Data Mining, Project 5 report
Mining Flickr

Armin Eftekhari, Alejandro Weinstein

December 4, 2009

1 Introduction

Flickr is one of the most popular image hosting website, with more than 4 billion pictures [3]. For each picture, the community can add comments, tags, or call the picture his or her favorite. Also, the details of the camera used to take the picture are available. So, an interesting question arise: is there any correlation between the popularity of a photo and the camera used to take it? The purpose of this project is to try to answer this, and to find other interesting patterns.

2 Generating the data set

There is no dataset directly available with the data needed to analyze Flickr. However, there is a Flickr API that allows to build such dataset. We wrote a Python script, using the Python "Flickr API kit" [2] to do that.

We decided to build a dataset with the following attributes:

Id: the unique id used by Flickr to identify a photo.

Views: the number of views of the photo.

Location: the location of the photo.

Comments: the number of comments of the photo.

Tags: the number of tags of the photo.

Favorites: the number of Flickr users that call the photo a favorite.

Make: the maker of the camera used to take the photo.

Model: the model of the camera used to take the photo.

Since it is not feasible to get information about all the pictures in Flickr, a key point at the moment of building the dataset is to determine how to sample the website. We decided to get the data of 100 picture for each of the most popular tags. Unfortunately, we didn't find an API function to get the most popular tags, so we copy the list of the most popular tags from the "All time most popular tags" page [1]. We also looked at photos at least one year old, to guarantee that the pictures has been exposed to the community for a time long enough.

We built the dataset in four steps:

1. Get the 100 ids for each of the popular tags: For each of the popular tags, we get the ids of 100 photos. In this step we ask for pictures at least one year old. We got 14400 records.
2. Eliminate the duplicate ids. Since a given photo can be associated to more than one tag, it is possible to have duplicated ids. We eliminated the duplicate ids, reducing the 14400 records to 7342.
3. Get the data for each photo id.
4. Write the data as a CSV file.

It is interesting to note that the time Flickr take to respond to each API call is roughly one second, and consequently, building the dataset took several hours.

The examination of the dataset revealed that the *Make* attribute was inconsistent, since we got slightly different names for a given maker. For instance, "Olympus" cameras were labeled as "Olympus Imaging Corp.", "Olympus optical Co. Ltd" or "Olympus corporation". We wrote another Python script to solve this inconsistency, reducing the number of unique manufacturers from 62 to 36.

3 Summary statistics

After finishing the preprocessing of the dataset, the next step on the data mining process is to generate some summary statistics, to get a general idea of the structure of the data. Table 1 shows the statistics for the numerical attributes. For the categorical attributes "make" and "model", the mode is "Canon" and "Nikon D200", respectively.

Figure 1 shows the histograms for the numerical attributes. As can be seen, all the numerical attributes share a *long tail* behavior, where most of the photos are unpopular (in terms of a low number of views, comments, etc), and a relatively small amount of the photos are very popular. Figure 2 shows an histogram for the "Make" attribute.

Table 1: Summary statistics for the numerical attributes.

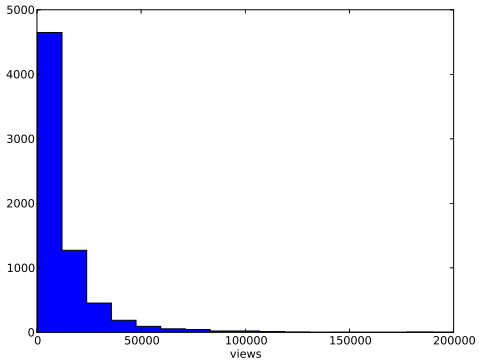
views	
Min	164
Max	1184897
Mean	14778.5
Std dev	35718.6

comments	
Min	6
Max	2891
Mean	150.2
Std dev	161.8

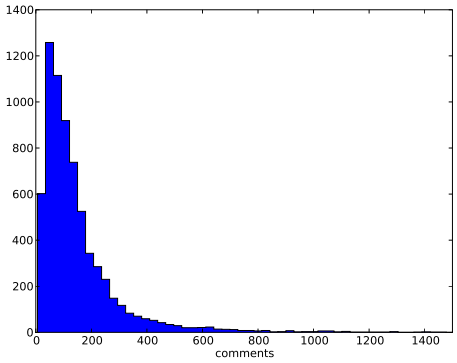
tags	
Min	2
Max	279
Mean	30.2
Std dev	17.5

