the knowledge about the cost of acquiring a particular kind of supervision. In the absence of any real data about this factor, we have arbitrarily chosen the following cost structure: Cost of acquiring a rating on an item is the same for all users and cost of a membership query on an item-genre pair is the same for all genres. Because users typically provide ratings for free, but genres are manually curated, we assume that the cost of acquiring a rating is 1/5th the cost of acquiring a genre supervision. We used 1 units cost for rating acquisition and 5 units cost for genre acquisition. For comparison with the TAL criterion, we use the following baselines

1. **Passive**: The passive baseline which randomly selects any of the candidates from the active set to acquire supervision
2. **Passive: Ratings only**: The passive baseline which randomly selects to acquire only a rating
3. **Passive: Genre only**: The passive baseline which randomly selects to acquire a genre information only

Note that item-focused Active Learning is a new area and consequently, we are unable to compare TAL to previously published baselines other than variants of our own method. The variants we compare are cost-agnostic preferential versions of TAL: **TAL: Ratings only** selects the most effective ratings and **TAL: Genre only** selects the most effective genres according to the TAL score.

Figure 10 shows a performance comparison of the approaches on the recommendation task, in terms of the number of queries performed versus the performance on the task. Figure 11 shows an alternate comparison that compares the expenditure (cost of all acquisitions made) to the performance on the task. Clearly, the performance of TAL baselines is much better than the passive baselines. In terms of the expenditure, the cost-sensitive TAL approach is much more parsimonious in achieving better performance as compared to its cost-agnostic counterparts. Figure 12 and Figure 13 show similar behavior on the classification task.

Several interesting observations can be made in these results. For a given task, in the absence of a cost function, the most suitable supervision to acquire is task-dependent. Thus, it can be seen in Figure 10 that for the recommendation task, ratings matter more than genre information. That means, the performances of *Passive: ratings only* and *tal: ratings only* are better than their *genre-only* counterparts. Conversely, for the classification task, in Figure 12, the most suitable supervision to acquire is genres, not ratings.

However, in the presence of variable costs, the TAL strategy is more cost-effective is choosing the right kind of supervision to acquire, as shown in Figures 11 and Figure 13. It can be observed that the seemingly better classification performance of *genres-only* over *ratings-only* approaches comes at a higher cost. As the results indicate, the TAL approach overcomes this limitation by selectively acquiring the most cost-effective supervision.

4.4 Summary

We presented a novel scenario for Active Learning: Acquiring supervision over one kind of task, with the goal of improving performance over another kind of task. To address this challenge, we presented the Transferable Active Learning strategy as a

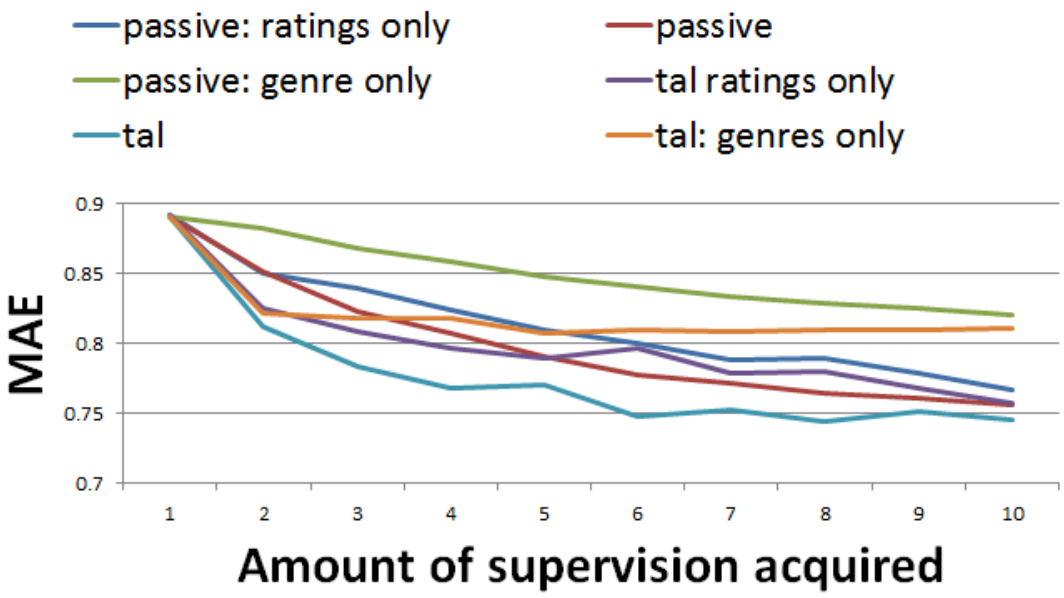


Figure 10: TAL: The effect of supervision acquisition on recommendation performance

mechanism of gauging cross-task benefit of acquiring supervision. We presented our ideas in the context of the real-world application of genre-based Collaborative Filtering. Our empirical analysis demonstrated the significant improvements made possible by learning the two kinds of tasks, Collaborative Filtering and Genre-Classification, jointly. Using the novel TAL strategy, we demonstrated significant reductions in the amount of the cross-task supervision required. Heterogeneous tasks are ubiquitous in several Machine Learning applications and further exploration is required to identify the best model and the best Transferable Active Learning strategy for improving such tasks with minimal supervision. On the theoretical front, it is crucial to identify a surrogate task that is a good indicator of the performance of the original tasks.

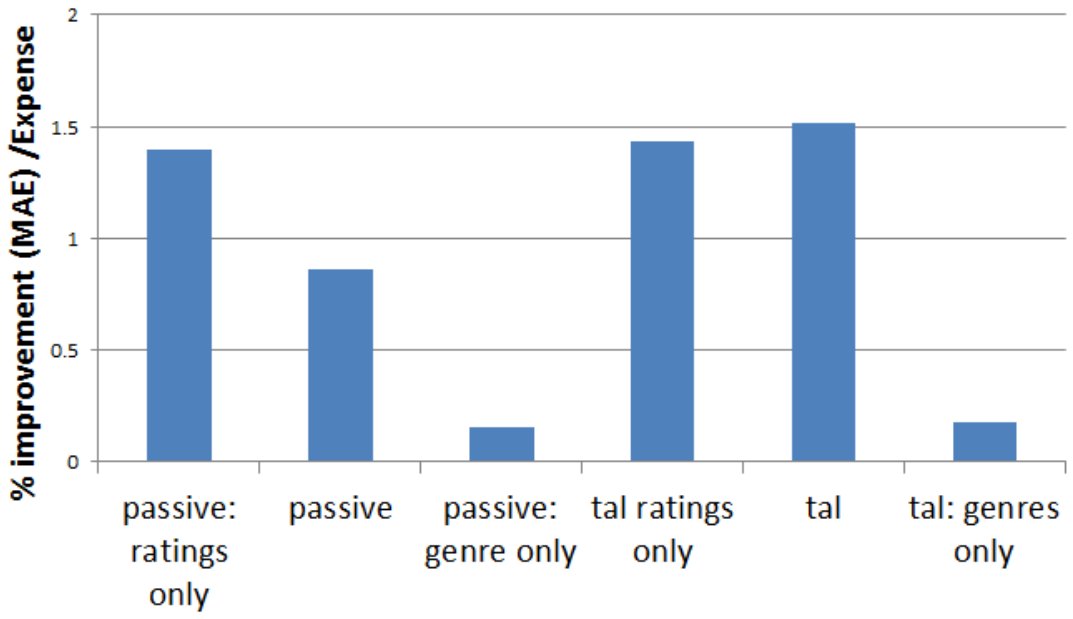


Figure 11: TAL: The effect of expenditure on acquiring supervision on recommendation performance. The numbers have been arrived at by dividing the percentage improvement in MAE-score by the corresponding total expenditure in acquiring supervision, using a particular strategy.

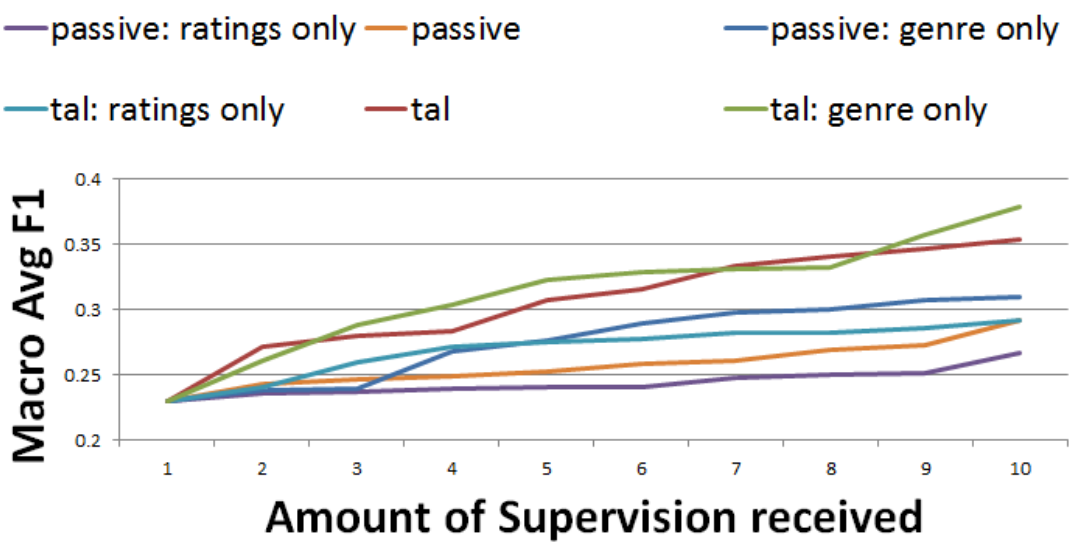


Figure 12: TAL: The effect of supervision acquisition on classification performance

5

Multi Task Active Learning for Hierarchical Classification

Hierarchical Classification (HC) Cai and Hofmann [2004]; Chen and Dumais [2000]; Liu et al. [2005b] is an important application of machine learning. HC deals with

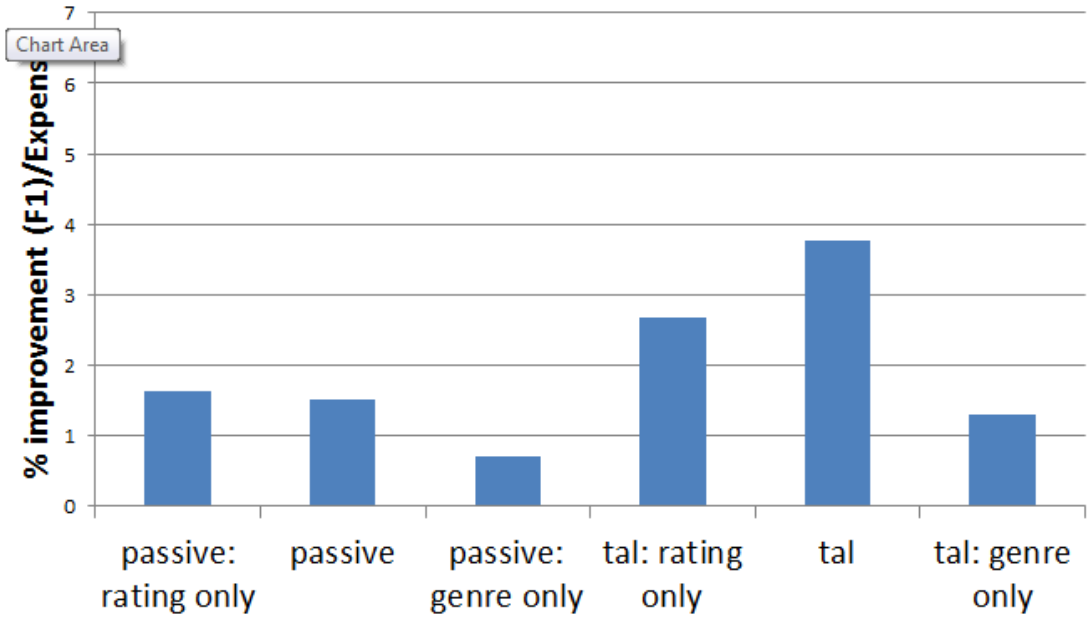


Figure 13: TAL: The effect of expenditure on acquiring supervision on classification performance. The numbers have been arrived at by dividing the percentage improvement in F1-score by the corresponding total expenditure in acquiring supervision, using a particular strategy.

classifying a given instance into one, or several, categories belonging to a pre-defined taxonomy or ontology. It is widely used for diverse domains such as text Sun and Lim [2001], proteins/genes/enzymes Zimek et al. [2010], advertisements and user-interests, chemical databases Rose and Eastman [1997], either as a final goal to facilitate information management, or as an intermediate task for important applications such as information retrieval Dumais and Chen [2000], disorder prediction, click-through prediction, personalization and collaborative filtering.

