

Documentació de la primera entrega

Projectes de Programació

Pol Casacuberta Gil
Edgar Moreno Martínez
Maria Prat Colomer
Pablo Vega Gallego

Q1 2021 - 2022
FIB - UPC

En aquest document justifiquem l'implementació de les classes i descrivim les estructures de dades i algorismes utilitzats per implementar les funcionalitats principals de la primera entrega de l'assignatura de Projectes de Programació.

Relació de les classes implementades per cada membre de l'equip

Pol Casacuberta Gil

- ConjuntUsuaris
- ConjuntValoracions
- Programa
- Sessio
- SessioIniciada
- SessioNoIniciada
- Usuari
- Valoracio

Edgar Moreno Martínez

- package jocs_de_prova
- Pair
- recomanador.metode_recomanador.ConjuntPunts
- recomanador.metode_recomanador.KMeans
- recomanador.metode_recomanador.KNN
- recomanador.metode_recomanador.MetodeRecomanador
- recomanador.metode_recomanador.MetodeRecomanadorCollaborative
- recomanador.metode_recomanador.MetodeRecomanadorContentBased
- recomanador.metode_recomanador.Punt
- recomanador.metode_recomanador.SlopeOne
- Main

Maria Prat Colomer

- libs/consola
- ConjuntIdentificat
- ElementIdentificat
- Id
- Item
- TipusItem
- recomanador.filtre.Filtre
- recomanador.filtre.FiltreExclusiu
- recomanador.filtre.FiltreInclusiu
- recomanador.Recomanador
- recomanador.RecomanadorCollaborative
- recomanador.RecomanadorContentBased
- atributs.TipusAtribut

- atributs.distancia.Distancia
- atributs.distancia.DistanciaDiferenciaDeConjunts
- atributs.distancia.DistanciaDiscreta
- atributs.distancia.DistanciaEuclidiana
- atributs.distancia.DistanciaLevenshtein
- atributs.distancia.DistanciaZebuero
- atributs.valors.ValorAtribut
- atributs.valors.ValorBoolea
- atributs.valors.ValorCategoric
- atributs.valors.ValorConjunt
- atributs.valors.ValorConjuntBoolea
- atributs.valors.ValorConjuntCategoric
- atributs.valors.ValorConjuntNumeric
- atributs.valors.ValorConjuntTextual
- atributs.valors.ValorNumeric
- atributs.valors.ValorTextual

Pablo Vega Gallego

- ConjuntItems
- Contenedor
- EscriptorDeFitxers
- LectorDeFitxers
- recomanador.ConjuntRecomanacions
- recomanador.Recomanacio
- csv.EscriptorDeCSV
- csv.LectorDeCSV
- csv.TaulaCSV

A més a més, per facilitar la implementació dels tests i els drivers, hem creat dues classes: UtilitatsDeLectura i UtilitatsDEscriptura. En aquestes classes hi hem treballat tots els membres de l'equip.

Cadascú ha implementat els tests de les classes que li corresponen.

Justificació i descripció de cada classe

Pol Casacuberta Gil

- **ConjuntUsuaris**

Classe que representa un conjunt d'usuaris. ConjuntUsuari hereta de la classe ConjuntIdentificador

- **ConjuntValoracions**

Classe que representa un conjunt de valoracions. `ConjuntValoracio` hereta de la classe `ConjuntIdentificador`.

- **Programa**

Classe que representa l'estat del programa. Conté l'estat de la sessió implementat de la forma `State Pattern` per a saber si un usuari ha iniciat sessió o no. Conté un atribut `instancia_unica` que representa l'única instància de `Programa` que podem crear, seguint el patró de `singleton`. Finalment, conté un `conjuntUsuaris` que conté tots els usuaris del programa en aquell moment.

- **Sessio**

Representa l'estat del programa en què s'ha iniciat sessió o bé en el que no, d'aquesta classe abstracta en poden esdevenir dos estats, `SessioIniciada` i `SessioNoIniciada`.

