CoBayes: Bayesian Knowledge Corroboration with Assessors of Unknown Areas of Expertise

Gjergji Kasneci
Microsoft Research
7 J J Thomson Avenue
Cambridge, UK
gjergjik@microsoft.com

Jurgen Van Gael
Microsoft Research
7 J J Thomson Avenue
Cambridge, UK
jvangael@microsoft.com

David Stern
Microsoft Research
7 J J Thomson Avenue
Cambridge, UK
dstern@microsoft.com

Thore Graepel
Microsoft Research
7 J J Thomson Avenue
Cambridge, UK
thoreg@microsoft.com

ABSTRACT

Our work aims at building probabilistic tools for constructing and maintaining large-scale knowledge bases containing entity-relationship-entity triples (statements) extracted from the Web. In order to mitigate the uncertainty inherent in information extraction and integration we propose leveraging the "wisdom of the crowds" by aggregating truth assessments that users provide about statements. The suggested method, CoBayes, operates on a collection of statements, a set of deduction rules (e.g. transitivity), a set of users, and a set of truth assessments of users about statements. We propose a joint probabilistic model of the truth values of statements and the expertise of users for assessing statements. The truth values of statements are interconnected through derivations based on the deduction rules. The correctness of a user's assessment for a given statement is modeled by linear mappings from user descriptions and statement descriptions into a common latent knowledge space where the inner product between user and statement vectors determines the probability that the user assessment for that statement will be correct. Bayesian inference in this complex graphical model is performed using mixed variational and expectation propagation message passing. We demonstrate the viability of CoBayes in comparison to other approaches, on real-world datasets and user feedback collected from Amazon Mechanical Turk.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Reliability, Algorithms, Experimentation

Keywords

Knowledge, User, Feedback, Expertise, Bayesian, Model

1. INTRODUCTION

In recent years, there have been several projects aiming at extracting and organizing Web content into large-scale knowledge bases [7, 6, 3]. The majority of these knowledge bases build on the Semantic Web formalism of the *Resource Description Framework Schema* (RDFS) [1], a schema language for knowledge representation. The popularity of RDFS can be explained by its flexibility and its practical logical reasoning capabilities, including reasoning over properties of relationships (e.g., reflexivity, transitivity, domain, and range). However, the framework is missing a crucial ingredient: it does not allow the representation and reasoning with uncertainty, which may come from various sources:

Extraction & Integration Uncertainty Information extraction and integration use inherently noisy techniques such as natural language processing, pattern matching, statistical learning, etc.

Information Source Uncertainty Web sources may be unreliable, non-authoritative or even deliberately misleading.

Inherent Knowledge Uncertainty Knowledge itself is often inherently uncertain; e.g., nobody knows when exactly Plato was born.

Consistency checking in uncertain knowledge is a very difficult task, and in fact, sound knowledge curation and provenance are considered to be major challenges by the community of probabilistic databases [8]. We argue that in order to achieve these goals we need a framework for information extraction, integration, corroboration, querying, and inference which has uncertainty management built in as a first-class citizen. As a first step towards these goals, in this paper, we demonstrate that the "wisdom of the crowds", as represented by user assessments, can be leveraged to quantify the uncertainty in state-of-the-art knowledge bases, while taking into account the specific areas of expertise of the human assessors.

The scenario we address is the following. Let us assume that m users $u_1, ..., u_m$ give feedback on n statements $f_1, ..., f_n$ from a knowledge base. The statements may depend on each other through logical deduction rules (e.g., such as the ones provided by RDFS). For simplicity, let us assume that the feedback will be in form of a truth assessment, i.e., users will say whether a statement is true or false. The underlying assumption of this work is that in general users will tend to report the truth. The goal then would be to exploit the feedback and the logical dependencies among statements in order to learn both the truth values of the statements and the reliabilities of the users. However, as the background knowledge of users may vary across knowledge domains, in some cases, it may be crucial to give preference to the expertise of specific users rather than to the "wisdom of the crowds". For example, the majority of the people from the "crowd" may not know that Barack Obama has won a Grammy Award. In such a case, it would be important to automatically identify the few experts in the "crowd" who may know the truth. This example also highlights the problem of majority voting techniques (in which the correct label is determined by the majority). One could address this problem by introducing a weight for the expertise of each user. But how could the performance of an expert be measured when there is no gold standard available?

1.1 Contributions and Outline

This paper presents a system, coined CoBayes. CoBayes exploits user feedback and logical deduction rules in a Bayesian corroboration process that jointly learns the truth values of knowledge fragments (i.e., statements) and the trustworthiness of users. As users may often act inconsistently or unreliably, and give inaccurate feedback across knowledge domains, the joint inference mechanism learns the latent affinity between users' expertise and statements by taking user and statement features into account. This is achieved by mapping users and statements into a common latent knowledge space. Finally, the logical deduction rules interconnect the statements under assessment and propagate the truth values thus mitigating feedback sparsity. CoBayes is implemented as a modular system. Each of the modules comes with its own Bayesian inference algorithm. Their powerful composition is achieved through efficient, approximate message passing. The different configurations of CoBayes are carefully evaluated in this paper. Our experimental evaluation on real-world datasets and feedback from Amazon Mechanical Turk demonstrates the system's viability.

The paper is organized as follows. Section 2 gives an overview of related work. In Section 3, we introduce our knowledge representation formalism and its abstraction into a Bayesian network. Section 4, explains the different components of CoBayes as well as their interaction. We present the experimental evaluation of our model in Section 5 and conclude in Section 6.

