Lab 1: Association Rules Mining

**Due date:** Thursday, April 12, midnight.

## Lab Assignment

In this assignment you will analyze collections of market baskets and will determine *frequent itemsets* and *association rules* present in the collections.

### Assignment Preparation

This is a pair programming assignment. Each student teams up with a partner. You get to select your partner at the beginning of the Thursday, April 5 lab.

**Note:** Enrolled students select as partners other enrolled students. Students on the wait-list select as partners other waitlisted students.

## Datasets

You are given three different datasets to work with on this assignment: `EXTENDED BAKERY`, Fantasy Bingo, and `TRANSCRIPTION FACTORS`. The first one is a synthetic dataset simulating actual market baskets (purchases made at a bakery). The second one consists of lists of authors whose books different people have read in a year-long reading challenge called "fantasy bingo". The third dataset comes from our colleagues in the Animal Science program (Dr. Dan Peterson). The dataset documents an experiment ran on mice and cows designed to establish how different conditions affect the abundance of genes sequenced in specific cells.

For each dataset we ask the questions that are translated into the need to find frequent itemsets, and, in some cases - association rules of specific type. We refer to these special frequent itemsets and association rules as Skyline Frequent Itemsets and Skyline Association Rules.

## EXTENDED BAKERY Dataset

The assignment is based on the `Extended BAKERY` dataset. The dataset is a modified version of the CSC 365 `BAKERY` dataset. The `Extended BAKERY` dataset describes the work of a *chain* of bakery shops that sell a variety of pastries and drinks to customers.

The data provided to you for this assignment is the information about purchases made by the bakery chain customers in various locations. The four sub-datasets contain information about 1000, 5000, 20,000 and 75,000 purchases.

For each sub-dataset we provide three files representing the same set of receipts. For simplicity, each file represents the exact purchases: i.e., which items were purchased on which receipt, but **omits other information from the dataset:** the store location, the employee who rang the purchase, the date of the purchase. Additionally, the *quantity* of the purchased item is omitted in two representations of the three listed.

The full description of the dataset is below.

**Access to the dataset.** All CSV files can be downloaded from the Lab 1 data page

http://www.csc.calpoly.edu/ dekhtyar/466-Spring2018/labs/lab01.html

The list of **market baskets** for each dataset size is available in **three formats**:

1. Sparse Vector format. Files `XXXX-out1.csv`. Each line of the file has the following format:

   - First column is the receipt Id.
   - Subsequent columns store a list of goods purchased from the bakery ordered by `Goods.Id`.

   **Example:**

   ```
   1, 7, 15, 44, 49
   ```

   *Receipt 1 contained purchases of a Coffee Eclair, a Blackberry Tart, Bottled Water and a Single Espresso.*

2. Full Binary Vector format. Files `XXXX-out2.csv`. Each line of the file has the following format:

   - First column is the receipt id.
   - 50 columns follow, with 0s or 1s as values. A 1 in column $i + 1$ means that a good with `Goods.Id` of $i$ was purchased on the receipt.

**Example:**

```
1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1
```

3. Items Table. Files `XXXXi.csv`. Each line of the file represents a single tuple from the `Items` table. The columns are:

   - Receipt(number), Quantity, Item

   **Example:**

   ```
   1,3,7
   1,4,15
   1,2,49
   1,5,44
   ```

   (note, that the item IDs may be out of order)

## Fantasy Bingo Dataset

The Fantasy Bingo Dataset is a relatively small dataset consisting of the information about the books read by 243 individuals for a challenge we refer to as Fantasy Bingo. Each person has read, as part of a year-long challenge around 25 different books[1].

We are interested in learning if there are frequent itemsets of writers (authors of the books) that occur in this data, and whether these frequent itemsets lend themselves nicely to association rules.

For this assignment, we are releasing two data files for you, `bingoBaskets.csv` and `authorlist.psv`.

**List of authors.** `autorlist.psv` contains a full list of authors who wrote the books read by the participants of the Fantasy Bingo. The file is in the Pipe-Separated Values format, and consists of records indexing individual unique authors. The format of the file is

```
Id | Name
```

(notice the spaces around the pipe character - there for readability, your parser needs to take that into account)

The first column is the numeric Id assigned to the author, and the second column is the name of the author. Sample lines are:

```
1 | *N/A
2 | Aaron, Rachel / Bach, Rachel
```

[1] There are caveats to this description, but for the purposes of this assignment they are not relevant.

```
3 | Aaronovitch, Ben
4 | Abbey, Kit
5 | Abbey, Lynn
6 | Abercrombie, Joe
```

**Note:** The value of `"*N/A"` is a special value that should be ignored if it is ever discovered in your market baskets (it means that a reader submitting their list of books failed to read one or more books to meet the challenge).

**Baskets.** The file `bingoBaskets.csv` contains the list of authors for each of the participants in the fantasy bingo. The file has the following format:

```
BingoCardId, AuthorId1, AuthorId2, ..., AuthorIdK
```

Here, `BingoCardId` is the unique Id of each basket (each reader's submission/bingo card), and `AuthorIdX` is an id of an author (see first column of the `authorlist.psv` file).