A popular approach to HC, especially in the large-scale scenario, is the hierarchical divide-and-conquer (HDC) Cai and Hofmann [2004]; Chen and Dumais [2000]; Liu et al. [2005b] strategy that associates a binary classifier with each node of the hierarchy. The classifier at each node acts as a filter which allows the positive instances to pass through and stems the negative ones. The descendant nodes are only concerned with

discriminating among the instances deemed positive by the parent node. During training, each node is trained in a one-vs-rest manner against its siblings, using only the positive instances of its parent. This simplicity of the approach makes it deployable on a large-scale using conventional state-of-the-art binary classifiers such as Support Vector Machines (SVM) Liu et al. [2005b], Naive Bayes (NB) Punera and Ghosh [2008] or Logistic Regression (LR) Zimek et al. [2010] classifiers as the building blocks at each node. This divide-and-conquer approach has been shown to significantly outperform direct one-vs-rest flat classification at the leaf level (which ignores the hierarchy), both in terms of classification performance as well as computational complexity Liu et al. [2005b], since each classifier gets trained on a fraction of the overall training instances as one goes down the hierarchy. A particular failure of the HDC (as well as the flat classification) approach, is its poor performance on nodes with scarce positive training instances Liu et al. [2005b]. Such scenarios are unavoidable at the lower-level nodes of the hierarchy and it is important to address this challenge.

Firstly, our approach implements a Multi-Task HDC (MT-HDC) strategy to leverage the learnt parameters of a parent node to regularize its children. Intuitively, our strategy is based on the fact that the parent node has been learnt on more training data, thereby leading to a more robust parent model. In comparison, a child node might tend to overfit the limited training data it has access to, and hence might benefit from the regularized guidance that it receives from the parent node. We derive a Stochastic Gradient Descent (SGD) based solution for training MT-HDC to make it scalable for large-scale hierarchiies and also enabling online-learning in scenarios where it is training instances arrive in a stream.

Secondly, we supplement the model with a novel Active Learning (AL) strategy that selectively solicits labels for instances that are deemed useful for learning a better HC model. Conventional AL approaches typically choose instances in a single-task manner, i.e., to improve the performance of a single classifier/node in isolation. In a hierarchical

setting, it is particularly important to select instances for labeling that lead to more cascading improvements, both in terms of the breadth and the depth of the hierarchy. Our AL approach selects instances taking such widespread improvements into account. We design this approach as an online scoring function that will select instances for labeling on the fly, as instances are presented to the classifier. Instead of computing the overall score of an instance to obtain its terminal label directly, the algorithm selectively performs membership queries at each node in the hierarchy to limit computational cost.

We first formally define the problem setting. The HC task consists of mapping instances $\mathbf{x} \in \mathbb{R}^D$ to their corresponding labels $\mathbf{y} \in \{0,1\}^Y$, where Y is the total number of labeling tasks. The labels, $\mathbf{y} = (y_1, \ldots, y_Y)$, are hierarchically structured in a way such that $y_c = 1$ implies $y_p = 1$ if y_p is the parent of the node y_c in the hierarchy, i.e. if $c \in \Lambda_p$, where Λ_p denotes the set of direct children of the node p. The inverse relationship is denoted by $p = \pi_c$, i.e. p is the parent node of c. Training data arrives in pairs of labeled instances of the form $(\mathbf{x}, \mathbf{y})$.

The popular HC strategy, for performance and computational complexity, is the Hierarchical Divide-and-Conquer strategy (HDC) that associates a classifier with each node in the hierarchy. For large-scale deployments, it is crucial to use simple but effective classifiers at each of the nodes; for example a binary Logistic Regression (LR) classifier should suffice. Training involves learning the LR weight vector $\mathbf{w} \in \mathbb{R}^D$ per node, usually by Maximum Likelihood Estimation (MLE). For classification, instances arriving a node i are passed on to their children only if they can overcome the probabilistic membership threshold, i.e. $P(y_i = 1|\mathbf{w}_i, \mathbf{x}) > t$. For LR, $P(y = 1|\mathbf{w}, \mathbf{x}) = (1 + \exp(-\mathbf{w} \cdot \mathbf{x}))^{-1}$, often denoted as $\sigma(\mathbf{w} \cdot \mathbf{x})$ as the *sigmoid* function. The threshold t is tuned for the desired evaluation metric such as accuracy, precision, recall or the composite F-score. Due to the one-to-one correspondence between a node, the corresponding classifier and its associated weight vector $\mathbf{w}$, we will use the terms interchangeably.

5.1 Our approach: Multi-Task HC

We adapt the HDC model using a novel structured regularization scheme that utilizes the learnt weight vectors of a node as Bayesian regularizers for the parameters of its children. We present two alternative regularization schemes, that involve the L1 and L2 regularization of the parameters of the child node.

$$\forall c \in \Lambda_p$$

$$\forall i \in \{1, \ldots, D\}$$

L1:

$$\mathbf{w}_c \sim \text{Laplace}(\mathbf{w}_p, \Sigma_p)$$
$$P(\mathbf{w}_{c,i} | \mathbf{w}_{p,i}, \nu^2) = \frac{1}{2\nu^2} \exp\left(-\frac{|\mathbf{w}_{c,i} - \mathbf{w}_{p,i}|}{\nu^2}\right)$$

L2:

$$\mathbf{w}_c \sim \text{Normal}(\mathbf{w}_p, \Sigma_p)$$
$$P(\mathbf{w}_{c,i} | \mathbf{w}_{p,i}, \nu^2) = \frac{1}{\sqrt{2\pi\nu^2}} \exp\left(-\frac{(\mathbf{w}_{c,i} - \mathbf{w}_{p,i})^2}{2\nu^2}\right)$$

L1 regularization is preferred for learning sparse models, while L2 regularization is preferred for avoid over-fitting by the virtue of limiting the region of search for the optimal parameters. In our model, we center the top-level nodes (parent-less nodes) to zero for getting a sparser(L1)/tighter(L2) model, and rest of the nodes follow the regularization rule set above. In this work, we use diagonal matrices for Σ and treat them as hierarchy-wide constants, i.e. $\mathbf{\Sigma}_j = \nu^2\mathbf{I}, \forall j \in \{1, \ldots, Y\}$. Such choices are made

possible by pre-processing methods such as scaling each of the D input dimensions to have unit variance and then normalizing each input vector $\mathbf{x}$ to unit length (L2 normalization).

Our regularization framework is similar in spirit to the one studied by Shahbaba and Neal [2007], with subtle differences. They study hierarchical classification in the multi-class setting, where each instance can belong to exactly one child of a given node. We study the multi-labeled setting, where each instance can belong to multiple children of a given node. Consequently, their model is based on multinomial logit classifiers at each node, while ours is based on binomial logit classifiers, one per node. Secondly, in their model, the parent-child regularization is achieved through the variance term while utilizing a zero mean (Normal$(0, \Sigma_p)$), while in our model, its achieved through the mean term. Furthermore, we develop our model to enable online learning for large datasets, while theirs may not be scale well to large applications.

We present a Stochastic Gradient Descent (SGD) solution for learning this joint Multi-task model. SGD, originally the preferred method for learning the perceptron model Bottou [1991], has now been adopted as the method of choice for speedier large-scale learning of Support Vector Machines Bottou [2010], Logistic Regression, and Conditional Random Fields (CRF) Bottou [2010]. The complete derivation is based on Maximum Likelihood Estimation of the joint model, achieved by setting the derivative of the complete log-likelihood with respect to a parameter to zero. We present here the gradient update steps.

$$w_{i,j} \leftarrow w_{i,j} + \eta \left(\Delta_{\text{instance}} + \Delta_{\text{parent}} + \Delta_{\text{child}}\right) \tag{45}$$

$$\Delta_{\text{instance}} = x_j(y_i - \sigma(\mathbf{w}_i \cdot \mathbf{x}))$$
$$\Delta^{(L1)}_{\text{parent}} = \frac{1}{\nu^2 N}\text{signum}(w_{\pi_i,j} - w_{i,j})$$
$$\Delta^{(L2)}_{\text{parent}} = \frac{1}{\nu^2 N}(w_{\pi_i,j} - w_{i,j})$$
$$\Delta^{(L1)}_{\text{child}} = \frac{1}{\nu^2 N}\sum_{c \in \Lambda_i} \text{signum}(w_{c,j} - w_{i,j})$$
$$\Delta^{(L2)}_{\text{child}} = \frac{1}{\nu^2 N}\sum_{c \in \Lambda_i} (w_{c,j} - w_{i,j})$$

Starting at the root node, upon receiving a training instance, the node's model is updated using the above update rules, before being passed over its children. The process cascades till the instance reaches the leaf nodes. Just like in HDC, a training instance is only passed on to the children, if it belongs to the node itself. Negative instances of a node do not make it to the children. In a strictly online setting, an instance is seen only once, but it is possible to apply this strategy in multiple epochs or iterations over the training set for better performance. Typically, smaller datasets require several epochs, while larger datasets, owing to the redundancy of training instances require fewer or single epoch.

Scalability

HC is usually deployed in large-scale scenarios, and it is crucial to understand the scalability aspects of the approach used. Our approach, MT-HDC, still permits the parallelism strategies offered by the original HDC approach. Each individual node can be learnt in parallel with its siblings. Intra-node communication is limited to simple message passing to get the parameters of a nodes' parent and children for use as regularizers. This message passing can be performed after training on each instance, or lazily after

processing a batch of instances, or at the end of each epoch depending on the scale and availability of resources. Message passing does not require sharing the full vectors, but only the differential (or changed components) since last message. Message passing is an inherent communication mechanism in massively parallel architectures such as MPI and even for achieving parallelism on single multi-core desktop machines using OpenMP. It is straightforward to deploy this algorithm on the Hadoop/Map-Reduce architectures, the most popular platform for processing data on a large-scale. Training for individual nodes could be performed during the *Map* phase (in parallel) and message-passing for sharing parameters can be achieved in the *Reduce* steps.

5.2 Our approach: Active Learning for MT-HDC

For our multi-task AL approach, we exploit the desirable properties of the model to derive the best instances for supplementing the training set. We describe these properties below.