2. RELATED WORK

The knowledge corroboration problem has previously been addressed in various contexts, such as user preferences, reliabilities, or authorities. [11, 12, 13, 14]. For example, Dawid et al. [11] propose an EM approach to estimating the error rates of patients with respect to yes-no classification of medical symptoms. For a given a list of symptoms, the patients (who are known to have a certain disease) identify and mark the symptoms they have. Based on the true symptoms of the disease, the EM algorithm can estimate error rates of the patients. Our work is more general in that the medical symptoms could be represented as features, and the trustworthiness (or, analogously, the error rates) of patients could be estimated through the Bayesian corroboration process. In general, we are interested in a joint corroboration process that can learn the truth values of knowledge fragments and the trustworthiness of users who give feedback. From this point of view, more related to our approach is the work presented in [15, 16, 17]. [15] presents three probabilistic fix-point algorithms for aggregating disagreeing views about statements and learning their truth values as well as the trust in the views. However, as admitted by the authors, their algorithms cannot be used in an online fashion, while our approach builds on a Bayesian framework and is inherently flexible to online updates. Furthermore, [15] does not deal with the problem of logical inference, which is a core ingredient of our approach. Neither does it consider the issue of inconsistent user performance across knowledge domains. A very recent article [16] proposes a supervised learning approach to the above problem. In contrast to our approach, the solution proposed in [16] is not fully Bayesian and does not deal with logical deduction rules. In general, our work distinguishes itself from prior work in this realm by dealing with uncertainty on top of the practically viable knowledge representation formalism of RDFS, which could be exploited by the Semantic Web community to integrate uncertainty as a first-class citizen into its formalisms. This work extends the approach of [17], which presents a family of Bayesian models for jointly learning the trustworthiness of users and truth values for statements in the presence of disagreeing user opinions and logical deduction rules. However, none of the presented models provide an expertise model for capturing the latent affinity between users' expertise and statements. We argue that a principled expertise model is very important, as without it the "ignorance of the crowds" could prevail.

3. KNOWLEDGE REPRESENTATION

3.1 From RDFS to RDFS#

The most popular Semantic-Web formalism for knowledge representation is the *Resource Description Framework Schema* (RDFS). RDFS allows the specification of a common syntax for data exchange. It builds on the entity-relationship (ER) formalism and enables the definition of domain resources (i.e., entities), such as individuals (e.g. *AlbertEinstein, NobelPrize, Germany,* etc.), classes (e.g. *Physicist, Prize, Location,* etc.) and relationships (or so-called properties, e.g. *type, hasWon, locatedIn*, etc.). Table 1 depicts the correspondence of ER and RDFS terminology.

The basis of RDFS is RDF which comes with three basic symbols: URIs (Uniform Resource Identifiers) for uniquely addressing resources, literals for representing values such as strings, numbers, dates, etc., and blank nodes for representing unknown or unimportant resources. Another

ER term	RDFS term
entity	resource
relationship (type)	property
relationship instance / fact	statement / RDF triple / fact

Table 1: Correspondence of ER and RDFS terminology.

important RDF construct for expressing that two entities stand in a binary relationship is a statement. A statement is a triple of URIs and has the form <*Subject*, *Predicate*, *Object*>, for example <*AlbertEinstein*, *bornIn*, *Ulm*>. An RDF statement can be thought of as an edge from an ER graph, where the *Subject* and the *Object* represent entity nodes and the *Predicate* represents the relationship label of the corresponding edge. Consequently, a set of RDF statements can be viewed as an ER graph. RDFS extends the set of RDF symbols by new URIs for predefined class and relation types such as *rdfs:Resource* (the class of all resources), *rdfs:subClassOf* (for representing the subclass-class relationship), etc. One of the strengths of RDFS is that it allows light-weight logical reasoning over the represented knowledge. For example, it allows the inference of the types of entities through the domains or ranges of relationships they occur in. Furthermore, RDFS enables reasoning over the reflexivity and transitivity of the relationships.

However, in the current specification of RDFS, reasoning over transitivity is defined only for *rdfs:subClassOf* and *rdfs:subPropertyOf*. This is too restrictive, as there are many more useful transitive relationships, such as *locatedIn*, *influences*, *partOf*, *ancestorOf*, etc. Furthermore, the RDFS formalism provides no means for representing uncertainty. The more expressive Web Ontology Language (OWL) [2], which builds on RDFS, allows the above properties to be defined for arbitrary relationships, but its expressive power makes consistency checking undecidable. The recently introduced YAGO model [7] permits the definition of arbitrary acyclic transitive relationships but has the advantage that it still remains decidable. Being able to define transitivity for arbitrary relationships can be a very useful feature for ontological models, since many practically relevant relationships, such as *isA, locatedIn, containedIn, partOf, ancestorOf, siblingOf*, etc., are transitive. Hence, we introduce a slightly different variant of RDFS that can represent the uncertainty of statements and reason about any transitive relationship.

DEFINITION 1 (RDFS#). *RDFS# [1] is the RDFS model, in which each statement f is assigned a probabilistic value $p(f)$, the blank nodes are forbidden, and the reasoning capabilities are derived from the following deductive rules. For all $X, Y, Z \in Ent, R, R' \in Rel$ with $X \neq Y, Y \neq Z, X \neq Z, R \neq R'$:*

1. $<X, \mathit{type}, Y> \wedge <Y, \mathit{subClassOf}, Z>$
   $\rightarrow <X, \mathit{type}, Z>$
2. $<X, R, Y> \wedge <Y, R, Z> \wedge <R, \mathit{type}, \mathit{TransitiveRel}>$
   $\rightarrow <X, R, Z>$
3. $<R, \mathit{subPropertyOf}, R'> \wedge <X, R, Y>$
   $\rightarrow <X, R', Y>$
4. $<R, \mathit{hasDomain}, \mathit{Dom}> \wedge <X, R, Y>$
   $\rightarrow <X, \mathit{type}, \mathit{Dom}>$
5. $<R, \mathit{hasRange}, \mathit{Ran}> \wedge <X, R, Y>$
   $\rightarrow <Y, \mathit{type}, \mathit{Ran}>$

It can be shown (by a straight-forward extension of the proof of tractability for RDFS entailment, when blank nodes are forbidden [5]) that the deductive closure of any RDFS# knowledge base can be constructed in polynomial time in the size of the knowledge base.

The problem setting is as follows. Consider an RDFS# knowledge base $\mathcal{K}$ with statements $f_1, f_2, ..., f_n$. Note that the deductive rules (as described in Definition 1) provide logical dependencies among the statements' truth values. Furthermore, consider users $u_1, u_2, ..., u_m$ who give feedback on the statements. Given descriptive user and statement features, we are interested in jointly learning the truth values of the statements and the expertise of the users by leveraging the logical dependencies among statements and the latent affinities between statements and users.

3.2 Graphical Model for Inference in RDFS#

An RDFS# knowledge base $\mathcal{K}$ in which $p(f) = 1$, for each statement f, is consistent when there is no cycle along deduction paths, or in other words, when no statement of the form $<X, R, X>$ (for any $X \in Ent$) can be derived by grounding the above rules. This is what we denote as *logical consistency*. However, when $p(f) \neq 1$ for some statements f in $\mathcal{K}$, logical consistency is no longer defined and we need an alternative notion of *probabilistic consistency*, in which case the deduction rules are viewed as soft constraints.

