Enhancing click-rates of online Advertisements by Identifying Patterns

(with emphasis on the use of Local Clustering for Rare Class Analysis and SVM's for classification)

Varun Mithal, Snigdha Chaturvedi,
Shashank Srivastava

1. Introduction:

The Komli Dataset (KD) offers many challenges to the machine learning community including dealing with large amounts of data, sequential data, issues of missing data and a rich domain complete with noise, hidden variables, and significant effects of context.

Preliminary Observations:

The dataset consists of about 5.3 million readings with 18 attributes, several of which appear to be irrelevant to the problem at hand. Since the data was not intended as a Machine-learning problem, it needs careful preprocessing. Issues like noisy data, an extremely high degree of skew (less than 0.8% of the datapoints are clicked), and an inherent heuristic nature of the problem need to be addressed.

Also, inspite of the huge volume of the data, the number of distinct Ad_id's is relatively small (223). Moreover, some Ad_ids are documented much more than others (in fact more than one-third of the data comes from a single advertisement (Ad_id: 29214).

The dataset consists of the following attributes in a tabular format:

Id, Pub_id, Site_id, Ad_id, Adserver_id, "Page", Click_count, "User_guid", "User_ip_addr", Ad_width, Ad_height, "Bg_color", "Border_color", "Link_color","Url_color","Text_color",Adserver_optimizer_id,"Timestamp"
The colors are given in hexadecimal html encoding, and so RGB values are recoverable.

Instance	Id	Pub_id	...	Page	Click_count	.....	Bg color	Other color encodings	....	Time Stamp
1	0	...	...	*4F7CD7	0	...	...	...	...	12:36:42
2	0	...	...	*E56BE....	0	...	...	...	...	12:36:42
.					.		.		.	.
.					.		.		.	.
.					.		.		.	.
.					.		.		.	.
.					.		.		.	.
5300000	0	...	...	...	1	...	...	...	...	12:36:42

Format of training* dataset (5300000 instances over 18 attributes)

Evaluation of Performance:

A major issue in skewed datasets includes measuring the performance of the classifiers. Success cannot be defined in terms of predictive accuracy because the minority class in the skewed data usually has a significantly higher cost.

Since the dataset contained an enormous imbalance (99.2% unclicked datapoints by instance), we intend to optimize a *Performance Evaluation measure* in favour of simple percentage accuracy for comparison of performances.

*Performance= {Clicked correctly predicted/total clicked +Unclicked correct predicted /total unclicked}*100%*

The present problem poses the challenge of predicting a very rare class, as most of the time ads go unclicked. Hence, it is challenging to build a learning model which has the predictive power to capture future clicks, with low false positive rates.

2. Feature Extraction and Selection:

Combinations of background, foreground and text colors intuitively seem to have a strong correlation with access rates of advertisements. Optimizing color palettes are understood to make ads more visible to users. Two colors provide good color visibility if the brightness difference and the color difference between the two colors are greater than a set range.

1)Color brightness is typically determined by the following formula:

*CBR= (Red value * 0.299) + (Green value * 0.587) + (Blue value * 0.114)*

This algorithm is taken from a formula for converting RGB values to YIQ values. This brightness value gives a perceived brightness for a color.

2)Color difference is determined by the following formula:

CDF= ||Red value 1-Red value 2|| + ||Green value 1- Green value 2|| + || Blue value 1- Blue value 2||

This suggests that creating new features for CBR and CDI for each of text, link, URL, background and border would make the problem more amenable to classification.

3)Color Fraction: Instead of RGB values, we use the relative fractions of the three colors.

4)Ad-area and Aspect-Ratio of the ad are used for training the classifier since this appears to be a better encoding of the same information.

Furthermore, some features seem irrelevant such as the first and last attributes that remain constant throughout. A little insight into the domain of the problem suggests that certain other attributes be ignored which might have coincidental strong correlations, within this very sparse sample set. (eg. User IP address is significant only in the third part of the problem)

We felt that a knowledge blind algorithm like a Support Vector Machine would likely and erringly give a higher weight to these characteristics. Hence they were not included in training.

Training the classifier:

There were two obvious approaches to go about the problem :

i) Aggregative calculation of click rates:

Datapoints within an Ad_id could be aggregated to form a single datapoint representative of the set. New features would represent means, extrema, general mode and variance in the size and colors for the ad. Outlier removal would be essential before aggregation. There would essentially be two types of outliers:

- Ad_ids with an extremely low click-rate
- or too few points(<100)

Such entries would need to be modified or removed as they would misrepresent the original data. Aggregation would have lead to a compaction, with less than 250 meaningful datapoints.

