A Bayesian Approach to Collaborative Dish Selection

Team 10

Introduction

As anyone who has ever planned a catered event can attest, attempting to satisfy the various palates, dietary requirements and tastes of a group of diners can be a daunting task. This is particularly true given the exponential number of dishes which can be created from a small number of ingredients, as well has hard constraints such as allergies and religious beliefs. Many professional catering services handle this problem by allowing guests to select from a very limited menu. We introduce a dish recommendation system based on Bayesian Networks modeling user preferences. We predict the meals from a data base of recipes that most likely match the varied tastes of the customers, using a limited set of ingredients. This type of expert system would be of great use to a catering service or restaurant which needs to rapidly decide on a small number of dishes which would be acceptable for a large dinner party, given diverse requirements and preferences.

Bayesian Catering: A use case. Imagine that you run a catering service and have to plan an event with a customer. You can create a variety of dishes and now you want to discuss with your clients which one to serve. In order to get a better idea of which preferences and needs you clients will have, you let them fill out a survey in advance, where they rate a small amount of your dishes on a scale from $1 - 10$ and inform you about hard constraints like allergies, religious constraints or vegetarians. You then use those results in order to predict the ratings for the rest of your dishes and present the clients the top k results. If such a system works this will save time and will lead to a better customer satisfaction since you can present them dishes they will most probably like but still surprise them (since you have not presented them what the already rated). After the dinner, participants could rate the dishes served at the party which would iteratively improve the process for future customers.

Related Work

Boekel and Corney propose using Bayesian Networks to model consumer needs in food production chains [6] [1]. Janzen and Xiang propose an intelligent refrigerator capable of generating meal plans based on inventory and past food choices [2]. We suggest that these approaches are limited in that they only consider the preferences of a single (or supposed 'typical') user rather than a group. Baysian networks have also been applied to recommendation systems before in on-line social networks [5] making predictions of the form "if you bought those items what is the probability you would like to buy that". This method also uses bayesian networks for prediction and our approach is similar or inspired by some of Truyens [5].

Approach

The approached problem is to pick a single meal which best meets the requirements and tastes of different people dining together. We learn a predictive Bayesian net from a survey distributed to participants of the meal as training data in order to capture their preferences. The dishes in the questionnaire are selected such that all ingredients are covered. The participants rate each dish on a scale from one to ten and give additional information like vegetarians. For new dishes we then predict the maximum likelihood rating given our model. In the following we will describe our approach in detail. First we will discuss the data selection, then the modeling of the user preference and in the end how we trained the modeled net from gathered data and predicted the value for different recipes.

Data acquisition We first accumulated a diverse collection of license-free sample recipes from web sites such as *Darkstar's Meal-Master Recipes* (http://home.earthlink.net/ darkstar105/). Next, we converted these recipes from flat text files to well-formed XML using the 'Krecipes' application for Debian Linux. Finally, we created a representative data set representing several diners' preference for 24 of these recipes, using a simple survey of the type 'rate on a scale of 1 to 10, 10 being favorite and 1 being least favorite'. Furthermore, users were allowed to specify a vegetarian or nut-free meal preference.

Knowledge Engineering We model the diners' various taste preferences using a Bayes net. The net consists of three node types. We call them "control nodes", "taste nodes" and "rating nodes". A "preference node" models the probability of a diners' preference towards an ingredient ($P(likestomato)$, $P(likespotato)$) or a category ($P(likesmeat)$). These variables are discrete. The ingredients are conditional independent from each other but conditioned by the food category they belong to (see Figure 1 the two top layers). A control node can definitely reject a dish, by evaluating to 0 in certain conditions. For example if someone is vegetarian and the presented dish contains meet, the control variable for vegetarian will evaluate to 0 and so the probability for the whole dish will become 0. So the vegetarian variable is conditioned by meat. The third type in the net is a preference node, it is continuous and models the dish rating given a set of ingredients. The overall net is shown in Figure 1.

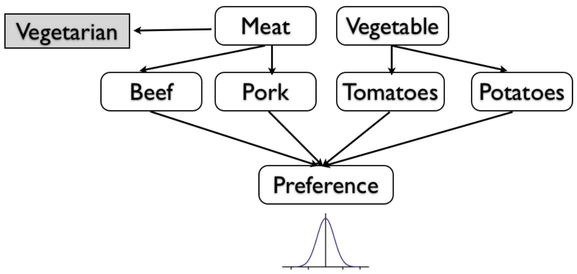

Figure 1. Our Bayesian net modeling user preferences. The top layer describes the categories Meat and Vegetable. We have a control variable vegetarian for Meat, such that it will always evaluate to 0 when there is meat involved in a dish and we have a vegetarian diner. The mid layer describes the preference for different ingredients. The last layer is a Gaussian predicting the users preferences.