Let us first consider the purely logical case. Let c be a statement in $\mathcal{K}$ and let

$$(a_1 \wedge b_1) \rightarrow c, ..., (a_l \wedge b_l) \rightarrow c \tag{1}$$

be all deductions of the conclusion c in $\mathcal{K}$, where the a_i and the b_i can be previously derived. The following must hold:

$$\bigwedge_{i=1}^{l} ((a_i \wedge b_i) \rightarrow c) \Leftrightarrow \left(\bigvee_{i=1}^{l} \underbrace{(a_i \wedge b_i)}_{ab_i}\right) \rightarrow c \Leftrightarrow \left(\bigvee_{i=1}^{l} ab_i\right) \rightarrow c$$

$$\Leftrightarrow \left(\bigvee_{i=1}^{l} ab_i\right) \vee d \leftrightarrow c \tag{2}$$

where d represents the missing evidence, i.e., all missing deductions that could lead to c and only c. Note that this semantic interpretation of the variable d makes the equivalence $\left(\bigvee_{i=1}^{l} ab_i\right) \vee d \leftrightarrow c$ possible.

Now we can turn the logical formula into a Bayesian network using deterministic conditional probability tables (CPTs) that represent the logical relationships. Figure 1 depicts the corresponding directed graphical model with

[1] Read: RDFS sharp.

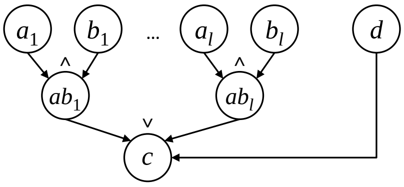

Figure 1: A graphical model illustrating the logical derivation for the formula $c = (a_1 \wedge b_1) \vee ... \vee (a_l \wedge b_l) \vee d$. Deterministic CPTs representing AND gates are marked as $\wedge$ and those representing OR gates as $\vee$.

additional auxiliary variables ab_i representing pairwise conjunctions.

The conditional probability at a node ab_i is given by:

$$P(ab_i = T|a_i, b_i) = \begin{cases} 1 & \text{if } a_i \wedge b_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This simplifies our disjunctive normal form to the expression $c = ab_1 \vee ... \vee ab_l \vee d$. Finally, we connect c with all the variables in the disjunctive normal form by a conditional probability:

$$P(c = T|ab_1, ..., ab_l, d) = \begin{cases} 1 & \text{if } ab_1 \vee ... \vee ab_l \vee d \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

4. THE COBAYES MODEL

The goal of the CoBayes model is to use assessments $a_{ij} \in \{T, F\}$ that users i make about statements j in order to infer truth values $t_j \in \{T, F\}$ of statements j. The model consists of three interacting model components:

1. An *assessment model* which relates an assessment a_{ij} with the truth value t_j of the statement, the correctness u_{ij} of user i's assessment, and the guessing probability q of users. This model is based on the assumption that users will tend to make correct assessments.

2. A *logical model* which describes the dependency between the truth value t_j of statement j and truth values $t \in \mathcal{D}_j$ of statements from which j can be derived.

3. An *expertise model* which models the expertise $\tilde{u}_{ij}$ of user i for statement j in terms of user features $\mathbf{x}_i$ and statement features $\mathbf{y}_j$, which interact via a latent expertise space.

We denote the partially observed matrix of assessments by $\mathbf{A} \in \{T, F\}^{n \times m}$, the vector of truth values of the statements by $\mathbf{t} \in \{T, F\}^m$, and the matrix of correctness of user assessments by $\mathbf{U} \in \{T, F\}^{n \times m}$. A variable $q \sim \text{Beta}(\alpha, \beta)$ represents the probability that users will guess the correct answer. The matrix of statement-based user expertise is denoted by $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times m}$, and the parameter tuple of the user-statement expertise model by $\Theta := (r_0, \mathbf{v}, \mathbf{w}, \mathbf{V}, \mathbf{W}) \in \mathbb{R} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \times \mathbb{R}^{k \times d_x} \times \mathbb{R}^{k \times d_y}$. Furthermore, the latent user expertise vectors are denoted by $\mathbf{s}_i \in \mathbb{R}^k$ and the latent statement vectors by $\mathbf{z}_j \in \mathbb{R}^k$. These variables are jointly modeled conditional on the observed user feature vectors $\mathbf{x}_i \in \mathbb{R}^{d_x}$, statement feature vectors $\mathbf{y}_j \in \mathbb{R}^{d_y}$, logical dependencies $\mathcal{D}_j \in 2^{\mathcal{K} \setminus t_j}$ and prior probabilities of truth $\pi_j \in [0, 1]$, as expressed by the following joint probability density:

$$p(\mathbf{t}, \mathbf{U}, \tilde{\mathbf{U}}, q, \{\mathbf{s}_i\}_{i=1}^n, \{\mathbf{z}_j\}_{j=1}^m, \Theta|\Omega, \Sigma) =$$

$$\underbrace{p(\mathbf{A}, \mathbf{U}, q|\mathbf{t}, \tilde{\mathbf{U}}, \alpha, \beta)}_{\text{Assessment Model}} \times \underbrace{p(\mathbf{t}|\{\mathcal{D}_j\}_{j=1}^m, \{\pi_j\}_{j=1}^m)}_{\text{Logical Model}} \times$$

$$\underbrace{p(\tilde{\mathbf{U}}, \{\mathbf{s}_i\}_{i=1}^n, \{\mathbf{z}_j\}_{j=1}^m, \Theta|\{\mathbf{x}_i\}_{i=1}^n, \{\mathbf{y}_j\}_{j=1}^m, \Sigma)}_{\text{Expertise Model}}, \tag{5}$$

where $\Omega = \{\{\mathcal{D}_j\}_{j=1}^m, \{\pi_j\}_{j=1}^m, \{\mathbf{x}_i\}_{i=1}^n, \{\mathbf{y}_j\}_{j=1}^m, \alpha, \beta\}$ represents the parameters of the prior distributions over the components of the expertise model Θ are jointly denoted by Σ. A complete reference of our notation is given in the Appendix, in Table 4.

