**FS-SSVD based perturbation method**

## Dataset

The dataset WDBC is taken from the UCI repository. We use the "wdbc\_30.mat" and "class.mat" files for input to MATLAB.

* **wdbc\_30.mat**: contain 30 real features of WDBC
* **Class.mat**: contain class labels of WDBC

## Running Environment

Both perturbation methods are implemented in MATLAB (7.11) and to implement classification algorithms and feature selection algorithm, WEKA (3.7) is used. Also test runs are performed on a personal computer with CPU.2.27 GHz and RAM 3GB.

## Executable Programs

## There are 24 MATLAB source files in Source Code folders

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | AttributeRankChange.m | 13 | WeightedAttributeRankChange.m |
| 2 | AttributeRankMaintanace.m | 14 | WeightedAttributeRankMaintanace.m |
| 3 | RankMaintanance.m | 15 | WeightedRankMaintanance.m |
| 4 | RankPosition.m | 16 | WeightedRankPosition.m |
| 5 | RelativeError.m | 17 | WeightedRelativeError.m |
| 6 | Thin\_SVD.m | 18 | ThinSVD\_Method.m |
| 7 | Single\_SSVD.m | 19 | SSVD\_Method.m |
| 8 | FS\_SSVD\_Method\_AllSetsOfPrivacyLevels.m | 20 | FS\_SSVD\_Method.m |
| 9 | Generate\_PrivacyWeightsOfFeaturesRandomly.m | 21 | ComputeDistortionMetrics.m |
| 10 | **ComputationalCostOF\_SVDMethods.m** | 22 | OSumNormalization.m |
| 11 | FeatureSelection.m | 23 | AverageOfMatrix.m |
| 12 | SubWDBCArffWrite.m | 25 | RankOfElements.m |
|  |  |  |  |

There are five main methods to implement the proposed method FS-SSVD and to compare it with   
Thin-SVD and SSVD methods as follows:

* Features valuation method
* FS-SSVD based perturbation method
* SSVD based perturbation method
* Thin-SVD based perturbation method
* Computational Cost

1. **Execution of Features valuation method**

#### To obtain the output of the features valuation phase in fig.1 of the paper and create Array of privacy levels of features (W) according of section 3.3.1, run the Generate\_PrivacyWeightsOfFeaturesRandomly.m using the following parameters:

#### [A]: The initial data matrix

#### [k]: The number of random generation

#### Run the function in MATLAB as shown in Fig. 1.

#### 

#### generate W.jpg

Fig.1. Screenshot for execution of features valuation method

The execution' result of the features valuation method based on above commands (data matrix W)  
 is shown in section E.

#### Execution of FS-SSVD based perturbation Method

#### First we need to implement the process of superior features selection. We use of Information gain algorithm in WEKA and produce array R (an array of features rank). Some sample screenshots on how feature selection algorithm was run on our machine can be found in the screenshots folder.

#### To implement the Algorithm1 of the paper, in accordance with the procedures provided in Table 1, run the FS\_SSVD\_Method\_AllSetsOfPrivacyLevels.m. In this function for every rank *k* and for every set privacy levels, the distortion amount of the perturbed matrix are obtained and then the average value ​​for each criterion is calculated. Also for every rank *k,* data perturbed matrix in ARFF format is produced. To run this function, use the following parameters:

#### [A]: The initial data matrix

#### [ev]: The sparsification parameter of matrix V in single-SVD method

#### [eu]: The sparsification parameter of matrix V in single-SVD method

#### [C]: The array of class labels

#### [R]: The array of features index according of their rank that are obtained from feature selection algorithm

#### [NSF]: The number of selected features

#### Run the function in MATLAB as shown in Fig. 2:

#### FS-SSVD.jpg

Fig.2. Screenshot for execution of FS-SSVD method

#### To calculate the mining accuracy, we use of SVM classification in WEKA based on the ARFF files are generated in the step 2. Some sample screenshots on how SVM classification was run on our machine can be found in the screenshot folder.

The execution' results of the FS-SSVD method (data distortion metrics, accuracy metric) is shown in section E.

#### Execution of SSVD based perturbation Method

#### To implement the SSVD method to compare with the proposed method FS-SSVD, run the SSVD\_Method.m. In this function for every rank *k*, the distortion amount of the perturbed matrix are obtained and then data perturbed matrix as an ARFF formatted file are produced. To run this function, use the following parameters:

#### [A]: The initial data matrix

#### [ev]: The sparsification parameter of matrix V in single-SVD method

#### [eu]: The sparsification parameter of matrix V in single-SVD method

#### [C]: The array of class labels

#### Run the function in MATLAB as shown in Fig. 3:

#### SSVD.jpg

Fig.3. Screenshot for execution of SSVD method

#### To calculate the mining accuracy, we use of SVM classification in WEKA based on the ARFF files are generated in the step 2. Some sample screenshots on how SVM classification was run on WEKA software can be found in the screenshot folder.

