Beyond the Boundaries of SMOTE

A framework for Manifold-Based Synthetic Oversampling

Colin Bellinger

Collaborators: Nathalie Japkowicz
School of Electrical Engineering and Computer Science,
University of Ottawa
Ottawa, Canada

Christopher Drummond
National Research Council of Canada
Ottawa, Canada

September 16, 2016

Overview

- Classification: induce $f(x) \rightarrow \{\omega_1, \omega_2\}$ such that it predicts the class ω_i of instance x
- Performance erodes quickly when data is imbalanced [Akbani et al., 2004, Wu and Chang, 2005]
  - oil spill classification, gene function annotation, medical and text classification, radiation classification
  - Costly mistakes on the minority class
- For this reason, class imbalance is an essential problem in machine learning
  - AAAI'00, ICML'03, ACM SIGKDD Explorations'04, PAKDD'09, ICMLA'12

Managing Class Imbalance

- Obvious solution to imbalance is more examples
  - This is impractical in many domains due to prior factors
- Synthetic oversampling provides an alternative which generates minority class instances
  - Balances the training set
  - Reduces the prediction bias by expanding the minority space
  - Avoids overfitting caused by random oversampling
  - Avoids information loss caused by random undersampling

Synthetic Oversampling: A brief history

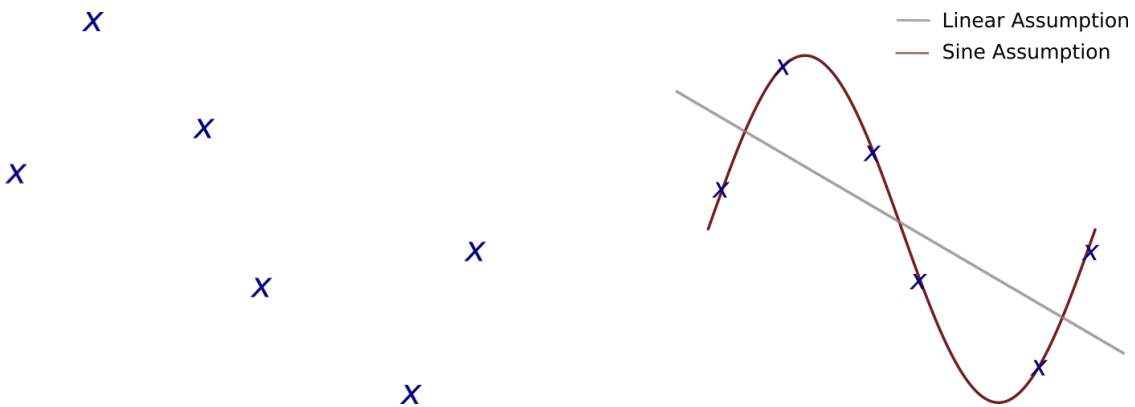
- SMOTE introduced data generation for class imbalance [Chawla et al., 2002]
- Designed to solve specific problems in existing methods
  - Overfitting caused by random oversampling
  - Information loss caused by random undersampling
- The initial success motivated many other studies involving it
- Subsequent studies:
  - Identified weaknesses
  - Offered variations on the standard algorithm

Claim

*To maximize the likelihood of **generating instances** that will **improve** the predictive **performance**, we should **design** synthetic oversampling methods **with generative biases** that **match** the properties of the **target data**.*

Generative Bias

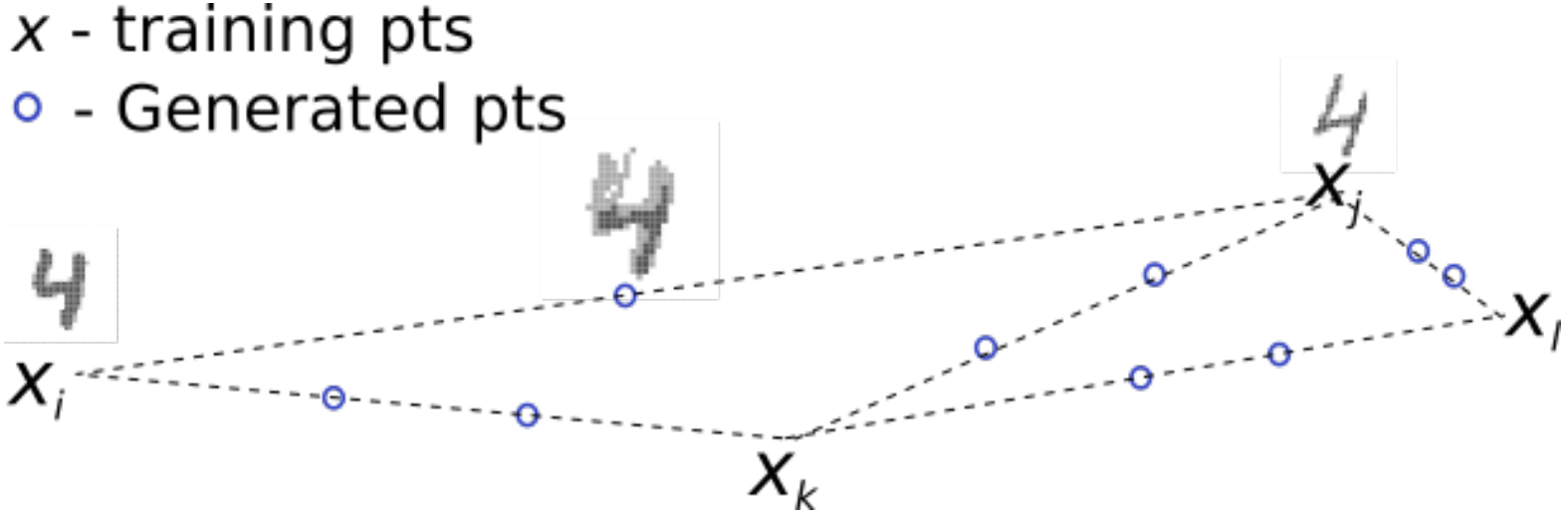
- Generative bias $\rightarrow$ how probability mass is spread through the feature space.
  - This dictates how the minority space is expanded
- When the minority class is small, an appropriate generative bias is essential
  - Maximizes the likelihood of benefiting classifier induction



SMOTE

1: **procedure** Synthetic Minority Oversampling TEchnique
   Input: Minority training instances $\mathbf{X}$
   Input: Number of nearest neighbours k
   Output: Synthesized minority instances $\mathbf{Y}$
2: **Do**
3: Sample $\mathbf{x}_i$ from $\mathbf{X}$;
4: Sample $\mathbf{x}_j \in kNN(\mathbf{x}_i)$ from kNN set of $\mathbf{x}_i$;
5: Synthesize instances as $\mathbf{x}_{new} = \mathbf{x}_i + (\mathbf{x}_j - \mathbf{x}_i) \times \delta$;
6: Add $\mathbf{x}_{new}$ to $\mathbf{Y}$;
7: **While** *generate more samples*
8: **Return:** Synthetic instances $\mathbf{Y}$
9: **end procedure**

SMOTE



- Key aspects of its generative bias
  - Use of k-nearest neighbours
  - Generation within the convex-hull of minority training instances

SMOTE and its Generative Bias

- Many weaknesses have been articulated
  - These haven't examined its generative bias in relation to data properties
- We have studied SMOTE in terms of its generative bias in relation to manifold data
  - Empirically, we have found it to be weak on such domains
  - Two properties give the theoretical basis for this:
    - Nearest neighbour querying performs poorly in high dimensional domains
    - Straight-line distance measures are inappropriate for sparse manifolds

Synthetic Oversampling and the Manifold

what we want and what we get

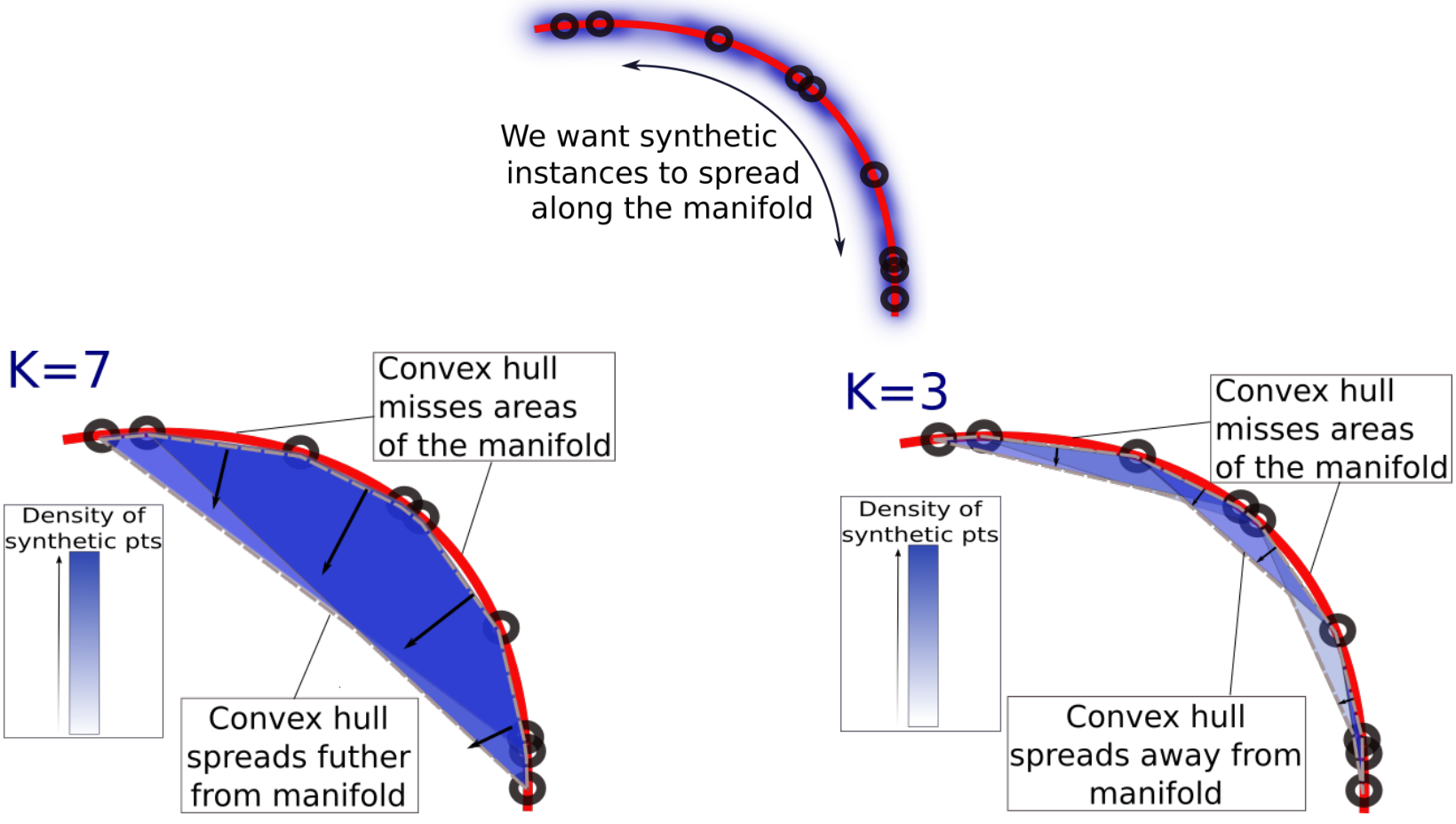


Figure: Manifold-based versus SMOTE for synthetic oversampling a manifold.