4.1 Logical Model

Each statement f_j in $\mathcal{K}$ is assigned a binary variable $t_j \in \{T, F\}$. In CoBayes, each truth value t_j of a statement that can be logically deduced by a set of premises $\mathcal{D}_j$ from $\mathcal{K}$ is connected with the truth values of statements in $\mathcal{D}_j$ through $p(t_j|\mathcal{D}_j, \tilde{t}_j)$ as described in the previous section. Each statement f_j for which there exist no premises in $\mathcal{K}$ is assigned $p(\tilde{t}_j) := \text{Bernoulli}(\pi_j)$ as a prior, where the binary variable $\tilde{t}_j$ accounts for the deduction through missing premises. Defining $p(t_j|\tilde{t}_j, \mathcal{D}_j, \pi_j) := \sum_{\tilde{t_j} \in \{T,F\}} p(t_j|\tilde{t}_j, \mathcal{D}_j) p(\tilde{t}_j|\pi_j)$ the ”prior” distribution for $\mathbf{t}$ factorizes as

$$p(\mathbf{t}|\{\mathcal{D}_j\}_{j=1}^m, \{\pi_j\}_{j=1}^m) = \prod_{j=1}^m p(t_j|\tilde{t}_j, \mathcal{D}_j, \pi_j), \tag{6}$$

where Equations (3) and (4) specify the conditional distribution $p(t_j|\tilde{t}_j, \mathcal{D}_j)$.

4.2 Assessment Model

Two of the assessment models presented in [16, 17] are based on the simple idea that the user is going to report the true truth value of a statement with probability p and the opposite with probability $1 - p$, where p represents the reliability value of the user. Such a model is too restrictive, as it assumes that the user is always going to report the opposite truth value with probability $1 - p$. Our model is more flexible in that it can capture cases in which users may guess the correct answer. In our model, as shown in the conditional probability table (CPT) of Figure 2, when user i assesses statement j, he will report the true truth value of j if $u_{ij} = T$, that is, if he knows j's truth value. Otherwise, with probability $q \sim \text{Beta}(\alpha, \beta)$ the user will guess the true truth value of j and with probability $1 - q$ he will guess the opposite truth value, where the α and β values are learned. Consider the set $\{a_{ij}\}$ of observed true/false feedback labels for statement-user pairs. The joint probability distribution for assessments $\mathbf{A}$ and correctness

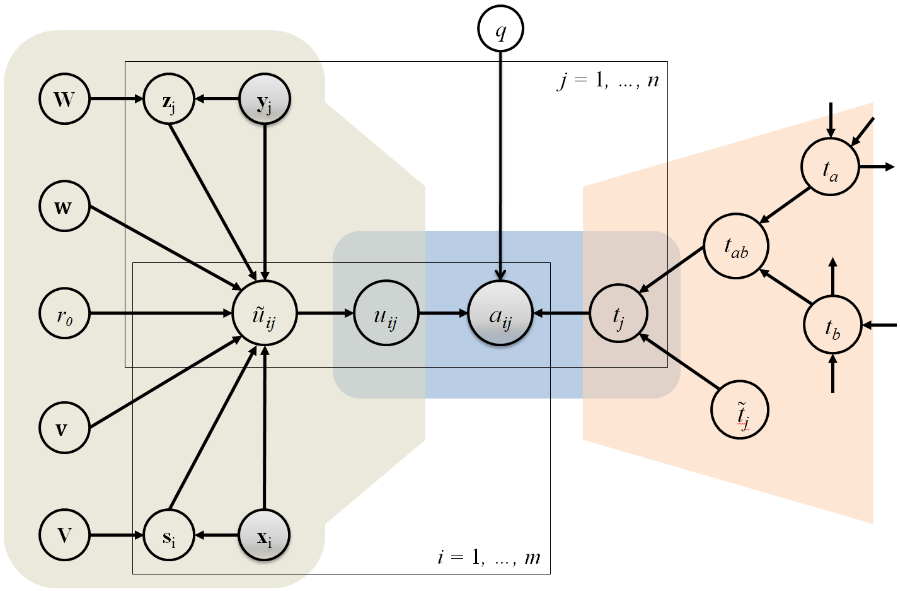

Figure 2: The graphical model of CoBayes. The subgraph highlighted on the left represents the Expertise Model, the one in the middle the Assessment Model, and the one on the right the Logical Model. The shaded nodes are observed; the remaining ones are latent variables. We are interested in the marginals of user expertise $\tilde{u}_{ij}$ and correctness u_{ij}, as well as the statement truths t_j. For visualization purposes it is also interesting to infer the knowledge space embeddings of $\mathbf{z}_j$ and $\mathbf{s}_i$.

t_j a_{ij}	T	F
T	$[\![u_{ij}]\!] + q(1 - [\![u_{ij}]\!])$	$(1 - q)(1 - [\![u_{ij}]\!])$
F	$(1 - q)(1 - [\![u_{ij}]\!])$	$[\![u_{ij}]\!] + q(1 - [\![u_{ij}]\!])$

Table 2: The deterministic conditional probability distribution $p(a_{ij}|t_j, u_{ij}, q)$ for feedback signal a_{ij} given assessment correctness u_{ij} and truth t_j. $[\![u_{ij}]\!]$ is 1 if $u_{ij} = T$ and 0 otherwise.

values $\mathbf{U}$ given truth values $\mathbf{t} \in \{T, F\}^n$ is

$$p(\mathbf{A}, \mathbf{U}, q|\mathbf{t}, \tilde{\mathbf{U}}, \alpha, \beta) = \prod_{i=1}^{n} \prod_{j=1}^{m} p(a_{ij}|t_j, u_{ij}, q) p(u_{ij}|\tilde{u}_{ij}) p(q|\alpha, \beta), \tag{7}$$

where $\tilde{\mathbf{U}}$ holds the corresponding prior parameters for the components of $\mathbf{U}$.

Figure 3 depicts the user feedback model. The CPT for $p(a_{ij}|t_j, u_{ij}, q)$ is depicted in Figure 2. The function $[\![u_{ij}]\!]$ maps T and F to 1 and 0, respectively.

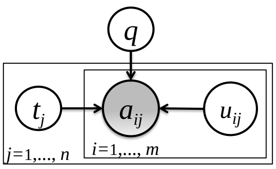

Figure 3: The graphical model for feedback signal a_{ij}, assessment correctness u_{ij}, and truth t_j.

The variables $\tilde{u}_{ij} \in \{T, F\}$ represent the expertise of user i for a specific statement j. In previous work [17], these have been modeled independently of the statement j as a general reliability $u_{ij} = u_i$ of user i. Instead, we are now proposing to model the specific areas of expertise of a user through a sub model similar to a recommender system. Note that in the above model, due to the assumed independence of assessments, the agreement of assessments acts as a truth amplifier.