- **SessioIniciada**

Representa l'estat del programa en què s'ha iniciat sessió.

- **SessioNoIniciada**

Representa l'estat del programa en què no s'ha iniciat sessió.

- **Usuari**

Classe que representa un usuari. Per a representar el nom i la contrasenya d'aquest hem utilitzat strings. Per a identificar de forma única als usuaris, ja que dos usuaris poden tenir noms iguals, fem servir una classe `Id` que conté un `int` que serà diferent de tots els altres usuaris, i un booleà que ens indica si l'usuari s'ha esborrat el compte o no. Finalment per a emmagatzemar els ítems valorats per l'usuari hem usat un `map` per a poder accedir amb un ítem, a la valoració que l'usuari ha fet a aquest ítem.

- **Valoracio**

Classe que representa una valoració. Conté el valor d'una valoració en un `double` i conté una instància d'`Usuari` i d'`Ítem` que representen l'usuari que ha valorat l'ítem amb aquell valor.

Edgar Moreno Martínez

- **package jocs_de_prova**

Conté un conjunt de jocs de prova per provar les funcionalitats de les classes.

- **Pair**

Classe bàsica que representa la unió de dos objectes de tipus qualssevol. S'ordena pel primer element i en cas d'empat pel segon. Ambdues classes han de ser *Comparable*.

- **recomanador.metode_recomanador.ConjuntPunts**

Aquesta classe representa un conjunt de punts, utilitzant com a base un map de int a punt per tenir els punts identificats en tot moment. Manté en tot moment calculat el baricentre dels punts del conjunt, fent-ho adient per utilitzar amb l'algorisme de k-means. També permet obtenir un identificador que no estigui utilitzat de forma eficient.

- **recomanador.metode_recomanador.KMeans**

Classe que permet processar un conjunt de punts per agrupar-los en clusters. L'únic destacable de l'algorisme és que assigna els clústers inicials als k primers punts tot i que es podrien explorar altres opcions.

L'única funció pública retorna la partició en k conjunts dels punts amb els quals s'ha construït l'objecte resultant de l'algorisme.

Hi ha un seguit de funcions i atributs privats per tal de modularitzar l'algorisme.

- **recomanador.metode_recomanador.KNN**

Classe que permet aplicar l'algoritme de k-Nearest Neighbours.

Es construeix amb el conjunt d'ítems desitjat i després es poden consultar els k ítems més propers a un ítem donat. Simplement, s'itera per tots els ítems, es calcula la distància a cadascun (mitjançant una funció de la classe ítem) i es retornen els més propers amb ajuda d'una cua de prioritat.

- **recomanador.metode_recomanador.MetodeRecomanador**

Classe abstracta que representa un Mètode Recomanador. Guarda un conjunt d'usuaris, ítems i valoracions que serveixen com a base per fer les recomanacions.

Permet obtenir recomanacions per un usuari concret, es poden especificar les valoracions que s'han d'usar i sobre quins ítems ha de ser la recomanació. Si no s'especifiquen s'utilitzaran els conjunts guardats.

- **recomanador.metode_recomanador.MetodeRecomanadorCollaborative**

Classe que exten MetodeRecomanador a un mètode basat en Collaborative Filtering.

Té un únic atribut que indica quants clústers s'han de crear al pas on s'usa l'algorisme k-means.

Al inicial precalcula les particions. Per això converteix els usuaris a punts a R^n en base a les valoracions que han donat al conjunt d'ítems sobre el que estem treballant. Aquests punts es processen fent servir l'algorisme de K-Means per obtenir clusters d'usuaris.

L'única funció pública genera una recomanació en rebre un usuari, valoracions d'aquest usuari, un conjunt d'ítems sobre els que recomanar i el nombre de recomanacions a generar.

Per això primer troba a quin clúster està l'usuari demanat i, per tant, els usuaris amb el que seguirà el procés.