Property 1. Hierarchical Consistency In a hierarchical setting, consistency among parent and child classifiers is paramount. Specifically, we are interested in the constraint given below:

$$\hat{y}_c \leq \hat{y}_p, \forall c \in \Lambda_p \tag{46}$$

The constraint (Equation 46) is a strict constraint and must be satisfied at all times. If an instance doesn't belong to a parent node, then it may not belong to any of its children either. It is reasonable to expect that a HC model that provides consistent predictions based on the applicable constraints is a better model than one that does not. Therefore, from an AL perspective, we are interested in selecting instances that will

attempt to lead to a more consistent predictive model. The instances which already obey the consistency rules cannot be expected to provide any additional benefit in improving the model consistency. Consequently, our AL strategy selects instances that violate the consistency constraints. Note that the consistency rules are *local* for each parent-child relationship and can be checked for an instance at each node to decide on whether to make a membership query at that node. To choose among a set of instances, all of which violate the constraints, we develop an AL consistency score for each of the conditions:

$$\mathcal{C}_p(\mathbf{x}) = 1 - \frac{\sum_{c \in \Lambda_p} \max(0, \sigma(\mathbf{w}_c \cdot \mathbf{x}) - \sigma(\mathbf{w}_p \cdot \mathbf{x}))}{|\Lambda_p|}$$

The score $\mathcal{C}_p$ is similar to the hinge-loss (used in Support Vector Machines) and only focuses on the situations where the child score is higher than its parent for an instance. An instance is consistent if the score is higher, and inconsistent if the score is lower. This AL strategy can be justified using a Version-Space reduction approach. Version-Space (VS) in a classification setting is the set of hypotheses that are consistent with the observed training data. The best classifier is (usually) contained in the VS and methods such as the *max-margin* approaches attempt to identify the best classifier using some heuristic. Naturally, in the absence of any training data, the VS contains all the models in the hypotheses space, and choosing the best one is difficult. As training set size increases, the hypotheses that are inconsistent with the data are removed from consideration (i.e. from VS), making it easier to choose a better classifier from the remaining ones. Thus, a good AL strategy involves choosing instances that lead to significant reduction in the VS for faster improvement of the classifier. The popular AL approach for SVM Tong and Koller [2000], i.e. choosing instances closer to the decision boundary, is based on this principle, as such instances are expected to reduce VS into half (the hypotheses that label the instance incorrectly are removed from the VS. For an instance at the boundary half of the hypotheses label the instance positive and the

other half negative). The space of all hypotheses of the overall HC model is significantly large, $\text{VS}_{HC} = \prod_{i=\{1,\ldots,Y\}} \text{VS}_i$, where VS_i is the size of the VS of node i. Since each individual component matters, a reduction in the VS of any given node leads to the reduction in the overall VS of the HC model. Training the model on an inconsistent instances will remove all the hypotheses that lead to such predictions from the VS. The faster that we can get rid of inconsistent hypotheses, the faster we can improve the model performance, making it necessary to choose such instances for labeling. Some previous approaches have studied the *smoothing* and regularization of raw classifier scores to achieve consistency Punera and Ghosh [2008]. We instead use it as a AL selection strategy, as a means of fixing the hypotheses.

As a side-note, consider that the consistency-based approach to Active learning may not always be effective in selecting the best instances. This is possible when the child and parent nodes are very different in their learnt models. Also, inconsistency in predictions is more likely when a node has fewer training instances. As a node accumulates more training data, the predictions might become consistent, thereby leading to lesser benefit of this strategy. But considering that Active learning is used mostly in the former case (when training data is less), we expect consistency-based approach to work well for Active learning.

Property 2. Influence In the Multi-Task setting, our primary expectation is that each of the children are fine-tuned versions of their parent. If the children of a node are not very different from a parent, then it means that the parent exerts significant *influence* on its children. Consequently, allocating additional labeled training instances to such an influential parent is expected to lead to cascading improvements among its children. Furthermore, a parent that influences *influential* children is the right candidate to receive additional supervision. Mathematically, we can express influence as a simple recursive function of the parameters of a node and its children as follows:

$$\mathcal{I}_p = \sum_{c \in \Lambda_p} e^{-\|\mathbf{w}_c - \mathbf{w}_p\|} \mathcal{I}_c \tag{47}$$

Note that influence [7] is independent of the unlabeled instance and can be computed offline before seeing an instance. This score is to be used in conjunction with other instance-dependent AL scores to supplement the influential node with a suitable training instance. Since labeling budget for the overall hierarchy is limited, with the influence score, we wish to avoid supplementing training data at nodes that have the least impact on the overall performance of the model, even if some instances might be beneficial at such places.

Online score for AL We adapt a popular online scoring function Cesa-Bianchi et al. [2006]; Saha et al. [2010] with the new AL scoring criteria as follows:

$$\zeta_p(\mathbf{x}) = \frac{\mathcal{I}_p}{\mathcal{I}_p + \delta_c \mathcal{C}_p(\mathbf{x}) + \delta_u \mathcal{U}_p(\mathbf{x})} \tag{48}$$

Incoming instances $\mathbf{x}$ at a node p are sampled for membership queries with probability $\zeta_p \mathbf{x}$. $\mathcal{I}_p$ and $\mathcal{C}_p(\mathbf{x})$ are the influence and consistency scores respectively. $\mathcal{U}_p(\mathbf{x})$ is the certainty of the prediction on the instance $\mathbf{x}$ at that node, and is computed simply as the distance from the classification boundary $\mathcal{U}_p(\mathbf{x}) = |0.5 - \sigma(\mathbf{w}_p \cdot \mathbf{x})|$: classifier is *certain* about instances farther from the boundary and uncertain about those in the proximity. Uncertainty of classification estimated this way has long been used as an AL criterion Tong and Koller [2000] for choosing the best instances to query. δ_c and δ_u are the relative weights for the consistency and certainty scores respectively, to provide weighting strategy [8].

[7] Our influence function should not be confused with the one popular in Robust Statistics Huber [1964]. In that context, the influence function estimates the infinitesimal impact of an observation situated at a point on the value of a functional being studied.

[8] For simplicity, we have used equal values for $\delta_c = 0.5$ and $\delta_u = 0.5$ in our experiments

Intuitively, higher influence scores lead to more aggressive sampling of instances for a node, as that term dominates the score. In the absence of significant influence, the system prefers to acquire supervision for only those instances which lead to uncertain and inconsistent predictions.

In scenarios with variable supervision costs at different nodes, appropriate modifications can be made to Equation 48 to give preference to nodes with cheaper supervision. One example modification could involve normalizing the score at a node with the corresponding supervision cost to determine the per-unit benefit of acquiring supervision at that node. Additional modifications can be made by introducing weighing factors for the three components (influence, uncertainty and consistency) to balance their effects. In our work, we study the uniform cost scenario with equal weighing of components and leave the weighted variable cost scenarios for further exploration.

5.3 Experiments

5.3.0.5 Datasets

We use a wide selection of publicly available benchmark collections popular in the HC community. There's a large diversity in the scale, type, depth, and label-cardinality of these datasets. The details are provided in Table 3.

Dataset	**instances**	**features**	**categories**	**Hierarchy Depth**
ImCLEF07D Dimitrovski et al. [2008]	11006	80	46	3.0
ImCLEF07A Dimitrovski et al. [2008]	11006	80	96	3.0
Enron Klimt and Yang [2004]	1648	1001	54	3.0
Reuters Lewis et al. [2004a]	6000	47236	100	4.0

Table 3: Hierarchical Datasets

5.3.0.6 MT-HDC performance

The results presented here have been averaged over 25 randomized splits of training/test sets. We evaluate the performance using the popular F-score (harmonic mean of Precision and Recall) metric, computed as a Macro-Average and Micro-Average. As a comparative baseline, we compare against vanilla HDC. HDC is the conventional approach described before in which each node is learnt in isolation with its own regularizer that is independent of the parent's parameters. In Figure 14 and Figure 15, we present the comparison of the 4 configurations: L1/L2 regularization and vanilla HDC versus Multi-Task HDC (MT-HDC). It can be observed that in all the cases, MT-HDC outperforms the conventional approach. Although L2 regularization is seemingly better, there is no clear winner.

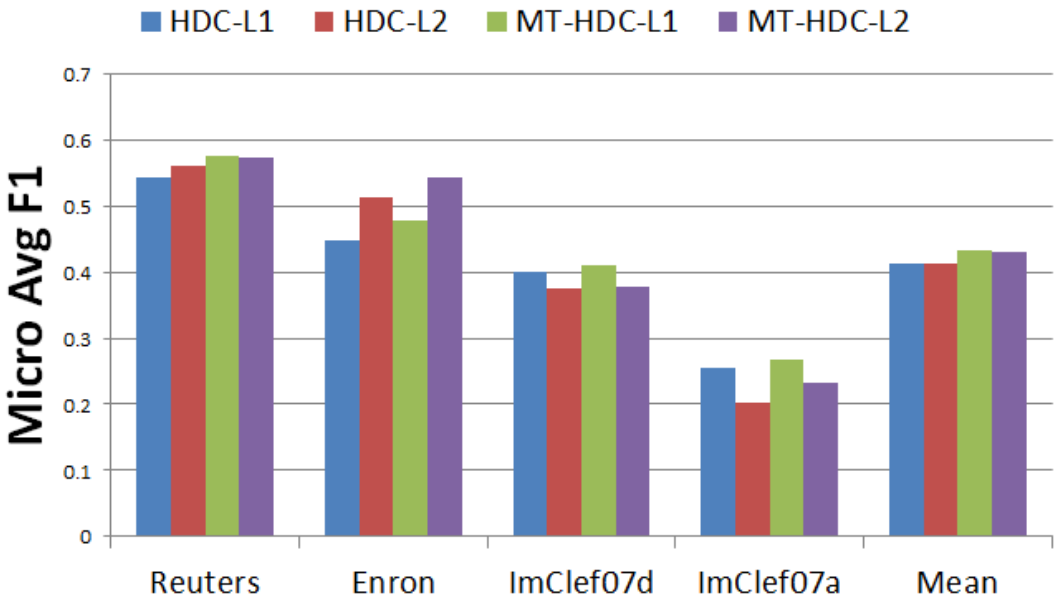


Figure 14: Classifier performance when trained on 60% of the available instances: Micro-Average F1

For further clarity, in Figure 16 and Figure 17, we compare the depth-wise performance of the various methods. It can be observed that the MT-HDC approach outperforms at

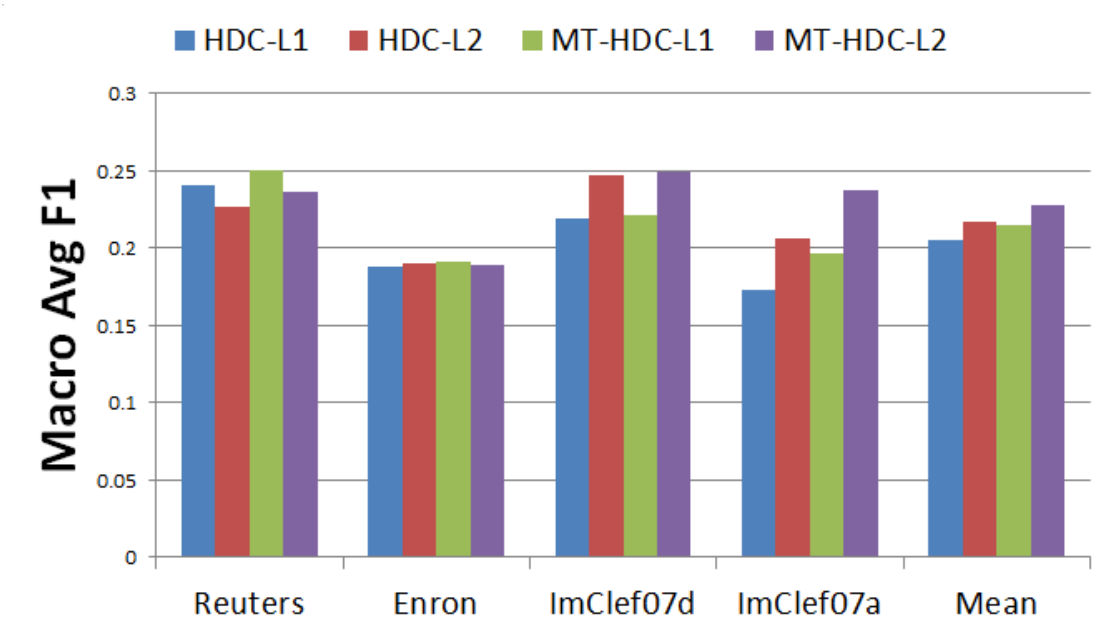


Figure 15: Classifier performance when trained on 60% of the available instances: Macro-Average F1

all levels, especially at the lowest levels where there is dearth of training instances.

To evaluate the performance based on skewness or imbalance of training data, we present a comparison of MT-HDC with HDC as a function of the class imbalance in training data in Figure 18. It can be observed that MT-HDC handles class-imbalance better than HDC.