Here are a few sample lines:

```
12,3, 68, 91, 166, 183, 224, 237, 259
13,2, 3, 48, 91, 166, 171, 216, 217, 218, 251
```

Please note that all author Ids in each line are sorted by author ID (and all author ids in general are sorted in lexicographical order by the author's last, and first names).

## Transcription Factors Dataset

The `TRANSCRIPTION FACTORS` dataset is a collection of information about the participation of specific proteins (called *transcription factors*) in the experssion of certain genes in the mammary gland tissues of different animals.

In short, a gene plays a role of a market basket (or, to be more exact - a role of a label of a market basket, the same way the receipt number playes the role of the label of a market basket in the `EXTENDED BAKERY` dataset). With each gene, a collection of transcription factor names is associated.

The dataset comes with four files, `genes.csv`, `factors.csv`, `factor_baskets_full.csv` and `factor_baskets_sparse.csv`.

`genes.csv` contains the list of genes used in the study. The first line of this file provides information about its format:

```
expgene,geneabbrev,experiment,species
```

Here, `expgene` is the unique Id of the gene (experiment) in this dataset (there are 47 experimental data points in this dataset), `geneabbrev` is the abbreviated name of the gene studied, experiment is a short label describing the experiment (usually "Laura Strand - May 2012" indicating, that the experiment was performed by Laura Strand - then an MS student in the Animal Science program in May 2012), and species is the species from which the gene was taken. This file is used to simply provide meta-data for each row/market basket of the dataset.

factors.csv has a list of 412 transcription factors (items) that were observed in the study. For each transcription factor, its numeric id is recorded in the first column, and its abbreviated name – in the second. This file essentially stores "column names" for the market basket dataset.

factor_baskets_full.csv and factor_baskets_sparse.csv store the collection of market baskets in two different formats. The full format, found in the factor_baskets_full.csv file is

```
expgene,tf_id,occurrences
```

Here, expgene is the id of the gene (experiment), tf_id is the id of the transcription factor, and occurrences is the number of times the transcription factor was observed. This value is not required for your work on this lab, but you are allowed to take it into account if you want to.

A single market basket is represented by a sequence of rows in the factor_baskets_full.csv file.

factor_baskets_sparse.csv stores each market basket in a single record in the format

```
expgene, tf_id_1, occurrences_1, tf_id_2, occurrences_2,...
```

Here each line starts with the unique id of the gene (experiment), and continues with the pairs of the transcription factor id and number of occurrences of the transcription factor for all transcription factors that had occurred in the experiment.

## Mining Frequent Itemsets and Association Rules

Your task is to discover the association rules that exceed specific given values of *minimum support* and *minimum confidence.*

As discussed in class, mining association rules is a two-step process. On step one, the goal is to discover *frequent itemsets* with support exceeding *minsup.* On step two, the goal is to discover specific *association rules* found within the discovered frequent itemsets.

Algorithms for discovery of both frequent itemsets (Apriori) and association rules (genRules) have been discussed in class together with implementation strategies.

## Skyline (Maximal) Frequent Itemsets and Association Rules

Association rules mining tends to discover **a lot** of rules in any given dataset. This is due to permutation properties of the rules (e.g., if $A, B \to C, D$ is an association rule, then so are $A, B, D \to C$, $A, B, C \to D$) and due to the large number of items in a typical dataset.

To make results of your work *observable*, we will be interested only in so-called **skyline** or **maximal** frequent itemsets and association rules.

**Definition.** A frequent itemset is called a **skyline** (**maximal**) frequent itemset, if *it is NOT a subset of any other frequent itemset.* An association rule is called a **skyline** association rule if its right side and its left side form a **skyline** frequent itemset.

*Informally*, **skyline** or **maximal frequent itemsets** are those, that cannot be extended further to form larger frequent itemsets. To constrain the output of your work, you need only to report **skyline** frequent itemsets.

Furthermore, to simplify the process of mining association rules, you shall report **only skyline** association rules in which the right side of the rule contains *a single item.*

## Minimum Support and Minimum Confidence

Each of the three datasets may require *tuning* the `minsup` and `minconf` parameters - i.e., the minimum support for the frequent itemsets, and the confidence for the association rules. *It is your goal for this assignment to properly tune these parameters and find the best values of `minsup` and `minconf`.*

For the `EXTENDED BAKERY` dataset, all association rules and frequent itemsets were seeded and they are separated from any randomly occurring itemsets by a fairly large gap in support. You need to discover the `minsup` threshold that surfaces these skyline frequent itemsets, as well as the `minconf` value that exposes the actual association rules.

For the `TRANSCRIPTION FACTORS` and `Fantasy Bingo` datasets, you need to explore the data in order to come up with reasonable ranges for `minsup` and `minconf`. You can start by finding the support for each singleton itemset in each dataset, and building the frequency historgram. This should give you an idea what type of support (in terms of absolute or relative values) can be considered "relevant" in each case. From there, proceed to use to code you develop as a research tool to find the best parameter values.

