**A new aware-context collaborative filtering approach by applying multivariate logistic regression model into general user pattern**

**Abstract**

Traditional collaborative filtering (CF) does not take into account contextual factors such as time, place, companion, environment, etc which are useful information around users or relevant to recommender application. So recent aware-context CF takes advantages of such information in order to improve the quality of recommendation. There are three main aware-context approaches: contextual pre-filtering, contextual post-filtering and contextual modeling. Each approach has individual strong points and drawbacks but there is a requirement of steady and fast inference model which supports to aware-context recommendation process.

This paper proposes a new approach which discovers multivariate logistic regression model by mining both traditional rating data and contextual data. Logistic model is optimal inference model in response to the binary question “whether or not a user prefers a list of recommendations with regard to contextual condition”. Consequently, such regression model is used as a filter to remove irrelevant items from recommendations. The final list is the best recommendations to be given to users under contextual information. Moreover the searching items space of logistic model is reduced to smaller set of items so-called general user pattern (GUP). GUP supports logistic model to be faster in real-time response. Because the (response or dependent) contextual variable is binary, this paper also proposes an extension of logistic function in order to process nominal values in contextual variable in case that context is quantized as discrete values, for example: “Monday”, “Tuesday”,… and “Sunday” in temporary data.

**1. Introduction**

Recent researches on collaborative filtering (CF) focus on inherent information about users and items and how to recommend such relevant items to such users. Database used to build up CF algorithms is in form of rating matrix composed of ratings that users give to items. Additional contextual factors such as time, place, condition and situation existing in real world are not considered in CF algorithms. For instance, if a user prefers to watch news program in morning and movies in evening then contextual information, namely temporary information, should be aware in recommendation tasks and so, it is inappropriate to provide movies to her/him in the morning even though such movies are the most relevant to her/him.

Given a training set which is a rating matrix and a user who requires recommendations, CF algorithm tries to predict rating value on items which are not rated by this user. After that CF algorithm makes a list of such items arranged in order of predictive rating values and recommends this user such list. In other words, CF algorithm constructs a predictive function *R2* [1] whose cross domain is the Cartesian product of a set of users *U* and a set of items *I*. The domain *U x I* is also called rating matrix. Co-domain of function *R2* is a set of predictive ratings denoted *R*.

*: U x I → R*

Function *R2* so-called traditional 2-dimension (2D) mapping doesn’t consider contextual factors now and it can lack of information necessary to highly accurate prediction. Suppose contextual information including location, time and companion are added to prediction process, the 2D function *R2* becomes the 3-dimension (3D) mapping denoted as below:

*: U x I x C → R*

Note that *C*, *U x I* and *U x I x C* represent context domain, 2D (cross) domain and 3D (cross) domain, respectively. In order words, function *R3* gives recommendations to user under circumstances which specified as contextual information.

Although context has many different types, we can reduce these types into three main types in order to answer three question forms: when, where and who.

* Time type indicates the time when user requires recommendation, for example: date, day of week, month, and year.
* Location type indicates the place where user requires recommendation, for example: theater, coffee house.
* Companion type indicates the persons with whom user goes or stays when recommendation task is required, such as: alone, friends, girlfriend / boyfriend, family, co-workers.

Contextual information is organized in two forms: hierarchical structure [2] and multi-dimensional data (MD) model.

According to hierarchical form, the context domain *C* is defined as a set of contextual dimension ***K*** = (*K1, K2, K3,…, Kn*). **K** is represented in hierarchy structure, whose attributes *Ki* (s) are associated with the ascending order of fine level. For example, given attributes *Ki* and *Kj* where *i < j*, *Kj* is finer than *Ki* and so *Ki* contains *Kj*. It is easy to recognize that *K1* is the coarsest attribute which contains all remaining attributes *K2, K3,…, Kn*. Each *Ki* contains values at the same level *i* and it can be split into finer levels.