5.3.0.7 Why does Structured Regularization work?

To further analyze the benefit of using structured regularization in MT-HDC, we study the actual learnt parameters of the model over different phases of learning. In Figure 5.3.0.7, we present a heatmap of the parameters of a node. Each column in the heatmap is the weight vector $\mathbf{w}$ of the same node at a given time, with the first column being the initial weight vector after being trained on only one instance and the last column being the final weight vector when all the training data has been observed

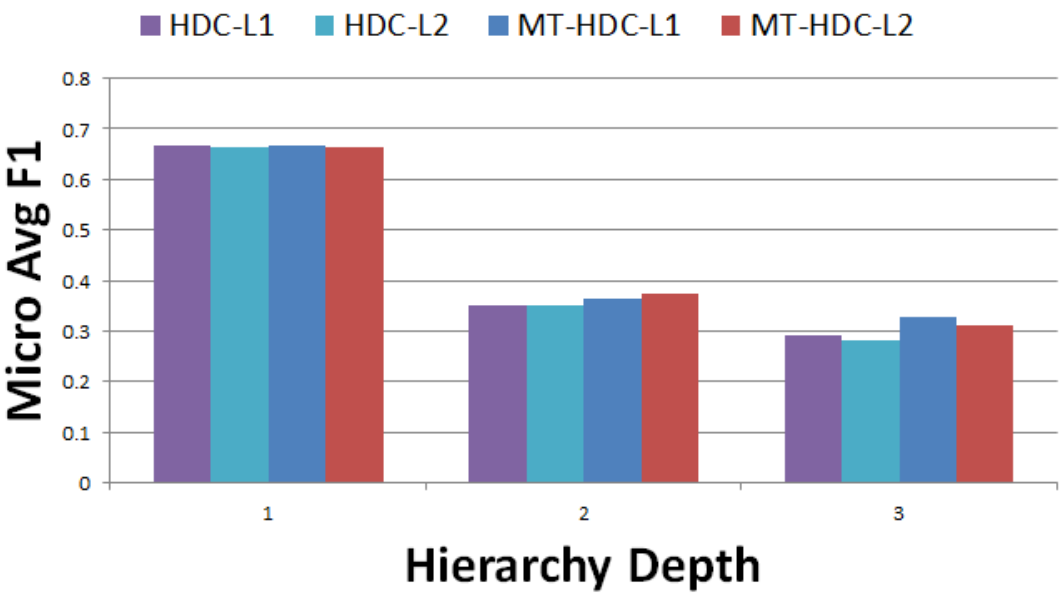


Figure 16: Depth-wise classifier performance when trained on only 1 positive instance per-leaf node: Micro-Average F1

by the node. In an ideal setting, to be able to learn from minimal data, the weight vector should stabilize to its final value as early as possible. In the Figure 5.3.0.7, it can be observed that the parent node stabilizes much earlier than the child node. This is primarily because at any given time, the parent node has access to more training instances than the child node. It can also be observed that the structurally regularized child becomes more discriminative among features more quickly than the one without parental guidance. It can be observed that the child is indeed a fine-tuned version of its parent, slightly different in the parameter space.

5.3.0.8 Experiments: Online Active Learning

For evaluating our novel Multi-Task Active Learning score (henceforth MTAL-HDC), we use the experimental setting popular in the online Active Learning community Beygelzimer et al. [2010]; Saha et al. [2010]. Specifically, we start with only one positive

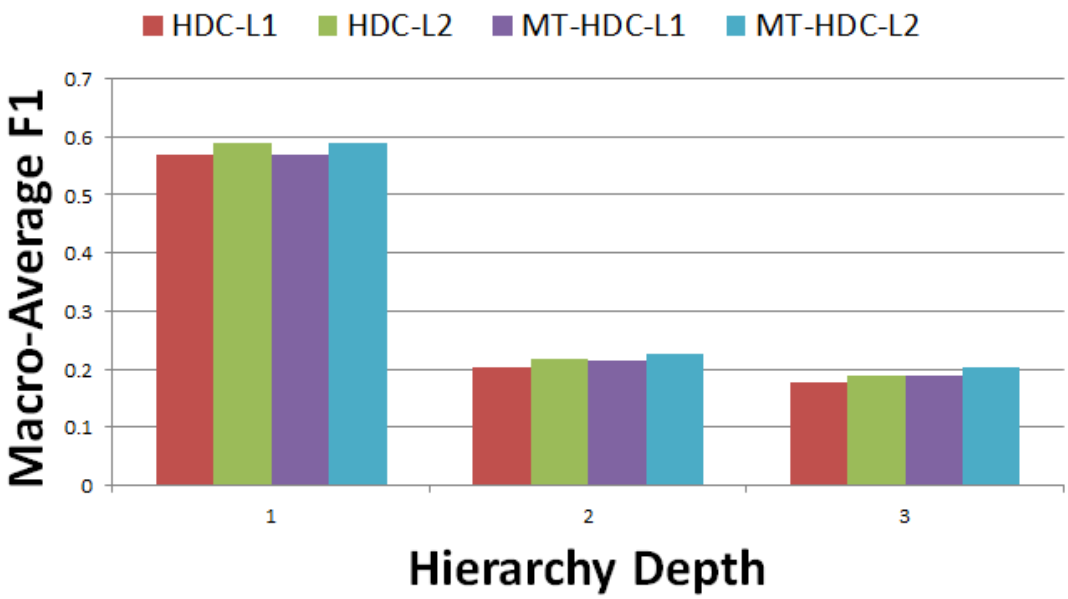


Figure 17: Depth-wise classifier performance when trained on only 1 positive instance per-leaf node: Macro-Average F1

instance per leaf node in the overall training set [9]. As a weak baseline, we compare the online AL strategies to getting all the labels on each unlabeled instance that arrives at a node. This we call the **Full supervision** baseline. As a strong baseline, we use the uncertainty-based sampling approach, using the formulation in Cesa-Bianchi et al. [2006]. This essentially means the same form as Equation 48, but without the consistency and influence components. We call this the **uncertainty** based approach. For these experiments, we have used only the MT-HDC-L2 model for all the comparative AL strategies. In Figure 20, we compare the chosen baselines with MTAL-HDC in terms of the classification performance as unlabeled data is streamed (and queried, actively) through the system. It can be observed that the MTAL-HDC performance is approximately upper bounded by the performance of obtaining full supervision, and is not too less than it. Also, MTAL-HDC performance is superior to the uncertainty-

[9] By virtue of being multi-labeled, some leaves might get more than one positive instances in our experiments, but the initial training set is same for all compared approaches

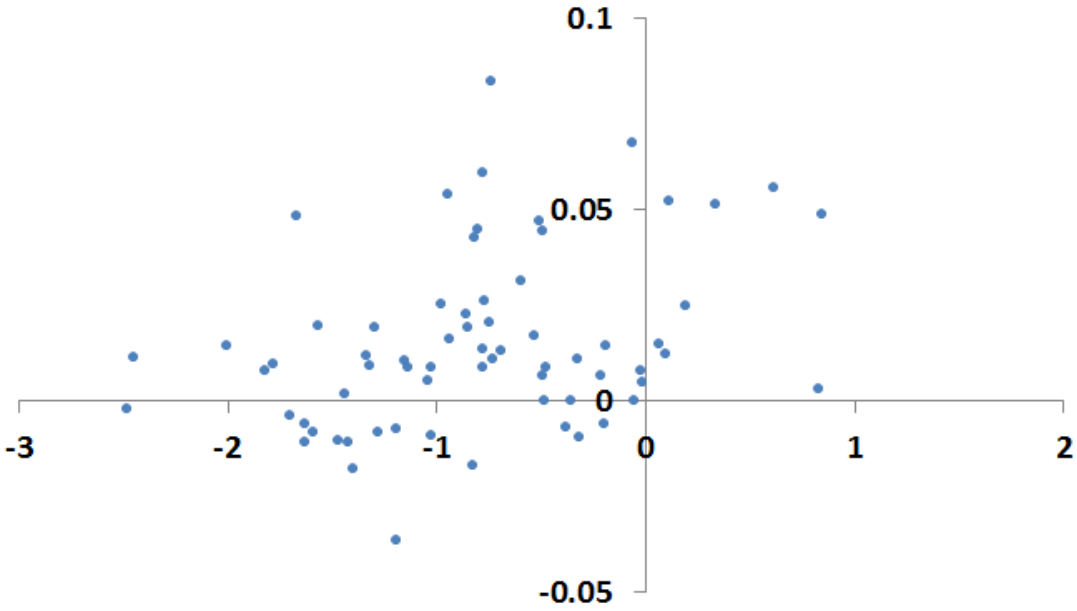
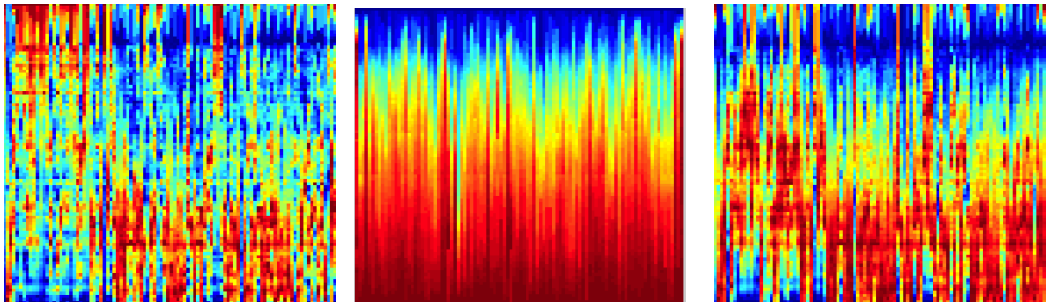


Figure 18: Training set class imbalance Versus Performance improvement using MT-HDC over HDC. The horizontal axis denotes the $\log\left(\frac{\text{Number of positive instances}}{\text{Number of negative instances}}\right)$. Thus left extreme shows scenarios with scarce positive instances and right extreme shows scenarios with scarce negative instances. The vertical axis shows the difference in the performance of the MT-HDC and HDC approach. Positive half indicates MT-HDC is better than HDC and negative half shows superiority of HDC over MT-HDC on that class.

based sampling strategy. To further validate the efficacy of our approach, we compare the actual number of active queries (membership queries) made by our approach to that of the baselines. (Full supervision always makes all possible queries). It can be observed in Figure 21 that the MTAL-HDC approach makes significantly lesser number of queries than the baselines. In conjunction with the earlier experiment in Figure 20, this would mean that despite the significantly lesser number of queries, there is not much performance sacrifice using the MT-HDC approach. This exemplifies the significant cost-reductions made possible by using MTAL-HDC.



(a) Child: Without Structural Regularization (HDC-L2)

(b) Parent

(c) Child: With Structural Regularization (MT-HDC-L2)

Figure 19: Comparison of the learnt parameters of a parent and its child when trained using L2 regularization. The left and right heat-maps are for the same child, but with and without structural regularization. The colors of the heat-map denote the magnitude of the learnt weight vector components. The horizontal-axis of the heat-map denotes the fraction of training instances observed (0 through 1). The vertical-axis of the heat-map is representative of the components of the weight vector (features).

5.4 Summary

We presented a novel combination of Multi-task Learning and Active Learning for the important problem of Hierarchical Classification. For Multi-task learning, we regularize the child nodes with the learnt parameters of the parent node, to appropriately guide the child parameters. For Active learning, we presented a novel concept based on hierarchical consistency that leads to further reduction in training instances than conventional active learning strategies that select instances for each node in isolation. Through experiments, we demonstrated significant reduction in the overall training data required for HC, with particular improvements in the performance of the nodes with scarce training data.