4.3 Expertise Model

In general, different users specialize in good assessments for particular groups of statements (i.e., for particular knowledge domains). The expertise model takes the form of a recommendation system like Matchbox [22]. However, instead of explicit user-statement ratings, the predictions r_{ij} feed via the expertise variables $\tilde{u}_{ij}$ into the correctness variables u_{ij}. In algebraic terms, the expertise model from (5) factorizes as

$$p(\tilde{\mathbf{U}}, \{\mathbf{s}_i\}_{i=1}^{n}, \{\mathbf{z}_j\}_{j=1}^{m}, \Theta|\{\mathbf{x}_i\}_{i=1}^{n}, \{\mathbf{y}_j\}_{j=1}^{m}, \Sigma) = \prod_{i=1}^{n} \prod_{j=1}^{m} \mathcal{N}(\tilde{u}_{ij}; \mathbf{s}_i^T \mathbf{z}_j + \mathbf{x}_i^T \mathbf{v} + \mathbf{y}_j^T \mathbf{w} + r_0, \beta^2) \times \prod_{l=1}^{k} \delta(s_{ik} - \mathbf{x}_i^T \mathbf{v}_k) \delta(z_{jk} - \mathbf{y}_j^T \mathbf{w}_k) \pi(r_0, \mathbf{v}, \mathbf{w}, \mathbf{V}, \mathbf{W}), \tag{8}$$

where $\pi(r_0, \mathbf{v}, \mathbf{w}, \mathbf{V}, \mathbf{W})$ is a fully factorizing Gaussian prior over r_0, $\mathbf{v}$, $\mathbf{w}$, $\mathbf{V}$, and $\mathbf{W}$ whose parameters are jointly denoted by Σ above. Intuitively, the model can be thought of as mapping both users $\mathbf{x}_i$ and statements $\mathbf{y}_j$ into a k-dimensional latent knowledge space, with $\mathbf{s}_i = \mathbf{V}^T \mathbf{x}_i$ and

$\mathbf{z}_j = \mathbf{W}^T \mathbf{y}_j$. The statement-dependent user expertise $\tilde{u}_{ij}$ is then modeled as the inner product $\mathbf{s}_i^T \mathbf{z}_j$ between the latent expertise vectors. In addition, purely user or statement related effects are modeled with linear models, $\mathbf{x}_i^T \mathbf{v}$ and $\mathbf{y}_j^T \mathbf{w}$, together with an overall threshold r_0.

The user and statement features, as represented by the vectors $\mathbf{x}_i$, $\mathbf{y}_j$, can be seen as characterizing descriptions that allow generalizations across users and statements. This leads to two main advantages. First, the features help mitigate the data sparsity. Second, as reported in [22], the features can be helpful when dealing with the cold-start problem, i.e., when new users join the feedback crowd or when new statements are added to the knowledge base. As we will see in the experimental section, the embedding of users and statements into a latent knowledge space remarkably improves the corroboration process.

4.4 Gaussian-to-Beta Approximation

The assessment model and the expertise model are connected by a sigmoid factor $p(u_{ij}|\tilde{u_{ij}}) = \pi = \frac{e^{\tilde{u}_{ij}}}{1+e^{\tilde{u}_{ij}}}$. Since the expertise model achieves efficient inference using a fully factorised Gaussian approximation we approximate the marginal distribution, $p(\tilde{u}_{ij})$ using a Gaussian. This is achieved by using a Laplace approximation to a Beta distribution $\mathrm{Beta}(\pi; \mathrm{a}, \mathrm{b})$ after changing the basis of the Beta distribution via the sigmoid, following [18]. For $p(\tilde{u}_{ij}) \sim \mathcal{N}(\mu, \sigma^2)$ this gives us $p(u_{ij}) = \mathrm{Ber}(u_{ij}, \frac{a}{a+b})$ where $a = \frac{e^{\mu}+1}{\sigma^2}$ and $b = \frac{e^{-\mu}+1}{\sigma^2}$.

4.5 Approximate Message Passing

We implemented the model based on the Infer.net[2] library for probabilistic inference in factor graphs. There are six types of factors in CoBayes: (1) logical factors connecting Bernoulli variables, (2) Beta-Bernoulli factors in the assessment model, (3) product factors, (4) linear combination factors, and (5) Gaussian factors in the expertise model, and (6) Gaussian-to-Beta factors for connecting the real-valued output of the expertise model with the Boolean variables of the assessment model[3]. Inference in the CoBayes model can be performed using approximate message passing based on a combination of expectation propagation (EP) [9] and variational message passing (VMP) [19]. VMP is necessary for the product factor in the inner product of the expertise model as discussed in [22], where the reader can find more details about the inference in this type of model. On the remaining part of the model (i.e., on the assessment and logical model), the inference is handled by EP. The inference in the expertise model is run based on Gaussian messages; the assessment model uses Beta and Bernoulli messages; and the logical model runs inference based on Bernoulli messages. Note that each of the models can be used as an independent module. This makes CoBayes a flexible compositional corroboration system. For the inference schedule across the modules, we start out by running inference on the expertise model and then switch iteratively between this and the remaining modules of CoBayes. The runtime complexity for the approximate message passing in CoBayes is linear in the number of user assessments. In this paper, however, the focus is on the prediction accuracy of CoBayes. A detailed investigation of CoBayes' efficiency and scalability are part of our future work.

5. EXPERIMENTAL EVALUATION

5.1 Dataset

For the empirical evaluation of the system we used the dataset of [17]. This dataset was constructed by choosing a subset of 833 interconnected statements about prominent physicists, philosophers, and politicians from the YAGO knowledge base [7]. Since the majority of statements in YAGO are correct, the dataset was extended by a subset of 271 false, but semantically meaningful statements (e.g., <*BarackObama*, *bornIn*, *Tirana*>), that were randomly generated from YAGO entities and relationships, resulting in a final set of 1,104 statements. The statements from this dataset were manually labeled as true or false, resulting in a total of 803 true statements and 301 false statements.

YAGO provides transitive relationships, such as *locatedIn, isA, influences,* etc. Hence, we are in the RDFS# setting. We computed the deductive closure of the dataset with respect to the transitive relationships. This resulted in 329 pairs of statements from which another statement in the dataset could be derived.

For the above statements, feedback labels were collected from Amazon Mechanical Turk (AMT). The users were presented with tasks of at most 5 statements each and asked to label each statement in a task with either true or false. This setup resulted in 221 AMT tasks to cover the 1,104 statements in the dataset. Additionally, the users were offered the option to use any external Web sources when assessing a statement. 111 AMT users completed between 1 and 186 tasks each. For each task the users were paid 10 US cents. At the end the total number of collected feedback labels was 11,031.