favorites	
Min	0
Max	8539
Mean	243.0
Std dev	348.2

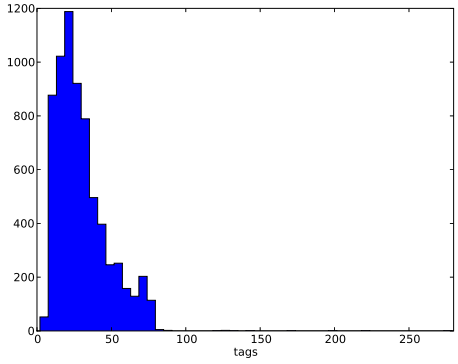
First of all, it can be noted that a significant amount of the photos (44%) don't have information about the maker of the camera used to take the picture. As we will discuss later in section 5, this will play a detrimental role in our data mining analysis. Secondly, we can see that the two mayor players are *Canon* and *Nikon*, followed by a handful of other brands with a significant share of the photos. Most of the remaining manufactures are associated with only one photo.



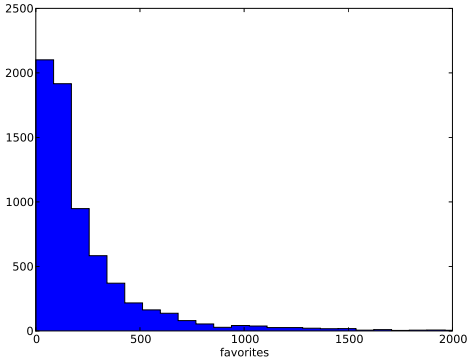
(a) "Views" histogram



(b) "Comments" histogram



(c) "Tags" histogram



(d) "Favorites" histogram

Figure 1: Histogram for "views", "comments", "tags" and "favorites".

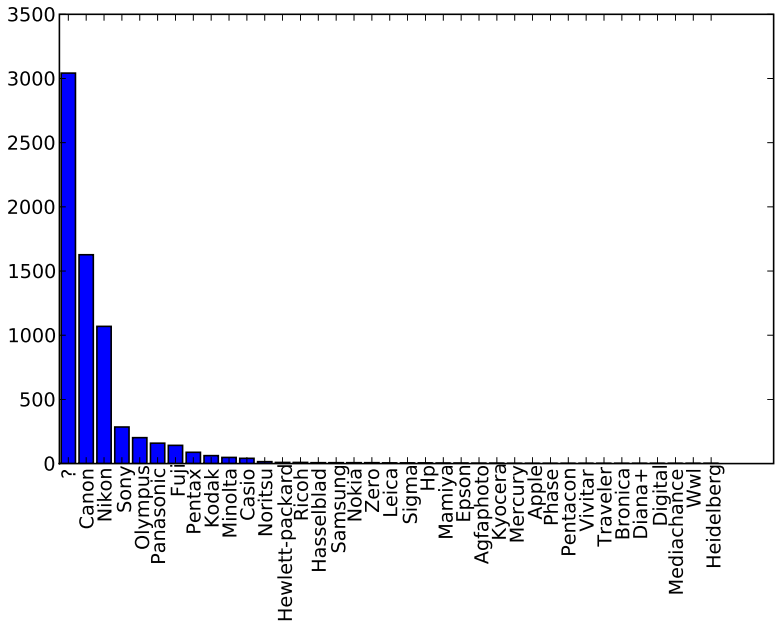


Figure 2: Histogram for the "maker" attribute.

Figure 3 shows a scatter plot, where the x axis correspond to the number of comments and the y axis correspond to the number of views. The color of each point is mapped to the logarithm of the number of favorites[1]. This plot illustrate, once again, the *long tail* behavior of the data, where we can see a big cluster of photos close to the origin, and a set of very popular outliers.

Notice that all the plots for the summary statistics were produced using Python and the Matplotlib library, since we think that the ones produced by Weka are very low quality.

4 Clustering

4.1 Location

Here we were interested in the location of Flickr users. In particular, we wanted to study any possible relations between the clusters and the location of users. Typically, social networks are more likely to form among people who live nearby, have the same language, and share the same culture. So, we expected to observe a distinct relation between the clusters and the location of Flickr users. First, user ID was omitted from our attributes.

