Final Year Project: Interim Report

Acquiring structured knowledge from Flickr

Alex Thomson

BSc Computer Science

1033702

Project No: 226

Supervisor: Steven Schockhaert

Moderator: Alun D Preece

Abstract

The goal of this project is to discover whether location based services can benefit from associative rules derived from a large set of Flickr photo metadata. The rules will reflect common associations between elements (specifically tags) within the dataset.

The purpose of this interim report is to document the research that has now identified the methodology I will be using to derive such rules, specifically:

- How photos will be clustered and using which clustering algorithms.
- What association rule mining is and how it can be applied to clustered data gathered from the Flickr API.
- How results can be pruned to remove unnecessary and undesirable tags to improve how meaningful the rules identified are to our goal.

I intend to initially work with UK subset of photos and subsequently cluster these photo's using a clustering algorithm to retrieve a workable dataset that can return robust results, I will then use an associative rule mining algorithm to retrieve rules from the remaining dataset. Feature selection and constraints will be used to prune and refine results to remove unwanted results and rules.

Table of Contents

1. Project Recap and Report Outline

This project aims to produce a list of common associations found between tags (or groups of tags) in photos obtained from Flickr.

The project is split into two major reports; the interim report and the final report. Being the interim report, this will cover the majority of the research into the techniques and methodology required to produce a manageable dataset and interrogate it for meaningful rules. The final report will cover the actual implementation and results produced by the methods discussed in this report.

There is a certain chronological order of steps to discovering rules within the dataset and as such, research surrounding these steps is included in this report. The major steps are:

1. Obtain dataset from Flickr
2. Prune the dataset and retain only photos within a manageable geographical location
3. Cluster the photos into meaningful groups
4. Interrogate the clusters for common associations
5. Investigate and refine the interrogation process to find more meaningful rules

Within these major steps, there are considerations that could drastically affect how long these steps takes to complete, such as the availability of existing tools for harvesting rules from datasets or whether a program will need to be written and implemented as part of the project.

The initial plan for this project outlined certain core objectives that translated directly into deliverables for this report, namely:

- A background study on Flickr
- A background study on the Flickr API
- A detailed section on ARM and how it can be used within the context of this project using previous research
- Available and existing tools for ARM
- Alternate data mining techniques
- Database suitability
- Research Hypothesis

Consequently during my research, it has become apparent there are other important topics that need to be covered such as:

- Expected dataset and pruning it for a manageable subset
- Clustering algorithms
- Feature Selection

These have also been covered in this report.

2. Background Studies and the Initial Dataset

2.1 Flickr Background

Flickr is a large, community driven image and video sharing website created by Ludicorp in 2004. It was acquired by Yahoo! In 2005 and has risen dramatically in popularity since then. As of today www.flickr.com has a global traffic ranking of 56 [1].

Flickr is a social media website that focuses primarily on user generated content (specifically images and videos). Registered users have tools at their disposal to help view, organise and edit content they have access to. For the purpose of this report I have created my own account so I can fully explore and understand the functionality Flickr offers users. Some of the key features for users of Flickr include:

Uploading content – Users can upload photos and videos from their local machine to their Flickr account so the content can be displayed and accessed online. When uploading, the content can be given descriptive tags to help Flickr best organise the content for when other users are searching for specific keywords. If the file being uploaded doesn't already contain geographical information, users are encouraged to 'plot' the location of the image/video onto a map.

Searching for content – Flickr incorporates a search bar that allows users to search the database for specific keywords, these keywords are compared against filenames and tags and the most relevant results are returned for the user.

Profiles – Like many social media websites, Flickr's users have the ability to set up personal profiles, users can set personal information, the privacy level of their profile, the privacy level of content they upload and what and which way content will be displayed to themselves and other users when looking at their profiles.

Users can also view the profiles of other users (depending on the privacy settings of the user) where content can be explored and if desired – contact can be made via Flickr's internal messaging system.

One of Flickr's main attractions is the ability to embed uploaded content to other webpages, this means users can use the direct link address of content they have uploaded to Flickr and use it in other forums/blogs/webpages – this alleviates the necessity for forums and blogs to allow users to upload files.