For each statement in the dataset, we use as features its relationship type (i.e., the relation label between the two entities), its topic, (i.e., physics, philosophy, politics, and general knowledge[4]), and its ID. For users we use only the ID as a feature. We also tried to collect a second dataset with user features such as age, location and gender (i.e. continent) from AMT, but controlling the quality of the collected features turned out to be quite difficult; for example, a large number of users would report that they were from Antarctica. As a result, the user features collected in the second dataset did not improve the learning process for the methods presented below. Hence, we present here the results from the first dataset in which we use only the above statement features and the user id as a user feature.

[2] Infer.net can be downloaded from `http://research.microsoft.com/en-us/um/cambridge/projects/infernet/`

[3] Note that the Beta distribution is a conjugate prior of the Bernoulli distribution

[4] e.g., <*physicist*, *isA*, *scientist*>

Approach ID	Features	# Trait Dim.
NULL	no features	0
Fact	only fact ID	0
User	only user ID	0
FactUser	fact ID and user ID	0
ALL	all fact and user features	0
MB1	all fact features except ID	1
MB2	all fact features except ID	2
MB_UF1	fact ID and user ID	1
MB_UF2	fact ID and user ID	2
MB_ALL1	all fact and user features	1
MB_ALL2	all fact and user features	2

Table 3: Different approaches resulting from different configurations of our system. The left-most column contains the ID of the approach. The middle column shows the features used by each method, and the right-most column reports the number of trait dimensions used.

5.2 Evaluated Approaches

We evaluated 11 different configurations of our system. Table 3 shows the resulting approaches, which we describe in the following paragraph.

The *NULL* approach does not take any statement or user features into account. It simply infers the posteriors of statement truths by considering the number of times a fact was labeled as true versus the number of times it was labeled as false. In this sense, the NULL approach is very similar to a majority voting approach (up to the priors used in the model). The *Fact* and *User* approaches learn truth values by considering only statement and user IDs, respectively. The *User* approach is similar to the corroboration approach presented in [17]. *FactUser* uses both, the user and the statement IDs. Finally, the approach coined *ALL* uses all available features. Note that for all approaches mentioned so far the model component which maps users and statements into a common latent expertise space is not used. This is different for the remaining models. All of them use either a one-dimensional or a two-dimensional trait space into which users and statements are mapped. We refer to the latter type of approaches as *MB approaches* (for Matchbox [22]). The *MB1* and *MB2* models use only statement features; they use all statement features except the statement ID. *MB_UF1* and *MB_UF2* use only the user and the statement IDs as features, and finally, *MB_ALL1* and *MB_ALL2* use all statement and user features.

5.3 Evaluation Measure

As a measure of accuracy for evaluating the learning methods, we choose the *normalized negative log score* (in bits). For a Bernoulli variable b_i with posterior p_i the negative log score is defined as

$$\text{nls}(p_i, b_i) := \begin{cases} -\log_2(p_i) & \text{if ground truth for } b_i \text{ is true} \\ -\log_2(1-p_i) & \text{if ground truth for } b_i \text{ is false} \end{cases}$$

The negative log score represents how much information in the ground truth is captured by the posterior derived by the corresponding learning method. More specifically, when $p_i = [\![b_i]\!]$ the negative log score is zero.

Let $p_1, ..., p_N$ be the posterior values for the Bernoulli variables $b_1, ..., b_N$. The *normalized negative log score* (NNLS) is defined as

$$\text{NNLS}(p_1, .., p_N, b_1, ..., b_N) = \frac{\sum_{i=1}^{N} \text{nls}(p_i, b_i)}{N} \qquad (9)$$

5.4 Experimental Results

The above approaches were evaluated based on two questions:

1. How well can they predict the truth values of statements?
2. How well can they predict the assessments users are going to give?

Predicting Truth Values. First, we evaluate how well the above methods predict the truth values of the statements with respect to the ground truth. For every approach the NNLS is computed for nested subsets of feedback labels. for each of the subsets, all 1,104 statements are used. Figure 4 shows the NNLS for each of the above approaches. It can be seen that the *MB* approaches have a lower NNLS, and outperform the non-*MB* approaches. Furthermore, the best performance is achieved by *MB_All2* and *MB2*, whereas, as expected, the *NULL* model performs worst. This indicates that the linear mapping of users and statements into a common latent knowledge space indeed improves the corroboration process.

In addition we compare the performance of the best-performing *MB* and non-*MB* approaches, namely *ALL* and *MB2*, when the logical deduction rules are employed. The corresponding approaches that use the logical deduction rules are denoted by *ALL T* and *MB2 T*. The results are depicted in Figure 5. It is interesting to see that the logical deduction rules already reduce the NNLS when there are no feedback labels at all. In general we observe that the deduction rules help reducing the NNLS for small subsets of feedback labels. However, when the amount of labels grows, the methods that use the deduction rules seem to become overconfident. We hypothesize that the rigidity of the rules thwarts the learning capabilities of the approaches. This happens at around 40-45% of the feedback labels (which corresponds to approximately 4 labels per statement). Interestingly, in Figure 6, where we plot the ROC curves of the *MB2 T* approach for various sparsity levels of feedback signals, we see that already with 40% of the feedback labels *MB2 T* achieves almost perfect accuracy. Furthermore, the AUC increases consistently as the number of feedback signals increases. These results are in lines with the findings of [17], where the logical deduction rules were reported to improve the learning process in the presence of fewer feedback labels.

Predicting User Assessments. In order to evaluate the predictive capabilities of the approaches with respect to the assessment that users may give, we treat the assessment

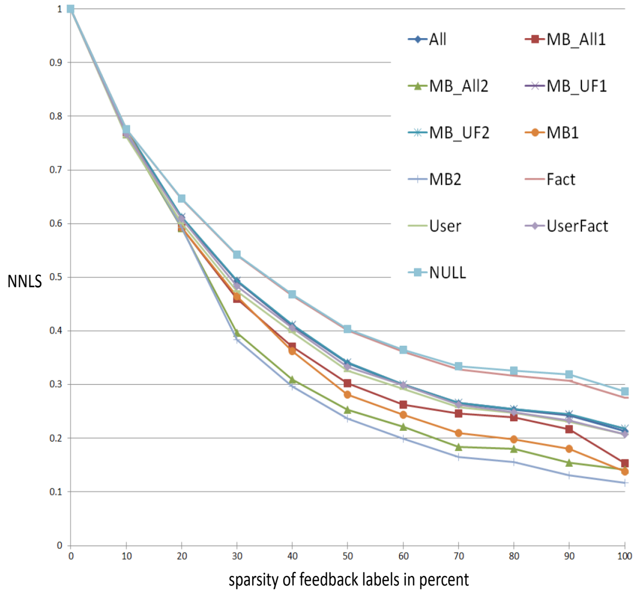

Figure 4: The NNLS for the task of predicting the truth values of statements computed on nested subsets of feedback labels. For each subset, the NNLS of each method is evaluated on all $1,104$ statements. 100% of feedback labels correspond to approximately 10 labels per statement.

