Kernel-based Principal Components Analysis on Large Telecommunication Data

Takeshi Sato[1] **Bingquan Huang**[1] **Guillem Lefait**[1] **M-T. Kechadi**[1]
B. Buckley[2]

[1] School of Computer Science and Informatics
University College Dublin,
Belfield, Dublin 4, Ireland
Email: [bingquan.huang, guillem.lefait, tahar.kechadi]@ucd.ie
takeshi.sato@ucdconnect.ie

[2] Eircom Limited
1 Heuston South Quarter,
Dublin 8, Ireland

Abstract

Linear Principal Components Analysis (LPCA) is known for its simplicity to reduce the features dimensionality. An extension of LPCA, Kernel Principal Components Analysis (KPCA), outperforms LPCA when applied on non-linear data in high dimensional feature space. However, on large datasets with high input space, KPCA deals with a memory issue and imbalance classification problems with difficulty. This paper presents an approach to reduce the complexity of the training process of KPCA by condensing the training set with sampling and clustering techniques as pre-processing step. The experiments were carried out on a large real-world Telecommunication dataset and were assessed on a churn prediction task. The experiments show that the proposed approach, when combined with clustering techniques, can efficiently reduce feature dimension and outperforms standard PCA for customer churn prediction.

Keywords: Kernel PCA, Churn Prediction, Clustering, Imbalanced Classification

1 Introduction

Due to the advances in data collection and storage capability, companies can record considerable amount of information on customers. However, the number of available information is so massive, that both automatic tools and experts face difficulties to analyse these data. Therefore, Dimension Reduction techniques have been developed to select the most adequate information out from high dimensional datasets. In telecommunication service sector, predicting customer churn has become a major focus for companies as churn can result in a huge financial loss. However, analyzing customer data can be troublesome as it contains substantial size of information. One solution is the Feature Extraction (FE) approach.

The main goal of FE approach is to discard redundant attributes and create a new set of attributes that captures the important information more effectively. The FE approach can be subdivided into three categories: Filter Approach, Heuristic Technique (Genetic Algorithm etc.) and Feature Transformation Approach. The last approach is applied in this paper as it has the advantages of exploring feature combination more effectively.

The Kernel Principal Component Analysis (KPCA) (Schölkopf, Smola & Müller 1998, Schölkopf, Mika, Burges, Knirsch, Müller, Rätsch & Smola 1998) is a renowned feature transformation approach that transforms the original features into new features with orthogonal transformations based on eigenvectors. KPCA is the extension of Linear Principal Component Analysis (LPCA) (I.T.Jolliffe 1986) which extends LPCA by mapping the input data into non-linear feature space and operates PCA with the support of Kernel trick. However, the computational complexity and the memory size required during the training process of KPCA depend on the size of the training dataset. Following issues are expected when applying KPCA on Telecommunication Data:

- Standard KPCA may not be a suitable solution to transform a very large dataset due to its high complexity and memory requirements.
- Similarly to LPCA, KPCA has very low discriminant ability to solve imbalanced classification problems as in LPCA (Chawla et al. 2004). Imbalanced classification is present when churn prediction is applied on Telecommunication data as 95% of the customers are non-churners and 5% are churners.

Several approaches have been proposed to solve these limitations by focusing on the simplification of the KPCA training procedure (Franc 2003, Kim 2005, Marukatat 2006). Kim et al. implement iterative KPCA by applying Generalised Hebbian Algorithm (GHA), which is a linear neural network model for unsupervised learning. The author kernelizes GHA to obtain a memory efficient approximation of KPCA and thus its training process can be achieved without storing a large kernel matrix.

Frank et al.(Franc 2003) introduces a simple greedy algorithm to iteratively add input vectors into a new training dataset until the prescribed limits are reached. These limits for input vector selection are the maximum number of vectors to store and the maximum approximation error rate ϵ. This approach reduces the workload of training process because it reduces the size of the training set, thus decrease the memory requirement.

In (Marukatat 2006), a clustering algorithm called Kernel K-Means is applied on a training dataset to simplify the training process. It applies K-Means clustering on input data after mapping them onto high dimensional feature space.