The Manifold Property

- Manifold property $\rightarrow$ probability density resides in a lower-dimensional embedded space
  - Text, images, speech, spectral data, *etc*.
- $g(\mathcal{F}) \rightarrow \mathcal{M}$: A mapping from the feature space $\mathcal{F}$ to the latent manifold space $\mathcal{M}$
  - Linear and non-linear representation are possible
- They improve performance on many machine learning problems [Zhang and Chen, 2005, Tuzel et al., 2007]
- Has not been applied in problems of synthetic oversampling
  - Increases the likelihood of generating novel instances that resemble to target domain

Framework for Manifold-Based Synthetic Oversampling

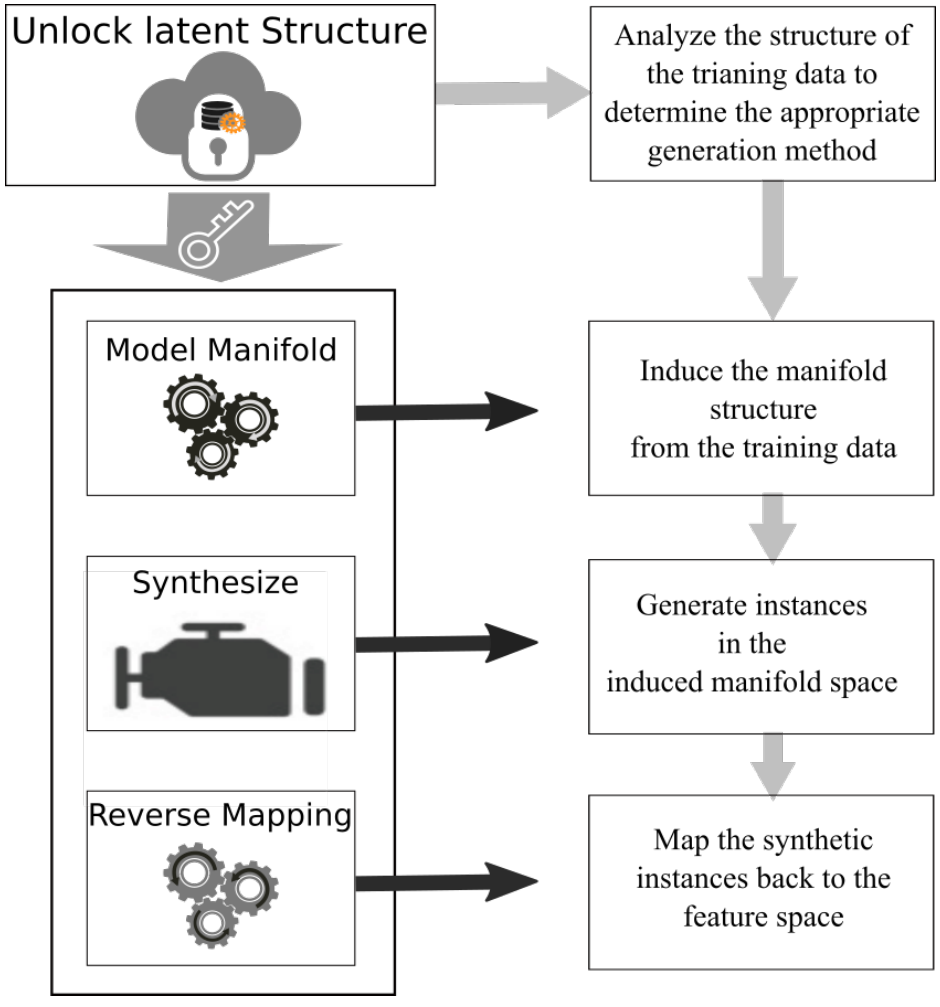


Figure: General framework for synthetic oversampling.

Framework for Manifold-Based Synthetic Oversampling - PCA Sampling

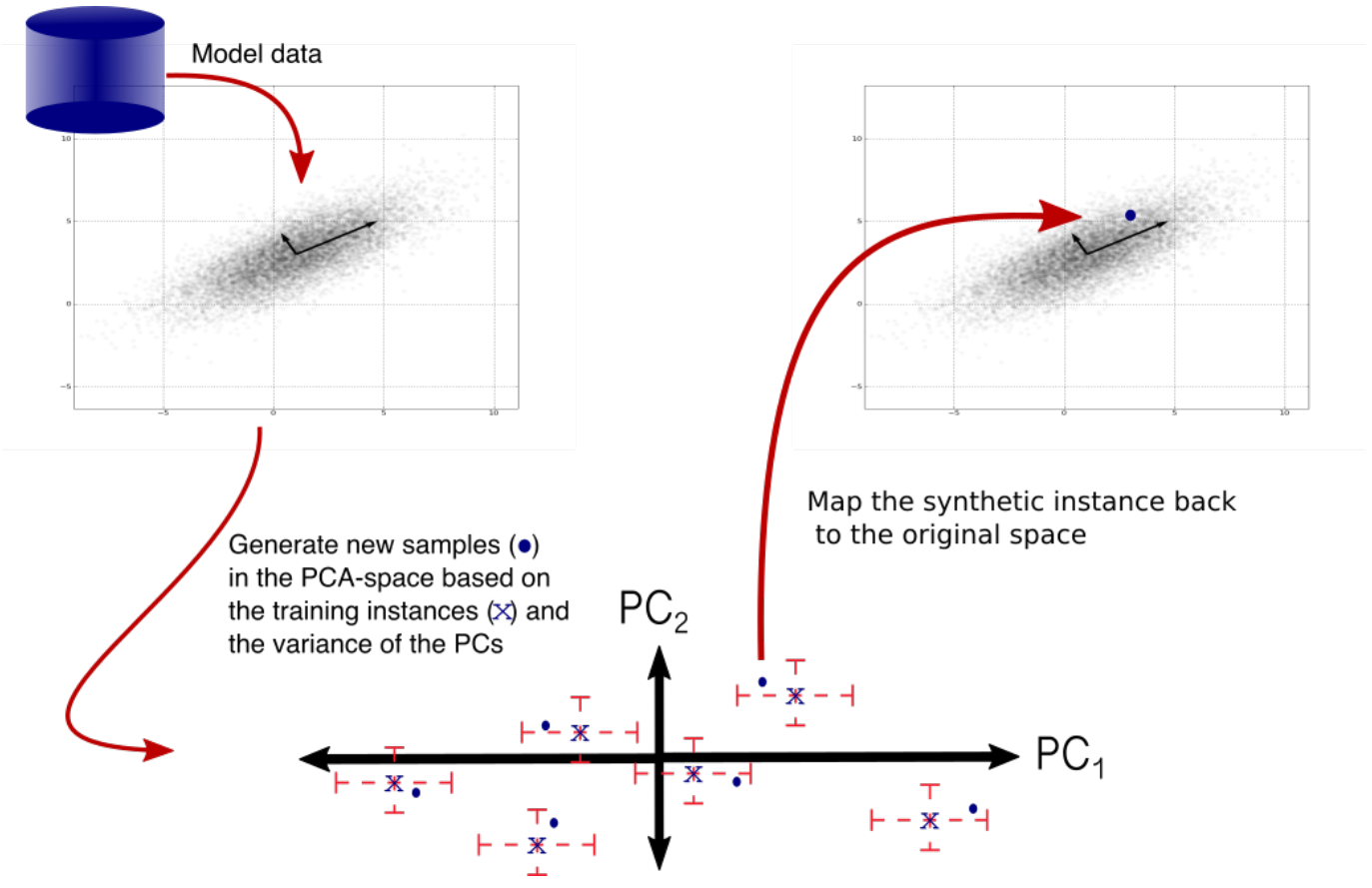


Figure: The process of synthesizing instances via PCA

Denoising Autoencoder

- Fully connected ANN
- Standard autoencoder learns to:
  - Compress $\mathbf{x}$ into hidden space
    $\mathbf{z} = f_\theta(x) = s(\mathbf{W}\mathbf{x} + \mathbf{b})$
  - Decompress back to feature space
    $\mathbf{y} = g'_\theta(\mathbf{z}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$
  - Trained to minimize the squared error
    $\sum(\mathbf{X} - g(f(\mathbf{X})))^2$
  - Denoising is a form of regularization

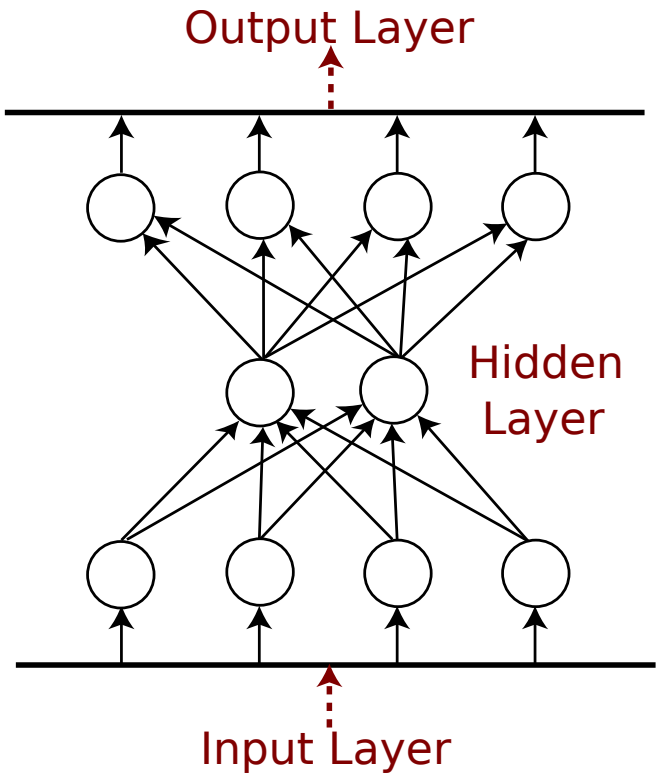


Figure: Structure of an autoencoder.