[1] A logarithmic map is used due to the huge range of the variable. Otherwise, most of the points end with the same color

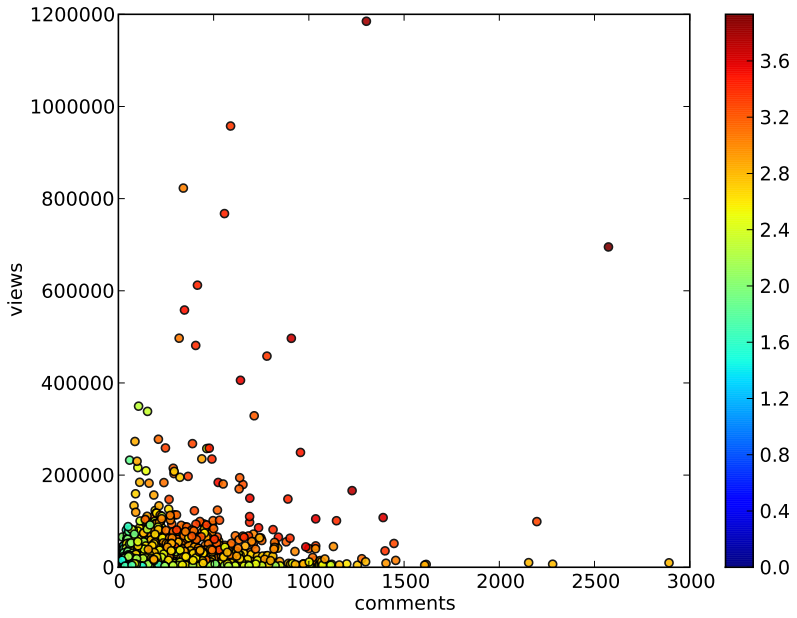


Figure 3: Scatter plot. The color represent the logarithmic of the "favorite" attribute.

Then typical k-means clustering was performed on data using 100 iterations. You can observe the cluster centers below.

We first see that all the locations are basically tourist destinations and that, as stated above, they belong to different clusters and different social networks. Interestingly these results also tell us that San Francisco in U.S. and Tokyo in Japan are very popular tourist destinations among others.

4.2 Comments

Even before this analysis, we expected a close relation between number of comments on a image and the number of people who marked that image as favorite. After clustering, it turned out that there is a semi-linear relation between them, as observed in Figure 4. However, this relation doesn't remain exactly linear and there are many images with many comments and yet few favorite marks, which is interesting in its own right.

Also it is instructive to study number of comments in each cluster, which is summarized in Figure 5. Going to information on the cluster centroid we find out that cluster 8 which has the highest number of comments in average corresponds to Tokyo Japan. Also, before running this analysis, we expected that photos which are more viewed are likely to get more comments. Interestingly, it turned out that there is not a clear and special relation between these two, as one may see in Figure 6

Cluster No.	Number of records	Location
1	532	San Francisco USA
2	440	San Francisco USA
3	726	Livermore & Antioch Calif. USA
4	817	New York U.S.A.
5	1352	San Francisco USA
6	478	Bonn Germany
7	278	the Island of Mactan Cebu Philippines
8	265	Berlin Germany
9	1507	Tokyo Japan
10	571	Barcelona Spain