2.2 Using APIs

Many popular websites now incorporate APIs (Application Programming Interfaces) due to the demand by external developers to access and use large datasets without crawling and scraping data from websites (which will likely be resource draining for the developer and website anyway) to gather the information. Requests through a web API typically extract data directly from the website's database and then return the data to the user in a pre-specified format, typically JSON or XML.

Datasets can then be analysed and utilised by developers for other purposes, often utilised in web mashups where multiple sources of data gathered from open or commercial API's are bought

together to create a new service. Although not necessarily a mashup, the concept of combining data from multiple sources online is popularly used by price comparison sites.

Access to API's is generally offered to developers under certain conditions that usually restrict developers from using the data in a way that may conflict with the website.

2.2.1 The Flickr API

A regular developer may utilise Flickr's API to request no more than 3600 requests per hour. This can be expanded at Flickr's discretion and is usually the case in research purposes (i.e. non-commercial). However, the aim of this project is not to re-use the Flickr API but instead gather one large initial dataset and use that.

There are multiple formats that the Flickr API can export requests in; XML, JSONP, JSON and PHP Serial. These formats are organised versions of the requested data, to make use of the data the returned file would need to be parsed to strip out unnecessary syntax and headers only leaving the raw data.

2.2.2 Available Data

When photos are taken, lots of information about the photo is stored by the device capturing the image, commonly referred to as its Exif data (Exchangeable image file format). Information stored is heavily dependent on the device, however common attributes are: manufacturer, model, orientation, software, date and time, compression, x-resolution, y-resolution, resolution unit, exposure time, exif version, flash.

Modern devices tend to also record the latitude, longitude and altitude of photos provided there is a GPS receiver present. This is of particular interest to the project as these location attributes can be used to locate the origin of the photo and this will be necessary to later cluster the photos.

For the purpose of this project I have been given access to a large dataset previously obtained from Flickr, the format of which is; unique photo id, owner of the photo, latitude, longitude, tags, time stamp, and the Flickr accuracy level. For example:

1,10000137@N04,-33.977041,25.648849,dolphin dolphins,1199284643000,16

It may be necessary at points to strip the dataset of certain unnecessary columns, such as the Flickr accuracy level, which is '16' for every photo. As this point I am familiar enough with java to write a small program that can iterate over the dataset and alter it as necessary.

2.3 The Dataset

The initial dataset is an extremely large file consisting of several million photos metadata, with all photos containing GPS origin data or user specified coordinates.

A majority of the algorithms and computation performed throughout this project will be a trade-off between computational difficulty and gaining meaningful results. Narrowing the initial dataset into something manageable is very important for the project. I plan to firstly narrow the dataset by only

using a subset of photos within Great Britain for this project; however it will be possible to expand this selection later in the project given time allowance and feasibility.

To select the UK subset I will need to iterate over the initial dataset and select only photos that fall within Great Britain, to do this the origin coordinates of the photo must fall within the most extreme points of Great Britain [8]:

Northern Most Point: 58.666667,-3.366667

Southern Most Point: 49.85,-6.4

Eastern Most Point: 57.583333,-13.683333

Western most Point: 52.481167,1.762833

3. Clustering Photos and Discovering Rules

3.1 Clustering Algorithms

There are several potential approaches to clustering the photos for analysis within the project; the important factor is deciding what types of cluster are desired. For example, would it be more beneficial if all clusters covered an equal area geographically, should each cluster contain the same number of photos regardless of size and how many clusters are needed to identify unique information pertaining to that particular cluster.

A very popular clustering algorithm is the 'K-means' algorithm used commonly within data mining. The K means clustering algorithm aims to create K clusters from a dataset whilst minimising the squared error in the clusters, which is the sum of distances from each point within the cluster and the cluster centre.

3.1.1 K-Means

The process for the algorithm is as follows:

- Set K to how many clusters are desired from the dataset
- The algorithm will randomly pick K positions within the dataset
- Each datapoint will associate itself with its closest K
- Each K will now revaluate and pick it's mean position (most central point given all datapoints in K) to be the new set of K
- The above two steps will repeat until the change in mean position (when revaluating) falls below a predefined threshold or a maximum number of iterations has been carried out. This is why this algorithm is known to only find the local optimum is usually ran multiple times to account for this.