Principal Component analysis could be used to reduce the number of relevant features. This approach appears ideal to identify attributes and their values that lead to higher click rates.

ii) Instancewise binary classification:

The dataset consists of less than 40000 clicked datapoints. Around one-fifth of the data has no documentation of colors (NCOLOR), which we intend to be a major source of our learning, and is discarded. Trying to learn a pointwise binary classifier on the entire set would be biased because of the high skew. With positive instances being so precious, our classifier must necessarily be trained on all clicked datapoints, and randomly choose a similar number of unclicked datapoints.

(This number can be chosen to engender a bias to maximize our performance evaluation function. However, previous works by various authors suggests that adjusting class-sizes alone can improve the predictive accuracy of the rare classes slightly, but at the cost of seriously reducing the recall rates of the large classes)

3. Upsampling the rare class:

The most direct method for dealing with highly skewed class distributions with unequal misclassification costs is to use cost-sensitive learning. An alternate strategy for dealing with skewed data with non-uniform misclassification costs is to use sampling to alter the class distribution of the training data so that the resulting training set is more balanced. Here we use both up-sampling and down-sampling, two common sampling methods. Up-sampling replicates minority class examples and down-sampling discards majority class examples.

Altering the class distribution of the training data aids learning with highly-skewed data sets is that it effectively imposes non-uniform misclassification costs. For example, altering the class distribution of the training set so that the ratio of positive to negative examples goes from 1:1 to 2:1, effectively assigns a misclassification cost ratio of 2:1. ***(2)***

Random down-sampling:

In our case the data set is enormous and therefore requires the size of the training set to be reduced. In this case, down-sampling seems to be a valid strategy. However, this might effectively lead to a loss of potentially useful data and we don't downsample than more than what is absolutely essential. Here, we randomly select 200,000 unclicked datapoints of a total of about 4,300,000 as representative of the larger class.

Issues with up-sampling:

Our up-sampling method generates duplicates of existing examples, overfitting is likely to occur in that classification rules may be formed to cover a single, replicated example. The number of positive training instances being very small, we use a mix of upsampling and cost-sensitive learning, in which we upsample less represented clusters of the rare class.

We performed EM clustering (unspecified number of clusters) on the 28500 positive instances, leading to three clusters, one of which accounted for 60% of all instances. The less represented clusters were replicated more so that all three clusters were equally represented.

4. Local Decomposition:

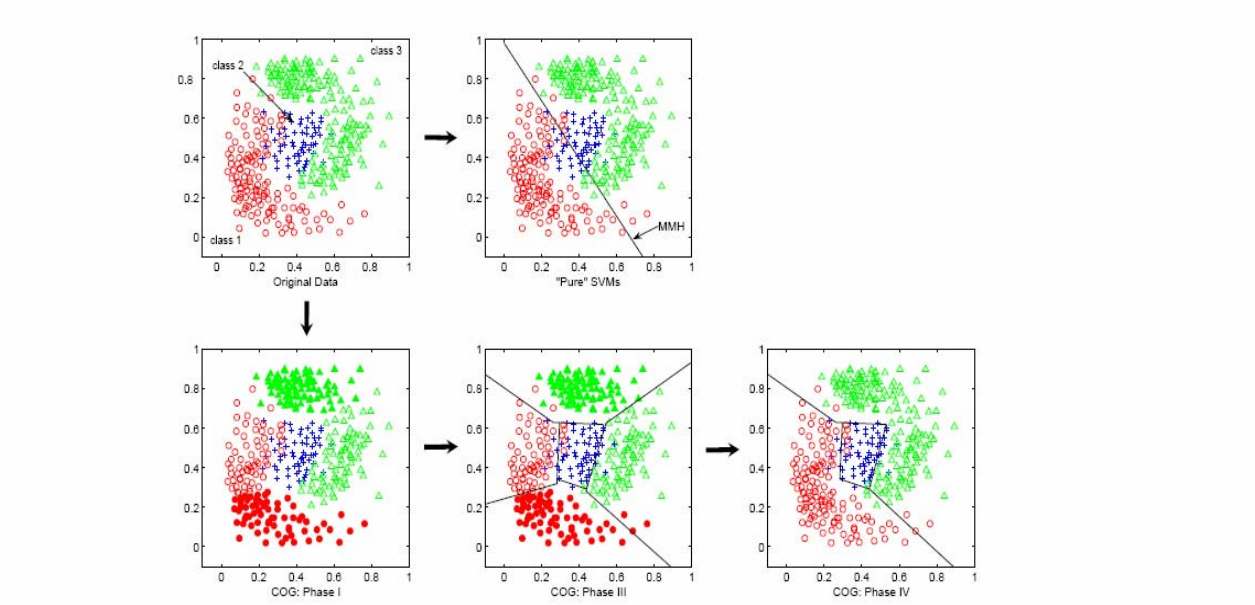
Balancing class sizes generally leads to some improvement in identifying the rare class, but is usually not adequate and can reduce precision (increase FP's). Hence we need a classifier that follows the following criteria:

i) has the ability to divide imbalanced classes into relatively balanced classes for classification
ii) has the ability to decompose complex concepts within a class into simpler concepts.