There are several direction for further exploration. In the current work, we have focused on the local parent-child disagreement for choosing instances. It would be interesting to extend the idea to computing more deeper disagreements, for example over a sub-tree rooted at a node to select instances that have more deeper impact in the hierarchy. That would further require novel algorithms for computing sub-tree disagreements more

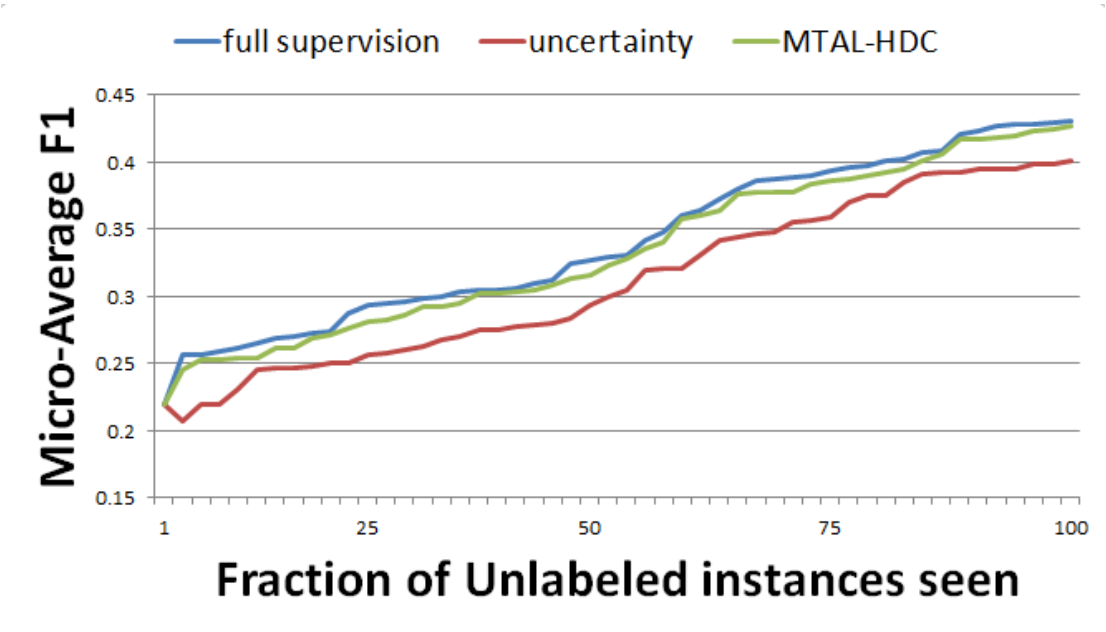


Figure 20: Comparison of online active sampling strategies in terms of the performance improvement as unlabeled data is streamed through the model.

efficiently. As a more convincing argument in favor of the proposed approaches, it is crucial to develop a theoretical analysis of the label complexities of the proposed methods.

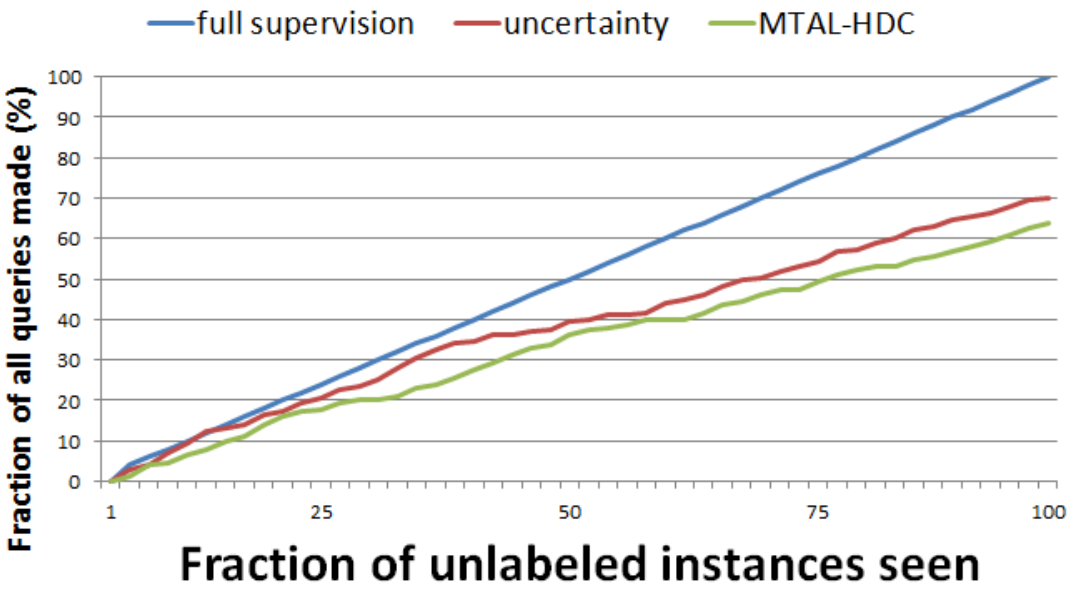


Figure 21: Comparison of online active sampling strategies in terms of the number of membership queries made by the approaches.

6

Oracle-sensitive MTAL: Active Collaborative Filtering

Active Learning for Collaborative Filtering is a relatively new concept; here the goal is to solicit ratings for minimal set of movies from a user to learn about the user's preference pattern. This is very essential because when new users join existing models, the system knows little about their preferences and the system would like to understand the user preference pattern with the least amount of training examples so as to not annoy the user with lot of questions.

Previous work on Active Learning applied to Collaborative Filtering Boutilier et al. [2003]; Jin and Si [2004], have made one common implicit assumption, that users would be able to provide rating to any item that is requested by the system. This assumption, in reality, is not true , because, to rate a movie, a person has to first procure the movie, and watch it. This can be very time consuming, as opposed to, for example, in text classification where users can quickly label text documents just by skimming through the snippets. Moreover, users would not like a system which solicits ratings for movies the user may not even watch and such a dialog can be frustrating. Since Collaborative Filtering deals with user-personalization, we believe Active Learning should also be

personalized to solicit ratings about items that a user would usually watch. In this paper, we provide one such *personalization* approach for active learning applied to Collaborative Filtering.

As part of this work, we demonstrate a successful approach to model oracle expertise. In this setting, each task is an item-recommendation task for a user, and there is a single-oracle per-task, the user itself. Note that the Proactive Learning approach [Donmez and Carbonell, 2008a] of clustering instances based on their content may not be directly applicable to this setting, as no other features are known about the items. Our approach instead utilizes the similarity among users (or oracles) in rating similar kinds of items. Previous work on Active Learning applied to Collaborative Filtering [Boutilier et al., 2003; Jin and Si, 2004], have assumed that users would be able to provide rating to any item that is requested by the system. In reality, users have experience with very few items resulting in very few successful active queries.

6.1 Personalized Active Learning

We define probability of obtaining a rating/supervision on item x from user/task m as $P(s|x,m)$. We estimate this probability by developing a variant of the Probabilistic Latent Semantic Analysis (PLSA) [Hofmann and Puzicha, 1999] model. We first cluster users into aspects $\mathbb{Z}$ based on their rating profiles and then discover the likelihood of obtaining supervision over an item in a particular interest group $p(s|x,z)$.

$$p(s|x,m) = \sum_{z \in Z} p(s|x,z)p(z|m) \tag{49}$$

$$p(s|x,z) = \frac{\sum_{m \in \mathbb{M}} p(z|m)I(m,x)}{\sum_{m \in \mathbb{M}} \sum_{x' \in \mathbb{X}} p(z|m)I(m,x')} \tag{50}$$

$$I(m,x) = \begin{array}{l} 1, \text{if user } m \text{ has rated item } x \\ 0, \text{otherwise} \end{array}$$

$I(m,x)$ is an indicator function that indicates whether a user has rated an item x. Thus, in modelling $p(s|x,m)$, we are not interested in the actual rating provided by a user, but whether the user provided a rating for an item. We estimate the desired components using the Expectation Maximization (EM) algorithm [Hofmann, 2003].

The probability $p(s|x,m)$ of obtaining rating for an item x from a particular user m can be used as a filter to first remove instances that are highly unlikely to receive ratings from the user. A chosen AL strategy can then choose items, to solicit ratings for, from the remaining unrated instances for that particular user. We reflect this *filtering* through the use of a soft-AND function selecting instances that have a high AL benefit $\mathcal{A}_m(x)$ and score higher on oracle expertise $p(s|x,m)$.

$$x_m^* = \arg\max_{x \in \mathbb{U}_m} \mathcal{A}_m(x)\ p(s|x,m) \tag{51}$$

In the context of Collaborative Filtering, the score $\mathcal{A}_m(x)$ can be the Bayesian selection criteria described in Equation 10.

6.2 Experiments

6.2.0.9 Experimental Setup

We use the MovieLens[10] and MovieRating[11] datasets for the empirical analysis of our approaches. The detailed characteristics of the two datasets are presented in Table 4.

The datasets were randomly split into training/test sets as shown in Table 5. The Aspect Model is first trained from training set. This gives the global model $p(r|m,z)$ as

[10] www.grouplens.org/system/files/ml-data.zip. This dataset is similar to the EachMovie dataset (prevalent in earlier literature) which is no longer available for usage. www.grouplens.org/node/76

[11] http://www.cs.usyd.edu.au/~irena/movie_data.zip

Table 4: Characteristics of MovieRating and MovieLens datasets

	MovieRating	**MovieLens**
Number of users	500	943
Number of movies	1000	1682
Average number of rated movies/user	87.7	106.05
Rating Scale	1-5	1-5

Table 5: Our Experimental Setup

	MovieRating	**MovieLens**
Number of Training-set users	200	343
Number of Test-set users	300	600
Number of Initial ratings per test user	3	3
Number of Preserved ratings (evaluation) per test user	20	20
Total number of Preserved ratings (evaluation) over all test users	6000	12000
Number of Active Selection Candidates per test user	977	1659
Number of user-classes	5	10

defined in Equation 7. Active learning involves soliciting ratings from new users. The test set forms this set of new users. Thus we start with very few initial ratings for each user, about 3, simulating a new user. The user-model θ_u is learnt from these ratings. For each test-user, remaining movies are split into two sets, the active-selection set and the evaluation set. Active learning algorithms select movies for rating from the set of active selection candidates. In previous approaches, the active-selection set consisted only of movies for which ratings are available in the dataset . This is unrealistic, since a system cannot know beforehand which movies will be rated by the user. This is also evident from the dataset characteristics reported in Table 4 in which average number of movies rated by a user is less than 10% of the total number of movies. We use a more realistic setting, in which the active-selection set for each test-user consists of all the movies (rated and unrated) which are not included in 1) the initial-ratings and 2) the evaluation set for that user. This makes active-selection more challenging and the

setting realistic. The evaluation set is a held-out set of movies for the test-user which is used for evaluating the performance of the Collaborative Filtering system.

We have set the number of user-classes to 5 and 10 respectively for the MovieRating and MovieLens datasets as reported in previous empirical studies and Active Collaborative Filtering literature Jin and Si [2004].

6.2.0.10 Evaluation Metrics

We evaluate the system using two kinds of metrics:

CF performance: This includes the usual CF metrics,i.e. the Mean Absolute Error (MAE) and Mean Squared Error(MSE). MAE and MSE measure the deviation of the predicted ratings of the movies in the test set from the actual ratings available in the dataset. In the equations below, M_{ue} denotes the evaluation set for the user u and $n(M_{ue})$ denotes the number of movies in the set.

$$MAE_u = \frac{1}{n(M_{ue})} \sum_{m \in M_{ue}} |r_m^{true} - r_m^{predicted}| \tag{52}$$

$$MSE_u = \frac{1}{n(M_{ue})} \sum_{m \in M_{ue}} (r_m^{true} - r_m^{predicted})^2 \tag{53}$$

Since we use multiple test-users, the reported MAE (and MSE) is the average over individual MAE (and MSE) for each test-user as shown in Equation 52, where U_t is the set of test users and $n(U_t)$ denotes the number of test users.

$$MAE = \frac{1}{n(U_t)} \sum_{u \in U_t} MAE_u \tag{54}$$

Failures: The system solicits ratings for movies from the user and the user may not provide ratings for some of them. Such instances are known as *failures* to obtain rating from the user. This metric is very essential for comparison of the personalized approach with previous approaches, since it clearly identifies the number of queries from the active

learning algorithms that will go unanswered, and thus not useful for the active learning system. High failure rate also means a system which is likely to annoy the user with questions for which the user does not have any answer. In our experimental setup, a failure occurs whenever the system solicits rating that does not occur in the dataset. The system thus cannot be re-trained and wastes an active-learning cycle and proceeds to the next iteration.

