Probability Density Function Estimation Based Over-Sampling for Imbalanced Two-Class Problems

Ming Gao[a], Xia Hong[a], Sheng Chen[b,c], Chris J. Harris[b]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK
ming.gao@pgr.reading.ac.uk x.hong@reading.ac.uk

[b] Electronics and Computer Science, Faculty of Physical and Applied Sciences,
University of Southampton, Southampton SO17 1BJ, UK
sqc@ecs.soton.ac.uk cjh@ecs.soton.ac.uk

[c] Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

IEEE World Congress on Computational Intelligence
Brisbane Australia, June 10-15, 2012

Outline

Outline

1. Introduction
   - Motivations and Solutions
2. PDF Estimation Based Over-sampling
   - Kernel Density Estimation
   - Over-sampling Procedure
   - Tunable RBF Classifier Construction
3. Experiments
   - Experimental Setup
   - Experimental Results
4. Conclusions
   - Concluding Remarks

Background

- Highly **imbalanced** two-class classification problems widely occur in life-threatening or safety critical applications
- Techniques for imbalanced problems can be divided into:
  1. Imbalanced learning algorithms:

     Internally modify existing algorithms, without artificially altering original imbalanced data
  2. Resampling methods:

     Externally operate on original imbalanced data set to re-balance data for conventional classifier
- Resampling methods can be categorised into:
  1. **Under-sampling**: which tends to be ideal when imbalance degree is not very severe
  2. **Over-sampling**: which becomes necessary if imbalance degree is high

Our Approach

- What would be ideal over-sampling:

  Draw **synthetic** data according to **same** probability distribution which produces **observed** positive-class data samples

- Our probability density function estimation based over-sampling
  1. Construct Parzen window or kernel **density** estimation from **observed** positive-class data samples
  2. Generate **synthetic** data samples according to **estimated** positive-class probability density function
  3. Apply our tunable radial basis function **classifier** based on leave-one-out misclassification rate to **rebalanced** data
- Ready-made PW estimator is low complexity in this application, as minority-class by nature is small size
- Particle swarm optimisation aided OFR for constructing RBF classifier based on LOO error rate is a state-of-the-art

Outline

Problem Statement

- **Imbalanced** two-class data set $D_N = \{\boldsymbol{x}_k, y_k\}_{k=1}^{N}$

$$D_N = D_{N_+} \bigcup D_{N_-} = \{\boldsymbol{x}_i, y_i = +1\}_{i=1}^{N_+} \bigcup \{\boldsymbol{x}_l, y_l = -1\}_{l=1}^{N_-}$$

  1. $y_k \in \{\pm 1\}$: **class label** for **feature vector** $\boldsymbol{x}_k \in \mathbb{R}^m$
  2. $\boldsymbol{x}_k$ are i.i.d. drawn from unknown underlying PDF
  3. $N = N_+ + N_-$, and $N_+ \ll N_-$

- Kernel **density estimator** $\hat{p}(\boldsymbol{x})$ for $p(\boldsymbol{x})$ is constructed based on **positive-class** samples $D_{N_+} = \{\boldsymbol{x}_i, y_i = +1\}_{i=1}^{N_+}$

$$\hat{p}(\boldsymbol{x}) = \frac{(\det \boldsymbol{S})^{-1/2}}{N_+} \sum_{i=1}^{N_+} \Phi_\sigma \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i) \right)$$

  1. **Kernel**:

$$\Phi_\sigma \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i) \right) = \frac{\sigma^{-m}}{(2\pi)^{m/2}} e^{-\frac{1}{2}\sigma^{-2}(\boldsymbol{x}-\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{x}-\boldsymbol{x}_i)}$$

  2. $\boldsymbol{S}$: **covariance** matrix of positive class
  3. σ: **smoothing** parameter

Kernel Parameter Estimate

- Unbiased estimate of positive-class **covariance** matrix is

$$\boldsymbol{S} = \frac{1}{N_+ - 1} \sum_{i=1}^{N_+} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\mathrm{T}}$$

  with mean vector of positive class $\bar{\boldsymbol{x}} = \frac{1}{N_+} \sum_{i=1}^{N_+} \boldsymbol{x}_i$