## Code

You can use any langauge you want. Python is a good choce due to powerful tools for parsing and manipulation of data. Java is a good choice because it will allow you to have a well-designed implementation of all functionality.

You are to write **the entirety of the code** from scratch without borrowing from any of the existing machine learning packages that may be available in your programming language of choice.

If you are using Python, you are allowed to use NumPy for data manipulation and parsing. You can use `Pandas` data frame manipulation functionality. In Java or other programming languages, you are allowed to use comparable packages/libraries/APIs.

## Deliverables

You shall discover **skyline frequent itemsets** and **skyline association rules** in each of the four `EXTENDED BAKERY` datasets. Additionally, discover **skyline frequent itemsets** in the Fantasy Bingo and `TRANSCRIPTION FACTORS` datasets ( you can also look for skyline association rules in the Fantasy Bingo dataset, but we do not need association rules for the `TRANSCRIPTION FACTORS` dataset).

Submit the following:

- A report containing the list of skyline frequent itemsets and the list of skyline association rules you discovered in each of the four datasets.

  For each *skyline frequent itemset* specify:

  1. All *items* in it. Use `Goods.Flavor` and `Goods.Food` attribute values to describe each item in the EXTENDED BAKERY dataset. Report author names for the Fantasy Bingo dataset. Report transcription factor names for the `TRANSCRIPTION FACTORS` dataset.
  2. The support of the itemset.

  Notice, that `Goods.Flavor` and `Goods.Food` attributes (as well as the author names and transcription factor names) are NOT present in the input market baskets (all the formats described above contain only numeric indexes). It is the job of your software to report these attributes given the good ids.

  For each *skyline association rule* specify:

  1. All items on the left side of the rule (`Goods.Food`+`Goods.Flavor`).
  2. The item on the right side of the rule (`Goods.Food`+`Goods.Flavor`).
  3. The support of the rule.
  4. The confidence of the rule.

- Any software you have written to discover association rules.

  In general, a program for association rules discovery should take as input the following parameters:

  1. Filename. Name of the CSV file containing the dataset. Your program can use any of the formats made available to you.
  2. minSup. The minimum support number for frequent itemset and association rule discovery.
  3. minConf. The minimum confidence number for association rule discovery.
  4. additional filenames. Names of any additional files needed to produce output.

  Optionally, the name of the output file may be passed to your program as well.

  Additionally, you may include any optional flags that specify whether:

  - all rules/frequent itemsets or skyline rules/frequent itemsets should be returned. (the default behavior is to print skylines).
  - only rules, only frequent itemsets or both rules and frequent itemsets shall be printed.

  Given the need to discover *only* the skyline frequent itemsets (but not the association rules) for `TRANSCRIPTION FACTORS` dataset, it makes sense to either have two executables, or to include the last flag from the list above.

  Generally speaking, you may elect to implement your software in any way you like: e.g., you can split reading/parsing data, frequent itemset search and association rules discovery into three separate pieces of code if this is more convenient for you.

- A `README` file which contains the following information (at least):

  - Names of all students in the pair/team.
  - Specification of which type(s) of input format your program(s) take(s).
  - Instructions on how your code should be run. This is especially important if you implemented association rules mining as a sequence of separate programs.

**Note:** Frequent itemset/association rule lists that you submit can be the output of your program(s), as long as your program output follows the guidelines specified above.

**Note:** Each `EXTENDED BAKERY` dataset incorporates a specific set of association rules (and frequent itemsets), that stand out. You may have to try your program with a number of minConf and minSup parameters until you discover all of them, but overall, the **separation between the frequent itemsets/association rules** and all other itemsets/candidate rules is **very robust**.

### Training Dataset

To help you calibrate your discovery process, we are providing one more dataset for you. The dataset contains 1000 market baskets and has the following **association rules** seeded in it:

Lemon Cake ⟶ Single Espresso
Blackberry Tart ⟶ Apple Danish
Napoleon Cake ⟶ Gongolais Cookie
Apple Tart and Berry Tart ⟶ Blueberry Tart

All these rules have support of at least 10% and a confidence of at least 90%. Note that other rules (e.g., Berry Tart and Blueberry Tart ⟶ Apple Tart will also exist in the dataset and would need to be reported).

The training dataset can be downloaded from the course web page. The `example.zip` file contains the following four files inside the `example` directory:

| | |
|---|---|
| `out1.csv` | market baskets in **sparse vector** format |
| `out2.csv` | market baskets in **full binary vector** format |
| `lab2-example-output` | output of the TA's rule mining program |
| `Rules2.xml` | an XML file specifying the rules found in the dataset |

### Submission Instructions

**Report submission.** While I prefer hardcopies of the reports, reports in soft copy only will also be accepted. Reports shall be word-processed, with the results of running your program included where necessary. Submit the reports by April 12, midnight. If you have hard copies of your report, bring them to class on **April 17**.

**Code submission.** You will use the handin tool to submit your. Each pair submits exactly one copy of all materials. Put all your files in a single archive (zip or gzipped tar), name it `lab01.zip` or `lab01.tar.gz` and submit as follows:

```
$ handin dekhtyar lab01-466 lab01.zip
```