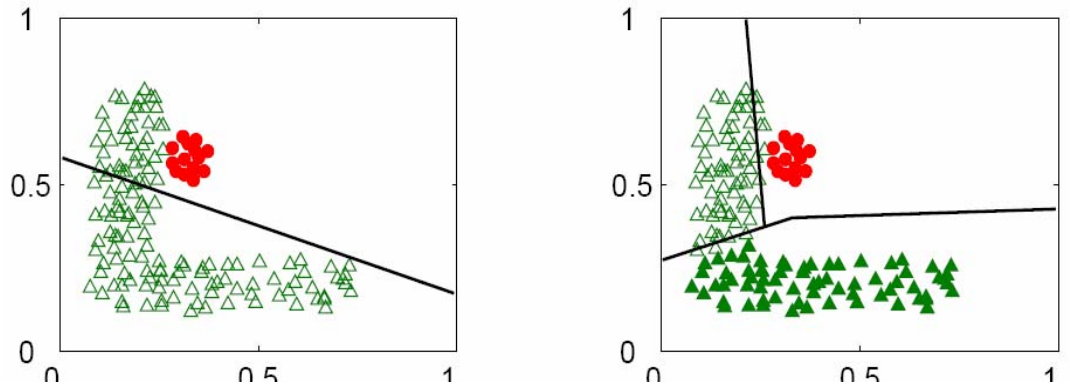
Classification using lOcal clusterinG (COG)

In this attempt, we change the binary classification problem to a multiclass one using EM clustering, and treat each cluster as a separate smaller class. This technique reduces the relative skew of the classes by producing clusters of relatively balanced sizes, and along with an intuitive oversampling technique of the rare class is seen to produce consistently better results than by sampling alone.

We then use Support Vector Machines (SVMs), for a *one–against–one* multiway classification since it is computationally less expensive than *one-against-rest*.



Intuitively, COG divides the imbalanced classes into relatively balanced and smaller subclasses. The key idea is to perform clustering within each class and produce linearly separable subclasses (which might not be possible in a two class situation) with relatively balanced sizes. Our experiments suggest that COG performs generally better in identifying the rare class, while doing much better on the predictive accuracy of the large class than sampling alone.



Since COG seems to be of help in improving accuracy with linear classifiers, we use an SVM instead of a C4.5 decision tree or Neural Nets.

The choice of clustering doesn't seem very relevant since the clusters would in themselves be separable, and misclassifications within clusters would anyhow not affect our final performance. However, we used the EM clustering instead of the more probabilistic K-means technique.

5. Training the SVM:

With a blend of sampling techniques described above, we created a dataset of 84000 positive and 200000 negative instances. Expectation Maximization clustering was used to group the negative class into 5 clusters. The number of positive instances was chosen to match the size of the largest cluster. Hence, the problem became a six way classification. Since the data is still not balanced, we experimented with the misclassification cost of the click class. (Class100)

Sample Confusion Matrix

Cluster0	594	0	0	0	0	0
Cluster1	0	1765	0	0	0	96
Cluster2	0	0	428	0	0	75
Cluster3	0	0	0	2792	0	0
Cluster4	0	0	0	0	488	282
Click	34	459	8	825	18	1436

Recall	0.52
Precision	0.72
%Accuracy	79.8

We extracted three random datasets, of 28000(E1), 37000(E2) and 45000(E3) instances, for training and tuning misclassification costs and parameters, since SVM training on the large set took a lot more time. *5-fold cross validation* and *66-33 split* were used to measure performance, and make a reasonable trade between rare class recall and precision.

Datasets:	E1	E2	E3
Number	28500	37000	45300
% Click	29.47	45.95	54.63
Optimum weight	~2-3	-	~1.15
Best Kernel	RBF/Linear	-	RBF/Linear
Best Performance Without local clustering: binary classification	68	68	67
Best Performance with local clustering (COG)	71	69	70
Optimum weight	~2-3	-	1.15

For an 'Ad in general' with 66/33 split

However, the performance significantly improved on training the entire dataset of 258,000 points. (which had the same composition as set E1), indicating that the smaller datasets in themselves are not adequately representative.

*(Results were significantly lower on a highly skewed random test set of 56,000 points with only 300 clicks (that after **all** positive instances were used on classification), suggesting effects of duplication. While precision rates for the rare class were drastically lower (3%), they were better than the randomly expected value of 0.6%, indicating that the problem is not entirely unamenable to machine learning. Recall rates were lower at about 20%)*

References:

- ‘Rare Class Analysis’ by Kate McCarthy, Bibi Zabar and Gary Weiss . Fordham University
- Elkan, C. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.
- Cristianni, Taylor, Shaw: *An Introduction to Support Vector Machines*
- CS674 Class notes, Prof. Karnick ☺