

A TrueSkill approach for movies recommendation

Laura Vanessa Cruz Quispe* and José Eduardo Ochoa Luna*†

*National University of San Agustín

Arequipa, Perú

Email: lcruzq@unsa.edu.pe

†Computer Science Department

San Pablo Catholic University

Arequipa, Perú

Email: eduardo.ol@gmail.com

Abstract—In this work a probabilistic approach based on TrueSkill for Preference Elicitation is presented. This approach allow us to tackle the “cold start” problem because relies on a content based recommendation system. In addition, it is valuable for handling high uncertainty due there is no dependency on the number of products and users. The only dependency is on ratings given by users on products. The proposal is highly scalable due to user preferences get richer as they are added.

Keywords—TrueSkill, Recommender Systems, Probabilistic Graphics Models.

I. INTRODUCCIÓN

La recomendación de información ha alcanzado un auge debido al incremento de aplicaciones prácticas tales como sistemas de recomendación de libros, música y otros productos en Amazon.com y películas en MovieLens [8], [12].

El Filtro Colaborativo (FC) es una de las técnicas más exitosas de recomendación hasta la fecha [2], [10]; se basa en el uso de datos de productos y las preferencias de usuarios para predecir un tópico o un producto que podría gustarle a un nuevo usuario. Sin embargo, esta técnica enfrenta muchos retos cuando se lidia con sistemas de gran escala. Esto se debe a que los usuarios y productos se van incrementando, lo que produce muchos más datos *esparcos* [10]. Esto provoca que el algoritmo se torne lento y no se obtenga una recomendación acertada.

Por este motivo se ha estudiado otro enfoque para la técnica de Filtro Colaborativo, como son las redes Bayesianas, en parte influenciado por la simplicidad y precisión del algoritmo Naive Bayes [10]. Diversos trabajos han presentado el modelado de preferencias de usuario usando redes Bayesianas como en [9], [2], [10], [7], así como procesos de *Dirichlet* y *Gaussianos* [1]. Estos abordajes tienen la ventaja que proveen una forma flexible para representar relaciones entre las preferencias del usuario y su contexto.

A diferencia de un método convencional, como el modelo de regresión, las redes Bayesianas no especifican una dirección de inferencia y cualquier variable aleatoria en la red puede ser la entrada, mientras que las demás pueden ser el objetivo de predicción que se calcula via probabilidades condicionales [7]. A pesar de su alta flexibilidad, las redes Bayesianas presentan dificultades en la construcción del modelo, varios enfoques para recomendación han sido propuestos pero suelen ser teóricos o sólo aplicables a problemas poco escalables [7].

En este trabajo se propone un modelo de red Bayesiana aplicado a sistemas de recomendación basado en contenido, para lo cual se adaptó el modelo de TrueSkill propuesto en [3]. En los experimentos realizados sobre el conjunto de datos MovieLens, al considerar TrueSkill, se obtuvieron valores altos de F1-score, en un número bajo de recomendaciones. El artículo está organizado de la siguiente manera. La Sección II describe el algoritmo original TrueSkill. La Sección III presenta trabajos relacionados. La propuesta es presentada en la Sección IV. Experimentos y resultados son relatados en la Sección V. Finalmente, la Sección VI concluye el artículo.

II. TRUESKILL

El primer modelo probabilístico aplicado a la puntuación de juegos fue *Elo rating* desarrollado en los años 1960 [6], este algoritmo procesa resultados de juegos de dos jugadores, incluyendo empates. TrueSkill [3] es una extensión de *Elo rating* y fue desarrollado por investigadores de Microsoft. Su inferencia bayesiana se basa en *factor graphs*. Este algoritmo compone equipos de acuerdo con las habilidades de los jugadores y los resultados de los partidos, en la Figura II se ilustra un ejemplo de TrueSkill.