The execution' results of the FS-SSVD method (data distortion metrics, accuracy metric) is shown in section E.

#### Execution of Thin-SVD Method

## To implement the Thin-SVD method to compare with the proposed method FS-SSVD, run the ThinSVD\_Method.m . In this function for every rank *k*, the distortion amount of the perturbed matrix are obtained and then data perturbed matrix as an ARFF formatted file are produced. To run this function, use the following parameters:

#### [A]: The initial data matrix

#### [C]: The array of class labels

#### Run the function in MATLAB as shown in Fig. 3:

#### Thi_SVD.jpg

Fig.3. Screenshot for execution of Thin-SVD method

#### To calculate the mining accuracy, we use of SVM classification in WEKA based on the ARFF files are generated in the step 2. Some sample screenshots on how SVM classification was run on WEKA software can be found in the screenshot folder.

The execution' results of the FS-SSVD method (data distortion metrics, accuracy metric) is shown in section E.

#### Computational Cost

In order to evaluate the computational cost of SVD based methods, run the **ComputationalCostOF\_SVDMethods.m.** This function measure the elapsed time of the Thin-SVD and SSVD methods. To run this function, use the following parameters:

#### [A]: The initial data matrix

#### [ev]: The sparsification parameter of matrix V in single-SVD method

#### [eu]: The sparsification parameter of matrix V in single-SVD method

### Screenshot

### Some sample screenshots on how feature selection algorithm and classification was run on WEKA software can be found in the screenshot folder.