Framework for Manifold-Based Synthetic Oversampling - Autoencoder Sampling

1: **procedure** DAE-MOS
   Input: Minority training instances $\mathbf{X}$
   Output: Synthesized minority instances $\mathbf{Y}$
2: Train DAE $g_{(\mathrm{W},\mathrm{b})}(f_{(\mathrm{W}',\mathrm{b}')}(\mathbf{X}))$;
3: Generate sample initiation points P in feature space;
4: Generate instance Y as $g_{(\mathrm{W},\mathrm{b})}(f_{(\mathrm{W}',\mathrm{b}')}(\mathbf{P}))$;
5: **Return:** Synthetic instances $\mathbf{Y}$
6: **end procedure**

Framework for Manifold-Based Synthetic Oversampling - Autoencoder Sampling

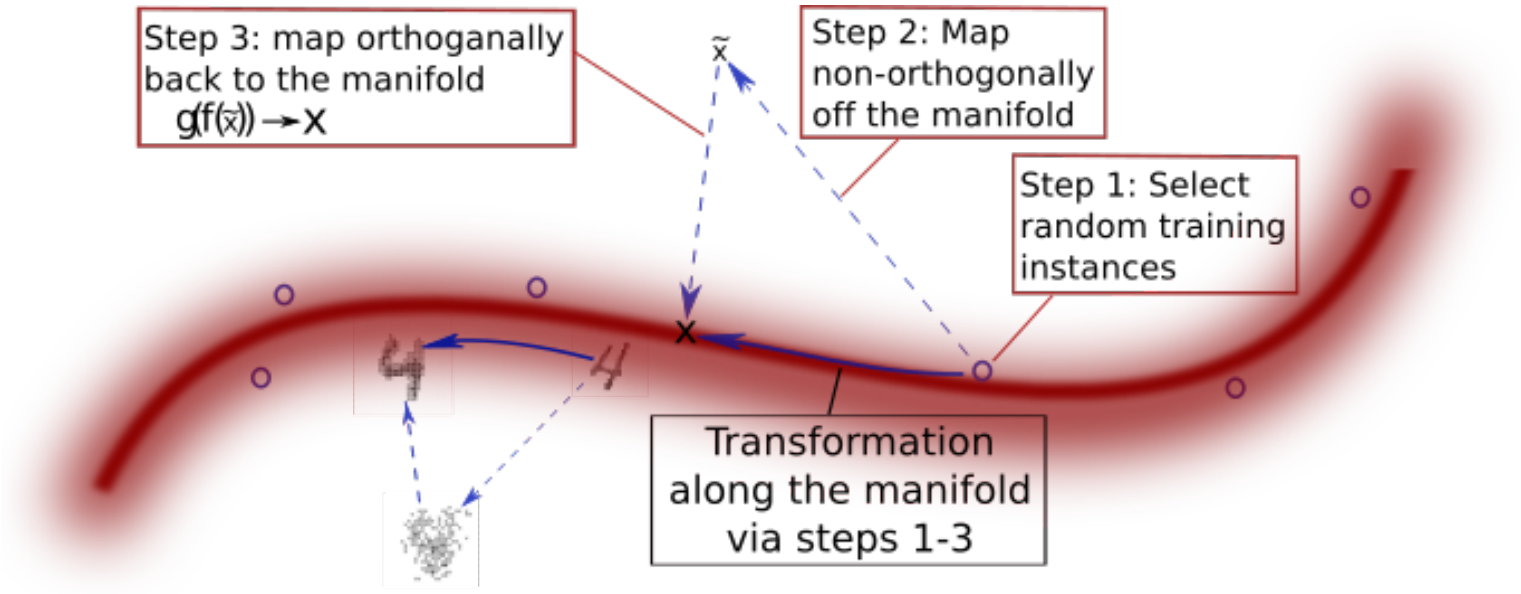


Figure: Three steps of synthesization for the autoencoder formalization.

Synthetically Oversampling Handwritten 4s

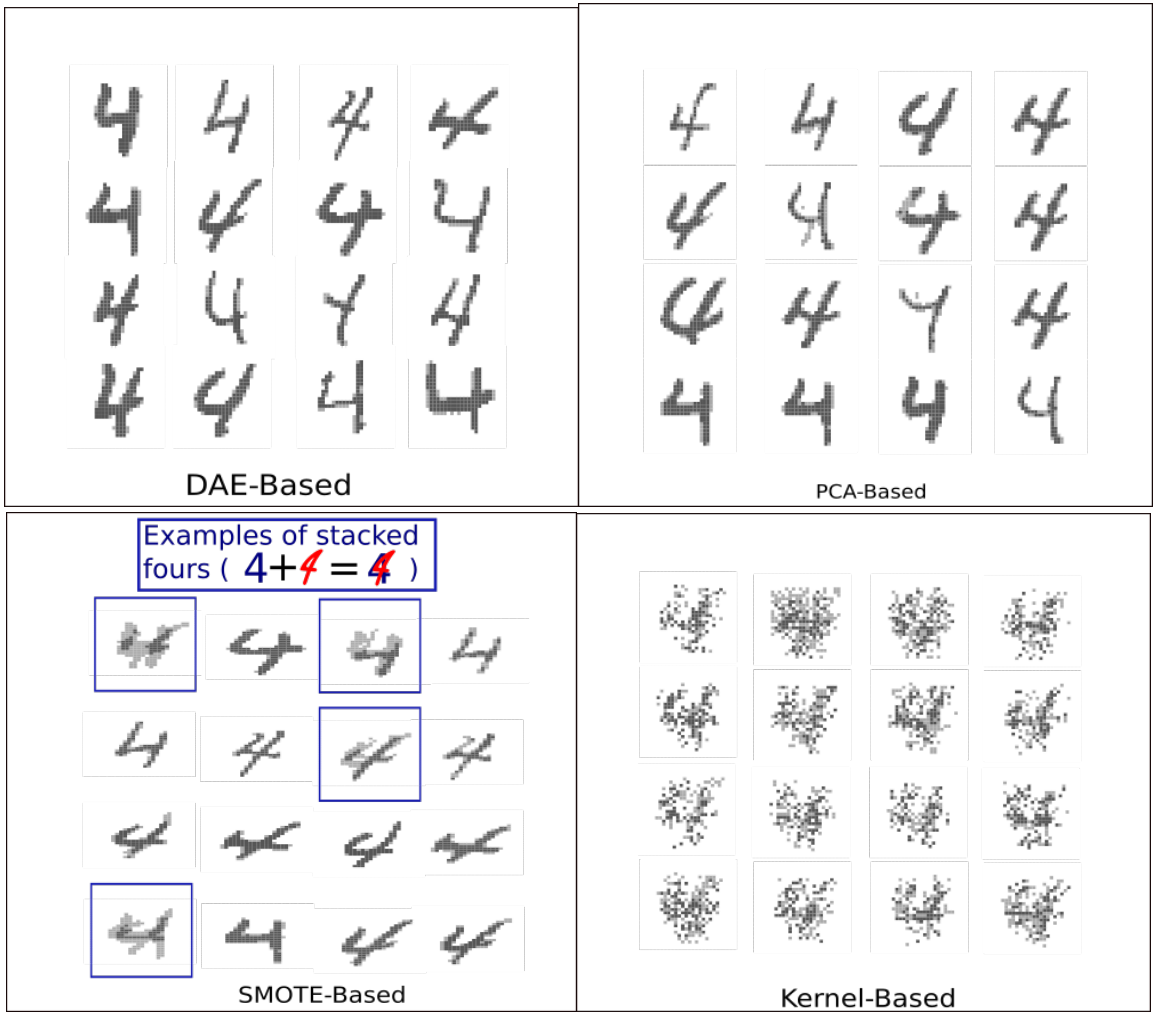
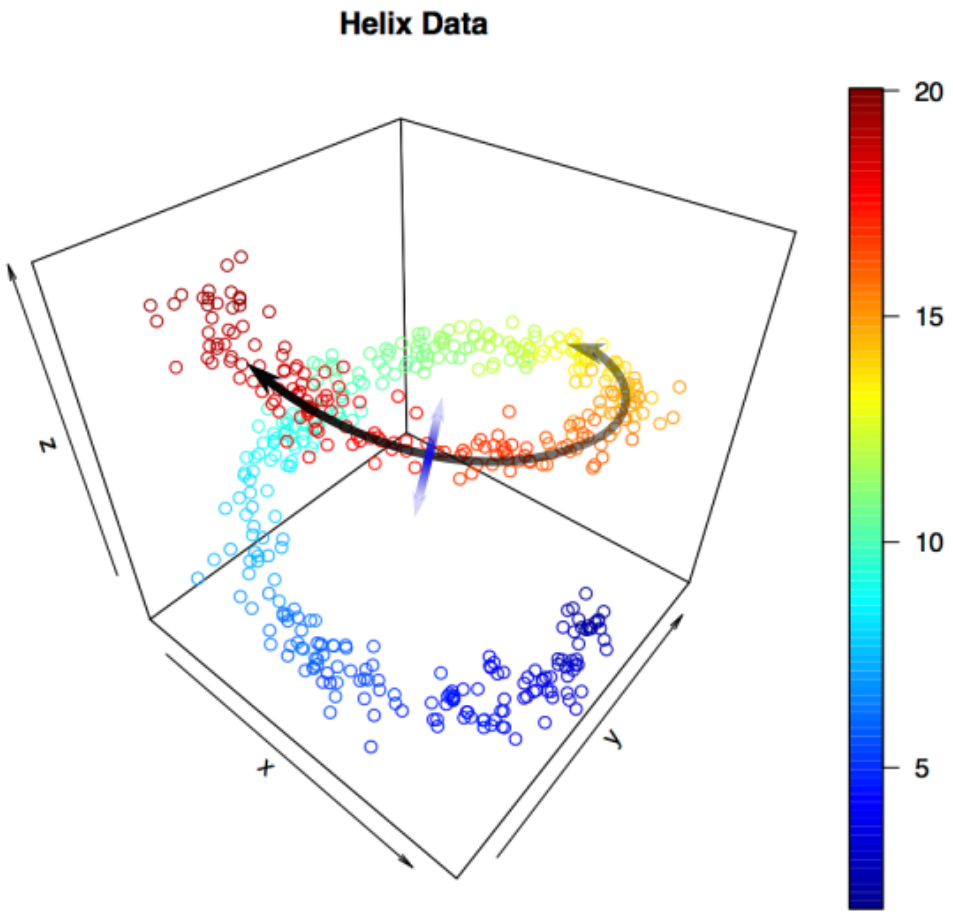
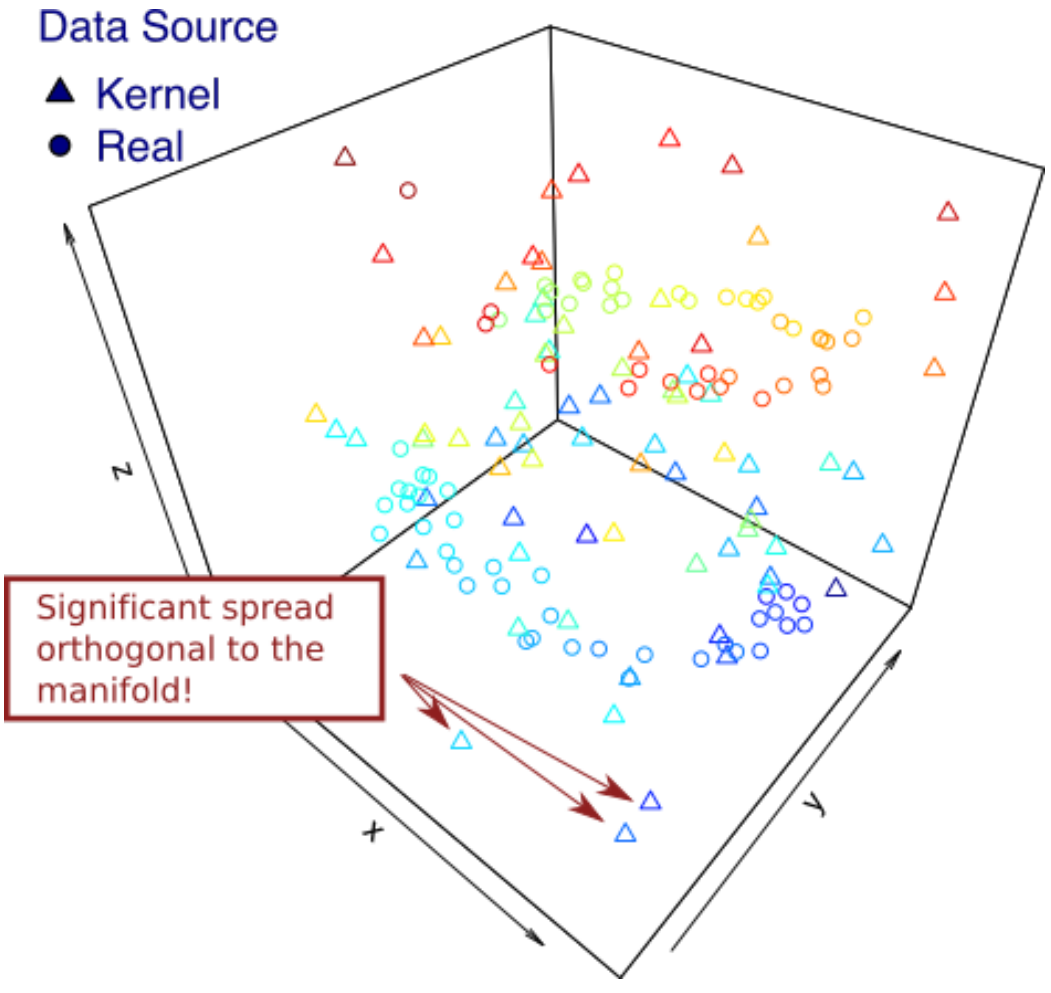


Figure: From left to right, handwritten fours synthesized by DAE, PCA, SMOTE and kernel-based methods.