Dado $A = i, j$ un conjunto de dos equipos i y j , definimos un conjunto de salida $r \in \{\text{team-}i\text{-win, team-}j\text{-win, draw}\}$. TrueSkill modela la probabilidad $p(r|s, A)$, la salida del juego r , dado un vector de habilidades s de cada jugador y su equipo asignado A . A partir de lo cual, se estima la distribución posterior del nivel de habilidad del jugador de acuerdo a la regla de Bayes:

$$p(s|r, A) \approx p(r|s, A)p(s) \quad (1)$$

donde la probabilidad Gaussiana *a priori* está dada por:

$$p(s) = \prod_{i=1}^n \mathcal{N}(s_i; \mu_i, \sigma_i^2) \quad (2)$$

Cada jugador adquiere una performance $p \approx \mathcal{N}(p_i; s_i, \sigma_i^2)$ dada sus habilidades s . La performance del equipo t esta dada por la suma de la performance de los jugadores que conforman el equipo $t_j = \sum_{i \in A_j} p_i$. Por otro lado, la probabilidad condicional $p(s|r, A)$ es calculada a partir de la distribución conjunta $p(s, p, t|r, A)$ integrando la salida del rendimiento individual p_i y el rendimiento del equipo t_i [3] que se puede escribir así:

$$p(s|r, A) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p(s, p, t|r, A) dp dt \quad (3)$$

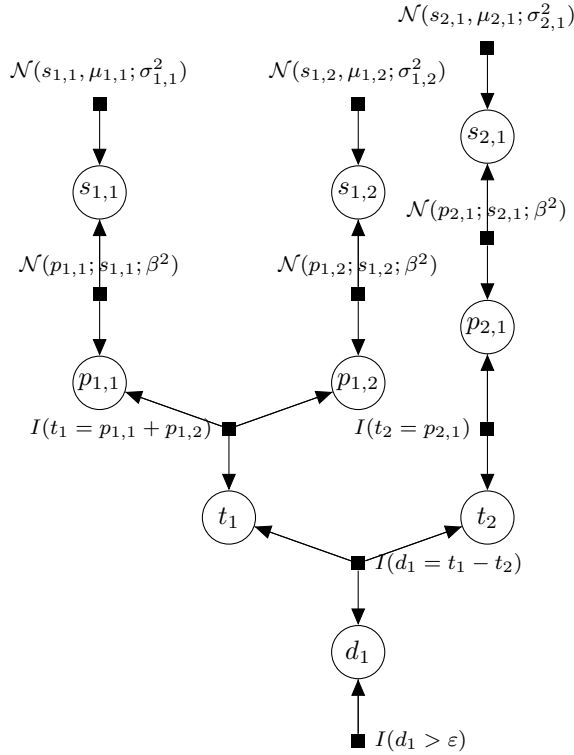


Figure 1. Modelo de TrueSkill con *Factor Graphs* [3]. Este grafo muestra la salida de un partido con 2 equipos, donde el equipo de la izquierda, que tiene dos jugadores ganó al segundo equipo que cuenta con uno sólo jugador.

III. TRABAJOS RELACIONADOS

En la literatura existen diversos trabajos sobre sistemas de recomendación que usan redes Bayesianas [8], [2], [6], [10], [11]. En particular, [5], [7] consideran las preferencias del usuario y su contexto. En [8] se propone un sistema de recomendación de música basado en el conocimiento del contexto usando una red Bayesiana *Fuzzy* y la Teoría de Utilidad. De esta forma, se mejora la recomendación de las redes Bayesianas debido a que estas, en general, requieren entradas discretas y existe una pérdida de información cuando se trabaja con variables continuas. Para satisfacer las configuraciones de tiempo real, multi-atributo, robustez y escalabilidad, [2] propone un framework de aproximación de elicitación de preferencias utilizando TrueSkill. A seguir, se detalla más el algoritmo utilizado en [2], ya que será base del presente trabajo. Se utiliza un vector de pesos sobre las preferencias del usuario $w = (w_{11}, \dots, w_{1|X_1|}, \dots, w_{D1}, \dots, w_{D|X_D|})$ los que describen su utilidad en cada dimensión. Cada dimensión indica las características latentes del producto. La meta de la elicitación de preferencias es aprender a estimar los pesos de cada atributo W , con la suficiente certeza. Dado que se quiere obtener el producto con utilidad más alta, ajustando los pesos de las preferencias de usuario. Se toma una perspectiva de aprendizaje Bayesiano en W manteniendo la distribución de probabilidad $P(W)$, que representa las preferencias del usuario w^* , como una distribución Gaussiana.