Figure 1.1 Contextual dimension example.

In above example, *K2* = {City, Province}, *K3* = (City → District, City → Suburb district, Province → District, Province → Suburb district)

According to MD form, context domain C is defined as the Cartesian product of *n* dimension, *C =* *D1 x D2 x… Dn.* Each dimension *Di*, in turn, is a set of attributes, *Di =* (*ai1, ai2,…, aik*). For example, suppose *C* has only one dimension of time denoted *D1 = Time* (*day of week*). So, the cross domain *U x I x C* of predictive function *R3* constitutes a 3D cube: *User* (*name*), *Item* (*book type*) and *Time* (*day of week*). Each block in this cube is assigned a rating which is the predictive outcome of function *R3*.

Figure 1.1 MD cube.

Above figure indicates that given user “John”, item “novel book” and time “Sunday”, the predictive outcome of *R3* gains the highest value 5. There are three approaches [1] to apply context into recommendation process:

* Contextual pre-filtering: Firstly, given context *c C* is used to select user-item pairs (*u, i*) which are more relevant to this context, leading to obtain the aware-context cross domain *U x I*. After that traditional 2D function *R2* is taken on such cross domain.
* Contextual post-filtering: Firstly, traditional 2D function *R2* is used to produce the list of recommended item. After that context *C* is used to fine-tune this list in order to remove irrelevant items according to concrete context.
* Contextual modeling: The 3D function *R3* is used directly on context-aware cross domain *U x I x C*.

The basic idea of contextual pre-filtering is to project the 3D domain *U x I x C* on 2D plane, based a concrete context *c C* so that such 3D domain is reduced to 2D domain *U x I*. Let ∏c be projection operation based on condition context *c*, we have:

*= U x I*

(*u, i, c*) *=* (*u, i*)

The concrete context *c C* which is strict projection condition can make the cross domain *U x I* small or sparse, causing low predictive accuracy. So generalization technique is used to make projection condition loose, namely the exact condition *c* is replaced by the more general condition *c’*. For example, the context *c* = ‘Saturday’ which indicates that “Mary prefers to go shopping on Saturday” is replaced by more flexible context *c* = “Weekend” because she can like going shopping on Sunday if she often goes on Saturday. The general context not only expends the space of potential recommendation solutions but also improve predictive accuracy.

The essence of contextual post-filtering is to fine-tune the raw recommendation results taken from predictive function *R2* which didn’t consider contextual factors before. Consequently, this method tries to figure out user’s context-aware interests, preferences or attributes by using some artificial intelligent and mining techniques and apply such attributes into raw results so as to remove out irrelevant items or change their ranks in final recommendation list. For example, given context *c* = ‘evening’ which indicates Mary’s interest “watching movies in the evening”, contextual post-filtering approach will remove all of news or sport events from her recommendation list.

The strong point of contextual pre-filtering and post-filtering approaches is the ability to take advantage of legacy recommendation algorithms not taking account into contextual factors.

The essence of contextual modeling is to incorporate directly contextual information into predictive function *R3* and so, *R3* is constructed as inference model such as data mining, machine learning, heuristic model or statistical model, etc. The 2D predictive function isn’t used in contextual modeling. It implies that the strong point of this method is pure and powerful inference mechanism. It opens a new trend in context-aware recommendation research, giving a lot of prospects although a few related techniques are extensions of 2D algorithms.

The approach in this paper is the hybrid of contextual post-filtering and contextual modeling. Hence a logistic inference model is used as the post filter to make recommendations more relevant to users according to contextual factors.

**2. Basic idea and details of the new approach**

The new approach is suggested by two comments:

* Although contextual information is necessary to improve the quality of recommendation process but it cannot replace essential rating information in collaborative filtering research. Inference model taking into account contextual factor should be used as the filter to adjust recommendations returned from predictive ratings so as to give users more appropriate items in concrete circumstances.
* When additional contextual information is considered, the speed of recommendation process is decreased. So inference mechanism should be fast in real-time response.

