UT DALLAS
CLASSIFYING ANONYMIZED DATA

Together with the anonymization toolbox, we also release the source code of our recent study on classifying anonymized data [1]. In this study, we proposed methods for building distance-based classification models over anonymized data. More specifically, investigated methods included instance-based classifiers (also called k-nearest neighbor classification), and support vector machines (SVMs). To implement anonymized versions of these models, we extended the $IB1$ classifier (i.e., 1-nearest neighbor) of the WEKA data mining library [2] and Java version of the $LibSMV$ library for SVM classification [3].

Most research in the area assume that classification models will be both trained and tested over anonymized data. However, as described in [1], various other alternatives are possible in real-life applications. For example, the model might be built on original data and queried with anonymized data or vice verse. Our experiments also focus on these distinct test scenarios.

Below we provide detailed information on how our experimental results can be re-generated using the $Census-income$ dataset [4] or any other dataset. In section 1, we discuss shared parameters of our implementation of IB1 and SVM classifiers over anonymized datasets. Then section 2 provides the details on IB1 classification and section 3 on SVM classification.

1 Common Parameters

Three global parameters are common to both SVM and IB1 classification experiments. These are:

- *testScenario* that specifies whether training and test datasets are anonymized or not. Four different scenarios are possible:
  - 00: training/testing over original data,
  - 11: training/testing over anonymized data,
  - 10: training over anonymized, testing over original data, and
  - 01: training over original, testing over anonymized data.

  If the parameter is set to 00, the performance of the corresponding classification over original data can be obtained. In this case, the toolbox calls the unaltered $IB1$ method of WEKA or SVM classifier of $LibSVM$.

- $arff$ that specifies the path to text file which contains schema information of the input dataset in ARFF file format [2]. This file should contain the $@RELATION$ specification for relation name, the $@ATTRIBUTE$ specifications for each attribute (in the original dataset) and the $@DATA$ mark indicating the end of attribute information. An example file is provided for the $Census - income$ dataset.

- $crossVal$ that specifies the number of folds of experiments to be performed. The default value is set to 2.

Apart from these parameters, also common to both methods are anonymization parameters of the toolbox (e.g., output format, anonymization method, privacy parameters, etc.). Please refer to the anonymization manual of the toolbox for details.

Among anonymization parameters, the option $-outputFormat$ should always be omitted. This is because in every possible test scenario, the toolbox will decide on the correct output format based on the other parameters. If set, the value of this option will simply be disregarded by the toolbox.

2 IB1 Classification

The only additional parameter necessary for IB1 classification over anonymized data is the path to the anonymization configuration file. Based on the configuration, we read the value generalization hierarchies (VGHs) of quasi-identifier attributes. These VGHs allow efficient and accurate calculation of expected distance between original and/or anonymized values from corresponding (possibly generalized) attribute domains.

Since our $IB1$ implementation extends that of the WEKA library, it is important that anonymized data be represented in ARFF file format. This we achieve simply by declaring quasi-identifier attributes as of type String. Then, our toolbox can resolve the specific value (either generalized or not) by itself.

In order to use our anonymized IB1 classification method like other classifiers of the WEKA library, one only needs to specify the configuration file used for generating the anonymized data through the $-config$ option. However, notice that, for proper handling of anonymized data, as described above, all quasi-identifiers within the ARFF file should be declared as of type String.

3 SVM Classification

SVM classification is much more trickier than IB1, because there are various different parameters regarding feature representation of anonymization data. On top these are the choices of SVM kernel and expected distance function.

As discussed in [1], all 4 well-known kernel types are supported. The choice can be specified through the $-kernel$ option according to the following keys: 0 (linear kernel), 1 (polynomial kernel), 2 (RBF kernel) and 3 (Sigmoid kernel). Unfortunately, in each case,

Option	Value	Description
-catRep	1	generalization is a new feature
-catRep	2	set all VGH values up to the suppression value as features
-catRep	3	set all ground domain values of the generalization
-numRep	1	generalization is a new feature
-numRep	2	replace with the mid-point
-numRep	3	set 2 features for upper and lower bound of generalization

Table 1: Feature representation options for categorical and numeric attributes

there is no way to specify kernel-related parameters. The toolbox currently uses the $LibSVM$ defaults.

In our previous studies, we proposed two methods of calculating expected distances over anonymized data. The first one appear in [5] and assumes that all ground-domain values of a generalization are uniformly distributed. For example, if $AnySex$ represents $\{Male, Female\}$, this expected distance method assumes $Male$ and $Female$ are equally likely. This expected distance calculation method can be invoked by setting the $-uni$ option to $true$ (i.e., include $-unitrue$ to your arguments). Automatically, the toolbox will set the feature representation parameters as well as the anonymization configuration's output format ($genVals$ will be assumed).

The other expected distance function was proposed in [1]. Here, the idea is to release additional statistical information with every quasi-identifier attribute to facilitate more accurate distance calculation without violating privacy. This method can be invoked by setting $-pdf$ option to $true$ (i.e., include $-pdftrue$ to your arguments). Again, the toolbox will set the output format $genValsDist$ automatically and take care of the parameters related to feature representation.

Next we discuss how various feature representation heuristics can be invoked through parameters. In [1] we proposed 3 heuristics each for categorical and numerical attributes. The choices are set through the $-catRep$ and $-numRep$ options respectively.

Notice that within such crowded a set of parameters, not every possible combination is valid. Whenever the parameters set by the user are invalid, the toolbox either fires an exception and exists, or tries to fix the situation by overwriting certain parameter values with correct ones. Some such cases are discussed below. Yet, the list definitely is not exhaustive.

- $-outputFormat$ option, if set, is always ignored.
- For anonymization methods that define their own VGHs (e.g., Mondrian [6], only certain feature representation heuristics can be used. These are $-catRep3$ in combination with $-numRep1$ or $-numRep2$.
- If $-pdf$ is set, $-catRep$, $-numRep$ and $-uni$ cannot be specified.
- Similarly, if $-uni$ is set, $-catRep$, $-numRep$ and $-pdf$ cannot be specified.

References

[1] A. Inan, M. Kantarcioglu, and E. Bertino, "Using anonymized data for classification," in *ICDE*, 2009, pp. 429–440.

[2] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.

[3] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] D. Newman, S. Hettich, C. Blake, and C. Merz, "UCI repository of machine learning databases," 1998.

[5] A. Inan, M. Kantarcioglu, E. Bertino, and M. Scannapieco, "A hybrid approach to private record linkage," in *ICDE*, 2008, pp. 496–505.

[6] K. Lefevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 25–36.