$$P(W) = \prod_{d=1}^D \prod_{i=1}^{|X_d|} p(w_{di}) = \prod_{d=1}^D \prod_{i=1}^{|X_d|} \mathbf{N}(w_{di}; \mu_{di}, \sigma_{di}^2) \quad (4)$$

Como dato apriori se tiene un modelo de consultas Q_{ij} al usuario, donde la respuesta indica la utilidad del producto i sobre el producto j . Dependiendo de las preferencias del usuario W se da correspondientes utilidades a los productos, $u(i|W)$ y $u(j|W)$. Para ello se realiza consultas de preferencia del producto i sobre j , $q_{ij} \in Q_{ij}$ representada en las siguientes ecuaciones:

$$P(q_{ij} = i \succ j|W) = I[u(i|W) - u(j|W) > \epsilon], \quad (5)$$

$$P(q_{ij} = i \prec j|W) = I[u(j|W) - u(i|W) > \epsilon] \quad (6)$$

$$P(q_{ij} = i \sim j|W) = I[|u(j|W) - u(i|W)| \leq \epsilon] \quad (7)$$

Donde el producto i , es preferido si la diferencia entre las utilidades de i es mayor que j dado un umbral ϵ definido, y viceversa. Si esta diferencia es menor que el umbral se considera que las utilidades de cada producto son iguales.

Dada la utilidad, $P(w|R^n)$ y un conjunto de $n \geq 0$ respuestas del usuario R^n , una nueva respuesta a la consulta q_{ij} es utilizada para ejecutar una actualización Bayesiana y obtener una probabilidad aposteriori $P(w|R^{n+1})$ donde $R^{n+1} = R^n \cup \{q_{ij}\}$:

$$P(W|R^{n+1}) \propto P(q_{ij}|W)P(W|R^n) \quad (8)$$

Asumiendo que la respuesta es que el producto i es preferido sobre el producto j , la probabilidad aposteriori $P(W|\{q_{ij} = i \succ j\})$ esta dada por:

$$P(W|\{q_{ij} = i \succ j\}) \propto I[u(i|W) - u(j|W) > \epsilon]P(w) \quad (9)$$

En relación a trabajos relacionados al algoritmo TrueSkill, el trabajo en [6] hace una mejora del algoritmo TrueSkill desarrollado en [3]. Para el soporte de empates y mejora del cálculo de performance del equipo se añade pesos a la performance de cada jugador en el equipo, ya que no es lo mismo si un jugador i juega un partido por media hora y un jugador j que juega por una hora. Esta mejora aún no fue aplicada por [2], pero sí en nuestra propuesta en el cálculo de performance del equipo para poder tener una certeza mayor sobre las preferencias del usuario. La siguiente sección detalla mejor nuestra propuesta, para después mostrar los resultados experimentales y conclusiones.

IV. PROPUESTA

La propuesta está presentada de manera esquemática en la Figura 2. Básicamente, la recomendación se realiza en dos etapas. En la primera etapa se realiza inferencia Bayesiana utilizando el modelo gráfico probabilístico de TrueSkill. El objetivo es obtener el vector de preferencias de usuario i , $S_{i,x}$, a partir de cada característica $x \in X$, extraída del contenido de los productos, y el rating otorgado por el usuario al producto. En la segunda etapa, se realiza una predicción de rating a partir de los vectores S y X , lo que conlleva a la recomendación final de productos. Para esta predicción se utiliza la fórmula generalizada de *Bredly Terry Luce model* [4].

$$P(T_i^+ > T_i^-) = \frac{\sum_{j \in T_i^+} p_j}{\sum_{j \in T_i} p_j} \quad (10)$$

Donde T_i^+ representa en nuestro caso al producto con mayor rating, entre dos productos elegidos. En la Figura IV, se ilustra el modelo TrueSkill de un producto, y dos usuarios