Amb aquests usuaris es crea una matriu amb les valoracions d'aquests usuaris als elements donats. Seguidament, aquesta matriu es processa amb l'algorisme SlopeOne per obtenir les valoracions inferides per cada ítem per l'usuari a qui recomanar.

Finalment, amb una cua de prioritat s'obtenen els ítems amb millor valoració que es retornen com a recomanació.

- **recomanador.metode_recomanador.MetodeRecomanadorContentBased**

Classe que exten MetodeRecomanador a un mètode basat en filtre per contingut.

Té un únic atribut que indica quin és el llindar mínim per tal que una valoració sigui considerada.

L'única funció genera una recomanació en rebre un usuari, valoracions d'aquest usuari, un conjunt d'ítems sobre els que recomanar i el nombre de recomanacions a generar.

L'algorisme és molt simple. Iterem per tots els ítems valorats per l'usuari i de cadascun busquem els més similars. Portem un comptador de quants cops ha sortit cada ítem ponderat per la valoració de l'ítem del qual és veí cada cop. Al final utilitzem una cua de prioritat per obtenir els ítems amb millor puntuació i retornem la recomanació.

- **recomanador.metode_recomanador.Punt**

Una classe senzilla que exten el concepte de ArrayList<Double> per poder representar punts de R^n i tenir operacions senzilles (suma, resta, multiplicació per escalar, distància i norma).

- **recomanador.metode_recomanador.SlopeOne**

Classe que processa un conjunt de valoracions per tal d'oferir prediccions sobre possibles valoracions seguint l'algorisme de SlopeOne.

S'ha de construir amb una array bidimensional de doubles que representa les valoracions. Es demana en aquest format per no dependre de la implementació de les classes Usuari, Ítem i Valoració.

En construir-se precalcula les desviacions entre ítems per fer més eficient la resta del procés.

Després es poden demanar totes les prediccions o una predicció per un ítem i usuari específic.

- **Main**

El Main ens permet carregar conjunts de dades i obtenir recomanacions per un usuari i avaluar les recomanacions que s'obtenen.

Maria Prat Colomer

- **libs/consola**

Aquesta classe conté funcions estàtiques que fan més fàcil la lectura de valors des de la consola. Permet imprimir missatges de consulta, missatges d'error i comprovar que els valors llegits estan en un determinat rang. És especialment útil a l'hora d'implementar els drivers.

- **ConjuntIdentificat**

Un conjunt identificat és un conjunt tal que els seus elements es poden identificar amb un Id. El conjunt identificat ens permet abstraure la idea de conjunts d'elements identificats i poder implementar així conjunts d'Ítems o d'Usuaris. Els conjunts identificats s'implementen com a TreeMaps ja que volem que les operacions d'inserció i de consulta siguin ràpides i, a més, en algunes ocasions hem d'iterar sobre els seus elements.

Pel que fa a les valoracions, no es poden implementar com a conjunts identificats ja que no tenen un identificador per si mateixes.

- **ElementIdentificat**

Un element identificat és una abstracció d'Item i Usuari, que són dos elements que es poden identificar amb un Id. És una interfície necessària per definir els conjunts identificats.

- **Id**

Un Id és un identificador. Té dos atributs: un valor enter i un booleà. El valor és el valor de l'identificador i és únic per cada element. És a dir, no hi ha dos elements identificats de la mateixa classe amb el mateix valor enter. Utilitzem valors enters ja que és més eficient que utilitzar Strings i el rang és suficient pels conjunts de dades que tractarem. El booleà ens diu si l'element està actiu o no. Això ens permet conservar les dades dels elements tot i que el client els esborri i fer-les servir pels algorismes de recomanació. És a dir, si un usuari es dona de baixa, en comptes d'esborrar-lo de la base de dades, fem el seu Id inactiu. Aleshores, el client no el veurà però les seves dades rellevants pels algorismes de recomanació (valoracions,...) no s'hauran perdut.