Learning user preferences In order to estimate the model parameters, the system will be trained with statistics about taste and preferences given a set of dishes with ratings from multiple users. The training set is generated from the questionnaires we distributed. An example for a survey output could look like this (Ingredients, Rating): "Pork, Potatoes, 8". In order to perform normal Maximum Likelihood Learning [3] we have to have information about all variables ("Pork, Potatoes, Tomatoes, Beef, Meat, Vegetables, Rating"). We perform several steps in order to transform from the survey input to a training instance. First we discretize the values such that all given variables (in our case Pork and Potatoes) are set to "true" if the value is above a certain threshold (in our experiments 5) and "false" otherwise. In that way "liking things rated > 5" appear more often in the training set and will be assigned with a higher probability. We add categories by including the category of each ingredient from the survey. If the ingredient is liked, the category is too and if it is not, the category is not liked too. The last step is to add all values that are not in the recipe as "false" to the training instance.

From a set of those preprocessed assignments, we can directly calculate the probabilities for the ingredients using Maximum Likelihood Learning [3]. For example for an assignment of a conditional variable $P(X = x \mid Y_1 = y_1, ...Y_2 = y2)$, we count how often we observe the configuration $X = x, Y_1 = y_1, ...Y_2 = y2$ and how often we count $Y_1 = y_1, ...Y_2 = y2$ in our data set. The maximum likelihood is then defined as

$$P(X = x|Y_1 = y_1, ...Y_2 = y2) = \quad (1)$$

$$\frac{N(X = x|Y_1 = y_1, ...Y_2 = y2)}{N(Y_1 = y_1, ...Y_2 = y2)} \quad (2)$$

where $N(A)$ is the number of times event Aoccurs in the data set. For a continuous variable like rating, we estimate a Gaussian for each combination of it's parents. For example if the rating variable is dependent on beef and tomatoes, we would estimate 4 Gaussians, one for each possible combination of beef and tomatoes. So during training we would estimate mean and variance for all cases where $(tomato = true, potato = true)$, $(tomato = false, potato = true)$ and so on.

Inferring maximum likelihood rating Having estimated the probabilities of such a net, we can infer the maximum likelihood rating of a unseen dish while observing only a set of ingredients. Therefore, we iterate over all possible ratings $(1 - 10)$ and compute the probability of this rating. The maximum probability is the maximum likelihood rating for that dish. We use the *enumerateAll* algorithm [4], for the probability calculations.

Implementation In order to model food preferences, we implemented a custom Bayesian net library in Java with minimal use of third party libraries (e.g. for XML input). We chose to implement our own Library, for maximum flexibility and to ensure that the learning algorithm functions precisely as follows: The library uses the sum-product algorithm for inference and maximum likelihood learning for parameter estimation. In our implementation we support discrete as well as continuous probability distributions. Discrete distributions can be modeled as tables or as trees. In our implementation only continuous distributions with discrete parents are supported. A continuous distribution is then modeled as a mapping of all possible combination of its parents to a Gaussian set.

Evaluation

In an experiment we collected 24 ratings from 4 persons. We trained the Bayes net using a sparse subset (50%) of the survey data. Then we evaluated the rest of the recipes (which are all unseen) and calculated the maximum likelihood rating. As shown below, the calculated preferences for recipes which were not used to train the Bayes net are quite close to the actual survey data, which essentially reflects the following preferences (Sample ratings are on a 1-10 scale):

Diner 1 Prefers all dishes equally (5)

Diner 2 Vegetarian, meat dishes are (1), remainder are (9)

Diner 3 Prefers meat (6) to vegetarian (4) to desert (3)

Diner 4 Prefers Pork and Desserts (9), remainder are (3)

Next, we calculate the error between the application's ranking of all dishes and the actual ranking as determined by the user surveys. We suggest that a low error indicates that the system has the potential to accurately appraise constrained group food preferences for dishes which are not part of the survey, given sufficiently detailed recipe information. As the Table and Figure 2 show, the estimated food preferences are quite close to the actual mean ratings over all diners for the dishes which were not used to train the Bayes net. The root mean-square-error for calculated vs. surveyed meal preferences is approximately 1.92.