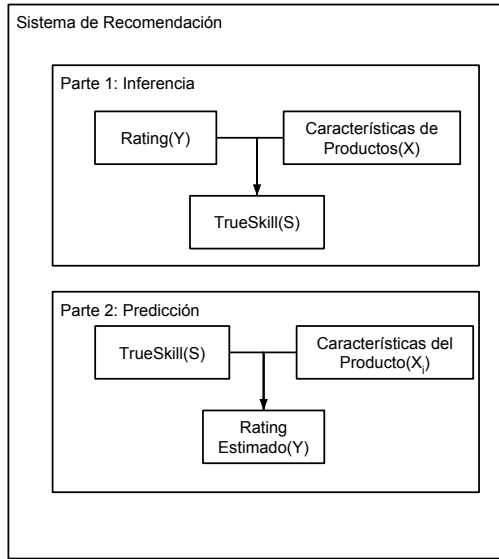


Figure 2. Modelo propuesto. Etapa 1. Inferencia de Bayesiana: Entradas del algoritmo TrueSkill, Rating(Y), características obtenidas de los productos(X). Salida: Vector de preferencias de usuario(S) sobre las características del producto. Etapa 2. Predicción y recomendación del producto utilizando el vector de preferencias del usuario extraídos del algoritmo TrueSkill.

que le dieron un rating, donde el rating del usuario u_1 es mayor que el usuario u_2 . El algoritmo ajustará las preferencias para cada usuario según el rating que este le asigna a cada producto. Si el usuario u_i hace una nueva calificación a un producto, este ajustará de nuevo las preferencias que tiene sobre las características de ese producto. Para mejorar la performance p_i del producto calificado, se hace uso del valor que tiene cada característica sobre el producto. Por ejemplo, para la película “Love”, dada las características “Romance” y “Acción”, la característica “Romance” tendrá un mayor valor que “Acción”, debido a que el contenido de la película expresa eso. Esto se traduce en coeficientes x_{j_1}, x_{j_2} sobre los valores de performance de cada característica p_i que, sumados, nos devuelve la performance total t_i del producto. De esta forma, cada vez que el usuario califique un producto, sus preferencias de usuario se actualizarán con mayor precisión. Como se puede ver este algoritmo no depende directamente del número de productos y usuarios que haya en el sistema, sino de cuantas veces se ha calificado un producto. De este modo, con cada calificación se obtiene un ajuste cada vez más exacto de las preferencias de usuario, y se pueda recomendar un producto no solo en base al rating sino también a su contenido.

V. RESULTADOS EXPERIMENTALES

Este experimento se ejecutó usando una máquina de con sistema operativo Ubuntu 14.04, procesador Intel Core i3 CPU M 330@2.13GHzx4. Los experimentos fueron realizados sobre el conjunto de datos “MovieLens”¹. “MoviLens” es una web colaborativa que administra un sistema de recomendación para películas, la cual esta basada en el algoritmo de Filtro Colaborativo, desarrollado por el grupo de investigación

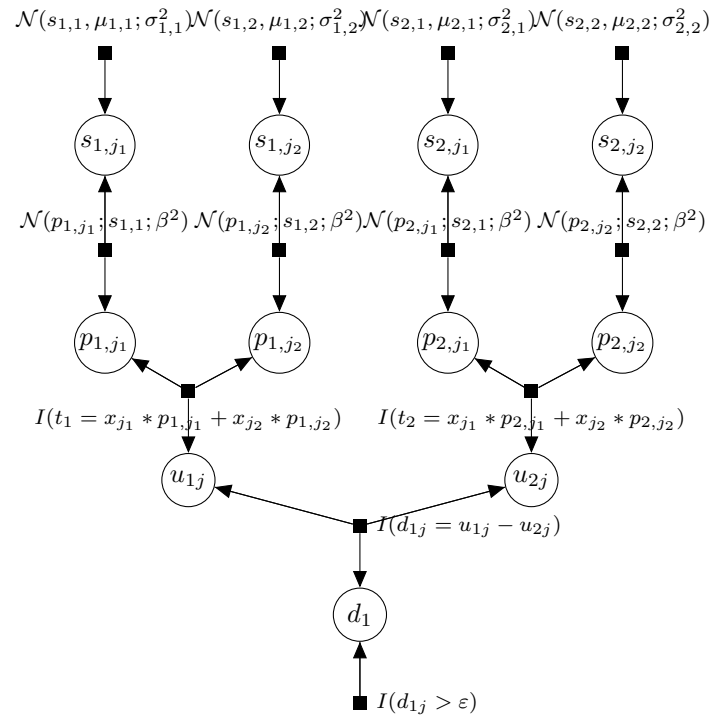


Figure 3. Modelo TrueSkill de calificación de un producto utilizando un vector con dos características. Este grafo muestra que el usuario u_{1j} prefiere al producto j , más que el usuario u_{2j} .