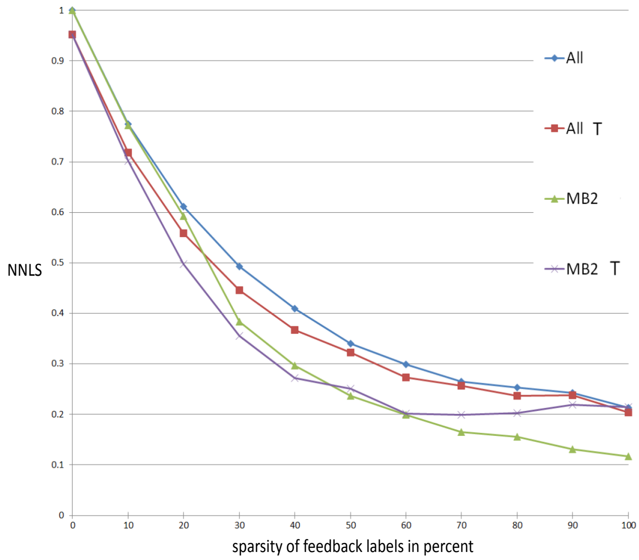

Figure 5: Comparison of *All* and *MB2* with their configurations *All T* and *MB2 T* which employ the logical deduction rules. The NNLS for each method is computed for nested subsets of feedback labels on all $1,104$ statements.

variable a_{ij} of our model as unobserved and compute their posterior probabilities. Note that this is an unambiguous evaluation since we have both, the actual assessments and our models prediction of the assessments. For each of the above approaches, the NNLS is computed with respect to the prediction of the assessments. Figure 7 depicts the results of this experiment. Again, the plots show that that the *MB* approaches achieve a lower NNLS and hence a better predictive performance than the non-*MB* approaches. As in the previous experiment, *MB_All2* and *MB2* show the best performance.

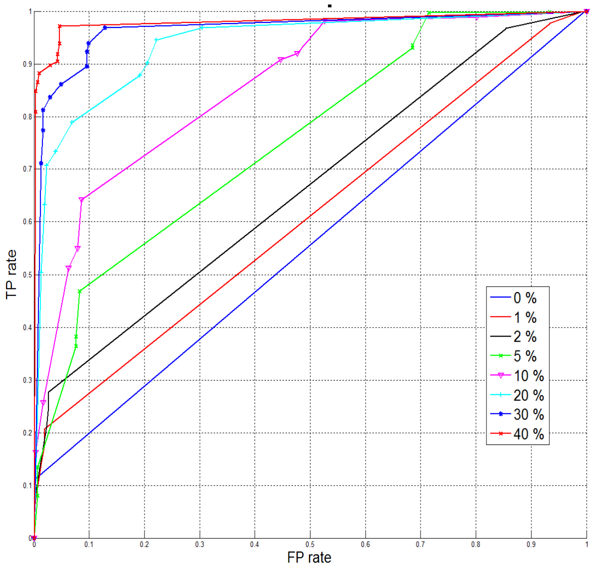

Figure 6: The ROC curves for *MB2 T* computed for different sparsity levels of feedback labels, i.e. from 0% to 40%, on all $1,104$ statements.

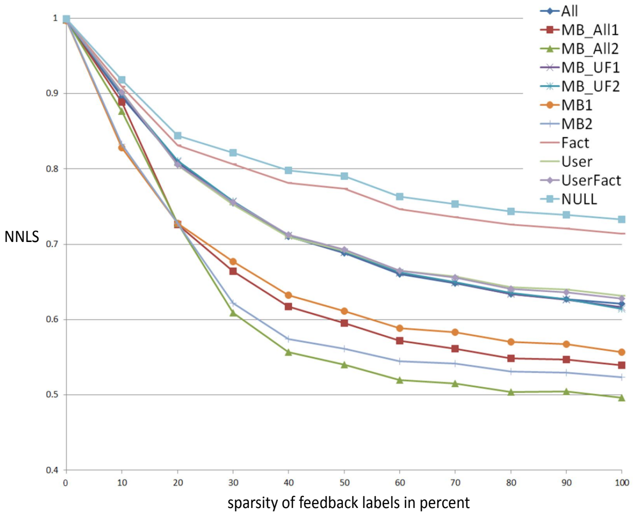

Figure 7: The NNLS for the task of predicting user assessments, computed on nested subsets of feedback labels on all $1,104$ statements.

To visualize the role of the linear mapping of users and statements into a common latent knowledge space, in Figure 8 we present the learned embedding of the users and statements into a 2-dimensional trait space. The small dots represent user traits and the bold dots represent the traits of the relationship labels for the statements in the dataset. Note that the CoBayes model employs the inner products to compute the similarity between the latent knowledge vectors of users and statements. This means that the similarity of users and statements is given by the cosine value of the angle between the corresponding vectors. For example, in Figure 8, small dots that fall in the shaded triangular area represent users who did a good job at answering historical questions such as when or where a person was born.

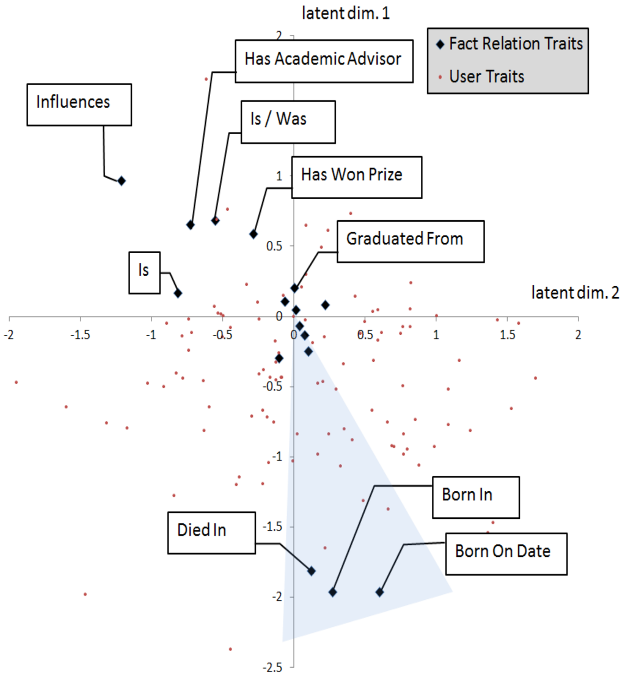

Figure 8: The latent embedding of users and statements for the *MB_All2* approach. The small dots represent user traits and the bold dots represent the traits of the relationship labels.