Table 2: Relation between cluster centroid and locations

Figure 4: Relation between the comments on an image and favorite status

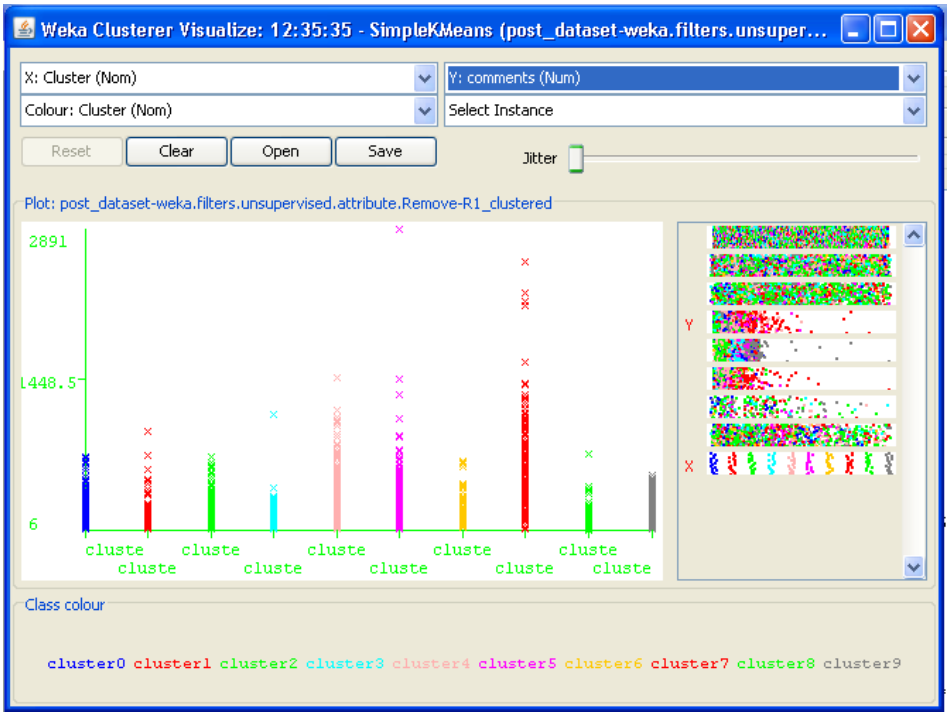


Figure 5: Number of comments represented by each cluster

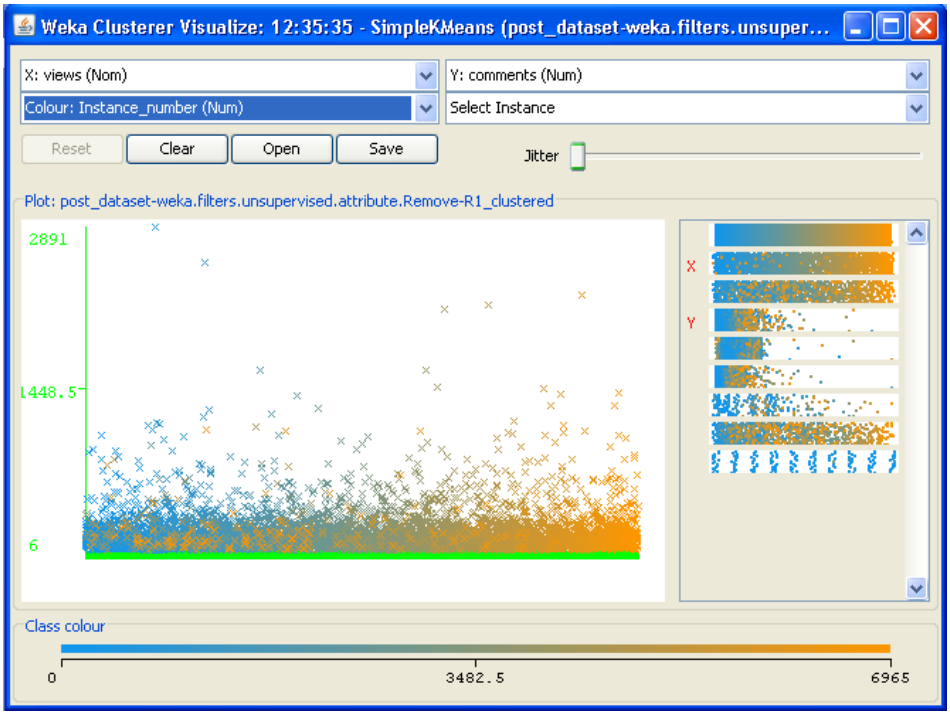


Figure 6: Relation between views and number of comments

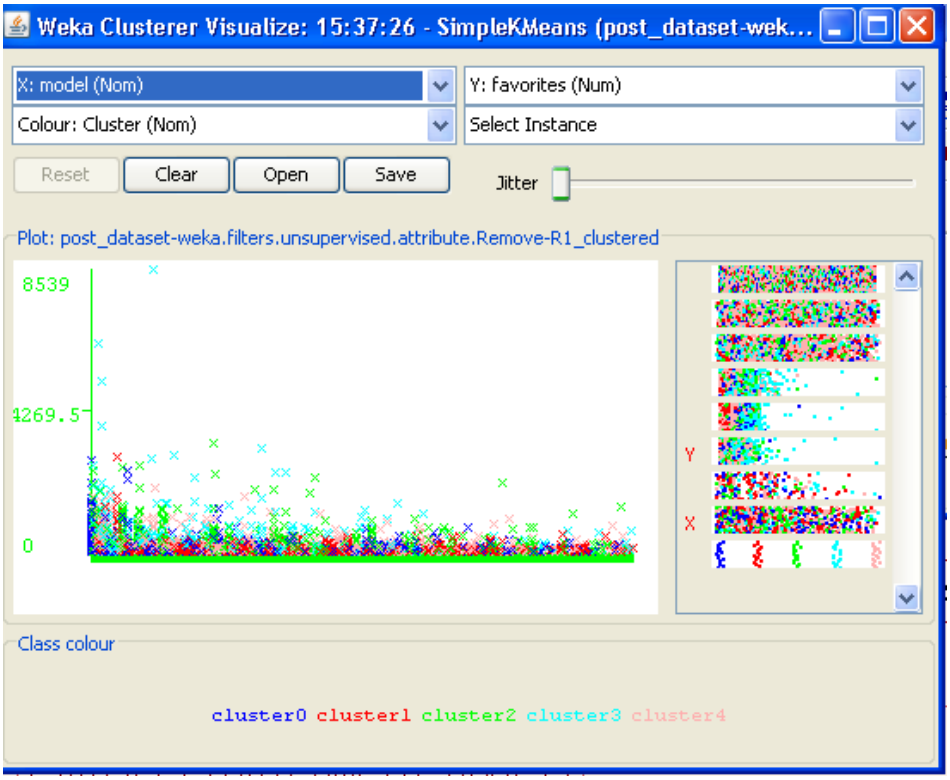


Figure 7: Relation between camera model and favorite status

4.3 Camera Model

Before we started the analysis, we hoped to find a relation between the camera model and the favorite status of a photo. In practice, however, we faced an unbalanced data which contained large number of photos from fairly few camera models, such as Nikon D200. As a result, patterns were not meaningful. In particular, let us show how model and favorite status are related using the k-means clustering (Figure 7). This figure shows that there is clusters with higher number of favorite images within, simply correspond to Nikon D200 and then Canon EOS 5D, which is not meaningful, because these are simply the most frequent camera models in dataset.

4.4 Other Clustering Algorithms

In our analysis, we have used simple k-means clustering algorithm. The main reason has been the efficiency and robustness of this clustering technique. In particular, we tried several other clustering techniques but the main problem was excessive computational requirements. In contrast, k-means gives good results and is very fast and efficient.

5 Classifiers