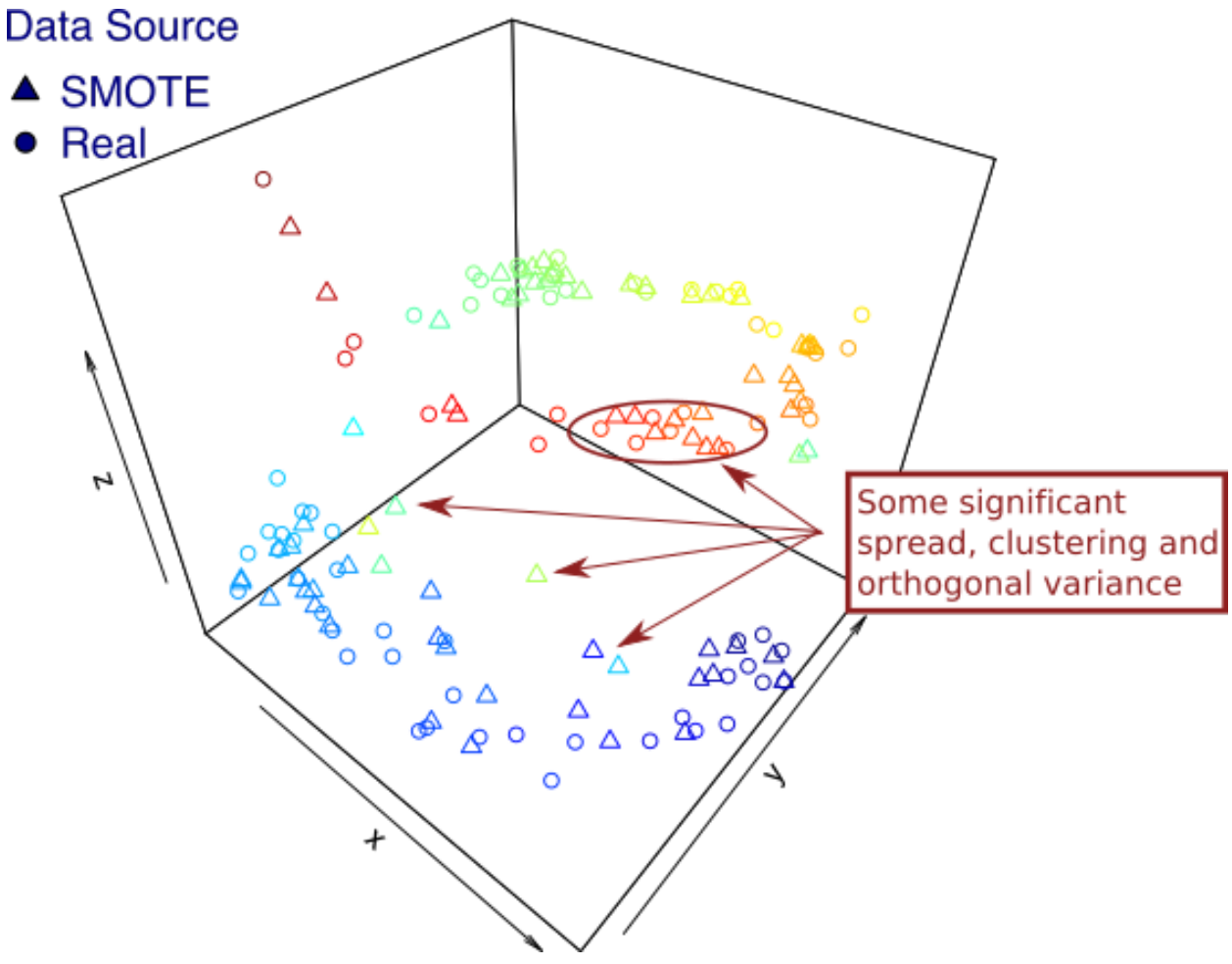
Noisy Helix



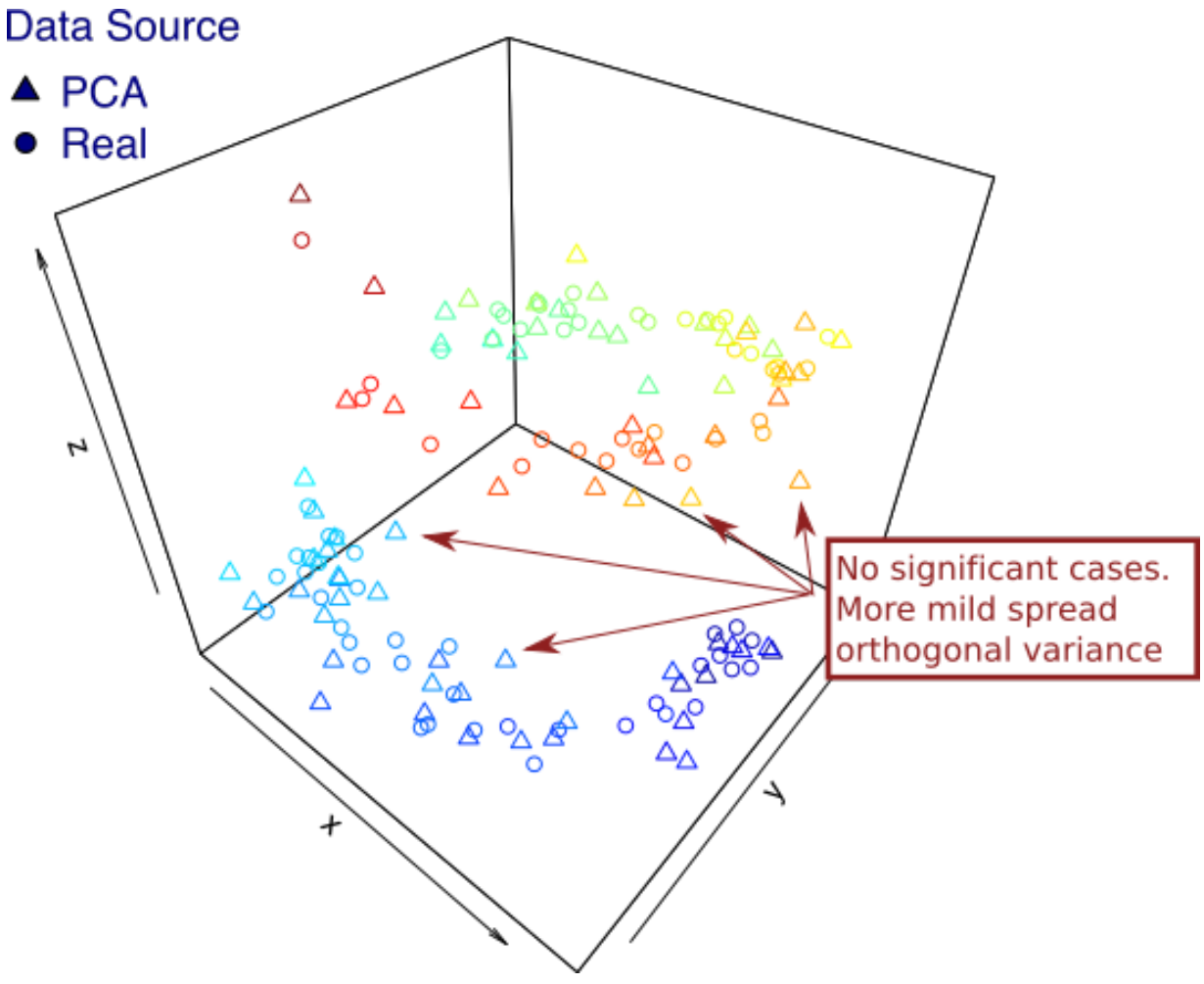
Noisy Helix


Data Source
Kernel
Real
z
x
y
Significant spread orthogonal to the manifold!

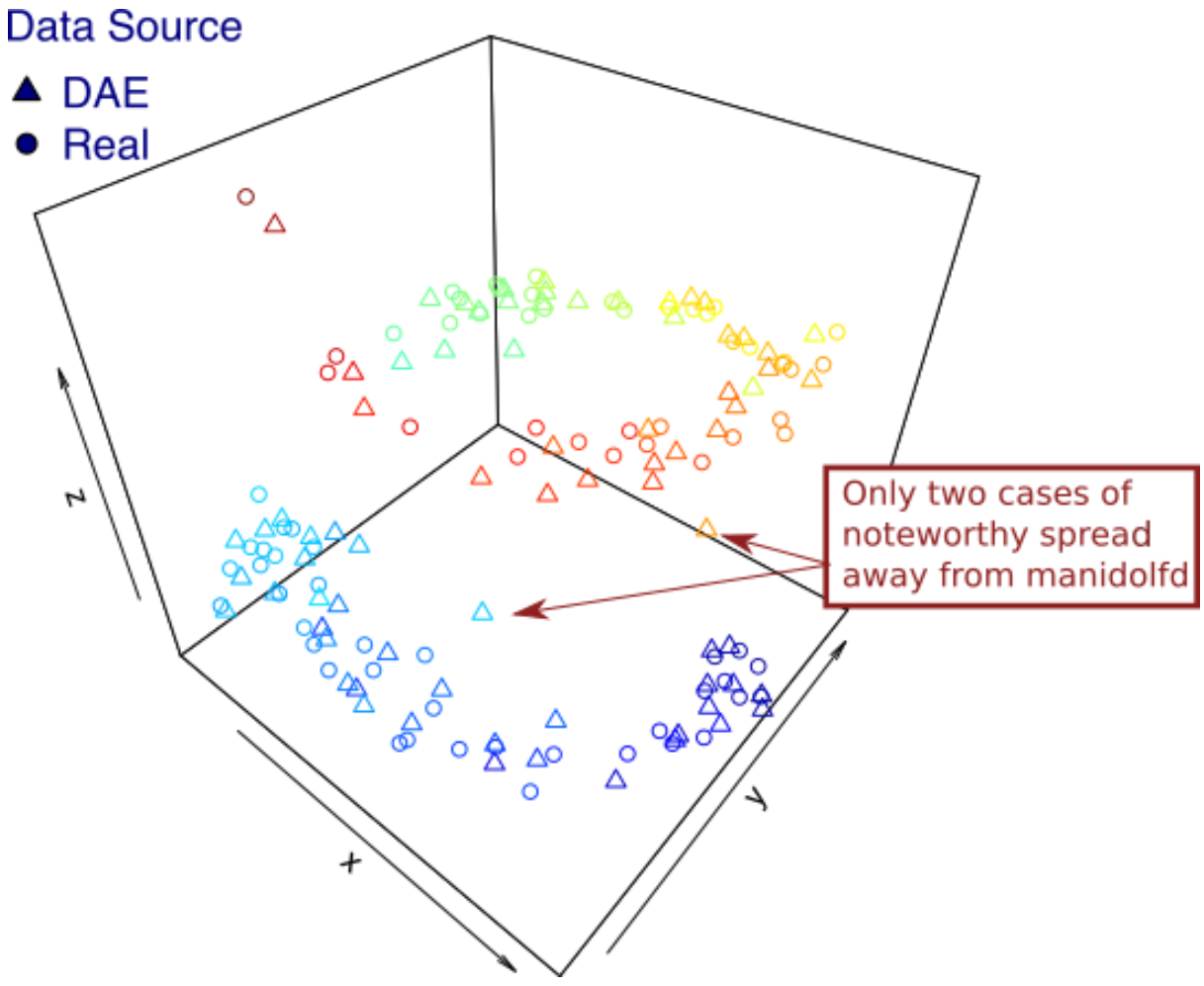
Noisy Helix


Data Source
SMOTE
Real
z
x
y
Some significant spread, clustering and orthogonal variance