However, these approaches have not been tested on a real application with a large dataset such as telecommunication data.

In this paper, we focus on condensation method through clustering and sampling techniques on training data before mapping them onto high dimensional feature space. The proposed approach employs a clustering/sampling algorithm to reduce the size of a training dataset, use the reduced dataset to train KPCA and then use the trained KPCA to reduce the feature dimensions (i.e. feature transformation). This process is validated by using a real-world dataset collected from Telecommunication Company.

The rest of this paper is organised as follows: the next section outlines details of the proposed KPCA approach. Experimental results along with discussion are presented in Section 4 and we conclude and highlight some future directions in Section 5.

2 Kernel PCA

KPCA is a feature extraction algorithm which combines the operations of LPCA approach and the Kernel trick technique (Schölkopf, Smola & Müller 1998) to transform data. LPCA has been known for its simplicity and minimal effort for dimension reduction due to the assumption of linearity. However, it is only limited to re-expressing a data as a linear combination of its basis vectors. KPCA extends LPCA by mapping the features into the high dimensional feature space F, which enables KPCA to separate non-linear data more clearly than LPCA that uses a linear combination. The procedure of KPCA feature extraction is explained in the followings.

Consider a dataset $\{x_{i,i=1,2,...,N}\}$ with dimensionality d. In order to find the separability of nonlinear data, the KPCA maps the data into a high (possibly infinite) dimensional feature space by using the Kernel trick, and then the KPCA solves the following eigenvalue decomposition problem:

$$\lambda\alpha = \mathbf{K}\alpha, \ \ subject\ \ to\ \ ||\alpha||_2 = \frac{1}{\lambda} \tag{1}$$

$\mathbf{K}$ is the kernel matrix, defined as:

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \cdot\ \cdot\ \cdot & k_{1N} \\ k_{21} & k_{22} & \cdot\ \cdot\ \cdot & k_{2N} \\ \cdot & & \cdot & \cdot \\ \cdot & & \ \ \cdot & \cdot \\ \cdot & & \ \ \ \ \cdot & \cdot \\ k_{N1} & k_{N2} & \cdot\ \cdot\ \cdot & k_{NN} \end{bmatrix} \tag{2}$$

The element k_{ij} of the kernel matrix can be computed by

$$k_{ij} = \mathbf{k}(x_i, x_j) \tag{3}$$

where $\mathbf{k}(.)$ is a kernel function. One of the most widely used kernel functions is the Gaussian kernel: $\mathbf{k}(x,y) = exp(-\frac{||x-y||^2}{2\sigma}), \sigma > 0$. Once the Eigenvalue Decomposition problem (See Eq.1) is solved, the eigenvalues and eigenvectors can be used to project a test data x by:

$$\pi_k(x) = \sum_{i=1}^{N} \alpha_i^k \mathbf{k}(x_i, x) \tag{4}$$

However, the described KPCA is considered only when the kernel matrix K is centred. The kernel matrix K is not centred in general case. To solve this problem, K needs to be updated by the following equation:

$$\widetilde{\mathbf{k}}(x_i, x_j) = k_{ij} - \frac{1}{N}\sum_{a=1}^{N} k_{ia} - \frac{1}{N}\sum_{a=1}^{N} k_{aj} + \frac{1}{N^2}\sum_{a,b=1}^{N} k_{ab} \tag{5}$$

Similarly, the data projection for this case is performed by

$$\pi_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{N} \alpha_i^k \widetilde{\mathbf{k}}(x_i, x) \tag{6}$$

where λ_k is the k^{th} largest non-zero Eigenvalue, α_i^k denotes i^{th} value of λ_k's corresponding Eigenvector and x_{test} is a test data.

In summary, following steps are required in order to obtain the principal components in KPCA (Schölkopf, Smola & Müller 1998): Firstly, decide which Kernel function to apply and then compute Kernel matrix $\mathbf{K}$ (Eq.2); secondly solve Eigenvalue problem; thirdly project the test data onto the Eigenvectors (Eq.6).