The basic idea is to apply a fast inference model, namely logistic regression function into a list of recommended items so as to achieve a better recommendation result under contextual information. The logistic regression model responses immediately the binary request “whether or not a list of items is relevant to concrete context or preferred by users”. Because there are various items and each item is associated with an individual regression function, the domain of regression function becomes huge, which decreases the speed of algorithm. In order to solve this problem, the items space is reduced to “general user pattern”.

General user pattern (*GUP*) is known as a set of items to which “many” users give ratings. Since rating values on *GUP* are not always high, it reflects solely user’s access or rating frequency. Given the threshold, let *n* and *nk* be the number of total users and the number of users who rate on item *ik*, we have:

*GUP* = { *ik | ni / n*  }

So *GUP* is defined as a set of items where the ratio of the number of total users to the number of users who rate on such each item is great than or equal to a given threshold.

Given a *GUP*, context *c* and a logistic regression function *f*, the regressive (or independent) variables of *f* are taken from *GUP*. The response of *f* takes only two values *0* and *1* where value *1* indicates that it is likely that user prefers *GUP* under context *c* and otherwise. Note that context *c* is contextual or dependent variable of *f*.

(**2.1)**

Given the raw recommended list *R*, an instance of *GUP* is initialized on *R*. This instance denoted *INS* is a set of predictive rating values taken from R with condition that respective items co-exist in both *R* and *GUP*. For example, if *R =* { *i1 = 5, i2 = 3, i3 = 4*} and *GUP =* { *i1, i3*} then *INS =* { *i1 = 5, i3 = 4* } because item *i1* and item *i3* exist in both R and GUP and their respective values are 5 and 4.

Consequently, regression model *f* is evaluated on *INS* with regard to context *c C*. If the outcome of *f* with is *0* then all items in *GUP* are removed from *R*. Finally, the list *R* is recommended to user after it was fine-tuned by pruning irrelevant items from it.

In general the algorithm has four steps:

1. A 2D predictive function is applied into rating matrix *U x I* so as to produce a raw list *R* of recommended items without existence of contextual factors.
2. *GUP* is discovered over contextual 3D cross domain *U x I x C*. Items in *GUP* are frequent items.
3. Multivariate logistic function *f* is learned from cross domain *U x I x C* by statistical technique.
4. Function *f* is used to remove irrelevant items from the list *R*. In other words, only aware-context items are kept in *R*. So the final filtered list is the best result which is recommended to users. This step includes two sub-steps:
   * The instance of GUP so-called INS is constructed by matching GUP with *R.*
   * Function f is evaluated on INS to perform the removal of redundant items from R.

Because step 3 is the most important, the method to construct multivariate logistic function is discussed in detailed now. Suppose that the probability that user likes *GUP =* {*i1, i2,…, in*} with context *c C* is p. The concept *odd* [3] is defined as the ratio of *p* to *1* – *p*. This ratio represents how much user likes *GUP* vice versa how much user dislikes item *GUP*.

**(2.2)**

The logarithm of odd is linear regression function of *n* variables being *GUP*

**(2.3)**

Note that *I1, I2,…, In* are regressive or independent variables whose values are obtained from recommended list *R* later. If *odd* is considered as response or dependent variable, equation 2.3 is re-written as following:

**(2.4)**

The probability that user likes item *GUP* is computed according to following function derived from equation 2.4:

**(2.5)**

Equation 2.5 represents the multivariate logistic model *f*. The approach in this paper uses this equation to estimate whether or not a user prefers a list of recommended items based on a concrete context. For example, given user being “John”, GUP being { “Gladiator”, “Golden Eye” }, recommended movie list being { “Gladiator” with predictive rating *5*, “Golden Eye” with predictive rating *4*, “Four Rooms” with predictive rating *4*} and time context being “evening”, if *p* (or *f*) is greater than or equal to *0.5* then it asserts that John likes such list and there is no film to be removed. This logistic model is built up in offline mode so as not to affect response time.