## Output

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | **privacy levels of features (W)** | | | | | | | | | |
| **Weighting1** | **Weighting 2** | **Weighting 3** | **Weighting 4** | **Weighting 5** | **Weighting 6** | **Weighting 7** | **Weighting 8** | **Weighting 9** | **Weighting 10** |
| 1 | 0.7642 | 0.2325 | 0.5404 | 0.3645 | 0.1495 | 0.3832 | 0.1428 | 0.7733 | 0.5751 | 0.7393 |
| 2 | 0.6073 | 0.2966 | 0.3874 | 0.0273 | 0.4395 | 0.5697 | 0.7537 | 0.5347 | 0.1652 | 0.3095 |
| 3 | 0.6133 | 0.9418 | 0.0364 | 0.6441 | 0.2336 | 0.922 | 0.0927 | 0.7217 | 0.723 | 0.1041 |
| 4 | 0.7599 | 0.6257 | 0.0877 | 0.0227 | 0.4078 | 0.9244 | 0.3356 | 0.8644 | 0.9879 | 0.0198 |
| 5 | 0.0984 | 0.2761 | 0.712 | 0.3469 | 0.5723 | 0.5648 | 0.7267 | 0.196 | 0.7986 | 0.5213 |
| 6 | 0.5896 | 0.0666 | 0.6268 | 0.8643 | 0.4727 | 0.8727 | 0.7446 | 0.9524 | 0.1283 | 0.0167 |
| 7 | 0.2086 | 0.5771 | 0.3407 | 0.0545 | 0.071 | 0.1758 | 0.7536 | 0.2931 | 0.8097 | 0.7212 |
| 8 | 0.1455 | 0.7578 | 0.7137 | 0.6161 | 0.4843 | 0.5993 | 0.578 | 0.8391 | 0.9855 | 0.0681 |
| 9 | 0.662 | 0.0642 | 0.9569 | 0.3952 | 0.4907 | 0.4972 | 0.9279 | 0.2917 | 0.3359 | 0.3558 |
| 10 | 0.2914 | 0.5698 | 0.5774 | 0.8665 | 0.9036 | 0.5047 | 0.6466 | 0.4834 | 0.1815 | 0.1951 |
| 11 | 0.4288 | 0.1889 | 0.6948 | 0.2079 | 0.4349 | 0.3694 | 0.7645 | 0.9245 | 0.3134 | 0.9427 |
| 12 | 0.3973 | 0.4717 | 0.0889 | 0.837 | 0.4273 | 0.1748 | 0.0195 | 0.0563 | 0.9942 | 0.672 |
| 13 | 0.7074 | 0.4867 | 0.0053 | 0.5315 | 0.7628 | 0.0027 | 0.3331 | 0.8445 | 0.0822 | 0.2321 |
| 14 | 0.1393 | 0.5445 | 0.6991 | 0.1549 | 0.3663 | 0.0935 | 0.275 | 0.0441 | 0.0648 | 0.3588 |
| 15 | 0.6684 | 0.6399 | 0.924 | 0.2203 | 0.5442 | 0.3775 | 0.7641 | 0.6392 | 0.8328 | 0.9421 |
| 16 | 0.2477 | 0.1295 | 0.6733 | 0.6814 | 0.9419 | 0.4155 | 0.7535 | 0.7782 | 0.5794 | 0.7552 |
| 17 | 0.8604 | 0.3469 | 0.1684 | 0.6176 | 0.0298 | 0.5498 | 0.4943 | 0.4377 | 0.6707 | 0.9659 |
| 18 | 0.8998 | 0.0967 | 0.2752 | 0.749 | 0.9832 | 0.8691 | 0.9809 | 0.3097 | 0.0142 | 0.4446 |
| 19 | 0.6002 | 0.4245 | 0.8265 | 0.6607 | 0.5892 | 0.3672 | 0.7521 | 0.1671 | 0.9428 | 0.6023 |
| 20 | 0.4417 | 0.8827 | 0.5481 | 0.1699 | 0.5569 | 0.9568 | 0.6799 | 0.7384 | 0.1129 | 0.5064 |
| 21 | 0.8347 | 0.2499 | 0.6741 | 0.9262 | 0.5326 | 0.8827 | 0.9544 | 0.8928 | 0.3413 | 0.224 |
| 22 | 0.7186 | 0.2562 | 0.3186 | 0.1779 | 0.3042 | 0.664 | 0.1631 | 0.8749 | 0.1046 | 0.692 |
| 23 | 0.6255 | 0.4604 | 0.6214 | 0.9296 | 0.853 | 0.1569 | 0.5295 | 0.3239 | 0.5174 | 0.6598 |
| 24 | 0.7197 | 0.6504 | 0.2737 | 0.9023 | 0.0496 | 0.6761 | 0.8677 | 0.7478 | 0.7332 | 0.2613 |
| 25 | 0.0509 | 0.3316 | 0.0533 | 0.826 | 0.0815 | 0.5942 | 0.9508 | 0.0316 | 0.2376 | 0.8622 |
| 26 | 0.4861 | 0.2999 | 0.5364 | 0.9386 | 0.8714 | 0.1997 | 0.2033 | 0.453 | 0.4124 | 0.2478 |
| 27 | 0.8758 | 0.0138 | 0.6502 | 0.7678 | 0.3234 | 0.0187 | 0.1293 | 0.0902 | 0.2728 | 0.2831 |
| 28 | 0.9589 | 0.1604 | 0.702 | 0.0296 | 0.3606 | 0.4818 | 0.9624 | 0.3842 | 0.1348 | 0.2769 |
| 29 | 0.4983 | 0.8437 | 0.7216 | 0.4655 | 0.8744 | 0.3595 | 0.3062 | 0.1399 | 0.5977 | 0.0024 |