3 Proposed Approach

3.1 KPCA applied to telecommunication data

When KPCA is applied to a dataset of N input data, a kernel matrix of size $N \times N$ is required to be computed. This matrix is required in both the training and the data projection parts. With increasing size of input data to train KPCA, the size of kernel matrix increases. This leads to an issue in the memory requirements and in the computational complexity. In order to solve these problems, sampling and clustering algorithms are applied on a large training dataset to create a dataset of limited size. Since the size of the new dataset is small, the kernel matrix becomes small and thus the computational complexity of solving Eq.1 and the memory to store this kernel matrix are greatly reduced.

The main objective of this paper is to find a solution to avoid KPCA high memory consumption. In order to evaluate our modified KPCA approach in pre-processing step, it is applied on a dataset from Telecommunication services. The accuracy of predicting churn customer is observed and used to compare the solutions. The experiments for this paper are conducted to observe: 1) the comparison of condensation methods, 2) the impact of size of training data and 3) KPCA versus LPCA.

We present an algorithm, a modified KPCA in pre-processing step, described in Algorithm 1 that reduce the size of the training set before applying KPCA. Two approaches are described in Algorithm 1: the class-based, which takes class label into account and the approach that is contrary to class-based. The latter approach selects data based on

Algorithm 1 Adapted KPCA to telecommunication with/without supervision

```
1. Divide $S^{tr}$ into groups $\{S_i^{tr}, i = 1, \cdots, M\}$ according to the class labels, where M is the total number of different labels.
2. For $i$=1 to $M$,
   2.1 Use either a clustering (e.g. K-Means algorithm) or a sampling algorithm to $S_i^{tr}$ to obtain K number of data, where K is the number of subsets after clustering $\{d_k, k = 1, \cdots, K\}$.
   2.2 For $k = 1$ to $K$,
       2.2.1 For sampling select an instance $x_{new}^k$ from $S_i^{tr}$ randomly.
             For clustering, calculate the centre of mass of cluster $d_k$ by
                 $x_{new}^k = \frac{\sum_{l=1}^{L} x_l}{L}$        (7)
             where $L$ is the size of $d_k$ and $x_l$ is a instances of $d_k$.
       2.2.2 Add $x_{new}^k$ to $S_i^{new}$.
   2.3 Store instances in $S_i^{new}$ dataset into $S^{new}$
3. Use $S^{new}$ to train KPCA by the training process described in section 2.
4. Apply trained KPCA to project the datasets $S^{new}$ and $S^{te}$ by Eq. (6).
```

condensation methods without considering the class label (churn or non-churn). There is a possibility that this approach leads to imbalanced classification problems due to disproportion in the number of class labels. The former approach, on the contrary, solves previous approach limitations by creating a dataset S^{new} that contains classes with same proportion (If K = 1024, 512 churn and 512 non-churn are allocated to form S^{new}). For the latter approach of Algorithm 1, initialise the algorithm from Step 2.1 as first step and remove Step 1 and Step 2 iteration. Replace S_i^{tr} and S_i^{new} as S^{tr} and S^{new}. On the other hand, Step 1 and Step 2 iteration are added for the class-based approach so that the size of churn and non-churn in S^{new} are due proportion.

Figure 1 illustrates the workflow of the experiments. The data provided in Step 1 is the original training dataset, S^{tr}, which contains 100,000 customers (non-churners and churners) in the ratio of 19:1. Clustering/Sampling techniques are then applied to this dataset to generate a reduced training dataset, S^{new}, of size K. In Step 3 the reduced dataset is used to train KPCA and transform the test dataset, S^{te} and possibly S^{tr}. The α in Step 3 refers to the number of attributes in transformed dataset, which is dependent on threshold parameter. These two datasets are employed for classification purposes afterwards. In addition, the transformed dataset can be visualized in Step 4.