- **Item**

Un Item té un identificador i és d'un determinat tipus i té un conjunt d'atributs amb nom i ha rebut un conjunt de valoracions. Els atributs i les valoracions són TreeMaps per poder iterar i fer consultes eficientment. Podem obtenir la distància entre dos ítems com la suma de les distàncies normalitzades de tots els seus atributs. Això ens permet donar la mateixa importància a tots els atributs independentment del rang de valors que puguin prendre les seves distàncies.

- **TipusItem**

Un ítem és d'un determinat tipus. Representem el tipus d'un ítem amb un nom i un conjunt de tipus d'atribut. Podem deduir el tipus d'un ítem a partir de la representació d'un Item com a conjunt de Strings intentant obtenir el valor de cada String. Si tenim un conjunt d'ítems, podem obtenir un tipus d'ítem que ens permeti representar tots els ítems (això és cert perquè, en el pitjor cas, podem considerar tots els atributs com a atributs categòrics o textuais).

- **recomanador.filtre.Filtre**

Un filtre ens permet obtenir recomanacions considerant només un subconjunt dels atributs d'un determinat tipus d'Item. Representem un filtre com el conjunt dels noms dels atributs.

- **recomanador.filtre.FiltreExclusiu**

Un filtre és exclusiu si volem incloure a la recomanació tots els atributs del tipus d'ítem excepte els indicats pel filtre.

- **recomanador.filtre.FiltreInclusiu**

Un filtre és inclusiu si volem incloure a la recomanació només els atributs del tipus d'ítem indicats pel filtre.

- **recomanador.Recomanador**

La classe recomanador ens permetrà aplicar el filtre al conjunt d'ítems abans d'aplicar el mètode recomanador. Els filtres són una funcionalitat extra que hem volgut implementar. L'estructura bàsica d'aquesta classe està implementada però la lògica no està completa i ho farem properament. També hi haurà un recomanador híbrid.

- **recomanador.RecomanadorCollaborative**

Representarà un recomanador amb el mètode col·laboratiu.

- **recomanador.RecomanadorContentBased**

Representarà un recomanador amb el mètode basat en el contingut.

- **atributs.TipusAtribut**

Cada atribut pot ser d'un tipus diferent. Un tipus s'identifica per quin tipus de valor tenen els seus elements i per quina distància s'aplica per trobar la distància entre dos atributs del mateix tipus.

- **atributs.distancia.Distance**

Una distància representa la funció de distància que s'aplica entre dos valors d'atributs. Com que volem les distàncies normalitzades, també guarden els paràmetres necessaris per obtenir els factors de normalització.

- **atributs.distancia.DistanceDiferenciaDeConjunts**

Representa la distància entre dos conjunts. El factor de normalització és el doble de la norma màxima, ja que donats conjunts amb mides inferiors o iguals a un valor, el valor màxim de la distància entre qualsevol parella de conjunts és el doble de la norma màxima (que té lloc quan els dos conjunts són totalment disjunts i cada un d'ells té la mida de la norma màxima).

- **atributs.distancia.DistanceDiscreta**

Representa la distància discreta entre dos elements simples. Aquesta distància ja pren valors entre 0.0 i 1.0.

- **atributs.distancia.DistanceEuclidiana**

Representa la distància euclidiana entre dos elements numèrics. El factor de normalització és el doble de la norma màxima, ja que donats valors en la bola de R^n centrada en el zero de radi la norma màxima, la distància entre qualsevol parella de punts és, com a màxim, el diàmetre de la bola.

- **atributs.distancia.DistanceLevenshtein**

Representa la distància de Levenshtein entre dues Strings. El factor de normalització és el doble de la norma màxima, ja que donades dues Strings de normes inferiors o iguals a un valor N, per passar de l'una a l'altra com a màxim necessitem esborrar N caràcters i afegir-ne uns altres N.