In this section we try to study the effect of different attributes and the level of information that they carry. One of the ideas suggested during our presentation was to design a classifier that predicts the camera model. In fact, due to large imbalance in prior probabilities, we can simply use the zero-R classifier and yet get pretty good classification result. This is due to the fact that most of the photos in the dataset are taken by few camera models.

So, we turn into another interesting problem, which studies the role of camera model on the status of photo. In particular, due to mentioned imbalance, we have to employ a classification technique that is least sensitive to prior probabilities. One candidate is support vector machines, because in SVM the classification boundary only depends on support vectors and not on the prior probabilities. This is best visualized in Figure 8. Note that only a very few number samples actually contribute to the boundary.

To get better results, we exclude the following attributes from the dataset: "ID, location, n_tags, maker". Kernel of SVM is selected to be RBF (radius basis function). Usually in real world applications, data tends to scatter around circular clusters and hence this type of kernel usually outperforms other kernels. An example of SVM with this kernel is depicted in Figure 9. Another major issue is that all classifiers need the target to be nominal, hence we need to change the favorite attribute from numerical to nominal. This is simply done by using discrete filter in Weka with 10 bins. Histogram after discretization is shown in Figure 10. As expected, classification results are really glamorous with SVM: 95.7% of samples are correctly classified. We note that 2/3 of data was selected as training set and the rest was used for testing. We also tested our technique with J48 tree, and with the same settings. This time we got 85.9% correct classification.