3.2 DataSet Description

139,000 customers were randomly selected from a real world database provided by Eircom for the experiments. The training set is composed of 6,000 churners and 94,000 non-churners. The test dataset contains 39,000 customers (2,000 churners and 37,000 non-churners). These data contain 122 features which describe individual customer characteristics. Following items describe features that are considered for customer churn prediction problem. See (Huang et al. 2009) for more detailed descriptions:

- Broadband Internet and telephone line information
- Broadband monthly usage information
- Demographic profiles
- Information of grants

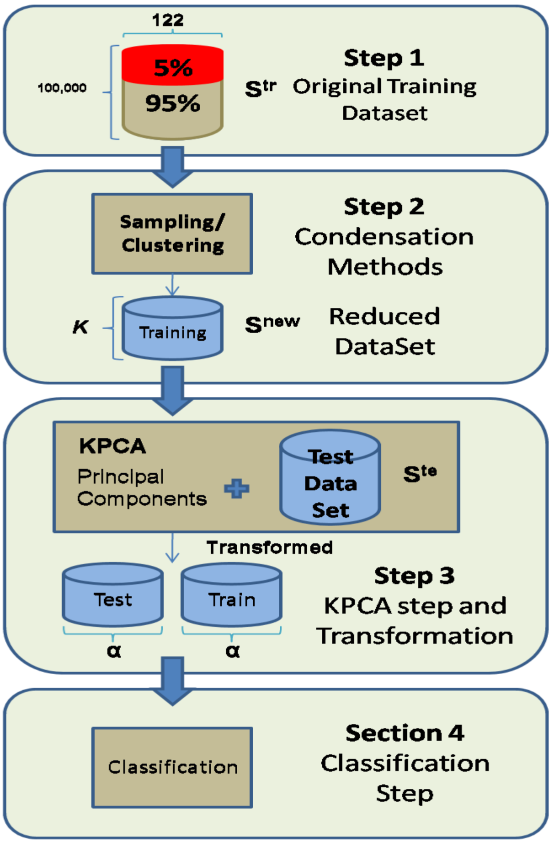


Figure 1: Workflow of the algorithm

- Account Information
- Service orders
- Henley Segments
- The historical information of payments and bills
- Call details

For example, features from *Broadband Internet and telephone line information* could give additional information as to why customer cease contract for the provided services. It includes information about voice mail service (provided or not), the number of broadband lines, the number of telephone lines, and so on. Due to confidentiality of Customer data, full details of each attribute sets cannot be explained.

3.3 Condensation Methods

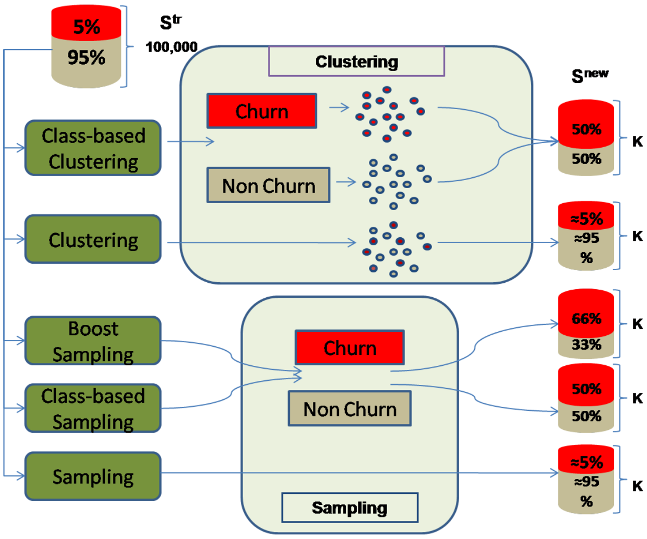


Figure 2: Condensation methods used in the experiments

Standard approach and Class-based approaches are applied on clustering and sampling techniques. Standard approach selects K cluster centre points but neglects the proportion of each class label. On the other hand, class-based approach takes the class label into account and therefore keeps a proportion of 1:1 between churners and non-churners.

There are a total of 5 different condensation methods as shown in Fig.2. First and second sets are Class-based and Standard clustering. In the clustering approach S^{tr}, the original imbalanced training dataset, is given to the K-Means algorithm, and is clustered into K groups. In Class-based Clustering, S^{tr} is separated into number of datasets based on the class label and clustering technique is applied on each subset. Although clustering a dataset may be computational expensive, this can be solved by less complex version of K-Means such as on-line K-Means.