| 30 | 0.4437 | 0.5286 | 0.6 | 0.7222 | 0.3483 | 0.5846 | 0.5381 | 0.5203 | 0.8772 | 0.4397 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| The FS-SSVD based perturbation method on WDBC (569\*22), ev=0.02, eu=0.038 | | | | | | |
| **rank** | **Average Data Value Distortion Metrics for 10 privacy level sets** | | | | | **Data Utility Metric** |
| **WRE** | **WRP** | **WRK** | **WCK** | **WCP** | **Accuracy** |
| 1 | 0.71898 | 255.77433 | 0.0005292 | 0.0284 | 6.67455 | 90.6854 |
| 2 | 0.7124 | 254.38722 | 0.0008987 | 0.0284 | 7.82976 | 90.6854 |
| 3 | 0.71174 | 253.16976 | 0.001 | 0.0284 | 7.82976 | 90.6854 |
| 4 | 0.71142 | 254.39693 | 0.0012 | 0.0455 | 3.46563 | 90.6854 |
| 5 | 0.71137 | 247.84104 | 0.0012 | 0.0455 | 3.46563 | 90.6854 |
| 6 | 0.71136 | 241.9468 | 0.0013 | 0.0511 | 3.59398 | 92.0914 |
| 7 | 0.71136 | 240.88158 | 0.0013 | 0.0511 | 3.46563 | 92.2671 |
| 8 | 0.71136 | 233.41681 | 0.0013 | 0.0341 | 3.97905 | 92.7944 |
| 9 | 0.71136 | 219.62086 | 0.0012 | 0.0398 | 4.23577 | 92.2671 |
| 10 | 0.71136 | 198.03349 | 0.0013 | 0.0341 | 3.33728 | 91.3884 |
| 11 | 0.71136 | 196.2313 | 0.0013 | 0.0341 | 3.46563 | 92.4429 |
| 12 | 0.71136 | 193.15233 | 0.0014 | 0.0341 | 3.46563 | 92.6186 |
| 13 | 0.71136 | 190.70339 | 0.0015 | 0.0398 | 3.20891 | 92.4429 |
| 14 | 0.71136 | 188.84391 | 0.0015 | 0.0455 | 3.20891 | 92.6186 |
| 15 | 0.71136 | 183.44957 | 0.0014 | 0.0398 | 3.33728 | 92.6186 |
| 16 | 0.71136 | 181.82221 | 0.0015 | 0.0398 | 3.33728 | 92.6186 |
| 17 | 0.71136 | 179.3453 | 0.0016 | 0.0511 | 3.08056 | 92.9701 |
| 18 | 0.71136 | 178.56794 | 0.0016 | 0.0511 | 3.08056 | 93.8489 |
| 19 | 0.71136 | 178.34147 | 0.0015 | 0.0511 | 3.08056 | 93.6731 |
| 20 | 0.71136 | 178.1838 | 0.0016 | 0.0455 | 3.20891 | 93.8489 |
| 21 | 0.71136 | 177.93429 | 0.0016 | 0.0455 | 3.20891 | 93.4974 |
| 22 | 0.71136 | 177.76284 | 0.0016 | 0.0455 | 3.20891 | 93.4974 |
| Average For All Ranks | 0.711774 | 209.264 | 0.001333 | 0.041332 | 3.94405 | 92.31506364 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **The SSVD based perturbation method on WDBC (569\*30), ev=0.02, eu=0.038** | | | | | | |
| **rank** | **Data Value Distortion Metrics** | | | | | **Data utility Metric** |
| **RE** | **RP** | **RK** | **CK** | **CP** | **Mining Accuracy (%)** |
| 1 | 0.5092 | 185.5859 | 0.0037 | 0.2 | 7.6 | 90.6854 |
| 2 | 0.5046 | 184.8655 | 0.0059 | 0.1667 | 8.6 | 90.6854 |
| 3 | 0.5041 | 184.233 | 0.0067 | 0.1667 | 8.6 | 90.6854 |
| 4 | 0.5039 | 184.8709 | 0.0074 | 0.2333 | 5.3333 | 90.6854 |
| 5 | 0.5038 | 179.9374 | 0.0077 | 0.3 | 4.6 | 90.6854 |
| 6 | 0.5038 | 176.8318 | 0.0079 | 0.2333 | 4.8667 | 92.2671 |
| 7 | 0.5038 | 175.9832 | 0.008 | 0.2333 | 4.7333 | 92.2671 |
| 8 | 0.5038 | 171.7193 | 0.0082 | 0.2333 | 4.8667 | 92.9701 |
| 9 | 0.5038 | 162.6518 | 0.0084 | 0.2 | 4.9333 | 92.4429 |
| 10 | 0.5038 | 160.2438 | 0.0089 | 0.3 | 5.3333 | 92.2671 |
| 11 | 0.5038 | 148.0288 | 0.009 | 0.2667 | 3.6 | 92.2671 |
| 12 | 0.5038 | 147.4946 | 0.0097 | 0.2667 | 3.6667 | 92.4429 |