6.2.0.11 Active CF cycle

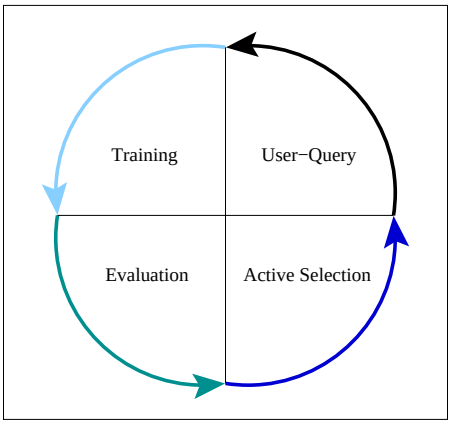


Figure 22: The Active Learning process

The Active learning cycle is shown in Figure 22. The global model is learnt over all the training-set users. The active learning cycle starts for each of the test users after the global model is learnt. The system learns a user-model for a test-user based on the available intial ratings, or the labeled-set and the system is then evaluated over the evaluation set. The system then selects a movie for rating from the active-selection set of the user. If the user provides rating for the movie, the system adds this movie-rating pair to the labeled-set and retrains the system model. If the user does not provide rating for the movie, then it is noted as a failure. The system cannot be retrained in case of a failure. In either case the system is evaluated in terms of MAE and MSE on the evaluation set of the user. This completes one active-iteration. In the next step the system asks rating for another movie from the user and the cycle continues.

6.2.0.12 Aspect Model Implementation

For the Aspect Model, we model $P(r|m,z)$ with gaussian distributions $N(r;\mu_{m,z},\sigma_{m,z})$. We use a multinomial model for the user-group mixing proportions $P(z|u)$, as proposed in Hofmann [2003]. Users generally do not follow the same distribution to provide ratings to items, and so it is common in the collaborative filtering techniques to normalize user-ratings to have zero mean and unit variance. It should be noted that the normalization parameters are learnt only over the ratings available to the system, not over the items in the active-selection candidates and evaluation set.

6.2.0.13 Comparative Baselines

The proposed Personalized Active Learning algorithm is compared against following approaches towards item selection:

- **Random Selection(RS)**: Items are selected randomly from the set of active-selection candidates for soliciting ratings from the user.
- **Bayesian Selection(BS)**: We implement the Bayesian approach outlined in Jin and Si [2004]. This approach has already been compared to other existing approaches like *Model Entropy based Sample Selection*, which selects item based on expected reduction in model entropy, and *Prediction based Sample Selection* which selects items based on the uncertainty in prediction. According to Jin and Si [2004] Bayesian Selection outperforms the other approaches.

We replicate the experiments in Jin and Si [2004] with our implementation of the baseline in Figure 24. In our comparative evaluation, the baseline performs similar to Jin and Si [2004] using the experimental setup reported in Jin and Si [2004]. It should be noted however that Jin and Si [2004] performs active-selection only on the subset of movies for which ratings are available in the dataset, thereby making the assumption that user can rate any movie. We call this setup *constrained* because the

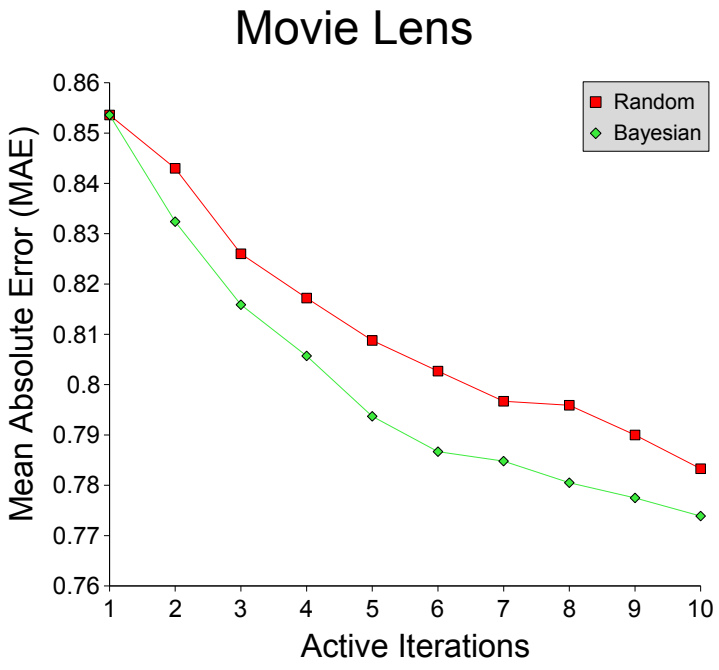


Figure 23: Constrained Baseline implementation: Active learning trends over 10 active-iterations for the MovieLens dataset

active-selection set is constrained to rated items. This is unrealistic, since the system will not know beforehand, which items will be rated by the user. Therefore, for comparison with *personalized* Active Learning, we will be using the entire set of movies, not just the ones for which ratings are available in the dataset. We call this the *unconstrained* setup.

For the sake of conciseness, we call our own approach the **Personalized Bayesian Selection (PBS)**.

6.2.0.14 Empirical Results and Discussion

The results for our experiments are presented in Figure 25, for the first 10 active iterations of the system.

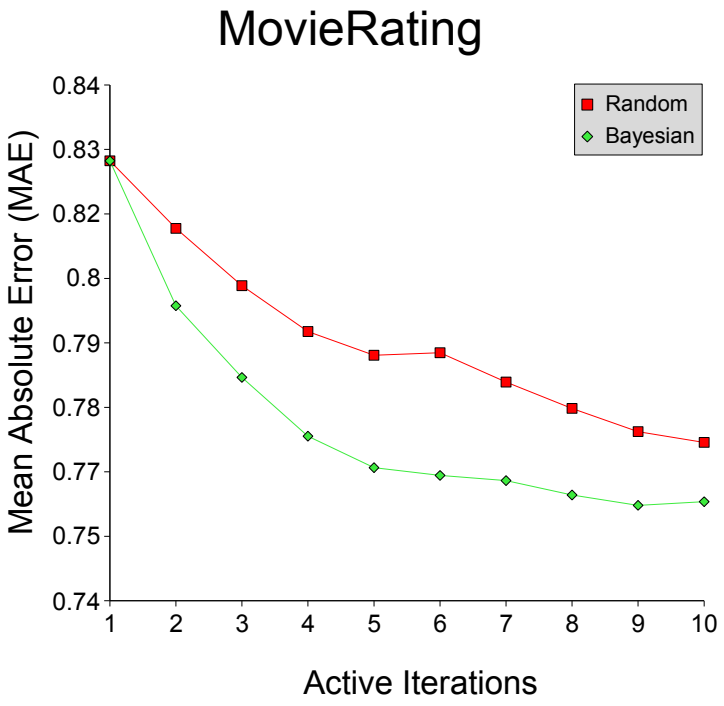


Figure 24: Constrained Baseline implementation: Active learning trends over 10 active-iterations for the MovieRating dataset

It can easily be observed that our approach, Personalized Bayesian Selection outperforms Bayesian and Random Selection on both datasets using both metrics, MAE and MSE. Moreover, since Bayesian Selection does not take the probability of getting a rating into account, it performs even worse as compared to plain Random Selection. This is expected because Bayesian selection tries to identify items that will provide the most information about user-model, but in the process often selects items which the user would not be able to provide a rating for. This is evident from the Table 6 where we report the mean number of failures, over all active-users for 10 active iterations.

As shown in Table 4, average number of ratings per user is about 87.7 (out of 1000) and 106.05 (out of 1620) for the MovieRating and MovieLens datasets respectively, each user has on an average only 1 item rated out of 10. Thus, it is expected that random

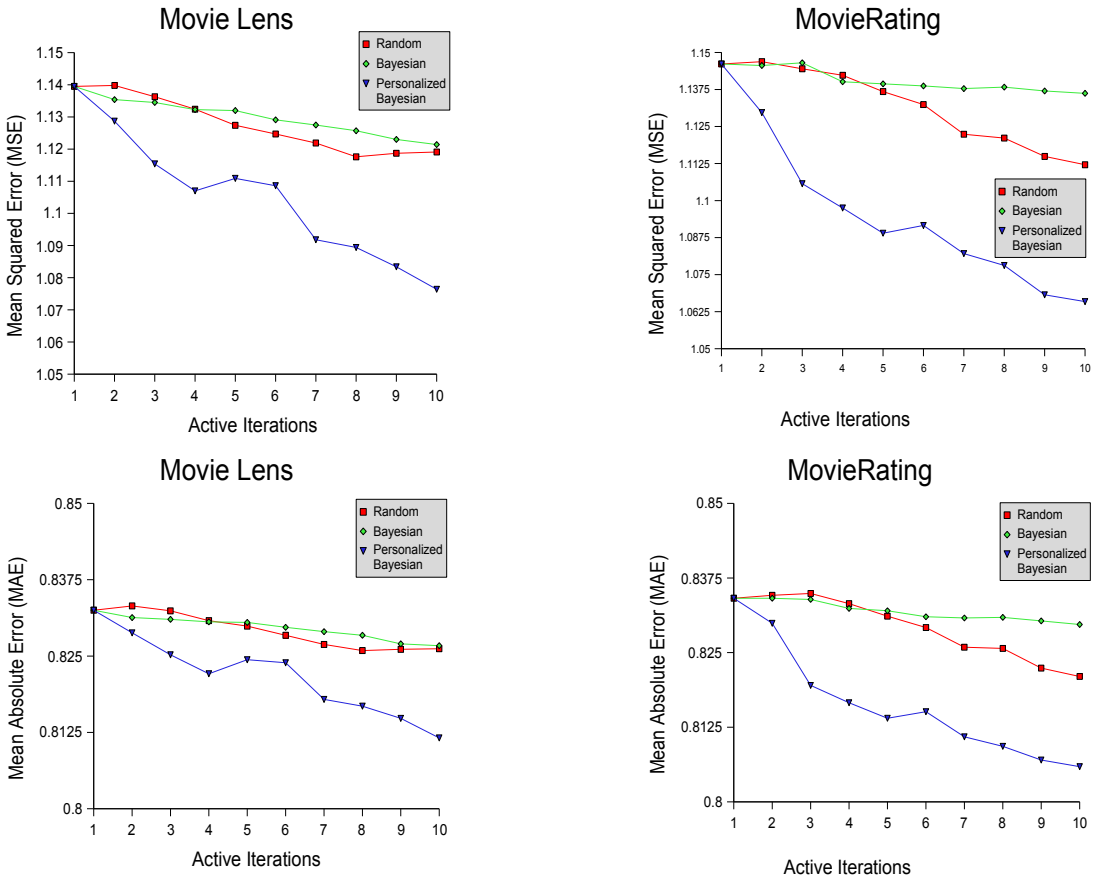


Figure 25: Unconstrained setup: Active Learning trends over 10 active-iterations

Table 6: Mean number of Failures over 10 active iterations (averaged over 300 active-users for MovieRating and 600 active-users for MovieLens)

	MovieRating	**MovieLens**
RS	9.0728	9.3268
BS	9.8398	9.8366
PBS	5.995	5.1267

selection has about 9 failures out of 10 active-iterations. Bayesian Selection has more failures than Random Selection because most informative items may not be rated by the user. Personalized Bayesian selection substantially reduces the number of failures and selects atleast 4 out of 10 items for which the active-user can provide a rating.

It should be noted that active CF systems will solicit ratings for new movies from the user

till the system has substantial number of ratings from the user. A system with higher failure rate will tend to ask more questions to obtain comparable number of ratings from the user. Users typically do not favor systems which ask too many questions. Thus, a *personalized* active CF system is more likely to be favored by the user than an unpersonalized one.

6.2.0.15 Significance Tests

We conducted paired t-tests Yang and Liu [1999] for comparing the performance of the PBS, BS and RS approaches. Given two systems (say A and B) and a set of n items, the MAE (or MSE) values were computed for both systems, for each test item, denoted as a_i, b_i for $i = 1, 2, \ldots, n$. We used the pairwise difference $d_i = a_i - b_i$ to examine the null hypothesis that expected difference $\overline{d} = \frac{\sum_i d_i}{n}$ is zero against the alternative hypothesis $\overline{d} > 0$, meaning system A is better than system B. The p-value is computed using the t-distribution with the degree of freedom $n - 1$ for

$$T \geq \frac{\overline{d}}{s.e.(\overline{d})} \tag{55}$$

Table 7: p-values for the significance tests comparing various active learning approaches

		p-value	
System A	**System B**	**MovieRating**	**MovieLens**
RS	BS	0.04	0.0019
PBS	BS	6.2e-05	2.1e-13
PBS	RS	0.0040	8.9e-08

Small p-value (≤ 0.01) means statistically significant evidence against the null hypothesis. Table 7 reports the p-values obtained in our tests using the MAE values of each system after 10 active learning iterations, i.e. after selecting 10 items and adding their ratings (if rated) to the current training set. It can be observed that our approach (PBS) significantly outperformed Bayesian Selection (BS) and Random Selection (RS).