In order to compare the interest of clustering-based condensation method, we compare it to usual sampling methods. The last three condensation methods are random Sampling, Boosted Sampling, Class-based Sampling. The first sampling method is a conventional method, which randomly selects K customers without any condition which leads to unknown size of churners and non-churners. On the contrary, the latter two methods conditionally select data. The class-based Sampling method selects equal number of customer for each class randomly to keep the ratio to 1:1. In Boosted sampling, which is also a class-based sampling, we increase the ratio of churners vs. non-churners to 2:1. Following the pre-processing step, we apply the newly generated dataset S^{new} to KPCA.

For both clustering and sampling approaches K is set to be in {2, 4, 8, 16, 32, 64, 128, 256, 512} where K is the number of customers in S^{new}, the dataset that will be given to the KPCA process.

In order to avoid biased classification 10 datasets were generated for each method for each K. These datasets are then given to KPCA to generate transformed datasets and used for churn prediction task. Following the task, the results of each dataset of same method are averaged for comparison purposes.

3.4 KPCA Configuration

Gaussian Radial Basis Function (RBF) is selected as the kernel function in the experiments. This method requires a parameter, the Gaussian width σ. The parameter σ is set to 46 throughout the experiments. The number of PC to keep can be decided either by specifying the exact number of features or setting a threshold. In the experiments, we set 0.9 as the threshold, which means that eigenvectors that account for 90% of total eigenvalues after solving Eq.1 are kept for feature projection.

The two parameters described earlier were employed based on previous experiments with smaller datasets and with different classifiers. However, the setting of these parameters should be assessed directly from the transformed dataset after KPCA.

3.5 Evaluation Criteria

According to the literature and previous experiments in (Huang et al. 2009), the Decision Tree C4.5 (R. 1996, 1993) performed the best results among other classifiers (SVM, Neural Networks) when applied on telecommunication data. Therefore, the Decision Tree C4.5 was employed.

The performance of the predictive churn model has to be evaluated. Table 1 shows a confusion matrix, where a_{11}, resp. a_{22} is the number of the correctly predicted churners, resp. non-churners, and a_{12}, resp. a_{21} is the number of the incorrectly predicted churners, resp. non-churners. Following evaluation criteria are used in the experiments (Hamilton et al. n.d.);