- **Smoothing** parameter by grid search to minimise **score** function

$$M(\sigma) = N_+^{-2} \sum_i \sum_j \Phi_\sigma^* \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{x}_i) \right) + 2N_+^{-1} \Phi_\sigma(\boldsymbol{0})$$

  with

$$\Phi_\sigma^* \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{x}_i) \right) \approx \Phi_\sigma^{(2)} \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{x}_i) \right) - 2\Phi_\sigma \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{x}_i) \right)$$

$$\Phi_\sigma^{(2)} \left(\boldsymbol{S}^{-1/2}(\boldsymbol{x}_j - \boldsymbol{x}_i) \right) = \frac{(\sqrt{2}\sigma)^{-m}}{(2\pi)^{m/2}} e^{-\frac{1}{2}(\sqrt{2}\sigma)^{-2}(\boldsymbol{x}_j - \boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{S}^{-1}(\boldsymbol{x}_j - \boldsymbol{x}_i)}$$

- $M(\sigma)$ is based on **mean integrated square error** measure

Outline

Draw Synthetic Samples

- Over-sampling positive class by drawing synthetic data samples according to PDF estimate $\hat{p}(\mathbf{x})$
- **Procedure** for generating a synthetic sample
  1) Based on discrete uniform distribution, randomly draw a data sample, $\boldsymbol{x}_o$, from positive-class data set D_{N_+}
  2) Generate a synthetic data sample, $\boldsymbol{x}_n$, using Gaussian distribution with mean $\boldsymbol{x}_o$ and covariance matrix $\sigma^2\boldsymbol{S}$

$$\boldsymbol{x}_n = \boldsymbol{x}_o + \sigma\boldsymbol{R}\cdot\boldsymbol{randn}()$$

  - $\boldsymbol{R}$: upper triangular matrix that is Cholesky decomposition of $\boldsymbol{S}$
  - $\boldsymbol{randn}()$: pseudorandom vector drawn from zero-mean normal distribution with covariance matrix $\boldsymbol{I}_m$
- Repeat **Procedure** $r \cdot N_+$ times, given oversampling rate r

Example (PDF estimate)

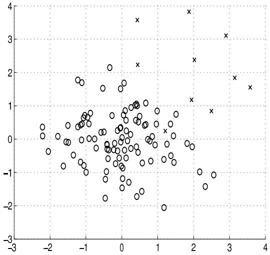
(a) Imbalanced data set: x denoting **positive-class** instance and $\circ$ **negative-class** instance

- $N_+ = 10$ positive-class samples: mean $[2\ 2]^{\mathrm{T}}$ and covariance $\boldsymbol{I}_2$
- $N_- = 100$ negative-class samples: mean $[0\ 0]^{\mathrm{T}}$ and covariance $\boldsymbol{I}_2$

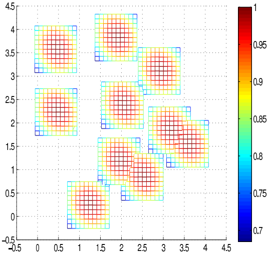
(b) Constructed PDF kernel of each positive-class instance

- Optimal **smoothing** parameter $\sigma = 1.25$ and **covariance** matrix $\boldsymbol{S} \approx \boldsymbol{I}_2$

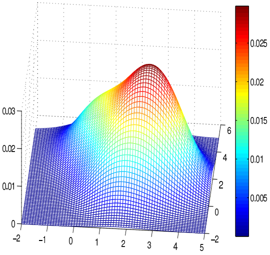
(c) Estimated **density** distribution of positive class



(a)



(b)



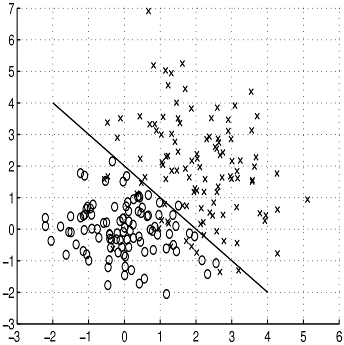
(c)

Example (over-sampling)

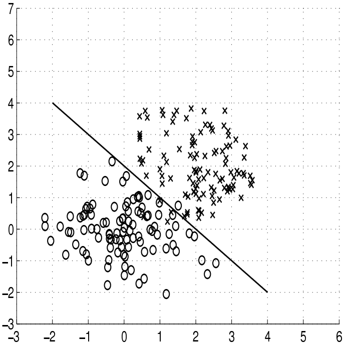
Over-sampling rate: $r = 100\%$, ideal **decision boundary**: $x + y - 2 = 0$

(a) Proposed PDF estimate based over-sampling: over-sampled positive-class data set **expands** along direction of **ideal** decision boundary