Table 8: Exemplary movie clusters sorted on $P(m|z)$ for the MovieRating dataset

Cluster A
Dante's Peak
Contact
Scream
Air Force One
Murder at 1600
The Game
Conspiracy Theory
Titanic
I know what you did last summer
The Devil's own

Cluster B
Star Trek: First Contact
Return of Jedi
Scream
Courage under Fire
Contact
Mission Impossible
The Godfather
Star Trek: The wrath of Khan
The fifth element
Star Trek: Generations

Random Selection performed better than Bayesian Selection with statistically moderate evidence, i.e., the p-values were less than 0.05 and above 0.01 for these comparisons.

6.2.0.16 Analyzing Personalization

The personalization term $P(m|u)$ in the active-selection Equation 49 consists of two terms, $P(z|x)$, the user-group mixing probabilities and $P(s|z,x)$, the probability of getting a rating for a movie m in group z.

In the Table 8, we present lists of movies in two exemplary interest-groups learnt for the MovieRating dataset. We show top 10 movies in each group sorted in descending order of $P(m|z)$. It can be observed that cluster A consists of movies which thriller fans will view and cluster B consists of movies which usually science-fiction fans will view. If a user has watched some of the movies in a cluster , it is very likely that the user has also watched other movies in that interest group. Thus it is reasonable to ask the user to rate the other movies in that cluster to learn a stronger user-model. Unpersonalized methods do not follow such an approach and movies will be identified for rating from any of the interest groups, even from those in which the user has never watched any movies. This is the main reason for the higher failure rate of the unpersonalized approaches.

6.3 Summary

We have shown that personalization of active learning is essential for getting benefit from a Bayesian active learning approach to collaborative filtering. Personalization can be achieved in a relatively simple way: by selecting the most informative items for a user and estimating the likelihood of the user to rate those items. Our experiments on benchmark datasets provide strong evidence for theoritical advantage of our approach over a baseline that is representative for well-established active learning methods without personalization in collaborative filtering. rate those items.

For future work, we plan to incorporate implicit feedback into the model. Thus, when a system selected item fails to get a rating, we can use failure as additional information to update the user-aspect probabilities $P(z|u)$. Google recently demonstrated techniques for making Aspect Model scalable to the number of news articles for collaborative-filtering based personalization of its news services Das and Garg [2007]. A similar scalability study on Active Learning techniques for Collaborative Filtering is required, in addition to the scalability analysis of our personalized approach.

7

Conclusion and Future work

7.1 Summary

In this thesis, we addressed the challenging problem of selectively acquiring training data for simultaneously improving multiple related tasks. Training data acquisition is expensive, and by choosing to acquire supervision that is beneficial for learning several tasks, one can mitigate the cost of learning such tasks. We proposed a novel Multi-Task Active Learning (MTAL) framework and studied it in the context of several practical scenarios.

In Chapter 3, we provided a general purpose solution to MTAL. This approach is applicable in general to the commonly used approach to Multi-Task Learning: shared Bayesian priors over tasks. By propagating the updates of supervision through the shared priors, it is possible to estimate the impact of acquiring supervision for one task on other tasks. In this context, we defined the *Circle of Influence* to allow the selection of the granularity of impact estimation, leading to several variants, namely, local, global and benevolent Active Learning.

In Chapter 3.5, we studied the proposed MTAL framework in the context of the important Adaptive Filtering application. Conventional Adaptive Filtering approaches

narrowly focus on immediate rewards and present only the relevant instances to the user. However, by taking a longer term strategy, our proposed active learning strategy also chooses to present instances that if labeled by the user will lead to future expected gain in the utility of the AF system. In this particular application, we also demonstrated the effectiveness of using a multi-pronged strategy of combining local, global and benevolent models for active learning.

In Chapter 4, we motivated the problem of scalability of MTAL, particularly in the context of heterogeneous tasks. We addressed the challenge by jointly learning the diverse tasks through a topic-model approach and then utilizing the topic-model for defining a surrogate active learning score. This approach, called Transferable Active Learning (TAL), avoided the exhaustive computation of impact of acquiring a supervision on all the tasks, while still leading to a performance benefit among all the tasks.

In Chapter 4.3, we studied the TAL in the context of the practically important problem of genre-classification and collaborative fitlering. We built joint model by enhancing the collaborative topic regression model with the genre classification component. The model lead to significant performance improvements in both the classification and recommendation tasks. For this particular model, we utilized the entropy of the topic model as the surrogate active learning score. The TAL approach chose to acquire supervision that would lead to the maximal decrease in the entropy of the topic model. We demonstrated the success of these approaches on the benchmark MovieLens dataset.

In Chapter 5, we described a novel MTAL strategy for hierarchical classification. By leveraging the parent-child relationships, we proposed a novel joint learning model for hierarchical classification. We also proposed two novel active learning strategies for leveraging the inter-task relationships. The first strategy, namely consistency, chose to acquire supervision for instances that could lead to predictions that were consistent with

the given hierarchy. The second strategy, namely influence, favored the acquisition of additional supervision for influential nodes, with the expectation that the influential nodes will lead to further cascading improvements in the *influenced* nodes.

In Chapter 6, we described a probabilistic model for estimating the possibility of acquiring supervision for a given task, based on historical data. Such modeling can potentially avoid the problems associated with failed queries in active learning, i.e. active learning based queries that the oracle is unable to answer. We studied this in the context of the Collaborative Filtering problem, using the popular Aspect Model with Bayesian Active Learning algorithm. Our experiments on benchmark datasets, namely Movielens and EachMovie, show significant reduction in the amount of failed queries, while still leading to significant performance improvements in the learnt model.

7.2 Research Contributions

The major research contributions of this thesis are:

- **Circle of Influence (CoI)**: CoI is a flexible strategy for estimating the impact of acquiring supervision for one task on other tasks. By varying the granularity of impact-estimation, we are able to develop several variants such as the local, global and benevolent strategies for Multi-Task Active Learning.
- **Benevolent Active Learning**: Benevolent Active Learning is a principled and exhaustive approach to estimate the impact of acquiring supervision for one task on other tasks being learnt jointly. The effect of supervision on one task is propagated through the appropriate paths in the Bayesian network.
- **Multi-Task Active Learning through surrogate scores**: To mitigate the computational intractability of exhaustively estimating the impact of supervision on the numerous tasks in the system, we proposed to utilize surrogate scoring criteria that instead choose to estimate the impact on a suitable central component

of the system. For example, we chose to minimize the topical entropy of the topic-model that is central to simultaneously learning numerous heterogeneous tasks.

- **Influence-driven Active Learning:** In several multi-task scenarios, such as hierarchical classification, it could be beneficial to selectively acquire supervision for influential tasks, while expecting resulting improvements in the *influenced* tasks. To this end, we presented a novel scoring function for estimating the *influence* a parent exerts on a child in a hierarchical Bayesian model for hierarchical classification.
- **Consistency-driven selection for structured tasks**: For structured tasks, we presented a novel strategy of selectively acquiring supervision for instances that violated the consistency of the given structure. In our experiments on hierarchical classification, we presented the efficacy of this approach over several benchmark datasets.

In addition to the aforementioned contributions to the field of Active Learning, as a consequence of our work, we have also made important contributions in implementing Multi-Task Learning approaches to several practical application.

- **Multi-task Adaptive Filtering**: We presented the first model for performing multi-task Adaptive Filtering. We utilized a Dirichlet-process based classification model, and chose a suitable filtering score strategy for filtering out irrelevant instances in the AF setting.
- **Genre-driven Collaborative Filtering**: We presented a novel topic-model based approach for jointly learning genre-classification and Collaborative filtering, using well-known building blocks such as the probabilistic matrix-factorization based Collaborative Topic Regression model and the Bayesian Logistic Regression model for genre-classification.
- **Bayesian framework for hierarchical classification**: Our joint model for

hierarchical classification leverages parent-child relationships through Bayesian priors for enhancing the popular hierarchical divide-and-conquer paradigm. By requiring the children to be fine-tuned versions of their parent, it is now possible to learn the children with minimal supervision, by leveraging the abundant supervision at the parent nodes.

7.3 Future work and Extensions

Multi-Task Active Learning is a new research area and there are several directions for future work.

Exploring Circle of Influence: We presented Circle of Influence as a general purpose flexibility for analyzing the impact of acquiring supervision on one task on other tasks and parameters of the model. There are many more possibilities than the local, global and benevolent CoI studied in this work. For example, in the scenarios with groupings and clusters of related works, it might be possible to select group-level CoI. Further exploration is required to study the trade-offs of such approaches.

Scalability and tractability: Multi-Task learning is inherently a challenging problem in terms of scalability. Jointly learning numerous tasks poses significant challenges in terms of inference algorithms. For special scenarios such as Hierarchical classification, previously preferred methods such as Hierarchical Divide and Conquer become significantly more challenging, in terms of scalability, when learnt as a joint model. As potential remedies, there has been a lot of interest in devising clever sampling strategies such as Collapsed Gibbs Sampling and approximation algorithms such as Variational inference to mitigate the problem of scalability. Nevertheless, performing Active learning in multi-settings introduces further complexity in terms of scalability. In addition to the usual challenge of a one-time batch-learning over training data for multiple tasks, we are now posed with the problem of significant computational expense for first identifying

instances beneficial for several tasks, and then additional computation is required to update the model based on the newly acquired supervision. In our work, the benevolent approach to MTAL is exhaustive and consequently scales poorly to large number of tasks. The scalability concern typically arises from having to estimate the Bayesian updates from one task to the other tasks through the shared priors and then estimating the consequent improvement in other tasks. For the heterogeneous setting, we proposed an alternative in the form of using surrogate scores for addressing this challenge. We also studied several intuitive heuristics in the form of consistency-based and influence-based approaches for Hierarchical classification. Such approaches may or may not work for other scenarios and identifying alternative approximations could be an important future research direction.

Theoretical complexity bounds: The work on computing theoretical label complexity bounds for single-task is leading to novel insights into new algorithms and strategies for Active Learning Balcan et al. [2008]; Beygelzimer et al. [2008]. Naturally, such complexity analysis for the approaches studied in this work can lead to more insights into improving and enhancing these approaches. This line of work is particularly challenging because, in addition to the chosen Active Learning strategy, much of the complexity analysis will be tied to the underlying joint representation of tasks, further complicating matters.

Alternative surrogate scoring functions: For our work on Transferable Active Learning, we chose entropy minimization of the topic model as the criteria for instance selection. While this approach was effective for our chosen case study of genre-driven Collaborative Filtering, it may not be equally effective for others. Further exploration of domain-specific alternative surrogate scoring functions is required in this regard. Additionally, our joint learning model for heterogeneous tasks was based on a topic-modeling approach and other alternate representations might be possible. Such representations and their corresponding Transferable Active learning strategies merit

further exploration.

CoI for consistency-based Active Learning: In our work on Hierarchical Multi-Task Active learning, we demonstrated the effectiveness of choosing instances that inconsistent in the parent-child predictions. This concept can be further extended to encompass a sub-tree or the entire hierarchy leading to further options in terms of the CoI. This would be an interesting direction to explore to study the trade-offs in terms of performance benefit versus scoring time for identifying such instances.

Task-selection: Training data acquisition for each task incurs expense and it is crucial to minimize this cost by focusing on the right tasks to solicit training instances for. We presented the influence-based approach to favor selection of instances for tasks that are more influential towards other tasks. However, several other criteria might be developed for first performing task-selection. These can be broadly grouped into intrinsic and extrinsic properties. Intrinsic properties of a task refer to properties that can be computed in isolation for each task. We can select tasks based on intrinsic factors such as starvation (task with the least number of training examples), performance (weakest performing task) and uncertainty (most uncertain task over unlabeled instances). It is questionable whether selecting a reference task this way might be beneficial for the improvement of the overall system. Alternatively, we can choose extrinsic properties such as choosing the most benevolent task, or the most influential task, or the most representative task in a group of tasks. Exploring these possibilities could be an interesting research direction.