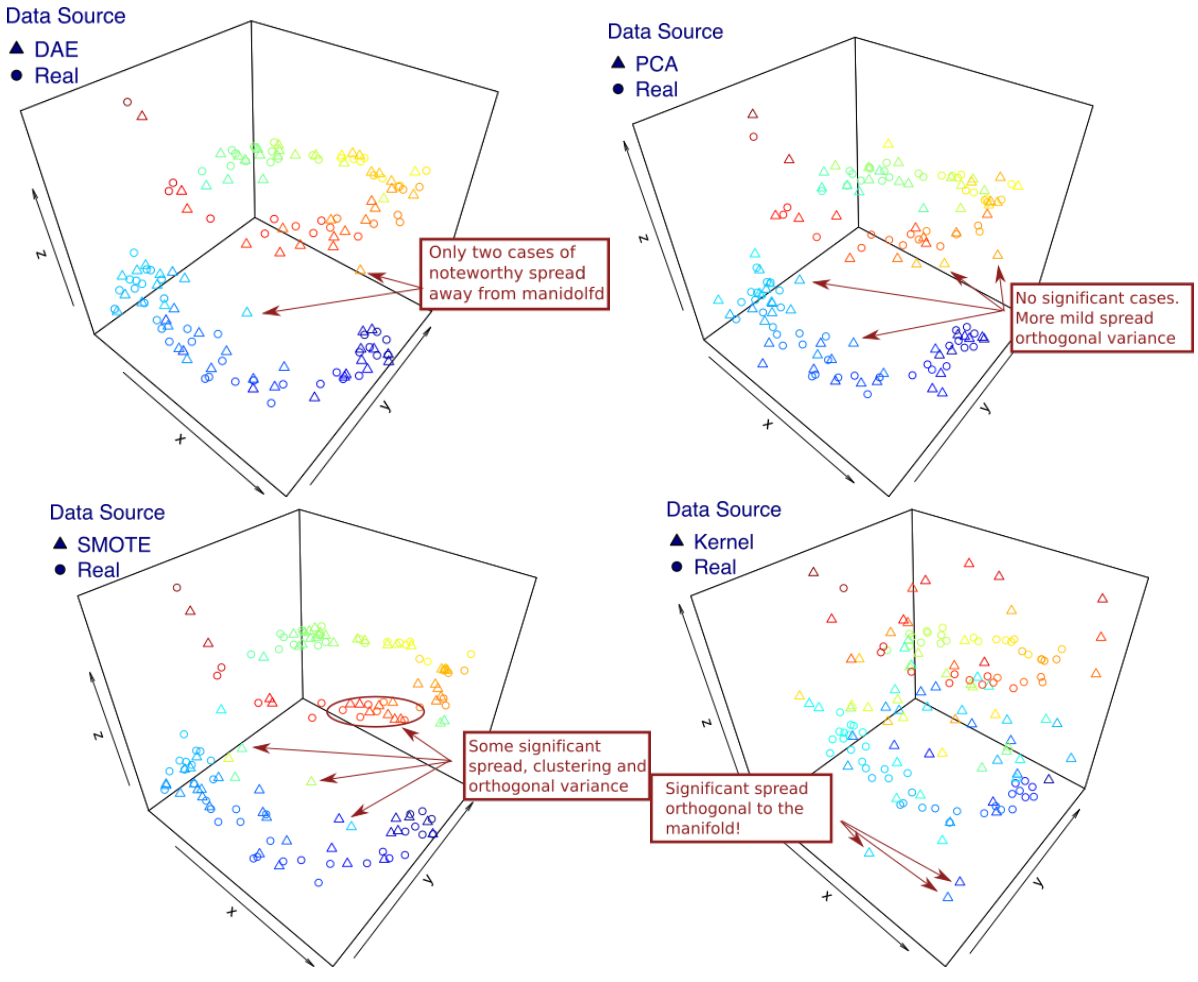
Noisy Helix


Data Source
PCA
Real
z
x
y
No significant cases.
More mild spread
orthogonal variance

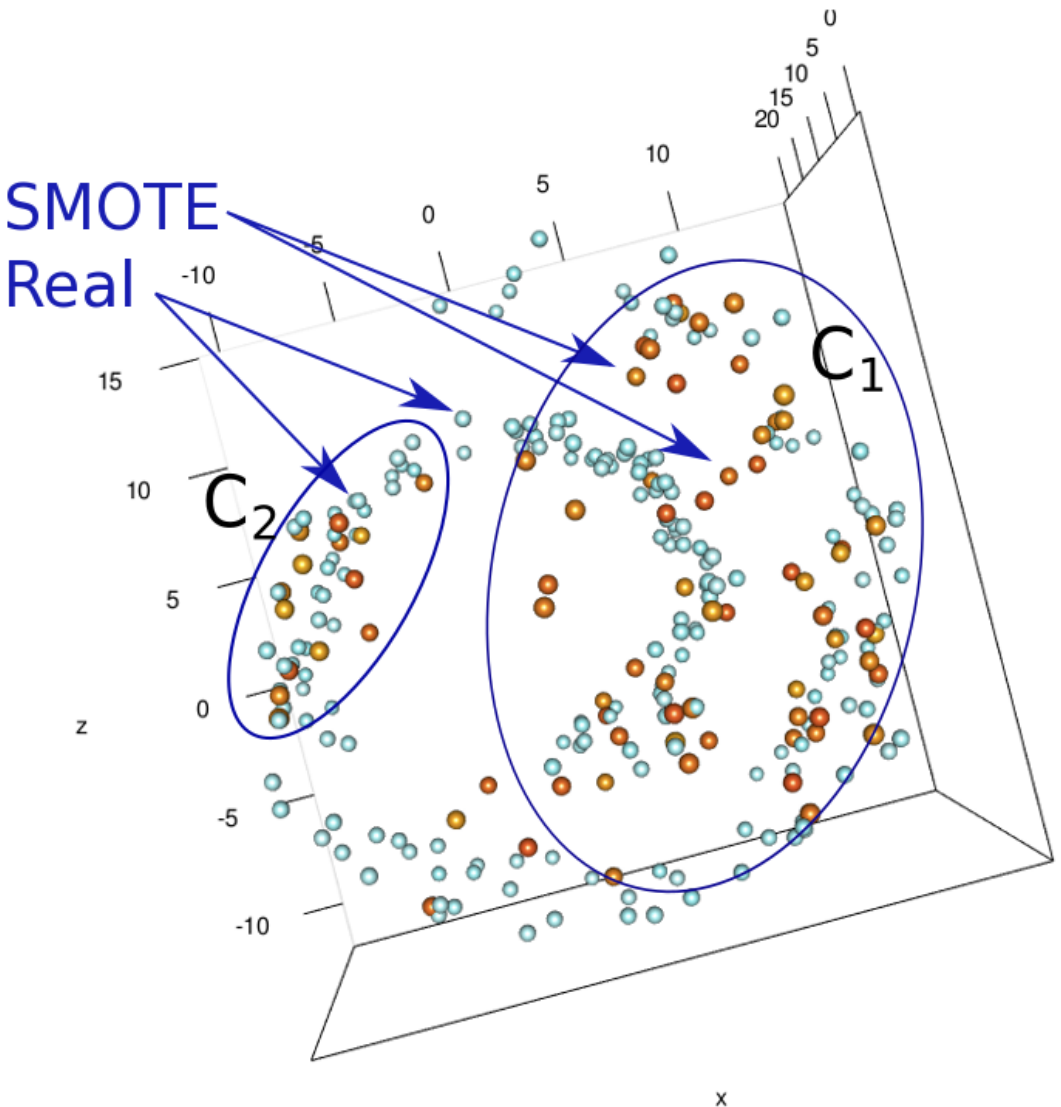
Noisy Helix


Data Source
DAE
Real
z
x
y
Only two cases of noteworthy spread away from manidolfd

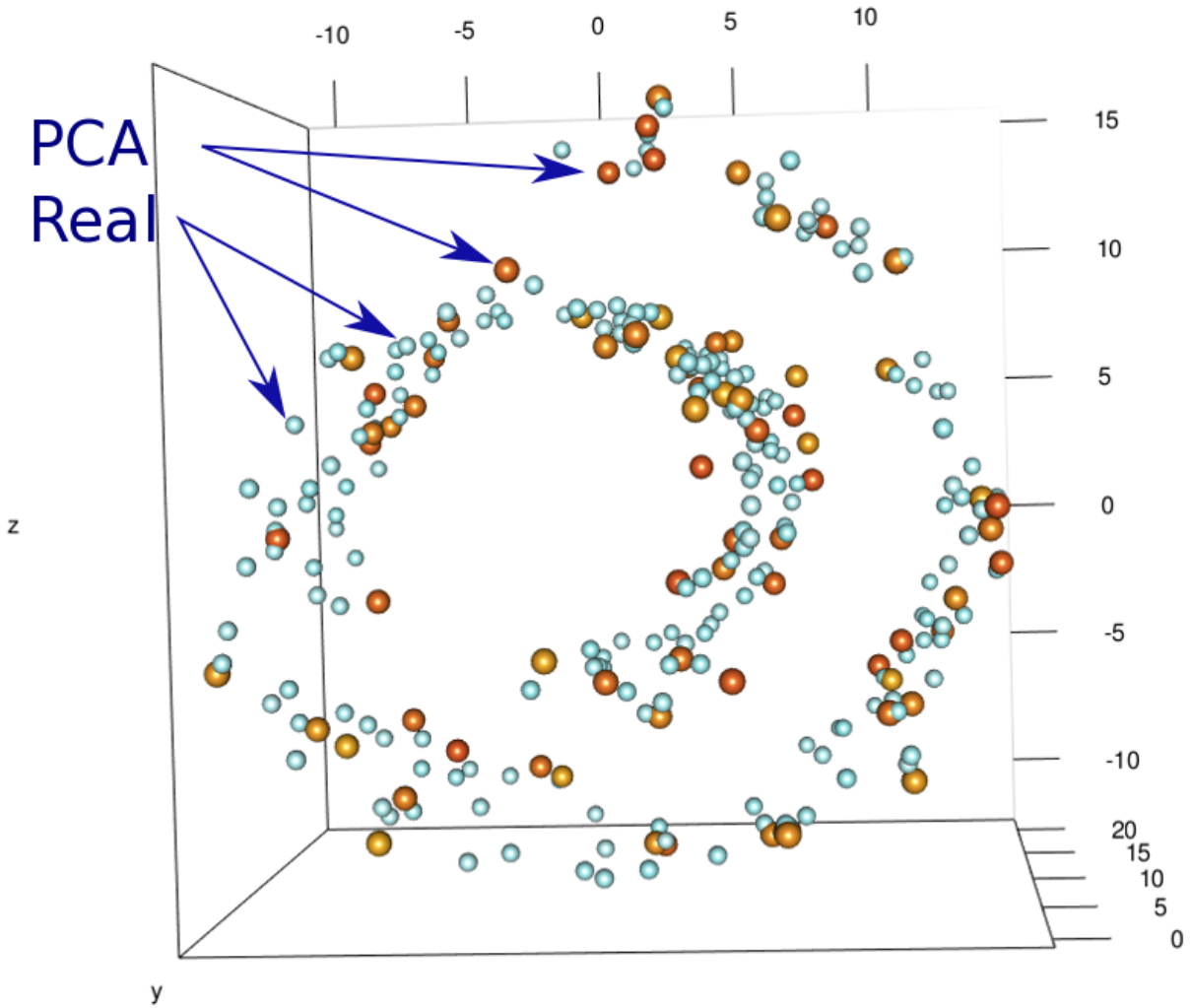
Noisy Helix


Data Source
▲ DAE
● Real
Only two cases of noteworthy spread away from manidolfd
Data Source
▲ PCA
● Real
No significant cases. More mild spread orthogonal variance
Data Source
▲ SMOTE
● Real
Some significant spread, clustering and orthogonal variance
Data Source
▲ Kernel
● Real
Significant spread orthogonal to the manifold!