(b) Synthetic minority over-sampling technique (SMOTE): over-sampled data set is **confined** in **region** defined by original positive-class instances



(a)



(b)

Outline

1. Introduction
   - Motivations and Solutions
2. PDF Estimation Based Over-sampling
   - Kernel Density Estimation
   - Over-sampling Procedure
   - Tunable RBF Classifier Construction
3. Experiments
   - Experimental Setup
   - Experimental Results
4. Conclusions
   - Concluding Remarks

Tunable RBF Classifier

- Construct **radial basis function** classifier from oversampled training data, still denoted as $D_N = \{\boldsymbol{x}_k, y_k\}_{k=1}^N$

$$\hat{y}_k^{(M)} = \sum_{i=1}^{M} w_i g_i(\boldsymbol{x}_k) = \boldsymbol{g}_M^{\mathrm{T}}(k)\boldsymbol{w}_M \quad \text{and} \quad \tilde{y}_k^{(M)} = \mathrm{sgn}(\hat{y}_k^{(M)})$$

  1. M: number of **tunable** kernels, $\tilde{y}_k^{(M)}$: estimated class label
  2. Gaussian kernel adopted: $g_i(\boldsymbol{x}) = e^{-(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}$
  3. $\boldsymbol{\mu}_i \in \mathbb{R}^m$: ith RBF kernel **center vector**
  4. $\boldsymbol{\Sigma}_i = \mathrm{diag}\{\sigma_{i,1}^2, \sigma_{i,2}^2, \cdots, \sigma_{i,m}^2\}$: ith **covariance matrix**

- **Regression** model on training data D_N

$$\boldsymbol{y} = \boldsymbol{G}_M \boldsymbol{w}_M + \boldsymbol{\varepsilon}^{(M)}$$

  1. $\boldsymbol{\varepsilon}^{(M)} = [\varepsilon_1^{(M)} \cdots \varepsilon_N^{(M)}]^{\mathrm{T}}$ with **error** $\varepsilon_k^{(M)} = y_k - \hat{y}_k^{(M)}$
  2. $\boldsymbol{G}_M = [\boldsymbol{g}_1\ \boldsymbol{g}_2 \cdots \boldsymbol{g}_M]$: $N \times M$ regression matrix
  3. $\boldsymbol{w}_M = [w_1 \cdots w_M]^{\mathrm{T}}$: classifier's weight vector

Orthogonal Decomposition

- **Orthogonal decomposition** of regression matrix $\boldsymbol{G}_M = \boldsymbol{P}_M \boldsymbol{A}_M$

$$\boldsymbol{A}_M = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

  $\boldsymbol{P}_M = [\boldsymbol{p}_1 \cdots \boldsymbol{p}_M]$ with orthogonal **columns**: $\boldsymbol{p}_i^{\mathrm{T}} \boldsymbol{p}_j = 0$ for $i \neq j$

- **Equivalent** regression model

$$\boldsymbol{y} = \boldsymbol{G}_M \boldsymbol{w}_M + \varepsilon^{(M)} \Leftrightarrow \boldsymbol{y} = \boldsymbol{P}_M \boldsymbol{\theta}_M + \varepsilon^{(M)}$$

  $\boldsymbol{\theta}_M = [\theta_1 \cdots \theta_M]^{\mathrm{T}}$ satisfies $\boldsymbol{\theta}_M = \boldsymbol{A}_M \boldsymbol{w}_M$

- After nth stage of orthogonal forward selection, $\boldsymbol{G}_n = [\boldsymbol{g}_1 \cdots \boldsymbol{g}_n]$ is built with corresponding $\boldsymbol{P}_n = [\boldsymbol{p}_1 \cdots \boldsymbol{p}_n]$ and $\boldsymbol{A}_n$
  - kth row of $\boldsymbol{P}_n$ is denoted as $\boldsymbol{p}^{\mathrm{T}}(k) = [p_1(k) \cdots p_n(k)]$

OFS-LOO

- **Leave-one-out** misclassification rate

$$J_{\mathrm{LOO}}^{(n)} = \frac{1}{N}\sum_{k=1}^{N} \mathcal{I}_d\left(s_k^{(n,-k)}\right)$$

Indication function: $\mathcal{I}_d(s) = 1$ if $s \leq 0$ and $\mathcal{I}_d(s) = 0$ if $s > 0$