3.1.2 K-medoids

Similar to K-means, K-medoids instead uses datapoints rather than mean positions for assigning clusters. It is suggested this could be more robust to noise and outliers (scarce and obscurely positioned photos in this circumstance). [2] [6]

3.1.3 Clustering by town/city names

If implementing a suitable clustering algorithm proves too difficult or less desirable, it is possible to cluster the photos using the same technique I will use to gather the subset of Great Britain photos; by this I mean identifying the boundaries of the major cities (an available coordinates list for cities is available online) and iterating over the dataset separating photos into clusters by city.

There two main reasons this is undesirable and effectively a last resort:

1. The list available only represents each city as a central point; it would be difficult to set boundaries to cluster photos without setting a variable that allows coordinates to be within a certain numerical proximity of the central point. However this again poses problems as some cities are larger than others thus defeating the possibility of a global variable.

   There is an API service available from Yahoo! that allows reverse geocoding where it can be passed pairs of coordinates and it will return which city the coordinates are located. This is again likely restricted to a certain number of requests per day and considering this project would need to use several hundred thousand as a minimum this method doesn't seem likely feasible.

2. It will be difficult to scale; it will be time consuming in itself to identify boundaries and implement a program that will automatically separate lots of photos into city clusters. If later in the project the results are unsatisfactory, it will be unfeasible to scale the clusters to encapsulate smaller regions without spending vast amounts of time manually identifying locations and their respective boundaries.

3.2 Association Rule Mining

The idea of mining for associative rules was proposed by Agrawal in 1993. It is aimed at discovering interesting and meaningful correlations and patterns between associations of elements within a set of transactions [3].

More generally ARM (Association Rule Mining) is the act of searching for common associations between elements within a large dataset. Any frequent and re-occurring associations between elements become known as rules and are given a confidence level (A percentage of how strong the association is) and all rules are then ordered by this confidence level. We then analyse each of the rules, starting with the strongest rule, to identify rules that could be meaningful to our goal.

Used across many contexts, ARM is a powerful tool for discovering unknown relationships that can be of great significance. It is commonly used by supermarkets to analyse customer purchasing habits

and use this knowledge to organise the shop floor, placing items frequently bought together close to each other on shelves. As an example of scope, it could also be used by car manufacturers to analyse customer repairs to look for associations between car model and faulty components. With such a variety in car models and components there will likely be thousands of rules found of varying interest and confidence.

Although this project differs greatly in context and motivation, it has the same meaningful decisions that will affect what results are produced and how these results could theoretically be interpreted and utilised in a corporate/business situation; this is an area I intend to explore near the conclusion of this project.

One of the key hurdles with ARM is analysing your dataset in a way to only retrieve meaningful rules whilst limiting the amount of iterations over the dataset. As ARM is typically used on large to massive datasets to gain robust results, simply comparing all elements against all elements consumes a lot of resources and isn't time effective. As a majority of any ARM analysis is automated, limiting the amount of data whilst maximising the likely hood of discovering meaningful rules is the main priority. This can be done by introducing mathematical limitations and feature selection.

3.2.1 Rule mining in general

Leaving aside the differing algorithms and approaches to generating rules from within a dataset, there is a fundamental and common structure for discovering meaningful rules.

A rule – A rule is a re-occurring association between elements of a transaction. It will have an antecedent and a consequent, that is to say a final rule may look something similar to:

> Given that X1,...,Xn occur, it is likely that Y also occurs.

How likely it is will be defined by how strong the rule is, which are governed by its support and confidence.

Support – the support of a rule is essentially how large a presence it has across the dataset, for example if we have a database with 1000 transactions and we have a rule we are analysing that contains 3 elements, if all 3 elements occur together in 100 of the transactions then the rule is said to have 10% support.

Confidence – The confidence of a rule is more about how often it is correct given its support. For example we have a rule that when A and B occur, C also occurs. To find the confidence of this rule, we need to find how many times A, B and C occur together in a transaction and divide it by how many times A and B occur together in transactions.

Constraints like support and confidence will directly impact the time taken to discover rules and their calibre. It may be wise for the first execution of the algorithm to set both variables at the same limit and change them separately afterwards to gain a distinction of how they each affect the type of rules being discovered.

3.2.2 AIS algorithm

In his initial paper, Agrawal initially proposed the AIS (artificial immune system) algorithm which is grounded around the concept of association, that is to say the algorithm itself was based on the concept of our immune system and its ability to recognise antigens and produce the associated antibodies.