| 13 | 0.5038 | 142.6239 | 0.0097 | 0.3333 | 2.9333 | 92.9701 |
| 14 | 0.5038 | 139.8204 | 0.0096 | 0.3333 | 2.9333 | 92.4429 |
| 15 | 0.5038 | 138.9829 | 0.0094 | 0.3333 | 2.9333 | 92.2671 |
| 16 | 0.5038 | 133.2498 | 0.0101 | 0.2667 | 2.9333 | 92.2671 |
| 17 | 0.5038 | 130.8121 | 0.0105 | 0.3667 | 2.7333 | 94.0246 |
| 18 | 0.5038 | 129.6974 | 0.0107 | 0.2667 | 2.6667 | 93.3216 |
| 19 | 0.5038 | 128.4674 | 0.0107 | 0.3 | 2.7333 | 93.3216 |
| 20 | 0.5038 | 127.5582 | 0.0115 | 0.3 | 2.7333 | 92.9701 |
| 21 | 0.5038 | 127.1537 | 0.0112 | 0.3 | 2.7333 | 92.9701 |
| 22 | 0.5038 | 125.85 | 0.0115 | 0.2667 | 2.7333 | 92.2671 |
| 23 | 0.5038 | 125.3793 | 0.0113 | 0.3 | 2.8 | 91.9156 |
| 24 | 0.5038 | 125.0069 | 0.0108 | 0.3 | 2.8 | 92.7944 |
| 25 | 0.5038 | 125.0141 | 0.0105 | 0.3333 | 2.6667 | 93.1459 |
| 26 | 0.5038 | 124.5877 | 0.011 | 0.3333 | 2.6667 | 93.1459 |
| 27 | 0.5038 | 124.2025 | 0.0116 | 0.3333 | 2.6667 | 92.6186 |
| 28 | 0.5038 | 124.0149 | 0.0117 | 0.3667 | 2.6 | 92.2671 |
| 29 | 0.5038 | 123.9049 | 0.0119 | 0.3667 | 2.6 | 92.4429 |
| 30 | 0.5038 | 124.1666 | 0.0122 | 0.3333 | 2.6667 | 92.6186 |
| Average For All Ranks | 0.50402 | 146.0979 | 0.0095 | 0.2844 | 3.875 | 92.325 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **The Thin-SVD based perturbation method on WDBC (569\*30)** | | | | | | |
| **rank** | **Data Value Distortion Metrics** | | | | | **Data Utility Metric** |
| **RE** | **RP** | **RK** | **CK** | **CP** | **Mining Accuracy (%)** |
| 1 | 0.0872 | 122.2534 | 0.0116 | 0.7 | 0.3333 | 92.2671 |
| 2 | 0.0341 | 128.1525 | 0.0374 | 0.8667 | 0.1333 | 92.2671 |
| 3 | 0.0188 | 121.3221 | 0.0504 | 1 | 0 | 92.4429 |
| 4 | 0.0054 | 87.9523 | 0.08 | 1 | 0 | 95.4306 |
| 5 | 0.0022 | 79.3115 | 0.1005 | 1 | 0 | 94.9033 |
| 6 | 0.0012 | 76.9913 | 0.1299 | 1 | 0 | 94.9033 |
| 7 | 0.0006 | 75.1917 | 0.1721 | 1 | 0 | 94.5518 |
| 8 | 0.0004 | 68.681 | 0.1882 | 1 | 0 | 95.0791 |
| 9 | 0.0003 | 58.6995 | 0.2028 | 1 | 0 | 95.2548 |
| 10 | 0.0002 | 56.0028 | 0.2343 | 1 | 0 | 95.4306 |
| 11 | 0.0001 | 47.85 | 0.2827 | 1 | 0 | 96.3093 |
| 12 | 0.0001 | 46.1452 | 0.3125 | 1 | 0 | 96.3093 |
| 13 | 0.0001 | 41.1896 | 0.319 | 1 | 0 | 96.3093 |
| 14 | 0 | 35.6657 | 0.3264 | 1 | 0 | 97.0123 |
| 15 | 0 | 34.5664 | 0.3454 | 1 | 0 | 96.4851 |
| 16 | 0 | 29.5415 | 0.3524 | 1 | 0 | 97.891 |
| 17 | 0 | 24.9743 | 0.356 | 1 | 0 | 97.7153 |
| 18 | 0 | 22.331 | 0.3729 | 1 | 0 | 97.5395 |
| 19 | 0 | 19.6499 | 0.3856 | 1 | 0 | 97.5395 |
| 20 | 0 | 17.2439 | 0.4057 | 1 | 0 | 97.7153 |
| 21 | 0 | 15.7708 | 0.4314 | 1 | 0 | 98.0668 |
| 22 | 0 | 12.355 | 0.4482 | 1 | 0 | 97.891 |
| 23 | 0 | 11.1664 | 0.4623 | 1 | 0 | 97.891 |
| 24 | 0 | 9.7513 | 0.4804 | 1 | 0 | 98.0668 |
| 25 | 0 | 8.6015 | 0.5241 | 1 | 0 | 97.891 |
| 26 | 0 | 6.4279 | 0.5901 | 1 | 0 | 97.7153 |
| 27 | 0 | 4.7058 | 0.6511 | 1 | 0 | 97.7153 |
| 28 | 0 | 3.4833 | 0.6918 | 1 | 0 | 97.891 |
| 29 | 0 | 1.5269 | 0.8096 | 1 | 0 | 97.891 |
| 30 | 0 | 0.1274 | 0.8999 | 1 | 0 | 98.0668 |
| Average For All Ranks | 0.0050 | 42.2541 | 0.3553 | 0.9855 | 0.0155 | 96.373 |