- LOO signed **decision variable** $s_k^{(n,-k)} = y_k \hat{y}_k^{(n,-k)} = \psi_k^{(n)} / \eta_k^{(n)}$ with recursions

$$\psi_k^{(n)} = \psi_k^{(n-1)} + y_k \theta_n p_n(k) - p_n^2(k) / \left(\boldsymbol{p}_n^{\mathrm{T}} \boldsymbol{p}_n + \lambda\right)$$
$$\eta_k^{(n)} = \eta_k^{(n-1)} - p_n^2(k) / \left(\boldsymbol{p}_n^{\mathrm{T}} \boldsymbol{p}_n + \lambda\right)$$

- Determine nth RBF **centre vector** and **covariance matrix**

$$\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{\mathrm{opt}} = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} J_{\mathrm{LOO}}^{(n)}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

  1. **Particle swarm optimisation** solves this optimisation
  2. OFS procedure **automatically** terminates at size M when $J_{\mathrm{LOO}}^{(M+1)} \geq J_{\mathrm{LOO}}^{(M)}$

Outline

Data Sets

Data set	m	N_+	N_-	ID	n-fold CV	σ
Pima Diabetes	7	268	500	1.87	10	0.47 ± 0.03
Haberman's survival	2	81	225	2.78	3	0.52 ± 0.03
Glass(6)	8	29	185	6.38	3	0.42 ± 0.06
ADI	8	90	700	7.78	8	0.56 ± 0.07
Satimage(4)	35	626	5809	9.28	10	0.90 ± 0.00
Yeast(5)	7	44	1440	32.73	3	0.10 ± 0.00

1. Glass, Satimage and Yeast turned into two-class problems, using class with class label in brackets as positive class, and other classes altogether as negative class
2. Imbalanced degree: ID $= N_- / N_+$
3. Each dimension of feature vector $\boldsymbol{x}_k = \left[x_{k,1} \cdots x_{k,m}\right]^{\mathrm{T}}$ is normalised using

$$\bar{x}_{k,i} = \frac{x_{k,i} - x_{\min,i}}{x_{\max,i} - x_{\min,i}}, \ 1 \leq k \leq N, 1 \leq i \leq m$$

with $x_{\min,i} = \min_{1 \leq k \leq N} x_{k,i}$ and $x_{\max,i} = \max_{1 \leq k \leq N} x_{k,i}$

4. Mean and standard deviation of smoothing parameter σ, determined by PW estimator for positive class, averaged over n-fold CV, are listed in last column

Benchmark Algorithms

1. **PFDOS+PSO-OFS**: proposed PDF estimation based oversampling with PSO-OFS based tunable RBF classifier
2. **SMOTE+PSO-OFS**: SMOTE based oversampling with same PSO-OFS based tunable RBF classifier

   M. Gao, X. Hong, S. Chen, and C. J. Harris, “A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems,” *Neurocomputing*, 74(17), 3456–3466, 2011
3. **LOO-AUC+OFS**: OFS based on LOO-AUC criterion for RBF classifier with weighted least square cost function

   X. Hong, S. Chen, and C. J. Harris, “A kernel-based two-class classifier for imbalanced data sets,” *IEEE Trans. Neural Networks*, 18(1), 28–41, 2007
4. **κ-means+WLSE**: κ-means clustering for RBF centres and same weighted least square cost function for RBF weights

- **Algorithms 1 and 2: oversampling rate r; Algorithms 3 and 4: weighting ρ**

Performance Metrics

1. **AUC**: area under receiver operating characteristics (ROC) curve
2. **G-mean**:

$$\text{G-mean} = \sqrt{\text{TP\%} \times (1 - \text{FP\%})}$$

   - True positive rate

$$\text{TP\%} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

   - False positive rate

$$\text{FP\%} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

   - Precision

$$\text{Pr} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3. **F-measure**:

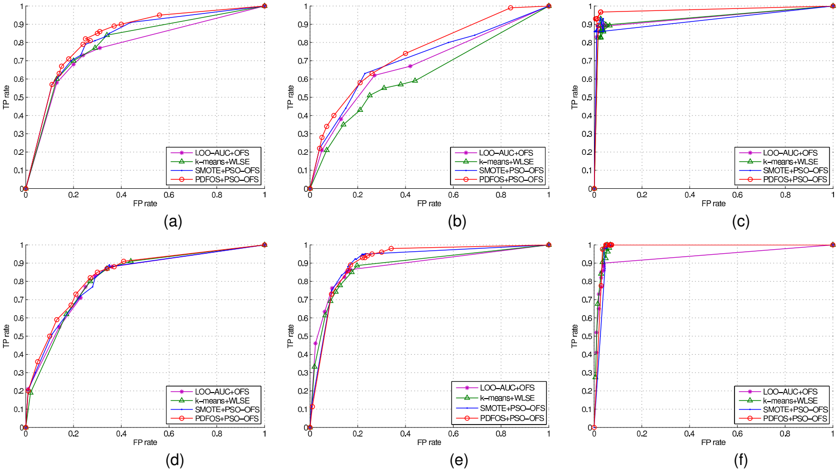
$$\text{F-measure} = \frac{2 \times \text{Pr} \times \text{TP\%}}{\text{Pr} + \text{TP\%}}$$