Unfortunately this algorithm was quickly superseded because of its nature to decline in performance and quality of results as the dataset got larger – its main disadvantage was that it iterates over too much meaningless data due to the lack of constraints.[4]

3.2.3 Apriori algorithm

A year later Agrawal proposed a much more efficient algorithm for mining associations [5]. This algorithm makes use of a manually defined support level to help remove infrequent itemsets.

This apriori algorithm works as follows:

Starting with an initial database of transactions, we count how many transactions each element appears in.

T1	{Item1, Item2, Item3}
T2	{Item2, Item3, Item4, Item5}
T3	{Item 1, Item4, Item5}
T4	{Item2, Item3, Item5, Item6}
T5	{Item2, Item3, Item5, Item6}

Item1	2 transactions
Item2	4 transactions
Item3	4 transactions
Item4	2 transactions
Item5	4 transactions
Item6	2 transactions

We then apply our predefined support level and remove elements that match or exceed it. For example >40% would mean appearing in more than 2 transactions so our resultant table would remove item1, item4 and item6.

Each combination of remaining 2 elements is then ordered and again the database is searched to see how many transactions each appear in.

Item2	4 transactions
Item3	4 transactions
Item5	4 transactions

{Item2, Item3}	4 transactions
{Item2, Item5}	3 transactions
{Item3, Item5}	3 transactions

Again any combinations that fall below our support level are dropped (none in this example) and we move on to making sets of 3 items.

Now comes an important step called the self-join, it's at this point to take the existing combinations with the same first item and join them, eventually leaving us with another list of combinations that need to be checked against our support prerequisite.

{Item2, Item3}	4 transactions
{Item2, Item5}	3 transactions
{Item3, Item5}	3 transactions

{Item2, Item3, Item5}	3 transactions

The process would be the same if we had enough sets of 3 items to perform another self join to create sets of 4 items, however it is important to note that after we exceed 3 items it is necessary to match all items except the last, for example:

{Item2, Item3, Item4}
{Item2, Item3, Item6}
{Item2, Item5, Item6}

We would only look to join the first two item sets as they have matching items except for the final one, the resultant set would look like:

{Item2, Item3, Item4, Item6}

This algorithm is ideal for this project because it is well researched and suited to the type of dataset expected, that is clusters (i.e. databases) of transactions (i.e. photos) that contain multiple and likely reoccurring items (tags).

3.3 Alternative Rule Mining Techniques

There are various contrasting approaches to data mining that can be utilised before/after/with each other for a multitude of reasons. For this reason I have identified some alternative methods of rule mining hoping that my familiarity with them could potentially assist later in the project when assessing the results or data from a different perspective.

3.3.1 Contrast set learning – Contrast set learning is a method of taking already classified sets and studying attributes to determine what separates them from the other sets; it is essentially reverse engineering sets of items to discover why they have been classified as they have.

3.3.2 K-optimal pattern discovery – somewhat an extension of ARM, K-optimal proposes that sometimes a minimum support threshold is not desired as rare and important rules may be missed. It focuses more on gaining meaningful constraints such as limiting the consequent to a single condition. This method seems particularly useful if you understand the dataset very well and have a good insight as to where the meaningful rules may be discovered.

3.3.3 Sequential Association Rules – The recognised style of identifying item associations within transactions does not usually account for temporal data, i.e. which order items were added to the transaction relative to other items.

3.4 Feature Selection

Feature selection within the context of this project is essentially the process of identifying and removing irrelevant and redundant tags assigned to photos in a cluster. Although this process will take place before the identification of rules, there will be insights gained from investigating the first set of identified rules that will lead to further pruning of tags.

3.4.1 Heuristics

The goal of the project is to identify rules that may benefit location based services, so although "if 'beach' then 'sun' is also likely" or "if 'wales' then 'millienium stadium' is also likely" are examples of rules that could be utilised to improve a service that locates sunny beaches or city hotspots, a rule like "if 'birthday' then 'balloon is also likely" is completely irrelevant for our context.

Working from this, it would be possible to build a list of likely irrelevant tags and have them removed from all of the photos before interrogation and not only does this decrease computing time but also how robust the results are.