Dish	Est.	Actual Avg.
Southwest Smoothie: DAIRY	5	5.5
Bayou Shrimp Creole: TOMATO	9	3.75
Crab Burgers: EGGS	5	3.75
Broiled Flounder: GENERIC NUTS, EGGS	5	3.75
Baked Steak And Lima Beans: TOMATO, SUGAR	2	3.75
Eggplant Lasagna: GLUTEN	5	5.25
Salisbury Steak: GLUTEN, DAIRY, BEEF	6	3.75
Meatless Loaf: SPICE	5	5.25
Lemon Pork Chops: PORK, SUGAR	5	5.25
Fava Bean Burgers: EGGS, POTATO	3	5.25
Angel Hair Pesto Primavera: GENERIC NUTS, SPICE	5	5.25

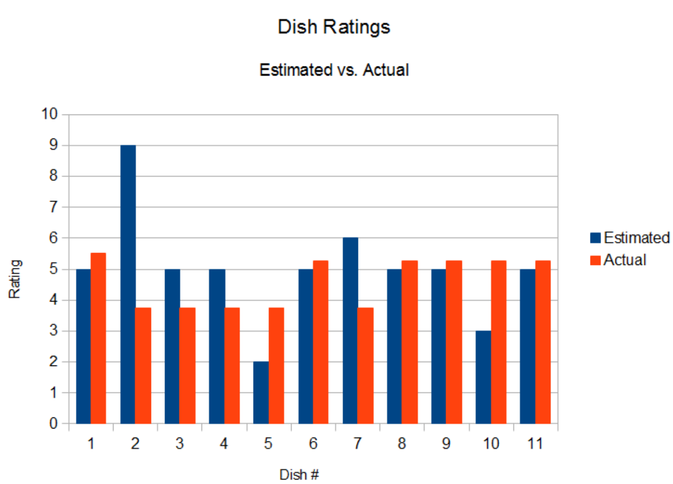

Figure 2. Estimated vs. surveyed dish ratings

Note the outlier at Dish #2 (Bayou Shrimp Creole). The strong preference for this dish is a result of the ingredient list containing primarily shrimp and tomato. Unlike beef and pork, the seafood category was not implemented in the knowledge engineering of the net. Consequently, this dish is incorrectly deemed to be vegetarian-compatible. The same issue had previously occurred at Dish #5 (Baked Steak and Lima Beans) until 'steak' was added to the recipe parser as a synonym for beef, and therefore a type of meat.

Conclusion

We proposed, implemented and evaluated a food preference prediction system that is capable of predicting how much a user would like a new, unseen recipes. We discussed how to encode user preference towards ingredients and categories in a Bayes Net and how to add control variables in order to exclude dishes that users have to avoid, such as meat in the case of vegetarians. Furthermore, we presented our learning scheme for such a Bayes net using data from a small survey and how to predict the user rating for unseen dishes. In an evaluation we showed that the net can predict preferences when learned from a sparse data set. So in a real life setting, where people plan a dinner with a catering service, a few participants could rate a small amount of recipes in an on-line service and the system could actually predict the scores on the rest of the caterers data base. The top k with the answers highest predicted rating of the system could be used to assemble the final dinner.

References

[1] David Corney. Designing Food with Bayesian Belief Networks. *ACDM*, 2000.

[2] M. Janzen and Y. Xiang. Probabilistic Reasoning in Meal Planning in Intelligent Fridges. *16th Conference of the Canadian Society for Computational Studies of Intelligence*, 2003.

[3] Kevin Murphy. A Brief Introduction to Graphical Models and Bayesian Networks. http://www.cs.ubc.ca/ murphyk/Bayes/bnintro.html, 1998. [Online; accessed 22-Feb-2012].

[4] S. Russel and P. Norvig. *Artificial Intelligence; A Modern Approach.* Prentice Hall, third edition, 2010.

[5] Tran The Truyen, Dinh Q. Phung, and Svetha Venkatesh. Preference Networks: Probabilistic Models for Recommendation Systems. *Proceedings of the Sixth Australasian Conference on Data Mining*, 2007.

[6] Stein A. van Boekel, M.A.J.S. and A.H.C. van Bruggen. Bayesian Statistics and Quality Modelling in the Agro-food Production Chain. *Proceedings of the Frontis workshop*, 3, 2004.