Outline

ROC Curves

Mean curves of (FP rate, TP rate) pairs averaged over *n*-fold CV, obtained for different over-sampling rates *r* of SMOTE+PSO-OFS and PDFOS+PSO-OFS or different weights ρ of LOO-AUC+OFS and κ-means+WLSE

(a) Pima Indians diabetes, (b) Haberman's survival, (c) Glass, (d) ADI, (e) Satimage, and (f) Yeast



AUC Metric

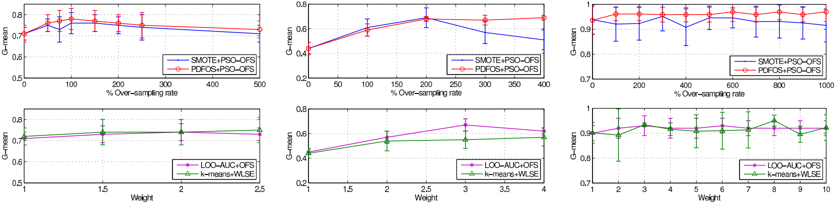
- Comparison of mean and standard deviation of AUCs

Data set	LOO-AUC+OFS	κ-means+WLSE	SMOTE+PSO-OFS	PDFOS+PSO-OFS
Pima Diabetes	0.77 ± 0.06	0.80 ± 0.06	0.82 ± 0.06	$\mathbf{0.84 \pm 0.06}$
Haberman's survival	0.68 ± 0.06	0.62 ± 0.06	0.71 ± 0.06	$\mathbf{0.74 \pm 0.06}$
Glass(6)	0.94 ± 0.05	0.93 ± 0.06	0.92 ± 0.06	$\mathbf{0.97 \pm 0.04}$
ADI	0.82 ± 0.03	0.82 ± 0.03	0.82 ± 0.03	$\mathbf{0.83 \pm 0.03}$
Satimage(4)	0.88 ± 0.03	0.88 ± 0.03	$\mathbf{0.91 \pm 0.03}$	$\mathbf{0.91 \pm 0.03}$
Yeast(5)	0.93 ± 0.04	$\mathbf{0.98 \pm 0.02}$	0.97 ± 0.03	$\mathbf{0.98 \pm 0.02}$

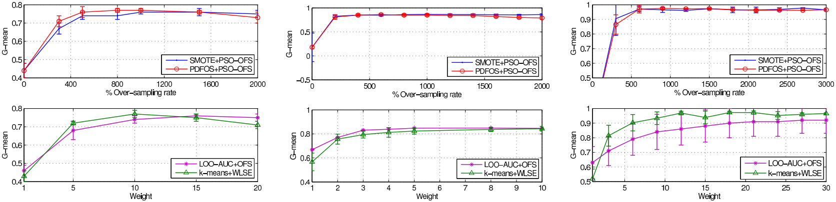
G-Means

G-mean metrics with respect to over-sampling rate r of SMOTE+PSO-OFS and PDFOS+PSO-OFS or weight ρ of LOO-AUC+OFS and κ-means+WLSE, averaged over n-fold CV

(a) Pima Indians diabetes, (b) Haberman's survival, (c) Glass, (d) ADI, (e) Satimage, and (f) Yeast



(a) (b) (c)



(d) (e) (f)

Best G-Means

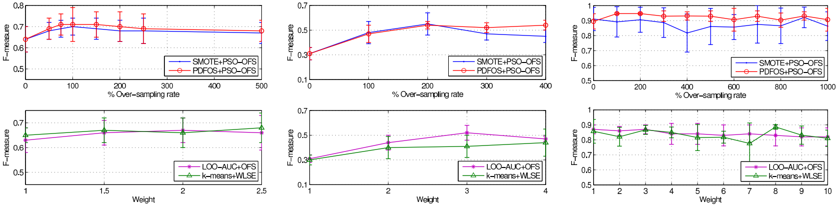
- Comparison of mean and standard deviation of best G-means