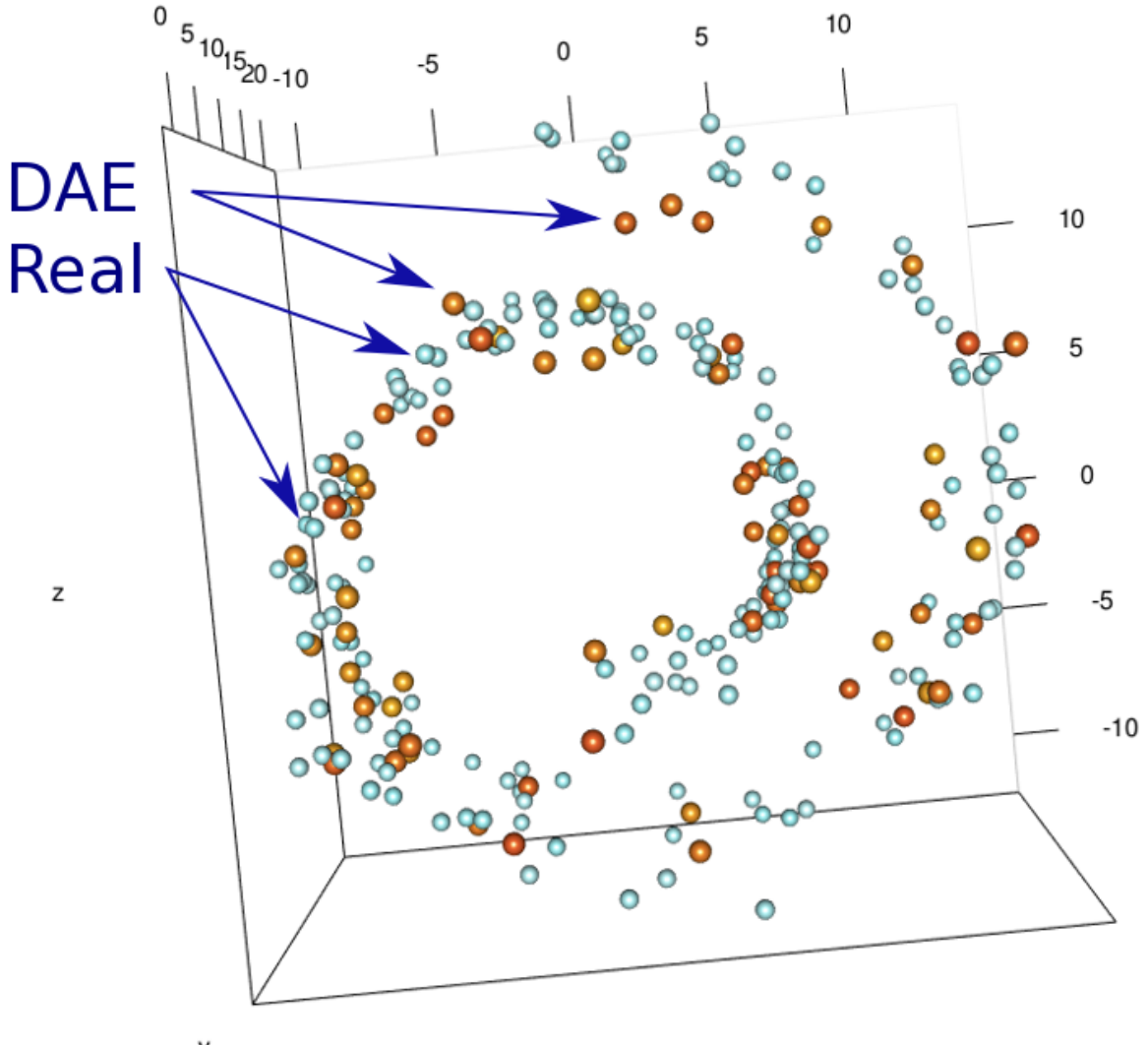
Noisy Swiss Roll



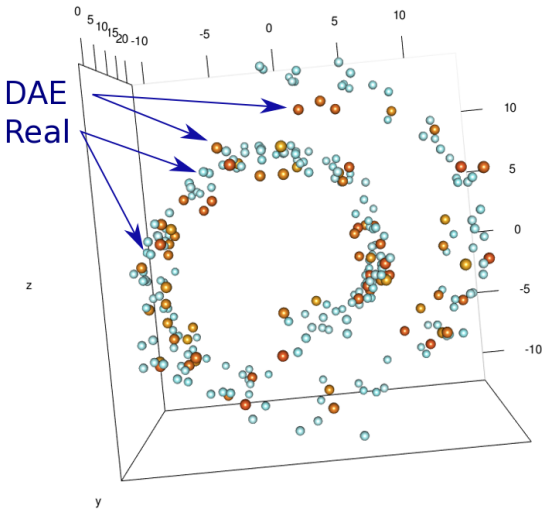
Noisy Swiss Roll

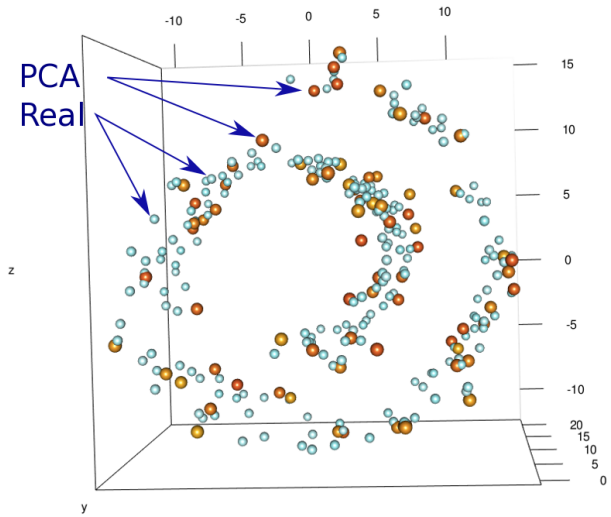


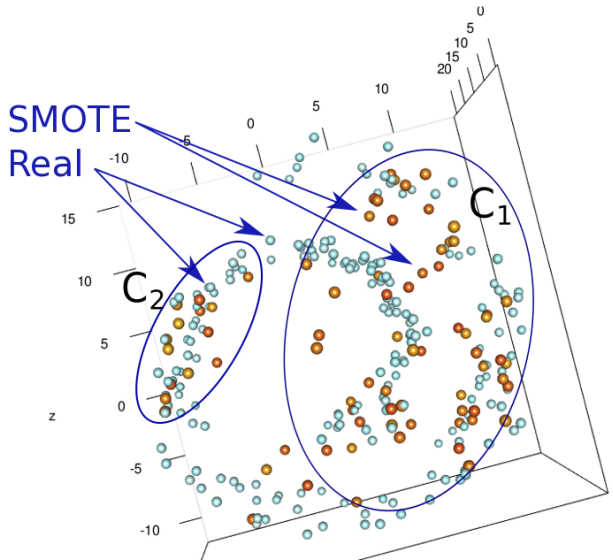
Noisy Swiss Roll



Noisy Swiss Roll







Gamma-ray Spectra Results

- Use spectra from NaI detectors to:
  - Detect radioactive threats at high-profile events
  - Perform national environmental monitoring