- **atributs.distancia.DistanceZero**

Representa la distància zero. Aquesta distància sempre retorna zero. Serveix per poder ignorar certs atributs quan calculem les distàncies perquè són irrellevants o perquè calcular la distància entre una parella és massa costós. Per exemple, si tenim un atribut que és un text, pot ser que calcular la distància de Levenshtein sigui massa lent i volguem ignorar-lo.

- **atributs.valors.ValueAtribut**

El valor d'un atribut pot ser d'un determinat tipus. ValueAtribut és la classe abstracta que representa el tipus d'un valor d'un atribut. Pot ser simple (booleà, categòric, numèric o textual) o bé pot guardar un conjunt de qualsevol dels tipus simples. Creiem que amb aquests quatre tipus podem incloure tots els atributs rellevants. La distinció entre categòric i textual és conceptualment clara, encara que ambdós es guardin com a Strings.

- **atributs.valors.ValueBoolea**

Guarda un booleà.

- **atributs.valors.ValueCategòric**

Guarda una categoria.

- **atributs.valors.ValueConjunt**

És un ValueAtribut que guarda un conjunt de ValueAtributs. És una classe abstracta.

- **atributs.valors.ValueConjuntBoolea**

Guarda un conjunt de ValuesBooleans.

- **atributs.valors.ValueConjuntCategòric**

Guarda un conjunt de ValuesCategòrics.

- **atributs.valors.ValueConjuntNumèric**

Guarda un conjunt de ValuesNumèrics.

- **atributs.valors.ValueConjuntTextual**

Guarda un conjunt de ValuesTextuals.

- **atributs.valors.ValueNumèric**

Guarda un valor numèric.

- **atributs.valors.ValueTextual**

Guarda un valor textual.

Pablo Vega Gallego

- ConjuntItems

Conjunt ítems crea un conjunt d'ítems del mateix TipusItem, a partir de, o bé una TaulaCSV que contindrà les dades trobades a un fitxer CSV o altrament directament amb un TipusItem ja definit i un TreeMap<Id, Item> on enllacem l'Id del ítem amb l'ítem en concret.

Hereta de la classe ConjuntIdentificat<Item>, per a tenir una estructura semblant a tots els conjunts.

- Contenedor

Classe abstracta que representa un contenidor de dades del fitxer desitjat a llegir.

D'aquesta manera reduïm la quantitat de relacions que pot haver-hi si decidim llegir o escriure altre fitxer que no fos del tipus CSV en el cas de la primera entrega.

- EscriptorDeFitxers

Classe abstracta que escriu un contenidor a un fitxer.

- LectorDeFitxers

Classe abstracta que representa una funció genèrica que emmagatzema les dades a un contenidor.

- recomanador.ConjuntRecomanacions

Classe que permet representar un conjunt de recomanacions.

Es construeix amb el conjunt de recomanacions desitjat sempre mantenint l'ordre de la llista pel valor de Recomendacio.seguretat. És a dir, la Recomendacio que tingui una seguretat més gran, serà la que trobem en primera posició.

- recomanador.Recomanacio

Classe que permet representar la seguretat d'una recomanació d'un ítem.

Es construeix la variable recomanació a partir d'un Id d'ítem i un double que representa la seguretat d'aquella recomanació.

- csv.EscriptorDeCSV

Donada una TaulaCSV, escriu el contingut de la taula a un fitxer amb el format CSV.

És una classe que només implementa funcions per escriure CSV a partir d'una TaulaCSV i la funció abstracta de lectorDeFitxers.

- csv.LectorDeCSV

Donat un fitxer que tingui el format .csv, llegeix el contingut del fitxer i el transforma en TaulaCSV.

Classe específica que implementa la funció abstracta de LectorDeFitxers i la de lectorCSV.

- **csv.TaulaCSV**

Representa el contenidor de dades per emmagatzemar el contingut d'un fitxer CSV.

Classe que implementa les funcionalitats per introduir les dades del CSV a una taula i manté la integritat d'aquesta perquè no hi puguin haver incongruències a l'hora de llegir els fitxers.