		Predicted	
		CHU	NONCHU
Actual	CHU	a_{11}	a_{12}
	NONCHU	a_{21}	a_{22}

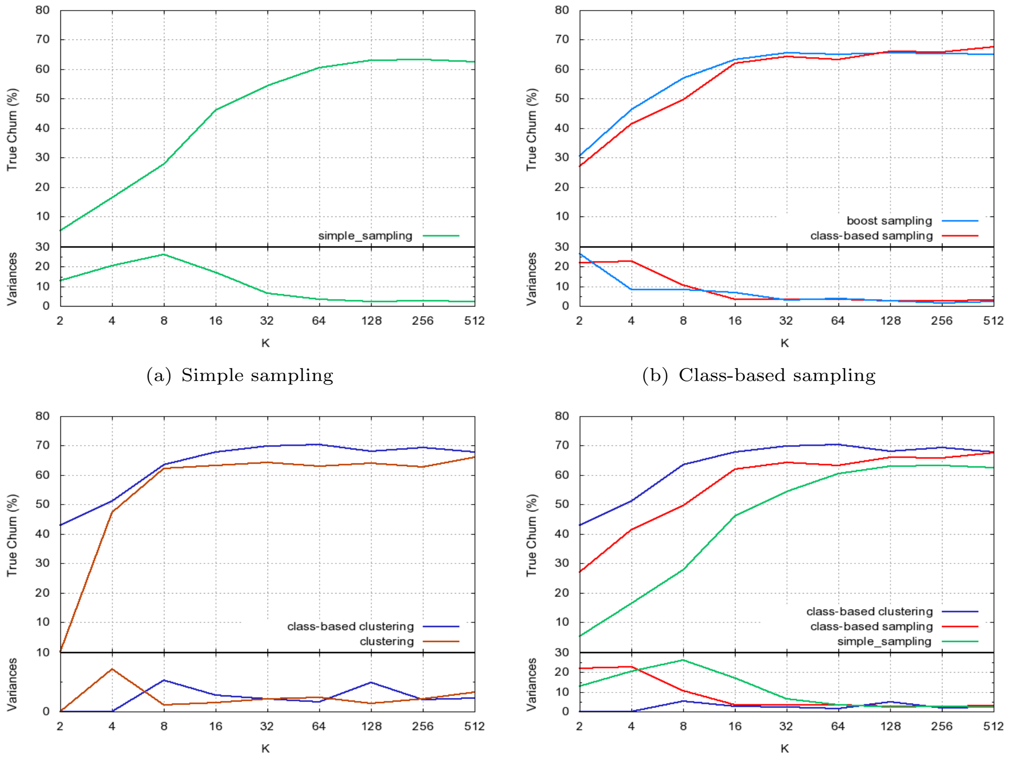
Table 1: Confusion Matrix

- The accuracy of true churn (TP) is defined as the proportion of churn cases that were classified correctly: TP = $\frac{a_{11}}{a_{11}+a_{12}}$.
- The false churn rate (FP) is the proportion of non churn cases that were incorrectly classified as churn: FP = $\frac{a_{21}}{a_{21}+a_{22}}$.

A good solution should have both a high TP and a low FP. When comparing two solutions, S_1 and S_2, if S_1'TP is above S_2'TP and S_1'FP is below S_2' FP, S_1 is considered to dominate S_2 and is considered as the best solution. When no solution is dominant, the evaluation depends on the expert strategy, i.e. to favour TP or FP.

4 Results and Discussion

Figure.3 presents the averaged results and the deviation of each condensation methods with a S^{new} of different size. Among the condensation methods, simple sampling produced the lowest churn prediction results with a maximum of 63% at K=256, resp. 62% at K=512, as shown in Fig.3(a). In Fig.3(b), the class-based sampling methods performed better



(a) Simple sampling

(b) Class-based sampling

(c) Clustering approach

(d) Approach comparison

Figure 3: TP and variance for diverse condensation methods with varying size

with 68% and 66% as the highest TP for class-based (1:1) and boost (2:1) sampling respectively. Although the boost sampling produced similar results to class-based sampling, increasing the number of churn in S^{new} do not increase the TP accuracy. The class-based clustering approach recorded the highest TP rate among other condensation methods with 70% at K=64 shown in Fig.3(c). On the other hand, standard clustering produced similar results to class-based sampling methods. Fig.3(d) compares the best approach from each previous figure to highlight the difference between sampling and clustering approaches. It includes simple and class-based sampling and class-based clustering methods.

The curve line in each graph in Fig.3 indicate that it is not necessary to consider all the data to obtain maximum accuracy. On all approaches after a given K, results tend to be of same order.

	LPCA	Class.Clus	Simple.Samp	Class.Samp
TP (%)	69.35	73.45	67	71.65
FP (%)	4.05	1.703	2.56	2.83

Table 2: The best Prediction rates vs. LPCA and the modified KPCA approach

Table 2 compares the prediction rates between the adapted KPCA and LPCA and Figure 4 plots the first PC against the second PC following the KPCA projections for 2D visualization. The dataset that produced the best results out of 10 datasets from each condensation methods in Fig.3 were used as training set and compared. The test set used for Figure 5 is described in Section 3.2. The red and green dots refer to churners and non-churners respectively. We can see that class-based clustering approach produced the best results with best separability of classes among the 4 approaches. Fig.5 plots the receiver operating characteristics (ROC) of TP against FP. It shows that the values of class-based clustering is above of both simple and class-based sampling (it dominates them except for K=2). Class-based Clustering with K = 16 has TP of around 70% and FP of below 1%. The remaining 30% of TP is classified as non churner while being churner. Similarly, the remaining 99% of FP is classified correctly as non churner. Therefore class-based clustering approach is the best approach among the condensation methods: *class-based clustering* dominates *clustering* and *clustering* dominates *sampling*.