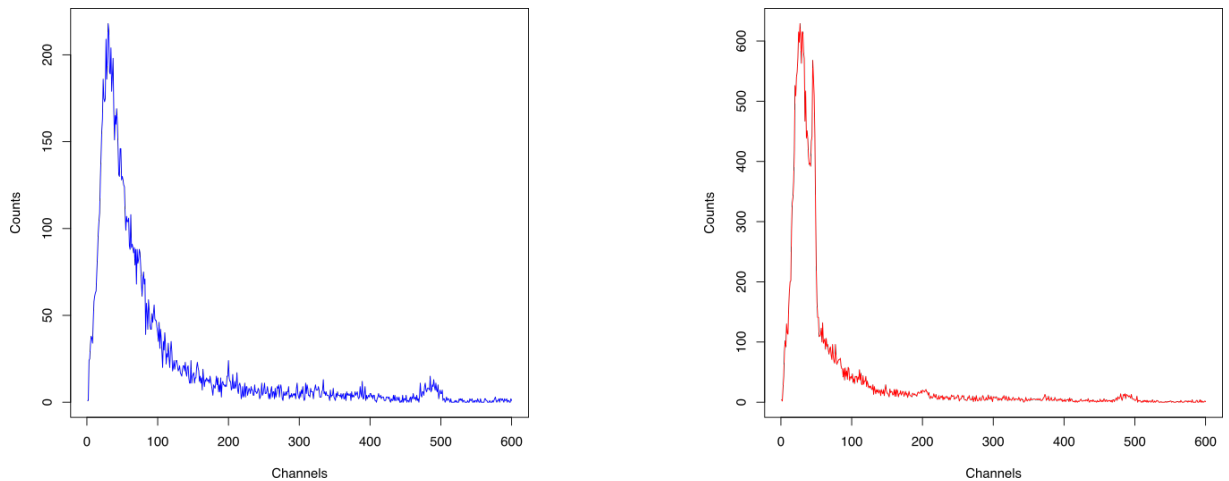
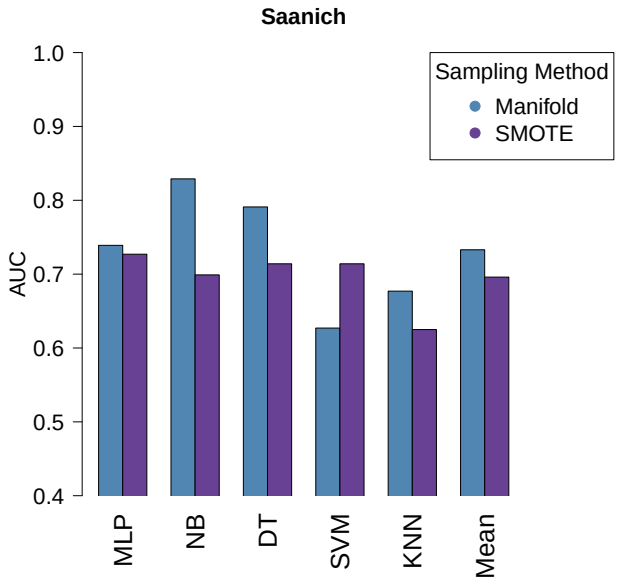
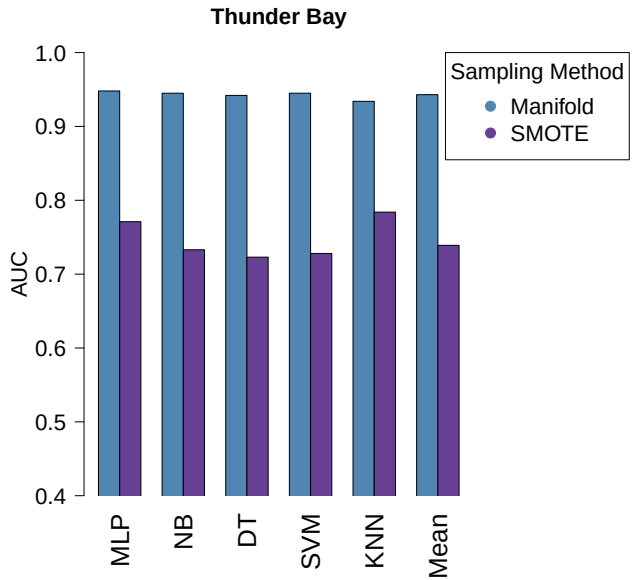
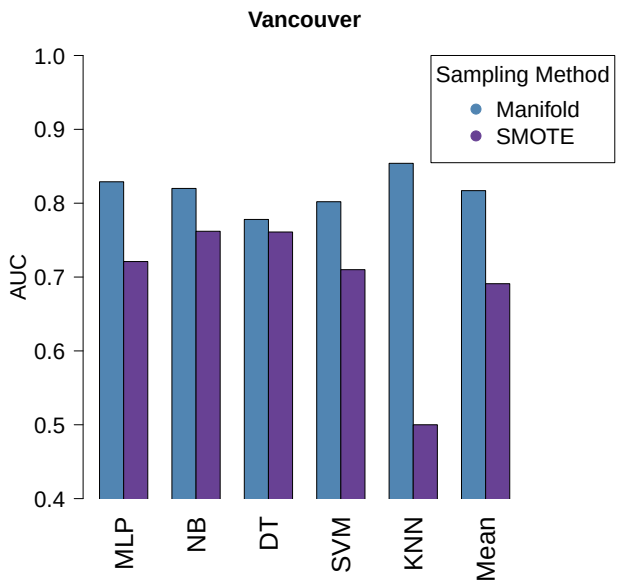


Figure: Example background spectra and target spectra.

Gamma-ray Spectra Results







UCI Results

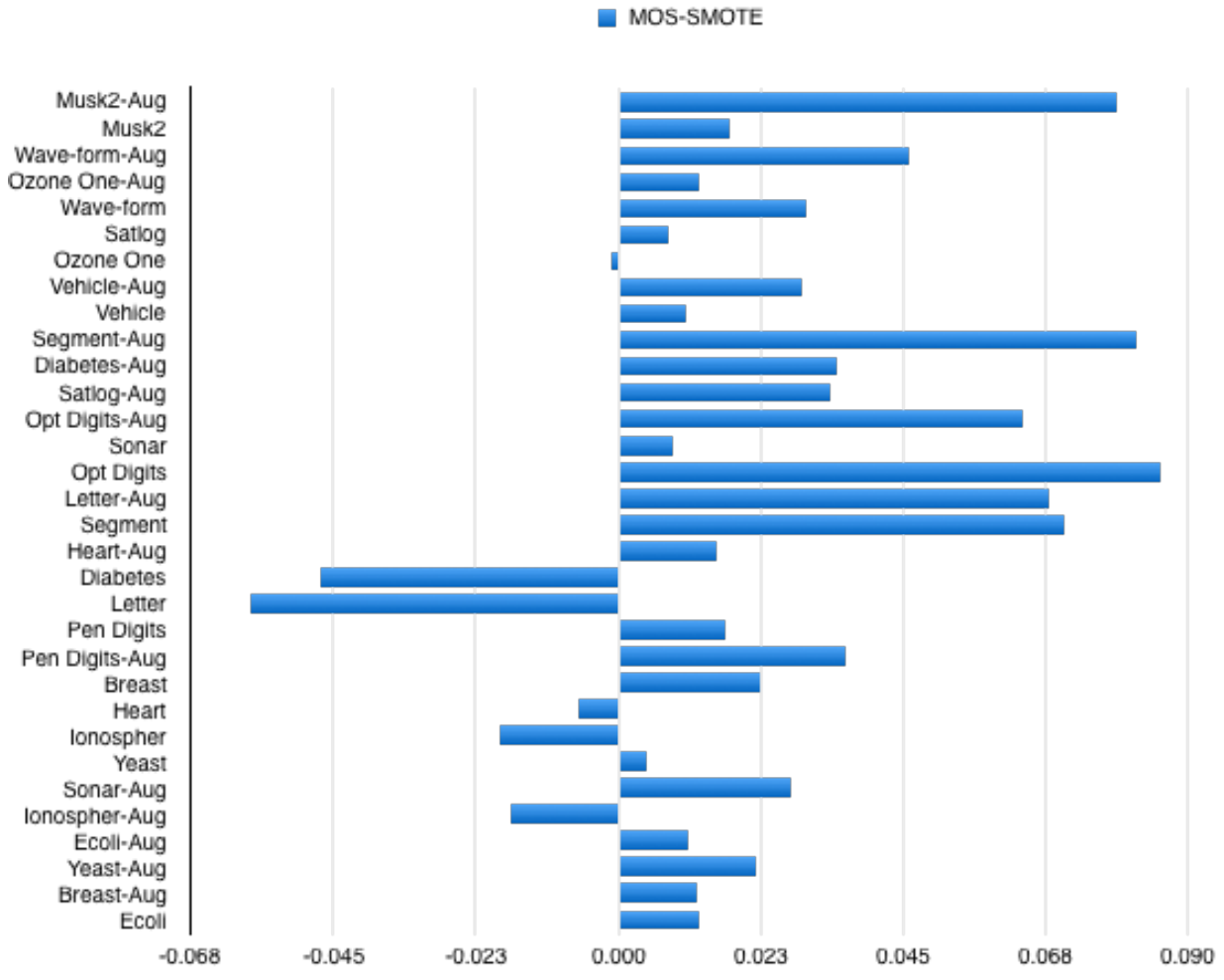


Figure: Bar plots of the performance difference $AUC(MOS) - AUC(SMOTE)$ sorted by the top $M(\cdot)$ methods .

Conclusions and Future Work

- Key point: consider the relationship between the generative bias and the target domain
  - Design accordingly
- Practically:
  - Demonstrated the weaknesses of existing methods on manifold data
  - Developed a framework for manifold-based synthetic oversampling
- We demonstrated the benefit of the framework in terms of the AUC.
- Future Work
  - Extend to multi-class and multi-label
  - Study the potential within deep learning

Publications

Publications:

- Bellinger, C., Japkowicz, N. and Drummond, C (2015). Synthetic Oversampling for Radioactive Threat Detection. In *Proceedings of 14th International Conference on Machine Learning and Applications* , 2015 (Best paper)
- Bellinger, C., Drummond, C and Japkowicz, N. (2016). Beyond the Boundaries of SMOTE: A Framework for Manifold-Base Synthetic Oversampling. In *Proceedings of European Conference on Machine Learning*, 2016

Under Review:

- Bellinger, C., Drummond, C and Japkowicz, N. (2016). Beyond the Boundaries of SMOTE: A Framework for Manifold-Base Synthetic Oversampling. In *Machine Learning Journal*, 2016

Questions?

Bibliography

Akbani, R., Kwek, S., and Japkowicz, N. (2004).

Applying Support Vector Machines to Imbalanced Datasets.
Machine Learning: ECML 2004, 3201(July):39–50.

Chawla, N., Bowyer, K., Hall, L., and W.P., K. (2002).

SMOTE: Synthetic Minority Over-Sampling Technique.
J. Artificial Intelligence Research, 16:321–357.

Humphreys, L. G. and Montanelli, R. G., J. (1975).

No Title.
An investigation of the parallel analysis criterion for determining the number of common factors, 10(2):193–205.

Raiche, G., Roipel, M., and Blais, J. G. (2006).

Non graphical solutions for the Cattell's scree test.
In *The International Annual Meeting of the Psychometric*.

Revelle, W. (2016).

psych: Procedures for Psychological, Psychometric, and Personality Research.

Revelle, W. and Rocklin, T. (1979).

Very simple structure - alternative procedure for estimat- ing the optimal number of interpretable factors.
Multivariate Behavioral Research, 4(14):403–414.

Ruscio, J. and Roche, B. (2012).

Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure.
Psychological assessment, 24(2):282.

Schwarz, G. (1978).

Estimating the dimension of a model.

Appendix - UCI Data

Table: UCI Datasets applied in these experiments.

	min class	dim
Breast	Malignant	9
Diabetes	Positive	8
Ecoli	1	7
Heart Statlog	Present	13
Ionosphere	B	34
letter	R	16
musK2	1	166
opt Digits	4	64
Ozone One	1	72
Pen Digits	3	16
Satlog	4	36
Segmentation	Brickface	19
Sonar	Rock	60
Vehicle	Saab	18
Wave	1	40
Yeast	MIT	8

Appendix - UCI Data

Table: AUC results on the benchmark UCI datasets for the baseline, SMOTE and the two manifold-based methods DAE and PCA.

	Baseline	SMOTE	MOS_{PCA}	MOS_{DAE}
Musk2	0.724	0.776	0.767	**0.793**
Opt Digits	0.828	0.830	0.850	**0.916**
Wave-form	0.657	0.725	0.733	**0.755**
Satlog	0.675	0.770	0.776	**0.778**
Ionospher	0.778	**0.855**	0.836	0.829
Sonar	0.724	0.733	**0.742**	0.740
Ozone One	0.625	**0.710**	0.709	0.702
Segment	0.864	0.889	0.879	**0.960**
Vehicle	0.581	0.657	0.665	**0.667**
Pen Digits	0.946	0.957	0.972	**0.974**
Breast	0.915	0.930	0.943	**0.953**
Yeast	0.602	0.703	**0.707**	0.653
Ecoli	0.887	0.937	**0.950**	0.923
Heart	0.755	**0.782**	0.776	0.770
Letter	0.762	**0.936**	0.878	0.870
Diabetes	0.569	**0.709**	0.662	0.652
Total Wins	0	5	3	8

Appendix - Manifold Augmented UCI Data

Table: Mean loss in AUC results between the benchmark UCI datasets and their augmented version for the baseline, SMOTE and the two manifold-based methods DAE and PCA.

	SMOTE	MOS_{PCA}	MOS_{DEAGO}
Wave-form	0.078	**0.038**	0.065
Segment	0.046	0.038	**0.035**
Ozone One	0.071	0.055	**0.051**
Sonar	0.082	**0.061**	**0.061**
Musk2	0.134	0.094	**0.073**
Vehicle	0.074	0.065	**0.056**
Satlog	0.092	0.095	**0.067**
Opt Digits	0.015	**0.001**	0.036
Letter	0.213	0.098	**0.078**
Ionospher	**0.033**	0.055	0.053
Breast	**-0.001**	0.002	0.009
Pen Digits	0.035	**0.016**	**0.016**
Ecoli	0.027	0.021	**0.016**
Yeast	0.066	0.023	**0.018**
Diabetes	0.106	0.045	**0.014**
Heart	0.028	0.007	**0.001**
Total Wins	2	4	10

Appendix - Manifold Conformance Test

Table: Factor analysis methods.