6. CONCLUSION

The efforts towards a more semantic Web in which knowledge is derived from content that is created and validated by users could highly benefit from sound evidence corroboration that treats uncertainty as a first-class citizen. In a joint learning process, CoBayes traces uncertainty from users to logically interdependent knowledge fragments and back again. It exploits features from users and statements to map both the users and the statements into a latent knowledge space, thus identifying the expertise of users on certain knowledge domains. Furthermore, because of its compositionality CoBayes can be used in various configurations on different corroboration tasks. We are currently investigating the extension of the CoBayes model to capture the trustworthiness of extraction tools and Web pages from which the statements were extracted. Note that this is different from the user feedback scenario, as extraction tools and Web pages give us mainly positive feedback (namely only the extracted triples). We are looking into more complex logical rules among statements for dealing with this problem. Finally, in an active learning scenario it would be important to identify the appropriate users for a given assessment task in an online fashion. The feature-based model of CoBayes offers a considerable potential for such scenarios. We are exploring active learning strategies to optimally leverage the feedback in an online fashion.

7. REFERENCES

[1] W3C: RDF Vocabulary Description Language 1.0: RDF Schema. `http://www.w3.org/TR/rdf-schema/`

[2] W3C: OWL Web Ontology Language. `http://www.w3.org/TR/owl-features/`

[3] W3C SweoIG: The Linking Open Data Community Project. `http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

[4] Infer.NET `http://research.microsoft.com/en-us/um/cambridge/projects/infernet/`

[5] Horst, H. J. T.: Completeness, Decidability and Complexity of Entailment for RDF Schema and a Semantic Extension Involving the OWL Vocabulary. In: Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 3(2–3), pp. 79–115, Elsevier (2005)

[6] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia: A Nucleus for a Web of Open Data. In: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC/ASWC 2007), pp. 722–735. Springer (2007)

[7] Suchanek, F. M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th International World Wide Web Conference (WWW 2007), pp. 697–706. ACM Press (2007)

[8] Dalvi, N. N., Ré, C., Suciu, D.: Probabilistic Databases: Diamonds in the Dirt. In: Communications of ACM, 52(7), (CACM 2009), pp. 86–94. ACM Press (2009)

[9] Minka, T. P.: A Family of Algorithms for Approximate Bayesian Inference. Massachusetts Institute of Technology (2001)

[10] Frey, B. J., Mackay, D. J. C.: A Revolution: Belief Propagation in Graphs with Cycles. In: Advances in Neural Information Processing Systems 10, pp. 479–485. MIT Press (1997)

[11] Dawid, A. P., Skene, M.: Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. In: Applied Statistics, 28(1), 1979. pp. 20–28. Blackwell Publishing (1979)

[12] Osherson, D., Vardi, M. Y.: Aggregating Disparate Estimates of Chance. In: Games and Economic Behavior, 56(1), pp. 148–173. Elsevier (2006)

[13] Jøsang, A., Marsh, S., Pope, S.: Exploring Different Types of Trust Propagation. In: 4th International Conference on Trust Management (iTrust 2006), pp: 179–192. Springer (2006)

[14] Kelly, D., Teevan, J.: Implicit Feedback for Inferring User Preference: A Bibliography. In: SIGIR Forum, 37(2), pp. 18–28. ACM Press (2003)

[15] Galland, A., Abiteboul, S., Marian, A., Senellart, P.: Corroborating Information from Disagreeing Views. In: 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010), pp. 1041–1064, ACM Press (2010)

[16] Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., Moy, L.: Learning From Crowds. In: Journal of Machine Learning Research, 11, pp. 1297–1322, MIT Press (2010)

[17] Kasneci, G., Gael, J. V., Herbrich, R., Graepel, T.: Bayesian Knowledge Corroboration with Logical Rules and User Feedback. In: ECML PKDD 2010, Springer (2010)

[18] MacKay, D. J. C.: Choice of Basis for Laplace Approximation. In: Machine Learning, 33(1), pp. 77–86, Kluwer Academic Publishers (1998)

[19] Winn, J. M., Bishop, C. M.: Variational Message Passing. In: Journal of Machine Learning Research, 6, pp. 661–694, JMLR (2005)

[20] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1997)

[21] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT press (2007)

[22] Stern, D., Herbrich, R., Graepel, T.: Matchbox: Large Scale Bayesian Recommendations. International World Wide Web Conference WWW 2009 (2009)

APPENDIX

A. NOTATION

Symbol	Meaning
$t_j \in \{T, F\}$	Truth value of statement j
$\mathcal{D}_j$	Set of truth values from which statement j can be derived
a_{ij}	Assessment of user i about statement j
$u_{ij} \in \{T, F\}$	Correctness of user i's assessment of statement j
$\tilde{u}_{ij} \in \mathbb{R}$	Expertise of user i when assessing statement j
$\mathbf{x}_i \in \mathbb{R}^{d_x}$	Feature vector describing user i
$\mathbf{y}_j \in \mathbb{R}^{d_y}$	Feature vector describing statement j
$\mathbf{s}_i \in \mathbb{R}^k$	Latent expertise vector of user i
$\mathbf{z}_j \in \mathbb{R}^k$	Latent expertise vector of statement j
$\mathbf{V} \in \mathbb{R}^{d_x} \times \mathbb{R}^k$	Linear mapping from user feature space to latent expertise space
$\mathbf{W} \in \mathbb{R}^{d_y} \times \mathbb{R}^k$	Linear mapping from statement feature space to latent expertise space
$r_{ij} \in \mathbb{R}$	Affinity of user i and statement j in latent expertise space
$\mathbf{v}_0 \in \mathbb{R}^{d_x}$	Weight vector of linear expertise model from user features
$\mathbf{w}_0 \in \mathbb{R}^{d_y}$	Weight vector of linear expertise model from statement features
$r_0 \in \mathbb{R}$	Threshold variable for expertise

Table 4: Notation