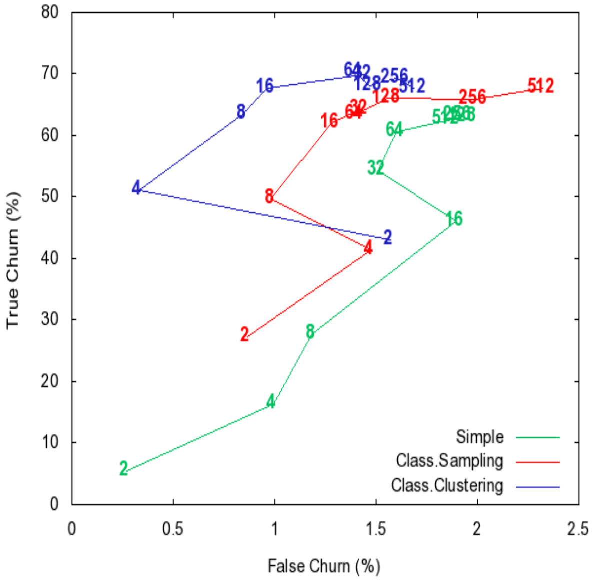
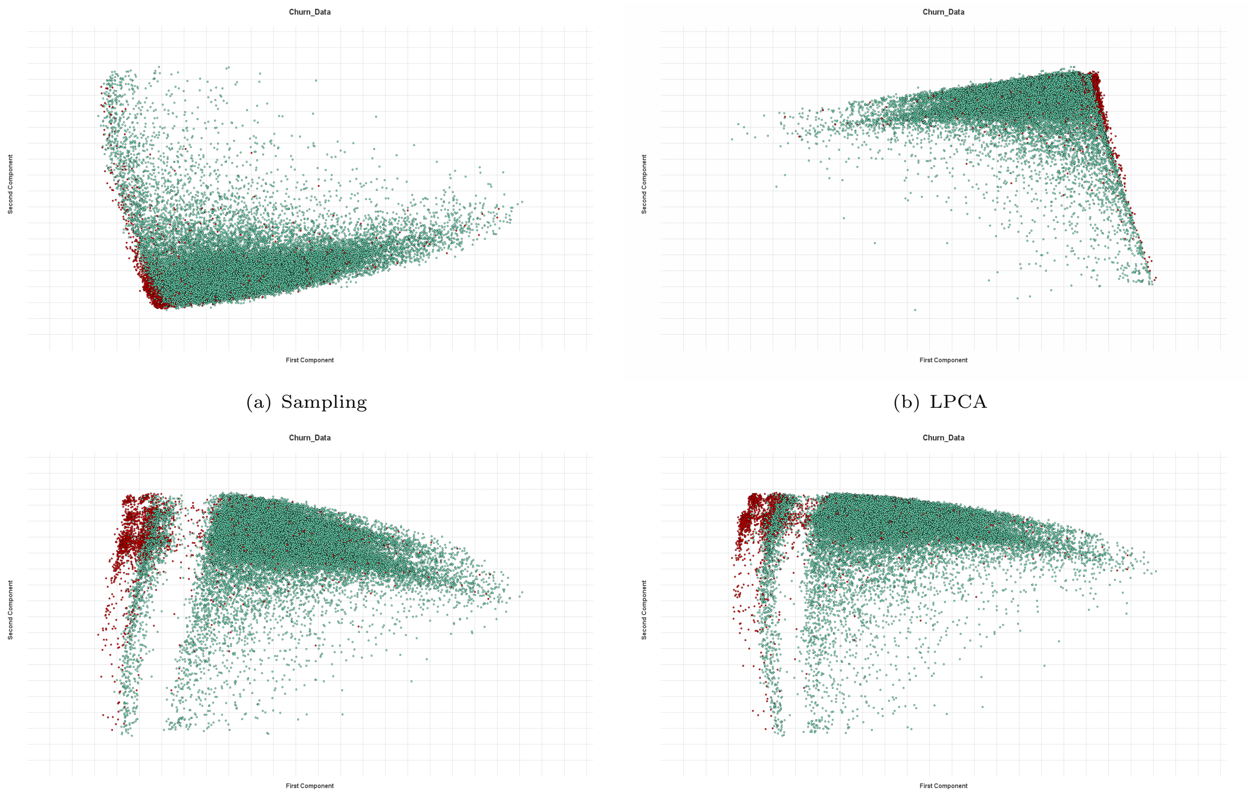