“GroupLens”. El conjunto de datos está compuesto por 6040 usuarios, 3952 productos y 100209 calificaciones de cada producto hecho por los usuarios. Sigue la siguiente estructura: *usuario, producto y calificación*. Además, tiene un vector de características que comprende los géneros al que pertenece cada película. Esto nos ayudará a definir valores sobre la performance de cada característica de la película, X . Estos géneros son: *Action, Adventure, Animation, Children’s, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western*. Cada valor de s_{i,j_k} en el modelo representa la preferencia del usuario u_i sobre la característica k del producto j , donde su distribución Gaussiana tiene: $\mu = 25$ y $\sigma = \frac{25}{3}$, como valor inicial. El valor de beta inicial es $\frac{\sigma}{2}$, y el margen de empate ϵ es $(\frac{p+1.0}{2}) \times \sqrt{N} \times \beta$ donde la probabilidad de empate p es igual a 0.10, y N es el número de usuarios que calificaron el producto j , estos valores iniciales fueron tomados de [3]. Los valores asignados a x_{j_k} son proporcionales al número de géneros contenidos en la película j . Se puede observar en la Figura 4, cómo los valores del vector de preferencias S del usuario u_i , se van modificando a medida que el usuario califica una película j , al evaluar películas que contienen solo dos géneros: *Action, Romance*. Se puede concluir que el usuario evaluado tiene preferencia por películas cuyo contenido es del género “Romance”. En la Figura 5 se observan las preferencias del usuario (los quince primeros) sobre cada característica después de realizar inferencia. Se han utilizado las 100209 calificaciones de usuarios y se puede observar que las preferencias están muy ajustadas, lo que puede provocar un problema de *overfitting*. En la Figura 6 muestra la recomendación de un producto, a un usuario aleatorio u_1 ,

¹<http://files.grouplens.org/datasets/movielens/ml-10m-README.html>

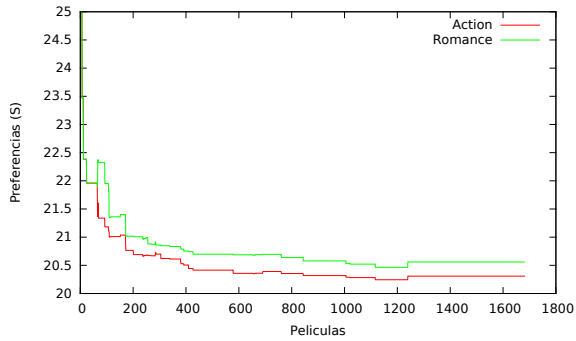


Figure 4. Aprendizaje de Preferencias del usuario sobre las características X "Romance" y "Action". Se puede observar que el usuario tiene preferencia por películas que contienen "Romance".

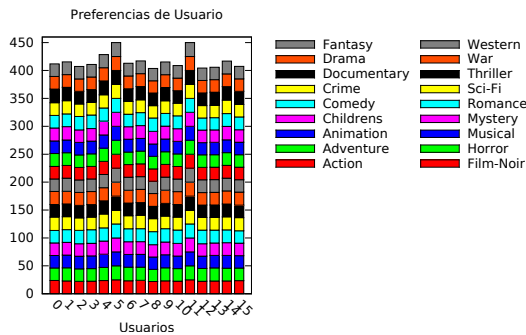


Figure 5. Preferencias del usuario sobre las características X

por lo que se elige a otro usuario u_2 para que compita con este, si el usuario u_1 consigue una mayor probabilidad de ganar, se le recomendará el producto, para evaluar la certeza en la recomendación usamos la métrica de F1-score, y se irá evaluando de acuerdo al número de recomendaciones que el sistema haga. Como se ve en la Figura 6, el sistema recomendador se va perfeccionando a medida que va haciendo recomendaciones, alcanzando un valor alto de F1-score con baja cantidad de recomendaciones, lo cual nos ayuda a superar el problema de cold start.