Now we get to our main purpose of this analysis, which is to study the role of camera model in the classification result. We exclude camera model from the attributes and use both SVM and J48 classifiers.

6 Discussion

The Flickr API provides a wonderful opportunity to explore very interesting datasets. However, an important point to consider in order to get meaningful results is the sampling strategy. While the sampling strategy we used is reasonable, there are other ways to sample the data that should be explored.

It was not surprising that our data showed a long tail behavior, since this is typical for data originated on social networks. This fact leads us to think that it would be interesting to investigate some techniques specially tailored to this kind of distribution.

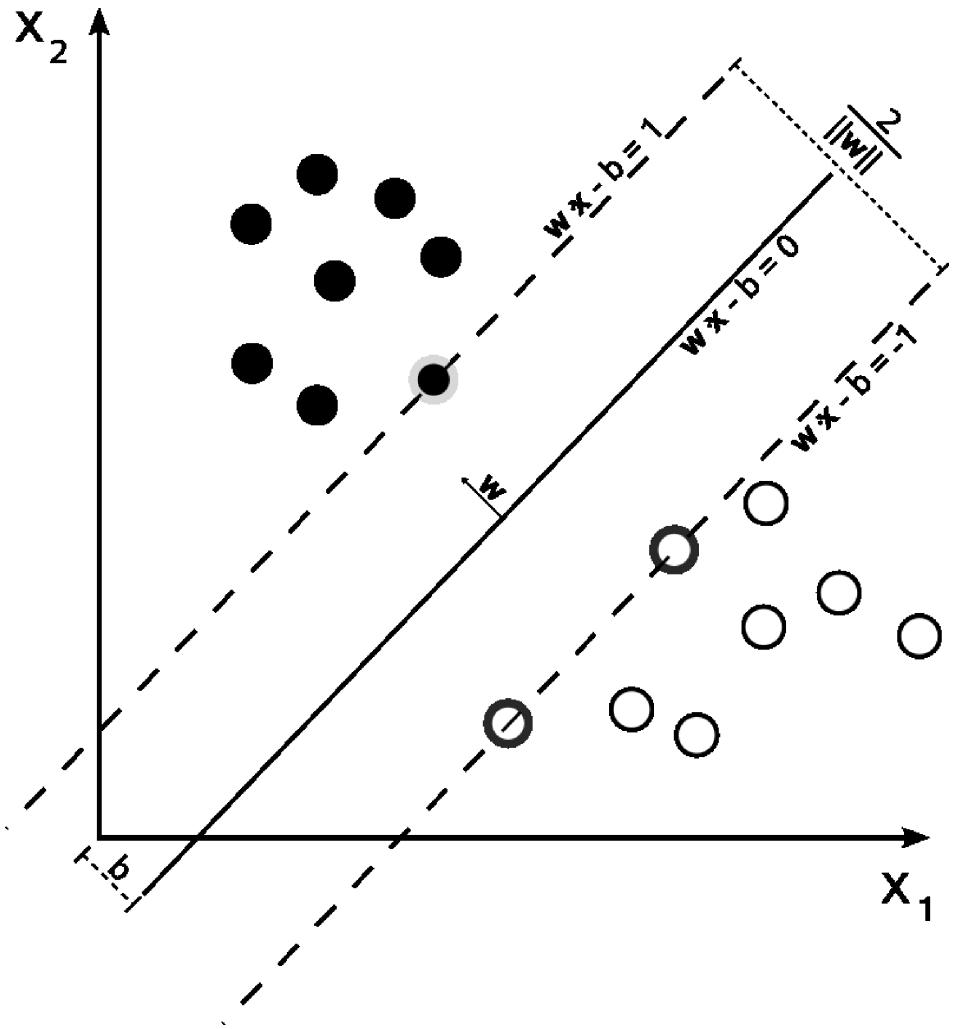


Figure 8: Intuitive explanation for independence of SVM from prior probabilities

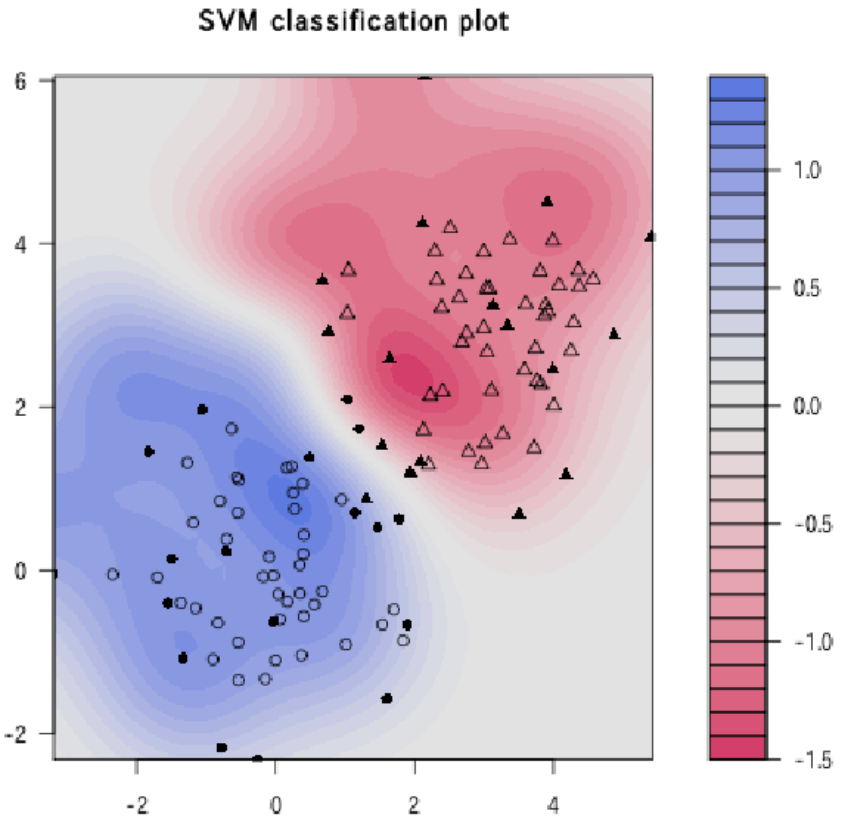


Figure 9: An example of SVM with RBF kernel, which shows the characteristics of this kernel.

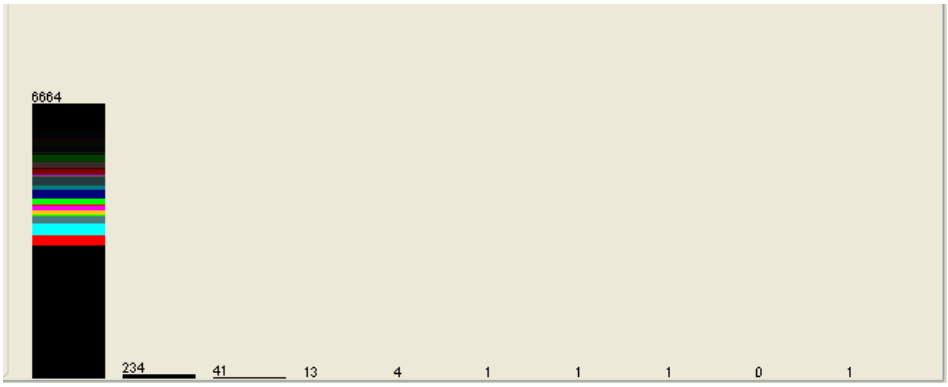


Figure 10: Histogram of favorite attribute after discretization

This work triggers some new interesting question that could be answering by mining Flickr, like for instance:

- Using geolocation information of the photos, can we identify a correlation between location and type of camera used? Producing, for examples, a statement like. "people in the USA use mostly Canon cameras, while people in Europe use mostly Nikon".
- So far we ignored the users, but some patterns may arise by looking at this dimension. For instance we could try to answer questions like, "are some users with a lot of popular photos?", "can we identify "different classes" of users?", etc.
- Can we found some semantic of the photos by looking at the data?

References

[1] Flickr. Popular tags on flickr, 2009. [Online; accessed 29-November-2009 http://www.flickr.com/photos/tags/].

[2] Stvel.eu. Projects—python flickr api kit, 2009. [Online; accessed 29-November-2009 http://stuvel.eu/projects/flickrapi].

[3] Wikipedia. Flickr — wikipedia, the free encyclopedia, 2009. [Online; accessed 29-November-2009 http://en.wikipedia.org/w/index.php?title=Flickr&oldid=326710624].