3.4.2 Mathematical Limitations

Putting a quantitative value against a tag (string of text) and being able to measure its meaningfulness against all other tags would help decide a cut-off numerical value for pruning the database of undesirable tags.

3.4.3 Chi Squared

Chi Squared is a statistical method that can be used to evaluate qualitative items; by that it is meant that items without a numerical value can be given a numerical value relative to other items based on evaluation criteria. As the tags used in this project can't be simply ordered by some numerical value that represents their usefulness, we can instead use chi squared to measure the distribution of a given tag in its cluster and the entire photo set relative to other tags in the cluster and entire photoset.

Prior to mining for rules (or afterwards for refinement) this method can be used to identify a numerical value for each tag in a cluster and how meaningful it is for that cluster (relative to the other tags). From here tags can be ordered numerically descending and any tags deemed below a certain threshold can be removed. It requires construction of a vocabulary list (all tags but none repeated) and knowledge of the number of photos, clusters and tags within each. [6]

3.5 Existing Association Rule Mining Tools

There are many available existing tools online for performing ARM analysis. Below I have documented the most useful of those available, based on its simplicity and what it offers (gathered from their respective documentation.).

Argui

Argui is small rule mining program that utilises the Apriori algorithm. It was built by Christian Borgelt and is freely available from his website [7]. The application takes an input transaction list and outputs the rules discovered on screen and to a file. As this program was fairly easy to become familiar with, I created some test transactions where elements 1 & 5 were obviously linked and have documented the output.

The test transaction file contained:

1	2	5
1	5	5
1	3	6
2	4	6
1	5	6
1	5	7
2	2	7

Where each row is a separate transaction and the columns represent the 1st, 2nd and 3rd item in their respective transactions. Argui discovered 16 rules in total (without constraints in place) but as expected the rule with the highest support was between item 1 and 5. Below is the output file from Argui.

File Sort Help

consequent	antecedent	ante. supp.	support	confidence	lift value	add. eval.
1	5	57.1/4	57.1/4	100.0	1.4	0.0
5	1	71.4/5	57.1/4	80.0	1.4	0.0
6	3	14.3/1	14.3/1	100.0	2.333	0.0
1	3	14.3/1	14.3/1	100.0	1.4	0.0
2	4	14.3/1	14.3/1	100.0	2.333	0.0
6	4	14.3/1	14.3/1	100.0	2.333	0.0
1	3 6	14.3/1	14.3/1	100.0	1.4	0.0
6	3 1	14.3/1	14.3/1	100.0	2.333	0.0
6	4 2	14.3/1	14.3/1	100.0	2.333	0.0
2	4 6	14.3/1	14.3/1	100.0	2.333	0.0
4	2 6	14.3/1	14.3/1	100.0	7.0	0.0
1	7 5	14.3/1	14.3/1	100.0	1.4	0.0
5	7 1	14.3/1	14.3/1	100.0	1.75	0.0
1	2 5	14.3/1	14.3/1	100.0	1.4	0.0
5	2 1	14.3/1	14.3/1	100.0	1.75	0.0
1	6 5	14.3/1	14.3/1	100.0	1.4	0.0

There were other tools that were investigated for the purpose of this report such as ARMiner and CBA but these both were unusable; ARMiner was built for handling client/server transactions and

analysing them and CBA was outdated and could not run on the windows platforms available (even in compatibility mode).

Links to both programs can be found under references [8] and [9] and a larger list of available data mining tools can be found at [10].

3.6 Database Suitability

Using a database to store the initial dataset has advantages; databases are often populated with millions of items because once databases are indexed, performance for searching and queries increases dramatically.

Once the database is populated, any desirable subsets of data can be acquired by writing SQL (Structured Query Language) queries (Which can be saved and reused) which is marginally different than having to construct variations of a java program that strips the dataset of certain data. It is noteworthy however that for the purpose of this project, I am more comfortable with Java than SQL.

The main reason for populating and having a database in this context would be to assist in the performance and simplicity of mining of rules, however as research for this report has identified a suitable rule mining program that expects a different type of input; namely a comma delimited text file. For this reason building and populating a database seems an unnecessary use of time.

4. Conclusion

Considering the research documented in this report, I have a clear understanding of the direction this project is heading in and what steps are necessary to return meaningful results that can be analysed to provide insight as to how they could theoretically improve location based services.