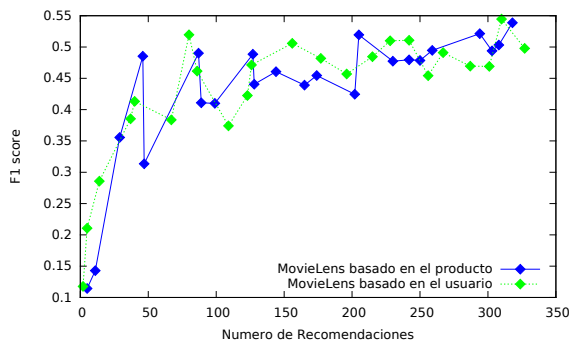


Figure 6. Cálculo del F1-score sobre la base de datos de MovieLens

VI. CONCLUSIÓN

Se ha propuesto un modelo de recomendación adaptando TrueSkill y sistemas de recomendación basados en contenido.

Este modelo resultó ser eficaz para enfrentar el problema de "cold start", ya que, utiliza el vector de características extraído del contenido del producto para realizar una inferencia de las preferencias del usuario. Asimismo, utiliza como entrada el "rating" de cada producto, alcanzando un promedio de 0.43 en F1-score, 0.54 de precisión, y recall de 0.38. para el conjunto de datos de "MovieLens". Esta propuesta es eficiente en un contexto de alta incertidumbre, ya que no depende directamente del número de productos y usuarios que haya en el sistema, sino de cuantas veces se ha calificado un producto. Esto permite obtener un ajuste cada vez más exacto de las preferencias de usuario sobre cada característica que los productos adquieran. Finalmente, la propuesta es altamente escalable ya que, las preferencias de usuario se van enriqueciendo a medida que se agregan más características de productos.

AGRADECIMIENTOS

Este trabajo ha sido apoyado y financiado por FONDECYT, a través del programa de maestría en informática de la Universidad Nacional de San Agustín.

REFERENCES

- [1] Ehsan Abbasnejad, Scott Sanner, Edwin V. Bonilla, and Pascal Poupard. Learning community-based preferences via dirichlet process mixtures of gaussian processes. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.
- [2] Shengbo Guo. *Bayesian recommender systems: Models and Algorithms*. 2011.
- [3] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems 20*, pages 569–576. MIT Press, January 2007.
- [4] Tzu kuo Huang, Ruby C. Weng, Chih jen Lin, and Greg Ridgeway. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.
- [5] Terje Nesbakken Lillegraven and Arnt Christian Wolden. Design of a bayesian recommender system for tourists presenting a solution to the cold-start user problem. Institutt for datateknikk og informasjonsvitenskap, 2010.
- [6] Sergey Nikolenko and Alexander Sirotkin. A new bayesian rating system for team competitions. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 601–608. New York, NY, USA, June 2011. ACM.
- [7] Chihiro Ono, Mori Kurokawa, Yoichi Motomura, and Hideki Asoh. A context-aware movie preference model using a bayesian network for recommendation and promotion. In Cristina Conati, Kathleen McCoy, and Georgios Paliouras, editors, *User Modeling 2007*, volume 4511 of *Lecture Notes in Computer Science*, pages 247–257. Springer Berlin Heidelberg, 2007.
- [8] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In Lipo Wang, Licheng Jiao, Guanming Shi, Xue Li, and Jing Liu, editors, *Fuzzy Systems and Knowledge Discovery*, volume 4223 of *Lecture Notes in Computer Science*, pages 970–979. Springer Berlin Heidelberg, 2006.
- [9] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 880–887. New York, NY, USA, 2008. ACM.
- [10] Xiaoyuan Su and T.M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, pages 497–504, Nov 2006.
- [11] Xiwang Yang, Yang Guo, and Yong Liu. Bayesian-inference based recommendation in online social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 551–555, April 2011.

- [12] Yi Zhang and Jonathan Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07*, pages 47–54, New York, NY, USA, 2007. ACM.