Following is pseudo-code of item-based *K*-nearest neighbors (KNN) algorithm for estimating one missing value.

|  |
| --- |
| Input: active item *v*1 = (*r*11, *r*21,…, *rn*1), user *ui* who does not rate item *v*1 (*ri*1 is missing), and a predefined number *K*.  Output: estimate value *ri*1 for user *ui*.  Let *L* be the sorted list of calculated similarities  For each item *v*2 = (*r*12, *r*22,…, *rn*2) in rating matrix  If user *ui* does not rate on item *v*2 then ignore.  Calculating the similarity measure sim(*v*1, *v*2) between *v*1 and *v*2.  If sim(*v*1, *v*2) is smaller than a predefined threshold then moving next item.  Add sim(*v*1, *v*2) to the list *L* such that *L* is sorted in descending order.  If size of the list *L* is larger than *K* then  Remove the last similarity from the list *L*.  End If  End For  *accum* = 0  *simTotal* = 0  For each item *v*2 = (*r*12, *r*22,…, *rn*2) associated with each sim(*v*1, *v*2) in *L*  *accum* += sim(*v*1, *v*2) \* (*ri*2 – )  *simTotal* += |sim(*v*1, *v*2)|  End For  *ri*1 = + *accum* / *simTotal* |

Where *U*1 and *U*2­ are index sets of users that rate on item *v*1 and item *v*2, respectively. This pseudo-code is coded as function *knn\_one*(*v*1, *ui*, *K*). The following is pseudo-code of item-based *K*-nearest neighbors (item-based KNN) algorithm based on function *knn*(*v*1, *ui*, *K*) for estimating many missing values of given active user *u*1.

|  |
| --- |
| Input: active user *u*1 = (*r*11, *r*12,…, *r*1*n*), all items that *u*1 does not rate, and a predefined number *K*.  Output: list *L* of estimate values of all items.  For each item *vj* whose rating values *r*1*j* is not rated by user *u*1  Call function *knn\_one*(*vj*, *u*1, *K*) for estimating *r*1*j* such that *r*1*j* = *knn\_one*(*vj*, *u*1, *K*).  Add *r*1*j* to list *L* such that *L* is sorted in descending order.  End for  Select some items whose values are highest in the sorted list *L* in order to recommend them to user *u*1. |