Acronym	Summary	Reference
PL	Profile Likelihood: Searches for the scree by finding the λ_n that maximizes the difference between the distribution of $1 \ldots n$ and $n+1 \ldots m$, where n is the number of eigenvalues.	[Zhu and Ghodsi, 2006]
Fact	Factors: Compares the scree of factors of the observed data with that of a random data matrix. Reports the number of factors with eigenvalues $>$ eigenvalues of random data.	[Revelle, 2016]
Comp	Components: Compares the scree of components of the observed data with that of a random data matrix. Reports the number of components with eigenvalues $>$ eigenvalues of random data.	[Revelle, 2016]
MAP	Velicer's Minimum Average Partial criterion: Applies principal components analysis and follows this by examining a series of matrices of partial correlations.	[Revelle and Rocklin, 19
VSS	Very Simple Structure criterion: Compares the original correlation matrix to that reproduced by a simplified version of the original factor matrix.	[Velicer, 1976]
BIC	Bayesian Information Criterion: Chooses the most likely model from a set of models.	[Schwarz, 1978]
ABIC	Sample Size Adjusted BIC: Chooses the most likely model from a set of models.	[Schwarz, 1978]
PA	Parallel Analysis: Creates a random data matrix and compares the eigenvalues values calculated on it to the eigenvalue calculated on the target domain. All components with eigenvalues greater than the mean of the eigenvalues for the random data are kept.	[Humphreys and Montan

Appendix - Manifold Conformance Test

Table: Factor analysis methods.

Acronym	Summary	Reference
CD	Data comparison: Variant on PA that reproduces the observed correlation matrix rather than generating random data.	[Ruscio and Roche, 2012
$\lambda > \mu$	$\lambda > mean(\lambda)$: Selects the end of the scree as the point where the eigenvalues become less than the mean of the eigenvalues.	[Revelle, 2016]
OC	Optimal Coordinate: Determines the location of the scree by measuring the gradients associated with eigenvalues and their preceding coordinates.	[Raiche et al., 2006]
AF	Acceleration Factor: Numerical solution for determining the coordinate where the slope of the curve changes most abruptly.	[Raiche et al., 2006]

Appendix - Manifold Conformance Test

$$m(D) = \mathcal{F}(D)/dim(D). \tag{1}$$

Table: Correlation between difference $AUC(MOS) - AUC(SMOTE)$ and $M(\cdot)$.

	FACT	COMP	MAP	BIC	ABIC	DC
DIFF	-0.507	-0.506	0.346	-0.139	-0.225	0.377

	$\lambda > \mu$	PA	OC	AF	PL
DIFF	-0.007	-0.448	-0.381	-0.520	-0.306

Appendix - Manifold Conformance Test

Table: Number of times each method produced the top mean AUC above and below $M(\cdot)$ score threshold for the top factor analysis methods.

			Total Wins			
	Threshold	Num DS	PCA	DAE	MOS	SMOTE
Comp	< 0.176	20	16	19	19	1
	≥ 0.176	12	7	4	7	5
Fact	< 0.206	19	15	18	18	1
	≥ 0.206	13	8	5	8	5
PL	< 0.125	18	14	17	17	1
	≥ 0.125	14	9	6	9	5
OC	< 0.154	17	13	14	16	1
	≥ 0.154	15	10	9	10	5
PA	< 0.164	16	12	15	15	1
	≥ 0.164	16	11	8	11	5
AF	< 0.016	8	6	7	7	1
	≥ 0.016	24	17	16	19	5

Appendix

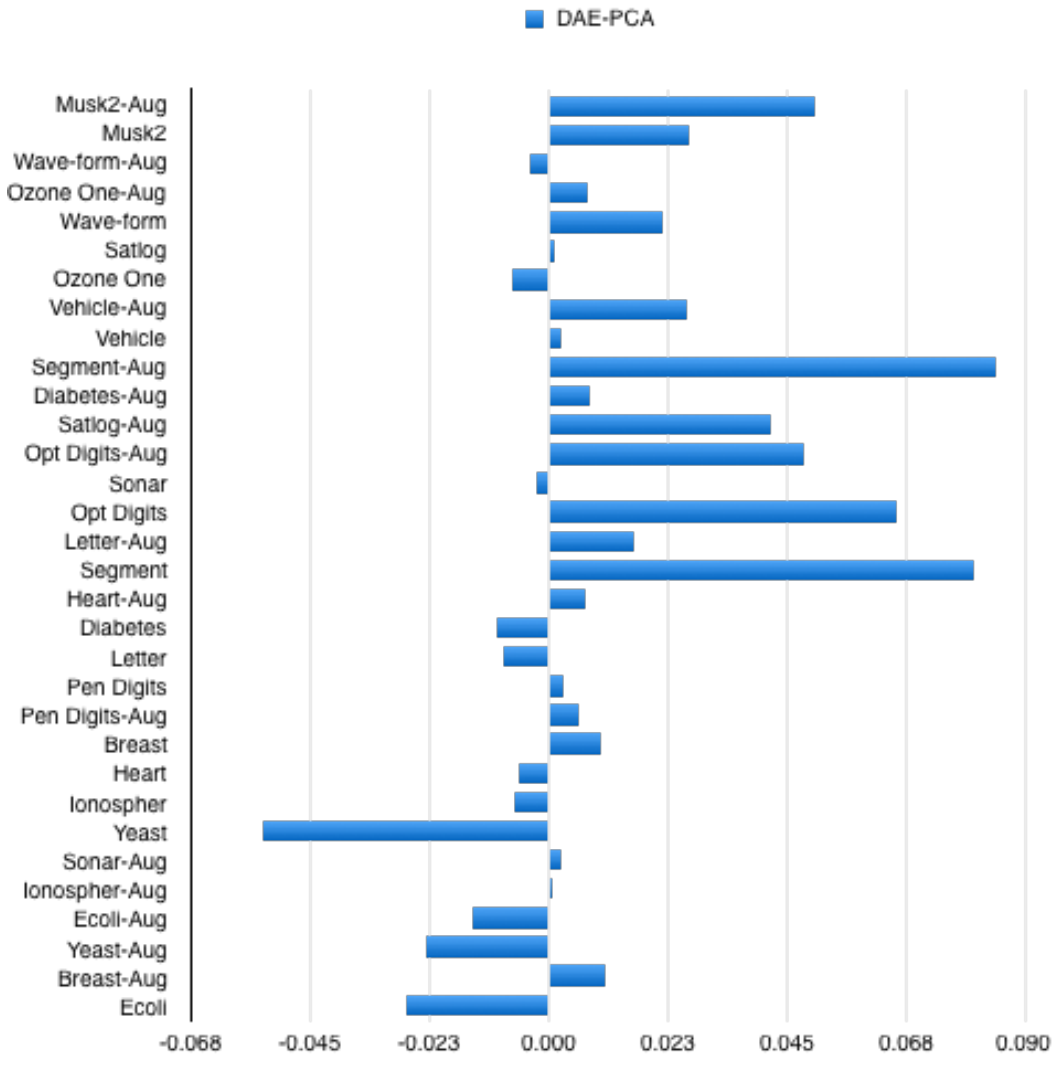


Figure: Bar plots of the performance difference $AUC(DAE) - AUC(PCA)$ sorted by $M(\cdot)$ using the Comp and PL methods .

Appendix

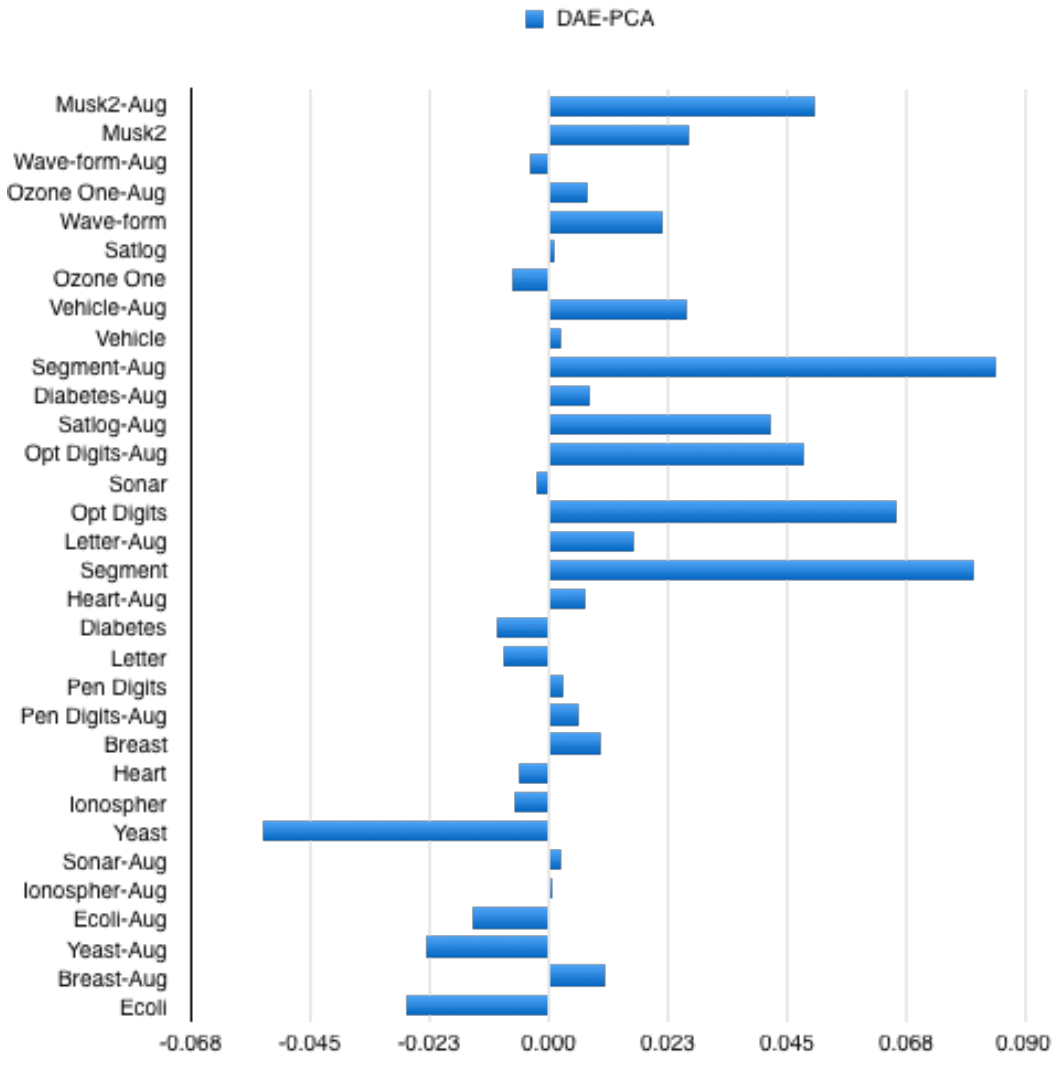


Figure: Bar plots of the performance difference $AUC(DAE) - AUC(PCA)$ sorted by $M(\cdot)$ using the Comp and PL methods .