Figure 5: ROC graph: True Churn vs. False Churn for different approaches



(a) Sampling

(b) LPCA

(c) Class-based Clustering

(d) Class-based Sampling

Figure 4: 2D visualization for different approaches after KPCA/LPCA

In summary, the experiments showed that clustering approach produced a more adequate summary of the dataset than sampling approach and leads to a better prediction rate both in TP and FP. To conclude, KPCA with clustering approach in pre-processing step is more efficient than LPCA and sampling approach.

5 Conclusions and Future Perspective

In this paper, we have proposed a modified KPCA approach to be able to process large datasets and to improve performances on imbalanced classification problem. The proposed KPCA approach focuses on reducing the memory requirements of the training process in the KPCA by reducing the size of the training dataset through sampling and clustering techniques. To solve KPCA limitations, Kernel matrix size is reduced as it accounts for most memory space in KPCA and the number of each class in a training dataset is adjusted for balanced classification purpose. Condensing techniques used were simple and class-based sampling and K-Means clustering.

The modified approach was tested on a telecommunication dataset on a churn prediction task. The results show that KPCA with clustering approach outperformed LPCA and sampling approach. In addition to this, the approach that takes class label into account performed better in both clustering and sampling.

However, there are some problems in using the proposed approach. Class-based clustering algorithm produced the best results but we need to investigate if a different clustering technique will produce better results than K-Means. We plan to use other clustering algorithms such as methods based on density, like DBSCAN, or based on subspaces, like CLIQUE.

Another limitation of using KPCA is the evaluation of the transformed features. It should be assessed directly without relying on the classifier results, because this evaluation is very costly. Therefore, we will measure and assess the separability of classes to be able to select internally the most suitable feature transformation.

Finally, the right parameters should be discovered automatically. In this paper, we set the Gaussian kernel width as 46 and the number of PC with a threshold of 0.9 based on previous experiments. In future works, we plan to develop a method to estimate automatically these parameters.

References

Chawla, N. V., Japkowicz, N. & Kotcz, A. (2004), ‘Editorial: special issue on learning from imbalanced data sets’, *SIGKDD Explor. Newsl.* **6**(1), 1–6.

Franc, V., V. (2003), Greedy algorithm for a training set reduction in the kernel methods, *in* ‘10th International Conference on Computer Analysis of Images and Patterns’, Groningen, The Netherlands, pp. 426–433.

This research was partly supported by Eircom of Ireland.

Hamilton, H., Gurak, E., Findlater, L., Olive, W. & Ranson, J. (n.d.). `http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html`.

Huang, B., Kechadi, M.-T. & Buckley, B. (2009), Customer churn prediction for broadband internet services, *in* '11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2009)', LNCS 5691, Linz, Austria, pp. 229–243.

I.T.Jolliffe (1986), *Principal Components Analysis*, Springer-Verlag.

Kim, K.I., F. M. S. B. (2005), 'Iterative kernel principal component analysis for image modelling', *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1351–1366.

Marukatat, S. (2006), 'Sparse kernel pca by kernel k-means and preimage reconstruction algorithms', *Trends in Artificial Intelligence* **Volume 4099/2006**, 454–463.

R., Q. J. (1993), 'C4.5: Programs for machine learning'.

R., Q. J. (1996), 'Improved use of continuous attributes in c4.5', *Journal of Artificial Intelligence Research* **4**, 77–90.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G. & Smola, A. (1998), 'Input space versus feature space in kernel-based methods', *IEEE Transactions on Neural Networks* **5**(10), 10001016.

Schölkopf, B., Smola, A. & Müller, K. (1998), 'Nonlinear component analysis as a kernel eigenvalue problem', *Neural Computation* **5**(10), 1299–13995.