It is easy to recognize that logistic function *f* is evaluated for only one contextual attribute. In case of many contextual attributes, we have two choices:

* Constructing many respective functions, each of them is applied into one contextual attribute. Or,
* Extending logistic model to support various contextual attributes.

Section 3 will discuss the second choice being an extension of multivariate logistic model which supports contextual attribute in case that context has nominal attributes such as “morning”, “afternoon” and “evening”.

**3. Logistic model and nominal attributes in contextual information**

Suppose that there are a set of *m* contextual attributes *C =* {*c1, c2,…, cm*} and a set of *n* item *GUP =* {*i0, i1, i2,…, in*} with note that *i0* is pseudo-item whose rating is *0*. Note that *c C* is nominal attribute. For each context *c C*, there is a set of respective coefficients which implies that is the parameter vector under context *c*. The logistic model *f* is re-written:

Where *Ac I* denotes the cross product of and *I*.

Model *f* is considered as decision maker when the outcome of *f* which is the posterior probability is used to determine whether or not context GUP belongs to context *ci*. What we need to do now is to calculate the parameter. Maximum likelihood estimation (MLE) function used to estimate A is:

Where *D* is the training data which is the instance of the aware-context 3D cross domain *U x I x C*. Because *mlei* is the partial MLE function with regard to context *ci*, the global MLE function over whole context domain *C* the product of all partial functions:

Note that the variable of *MLE* is parameter vector . The best estimator of parameter A is maximum peak of the curve *MLE*. It is easier to find such peak by taking logarithm of MLE. We have:

Where ***A*** is the parameter matrix where each row ***A*** is parameter vector. The estimator of ***A*** isdenoted ***Â****:*

The estimator of *kth* parameter is found by setting the first-order partial derivative of *K*(***A***) with respect to each parameter to 0 and solving this equation. The first-order partial derivative of *K*(***A***) is:

The solution of this equation is the estimator of .

**4. A case study: using multivariate logistic model**

**5. Conclusion**

The approach in this paper is the hybrid of contextual post-filtering and contextual modeling where logistic model is applied in the post stage of recommendation process. The thinking behind this approach is that rating values obtained explicitly by questionnaires or implicitly by inferring users’ behaviors are the most important and contextual factor around users or related to application is additional information which is useful but not essential. Comparing to contextual pre-filtering, this approach restricts the loss of rating information in rating matrix by ignoring data pre-filtering. At the post stage, this approach removes only items which are asserted that users do not like them under contextual condition. Such assertion is the outcome of steady inference model, namely logistic model.

The removal restriction increases recall metric due to reserving solutions space but doesn’t lessen precision metric in comparison of pre-filtering method. Comparing to traditional or post-filtering method, this approach is more accurate because of the stead inference mechanism of logistic function when logistic model is appropriate to binary request such as following yes/no question “whether or not Mary prefers to browse commercial websites in the evening”. Moreover this approach exploits the relationship among items in general user pattern, which is necessary to recommendation process but is not considered in contextual pre-filtering or post-filtering.

**Reference**

1. [Ricci, Shapira, Kantor 2011]. Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor. Recommender Systems Handbook. Springer Science + Business Media, LLC 2011, ISBN: 978-0-387-85819-7.
2. [Palmisano, Tuzhilin and Gorgoglione 2008]. Cosimo Palmisano, Alexander Tuzhilin and Michele Gorgoglione. Using Context to Improve Predictive Modeling of Customers in Personalization Applications. IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 11, November 2008.
3. [Montgomery, Runger 2003]. Douglas C. Montgomery, George C. Runger. Applied Statistics and Probability for Engineers, 3rd. ISBN: 0-471-20454-4. John Wiley & Sons, Inc. 2003.