List of Figures

List of Tables

References

Abe, N. and Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceeding of International Conference on Machine Learning (ICML)*, pages 1–90.

Agarwal, A., Gerber, S., and Daumé III, H. (2010). Learning multiple tasks using manifold regularization. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Ando, R. K., Zhang, T., and Bartlett, P. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2008). A spectral regularization framework for multi-task structure learning. In *NIPS*.

Balcan, M.-F., Hanneke, S., and Wortman, J. (2008). The true sample complexity of active learning. In Servedio, R. A. and Zhang, T., editors, *COLT*, pages 45–56. Omnipress.

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198.

Ben-david, S. and Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *Proceedings of Computational Learning Theory (COLT)*.

Bertsekas, D. P. (1996). *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*. Athena Scientific, 1 edition.

Beygelzimer, A., Dasgupta, S., and Langford, J. (2008). Importance weighted active learning. *CoRR*, abs/0812.4952.

Beygelzimer, A., Hsu, D., Langford, J., and Zhang, T. (2010). Agnostic active learning without constraints. *CoRR*, abs/1006.2588.

Blei, D. M. and Jordan, M. I. (2005). Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory (COLT)*, pages 92–100. Morgan Kaufmann Publishers.

Bottou, L. (1991). Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France. EC2.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Lechevallier, Y. and Saporta, G., editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France. Springer.

Boutilier, C., Zemel, R. S., and Marlin, B. (2003). Active collaborative filtering. In *Proceedings of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 98–106.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Technical Report MSR-TR-98-12*, pages 43–52. Morgan Kaufmann.

Breiman, L. and Friedman, J. (1997). Predicting multivariate responses in multiple linear regression. In *Journal of Royal Statistical Society B*, pages 59(1):3–54.

Cai, L. and Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM '04), November 8-13, 2004, Washington, D.C., USA*. ACM Press, New York, NY, USA.

Caruana, R. (1997). Multitask learning. In *Machine Learning*, pages 41–75.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. (2006). Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research*, 7:1205–1230.

Cesa-bianchi, N. and Zaniboni, L. (2006). Hierarchical classification: Combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 177–184.

Chen, H. and Dumais, S. T. (2000). Bringing order to the web: automatically categorizing search results. In Turner, T. and Szwillus, G., editors, *CHI*, pages 145–152. ACM.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1995). Active learning with statistical models. In *Journal of Artificial Intelligence Research (JAIR)*.

Das, A., D. M. and Garg, A. (2007). Google news personalization: Scalable online collaborative filtering. In *Proceedings of ACM SIGIR Conference on World Wide Web (WWW)*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Dimitrovski, I., Kocev, D., Loskovska, S., and Džeroski, S. (2008). Hierchical annotation of medical images. In *Proceedings of the 11th International Multiconference - Information Society IS 2008*, pages 174–181. IJS, Ljubljana.

Donmez, P. and Carbonell, J. (2008a). Proactive learning: Towards learning with multiple imperfect predictors. In *CIKM*.

Donmez, P., Carbonell, J., and Schneider, J. (2009). Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, page 259Ű268.

Donmez, P. and Carbonell, J. G. (2008b). Optimizing estimated loss reduction for active sampling in rank learning. In *ICML*.

Donmez, P., Carbonell, J. G., and Bennett, P. N. (2007). Dual strategy active learning. In *Proceedings of the European Conference on Machine Learning (ECML)*.

Dumais, S. and Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 256–263, New York, NY, USA. ACM.

Esuli, A. and Sebastiani, F. (2009). Active learning strategies for multi-label text classification. In *Advances in Information Retrieval (ECIR)*.

Evgeniou, T., Micchelli, C. A., and Pontil, M. (2005). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637.

Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168.

Godbole, S., Harpale, A., Sarawagi, S., and Chakrabarti, S. (2004). Document classification through interactive supervision of document and term labels. In *ECML PKDD-04*, pages 185–196.

Guo, Y. and Greiner, R. (2007). Optimistic active learning using mutual information. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, page 823Ű829.

Harpale, A. and Yang, Y. (2010). Active learning for multi-task adaptive filtering. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

Harpale, A., Yang, Y., Gopal, S., He, D., and Yue, Z. (2010). Citedata: A new multi-faceted dataset for evaluating personalized search performance. In *CIKM*.

Hofmann, T. (2003). Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 688–693.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining (ICDM 2008)*, pages 263–272.

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

Jaakkola, T. S. and Jordan, M. I. (1996). A variational approach to bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*.

Jacob, L. and Bach, F. (2008). Clustered multi-task learning: a convex formulation. In *NIPS*.

Jebara, T. (2004). Multi-task feature and kernel selection for svms. In *Proc. of ICML 2004*.

Jian Zhang, Z. G. and Yang, Y. (2005). Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems (NIPS)*.

Jin, R. and Si, L. (2004). A bayesian approach toward active learning for collaborative filtering. In *Uncertainty in Artificial Intelligence (UAI)*.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, pages 143–151.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 200–209. Morgan Kaufmann.

Kapoor, A., Horvitz, E., and Basu, S. (2007). Selective supervision: Guiding supervised learning with decision theoretic active. In *International Joint Conference on Artificial Intelligence*, pages 877–882.

Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. *Machine Learning: ECML 2004*, Volume 3201/2004:217–226.

Lawrence, N. D. and Platt, J. C. (2004). Learning to learn with the informative vector machine. In *Proceedings of the International Conference in Machine Learning*. Morgan Kaufmann.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. Springer-Verlag.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004a). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Lewis, D. D., Yang, Y., Rose, T. G., Li, F., Dietterich, G., and Li, F. (2004b). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Liu, Q., Liao, X., and Carin, L. (2009). Semi-supervised multitask learning. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, T., Yang, Y., Wan, H., jun Zeng, H., Chen, Z., and ying Ma, W. (2005a). Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations*, 7:2005.

Liu, T.-Y., Yang, Y., Wan, H., Zhou, Q., Gao, B., Zeng, H.-J., Chen, Z., and Ma, W.-Y. (2005b). An experimental study on large-scale web categorization. In *Special interest*

tracks and posters of the 14th international conference on World Wide Web, WWW '05, pages 1106–1107, New York, NY, USA. ACM.

MacKay, D. J. (1992). The evidence framework applied to classification networks. *Neural Computation*, 4:720–736.

Mccallum, A. and Nigam, K. (1998). Employing em in pool-based active learning for text classification. In *Proceedings of International Conference on Machine Learning (ICML)*.

Melville, P. and Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004*, pages 584–591.

Melville, P. and Provost, F. (2005). Economical active feature-value acquisition through expected utility estimation. In *Proceedings of the KDD05 Workshop on Utility-Based Data Mining*, pages 10–16.

Melville, P., Saar-Tsechansky, M., Provost, F., and Mooney, R. (2004). Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM)*, pages 483–486. IEEE Computer Society.

Micchelli, C. A. and Pontil, M. (2004). Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 109–117.

Minka, T. P. and Picard, R. W. (1997). Learning how to learn is learning with point sets. In *Web. Revised 1999, available at http://www.stat.cmu.edu/ minka/*.

Muslea, I., Minton, S., and Knoblock, C. A. (2002). Active semi-supervised learning = robust multi-view learning. In *Proceedings of ICML-02, 19th International Conference on Machine Learning*, pages 435–442.

Punera, K. and Ghosh, J. (2008). Enhanced hierarchical classification via isotonic smoothing. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 151–160, New York, NY, USA. ACM.

Reichart, R., Tomanek, K., and Hahn, U. (2008). Multi-task active learning for linguistic annotations. In *In ACL*.

Robertson, S. and Soboroff, I. (2002). The trec 2002 filtering track report. In *Text Retrieval Conference*.

Rose, J. R. and Eastman, C. M. (1997). Hierarchical classification as an aid to browsing. *Informatica*, pages 49–57.

Roth, D. and Small, K. (2006). Active learning with perceptron for structured output. In *ICML-06 Workshop on Learning in Structured Output Spaces*.

Roy, N. and Mccallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann.

Saar-tsechansky, M. and Provost, F. (2004). Active sampling for class probability estimation and ranking. In *Machine Learning*, pages 153–178.

Saha, A., Rai, P., III, H. D., and Venkatasubramanian, S. (2010). Active online multitask learning. In *Budgeted Learning Workshop, ICML*, Haifa, Israel.

Salakhutdinov, R. and Mnih, A. (2008a). Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

Salakhutdinov, R. and Mnih, A. (2008b). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the International Conference on Machine Learning*, volume 25.

Sarawagi, S. and Bhamidipaty, A. (2002). Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, pages 269–278.

Seeger, M., whye Teh, Y., and Jordan, M. I. (2004). Semiparametric latent factor models. Technical report, Workshop on Artificial Intelligence and Statistics 10.

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Settles, B., Craven, M., and Friedl, L. (2008). Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Computational Learning Theory*.

Shahbaba, B. and Neal, R. M. (2007). Improving classification when a class hierarchy is available using a hierarchy-based prior. *Bayesian Anal.*, 2(1):221–237.

Si, L. and Jin, R. (2003). Flexible mixture model for collaborative filtering. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 704–711. AAAI Press.

Silver, D. and Mercer, R. (2001). Selective functional transfer: Inductive bias from related tasks. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2001)*, pages 182–189.

Singh, M., Curran, E., and Cunningham, P. (2009). Active learning for multi-label image annotation. Technical report, UCD School of Computer Science and Informatics.

Sun, A. and Lim, E.-P. (2001). Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 521–528, Washington, DC, USA. IEEE Computer Society.

Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems*, pages 640–646. The MIT Press.

Tong, S. and Koller, D. (2000). Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, pages 999–1006.

Tong, S. and Koller, D. (2001a). Active learning for parameter estimation in bayesian networks. In *NIPS*, pages 647–653.

Tong, S. and Koller, D. (2001b). Active learning for parameter estimation in bayesian networks. In *Neural Information Processing Systems (NIPS)*, pages 647–653.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. In *International Journal of Data Warehousing and Mining*.

Tur, G., Hakkani-tür, D., and Schapire, R. E. (2005). Combining active and semi-supervised learning for spoken language understanding. In *Speech Communication*.

Wang, C. and Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In Apté, C., Ghosh, J., and Smyth, P., editors, *KDD*, pages 448–456. ACM.

Xu, Z., Yu, K., Tresp, V., Xu, X., and Wang, J. (2003). Representative sampling for text classification using support vector machines. In *ECIR*.

Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:2007.

Yang, B., Sun, J.-T., Wang, T., and Chen, Z. (2009a). Effective multi-label active learning for text classification. In *The 15th ACM SIGKDD Conference On Knowledge Discovery and Data Mining (KDD)*.

Yang, J. M., Li, K. F., and Zhang, D. F. (2008). Adaptive collaborative filtering based on user-genre-item relation. *Int. J. Commun. Netw. Distrib. Syst.*, 1:216–230.

Yang, L., Hanneke, S., and Carbonell, J. (2012). A theory of transfer learning with applications to active learning.

Yang, X., Kim, S., and Xing, E. P. (2009b). Heterogeneous multitask learning with joint sparsity constraints. In *Advances in Neural Information Processing Systems (NIPS)*.

Yang, Y. and Gopal, S. (2012). Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning*, 88(1-2):47–68.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Yu, K., Tresp, V., and Schwaighofer, A. (2005). Learning gaussian processes from multiple tasks. In *Proceedings of 22nd International Conference on Machine Learning (ICML)*, pages 1012–1019.

Yu, K., Tresp, V., and Yu, S. (2004). A nonparametric hierarchical bayesian framework for information filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*.

Zhang, J., Ghahramani, Z., and Yang, Y. (2008). Flexible latent variable models for multi-task learning. In *Machine Learning*.

Zhang, Y. (2010). Multi-task active learning with output constraints. In *AAAI*.

Zhang, Y., Xu, W., and Callan, J. (2003). Exploration and exploitation in adaptive filtering based on. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003*, pages 896–903.

Zhu, X., Lafferty, J., and Ghahramani, Z. (2003). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65.

Zimek, A., Buchwald, F., Frank, E., and Kramer, S. (2010). A study of hierarchical and flat classification of proteins. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 7(3):563–571.