Data set	LOO-AUC+OFS (ρ)	k-means+WLSE (ρ)	SMOTE+PSO-OFS (r)	PDFOS+PSO-OFS (r)
Pima Diabetes	0.74 ± 0.04 (2.0)	0.75 ± 0.06 (2.5)	0.76 ± 0.05 (100%)	**0.78 ± 0.05** (100%)
Haberman's survival	0.67 ± 0.05 (3.0)	0.57 ± 0.07 (4.0)	**0.69 ± 0.08** (200%)	**0.69 ± 0.02** (400%)
Glass(6)	0.93 ± 0.03 (3.0, 6.0)	0.95 ± 0.02 (8.0)	0.95 ± 0.06 (600%)	**0.97 ± 0.04** (600%)
ADI	0.76 ± 0.01 (15.0)	**0.77 ± 0.02** (10.0)	0.76 ± 0.02 (1000%, 1500%)	**0.77 ± 0.01** (800%, 1000%)
Satimage(4)	0.85 ± 0.03 (8.0)	0.84 ± 0.02 (10.0)	**0.86 ± 0.01** (1000%)	**0.86 ± 0.02** (600%)
Yeast(5)	0.92 ± 0.09 (27.0, 30.0)	0.97 ± 0.01 (18.0	**0.98 ± 0.00** (2700%)	**0.98 ± 0.01** (900%)

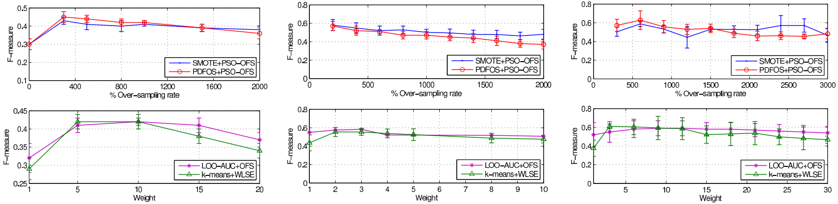
F-Measures

F-Measure metrics with respect to over-sampling rate r of SMOTE+PSO-OFS and PDFOS+PSO-OFS or weight ρ of LOO-AUC+OFS and κ-means+WLSE, averaged over n-fold CV

(a) Pima Indians diabetes, (b) Haberman's survival, (c) Glass, (d) ADI, (e) Satimage, and (f) Yeast



(a) (b) (c)



(d) (e) (f)

Best F-Measures

- Comparison of mean and standard deviation of best F-measures

Data set	LOO-AUC+OFS (ρ)	k-means+WLSE (ρ)	SMOTE+PSO-OFS (r)	PDFOS+PSO-OFS (r)
Pima Diabetes	0.67 ± 0.05 (2.0)	0.68 ± 0.06 (2.5)	0.70 ± 0.04 (100%)	**0.71 ± 0.06** (100%)
Haberman's survival	0.52 ± 0.06 (3.0)	0.44 ± 0.11 (4.0)	**0.55 ± 0.09** (200%)	0.54 ± 0.03 (200%, 400%)
Glass(6)	0.87 ± 0.03 (3.0)	0.89 ± 0.02 (8.0)	0.92 ± 0.07 (900%)	**0.95 ± 0.01** (100%, 200%)
ADI	0.42 ± 0.01 (10.0)	0.42 ± 0.02 (5.0, 10.0)	0.43 ± 0.02 (300%)	**0.45 ± 0.03** (300%)
Satimage(4)	**0.58 ± 0.03** (3.0)	0.55 ± 0.05 (2.0)	**0.58 ± 0.06** (200%)	0.57 ± 0.05 (200%)
Yeast(5)	0.59 ± 0.08 (9.0, 12.0)	0.61 ± 0.03 (3.0)	0.59 ± 0.03 (600%)	**0.63 ± 0.10** (600%)

Outline

Summary

- Our over-sampling method re-balances skewed class distribution according to original statistical information in observed data
  1. Parzen window density estimator using observed positive class data samples
  2. Draw synthetic samples according to estimated PDF to re-balance data
- Construct tunable RBF classifier based on rebalanced data set using efficient PSO aided OFS procedure
  - State-of-the-art for balanced classification problems
- Experimental results demonstrate that our approach offers a very competitive technique
  - Compared favourably with many existing state-of-the-art methods for dealing with highly imbalanced problems