All tasks for this report outlined within the initial plan have been documented including some additional content that turned out to be more pertinent than some of the initially planned areas of research, specifically the clustering algorithms and feature selection.

The project will continue to follow the initial plan although the stages of the project now have specific methods against them. At the time of writing, these are the specific stages I now expect to undertake to eventually discover rules from the dataset:

- Write small programs in java to iterate over the dataset and leave only necessary data.
- Using a variation of the above to remove all photo's whose coordinates do not fall between the limits of Great Britain.
- Implement a K-medoids algorithm to cluster the photos (using the coordinates) to separate the photos in close proximity into groups.
- Impose the chi-squared feature selection method to rank tags in clusters by their relevance and decide a threshold to remove unnecessary tags.
- Alter the datafile as necessary so it meets the requirements of Argui (which implements the Apriori algorithm) so rules can be discovered and analysed.
- Revisit stages as necessary to modify the dataset so more relevant rules can be discovered.

- Given time allowances, experiment with cluster sizes and constraints within the apriori algorithm to contrast rules discovered.
- Analyse interesting rules and theorise on how they could be used to improve location based services.

4.1 Research Hypothesis

Given the above steps, I would expect many rules to be found of varying support and confidence. The cluster sizes will directly impact the nature of the rules, for example if the number of clusters was the same as the amount of cities, I would expect clusters to centralise around pockets of photos like taken in cities (excluding the occasional tourist hotspot in the countryside) and lots of rules stemming from location names because only those clusters will have tags of that city name.

If the clusters are scaled up to the point where there are seemingly lots of clusters within cities I would expect to see much more specific rules, for example village names and specific events that are unique to the cluster areas.

Almost any of these rules could prove beneficial for a location based service providing they are of a geographical nature (or at least either the antecedent or consequent is), especially if the service is aimed at suggestions based on a location input.

Further rules such as associations between tags with generic names could be used to improve or create an automated tag service; however this would mean generalising the dataset and purging it of geographical locations.

Glossary

Metadata - A set of data that describes and gives information about other data.

Web Crawling – A computer program that automatically browses the World Wide Web

Web Scraping – A computer program that automatically downloads information from Webpages

Computational Difficulty – How difficult a task is for a computer in terms of how long it is expected to take.

Clusters – A subgroup of a population

Geocoding - A geographical code to identify a particular point or area

Antecedent – A thing or event that existed before or logically precedes another

Consequent - The second part of a conditional proposition, whose truth is stated to be conditional upon that of the antecedent

Comma Delimited – Values separated by a comma

Table of Abbreviations

JSON - JavaScript Object Notation

XML - Extensible Markup Language

GB – Great Britain

ARM – Association Rule Mining

API – Application Programming Interface

SQL – Structured Query Language

References

[1] www.alexa.com. Global website traffic rankings. Available: http://www.alexa.com/siteinfo/flickr.com [accessed: 01/11/2012]

[2] E.M. MIRKES. 2011. K-means and K-medoids applet. University of Leicester. Available: http://www.math.le.ac.uk/people/ag153/homepage/KmeansKmedoids/Kmeans_Kmedoids.html [accessed: 20/11/2012]

[3] R.AGRAWAL, T.IMIELINSKI, A.SWAMI. 1993. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, pp.1-3.

[4] http://www.dataminingarticles.com. 200-. Algorithms for Mining Frequent Itemsets. Available: http://www.dataminingarticles.com/association-rules-algorithms.html [accessed: 11/12/2012]

[5] R. AGRAWAL, R.SRIKANT. 1994. Fast Algorithms for Mining Association Rules. Proc. of 20th Intl. Conf on VLDB . Available: http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf [accessed: 11/12/2012]. pp. 3-5.

[6] O.V.LAERE, S.SCHOCKAERT, B.DHOEDT. 2012. Georeferencing Flickr resources based on textual meta-data. Note: Paper still under review.

[7] C.BORGELT. N.D. Association Rules GUI and Viewer. Available: http://www.borgelt.net//argui.html [accessed: 13/12/2012]

[8]E, KVALEBERG. Modification of 'Map Sources' known as GEOHACK. Available: https://wiki.toolserver.org/view/GeoHack [accessed: 13/12/2012]