Anomaly detection using Auto-encoder based on Skew Normal Mixture Model

Narin Park

Department of Statistics, Sungkyunkwan University

Dec 15, 2023

OUTLINE

1. Introduction
2. Deep Autoencoding Skew-Normal Mixture model (DASKNMM)
3. Simulation
4. Real Data

AUTOENCODER

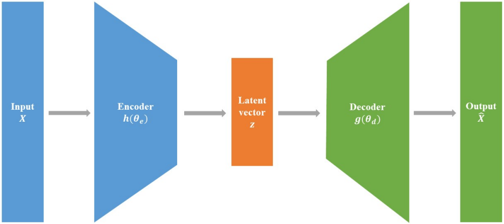


Figure 1: Illustration of autoencoder.

- **Encoder** : compresses the input into a latent variable

$$z_l = h\left(\mathbf{W}_1 x + b_1\right) \in \mathbb{R}^m \text{ where } m < d,$$

where d is the input dimension, m is the number of nodes, W_1 is (m, d) weight matrix, $\mathbf{b}_1$ is (m, 1) bias vector

AUTOENCODER,(Cont.)

- **Decoder** : reconstructs the input

$$\hat{\mathbf{x}} = g\left(\boldsymbol{W}_2 z_l + b_2\right) \in \mathbb{R}^d$$

where W_2 is (d, m) weight matrix, $\boldsymbol{b}_2$ is $(\mathrm{d}, 1)$ bias vector

- **Loss function**

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i - x'\left(\theta_e, \theta_d\right)\right\|^2,$$

where θ_e and θ_d are $\boldsymbol{W}$ and $\boldsymbol{b}$ from encoding and decoding network respectively

Architecture of DASKNM-EM

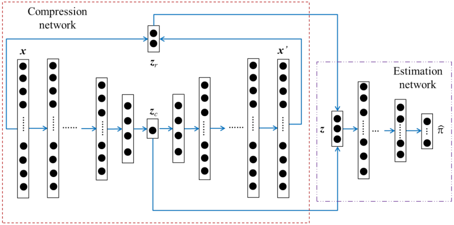


Figure 2: An overview of DAGMM (Zong et al. (2018))

- The Deep autoencoding skew-normal mixture model (DASKNM-EM) architecture consists of 2 main components :
- (1) Compression network : Dimension reduction & Generate reconstruction error by autoencoder
- (2) Estimation network : Estimate the probability distribution of compressed data using skew-normal mixture (SKNM) with the Expectation-Maximization(EM) algorithm

Compression Network

- To feed low-dimensional representation z to estimation net
- Output from compression network.

$$\mathbf{z} = [\mathbf{z}_l, \mathbf{z}_{d_1}, \mathbf{z}_{d_2}; \boldsymbol{\theta}_e, \boldsymbol{\theta}_d].$$

$\mathbf{z}_l = h(\boldsymbol{W}_1 x + b_1)$ (low-dimensional representation).

$\hat{\mathbf{x}} = g(\boldsymbol{W}_2 \mathbf{z}_l + b_2)$ (reconstructed vector)

$\mathbf{z}_{d_1} = \frac{\|x-\hat{x}\|_2}{\|x\|_2}$ ($x, \hat{x}$, relative Euclidean distaince).

$\mathbf{z}_{d_2} = \frac{\mathbf{x}\cdot\hat{\mathbf{x}}}{||\mathbf{x}||_2 \ ||\hat{\mathbf{x}}||_2}$ (Cosine similarity).

$\boldsymbol{\theta}_e = (\mathbf{W}_1, \mathbf{b}_1)$ (Parameter of encoder)

$\boldsymbol{\theta}_d = (\mathbf{W}_2, \mathbf{b}_2)$ (Parameter of decoder)

Estimation Part

- DAGMM rely on the assumption that the data follows a Gaussian distribution so it often struggle about strict inclusion of a normal density (Azzalini (1985)).
- For capturing wide range of the indices of skewness and kurtosis (Azzalini and Valle (1996)), skew normal mixture distribution seems to be sutiable for density estimation.

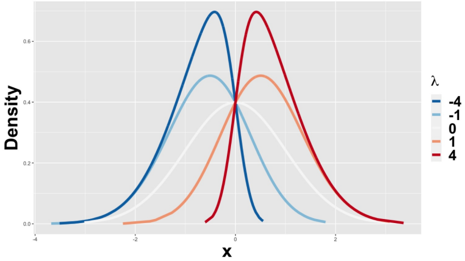


Figure 3: Density of SN(0,1,λ)

ESTIMATION PART,(Cont.)

Skew Normal Mixture Model (SNMM) (Lin (2009))

- Model: $\mathbf{z}_j \sim \sum_{i=1}^{g} \pi_i f\left(\mathbf{z}_j \mid \xi_i, \Sigma_i, \Lambda_i\right), \quad \pi_i \geq 0, \quad \sum_{i=1}^{g} \pi_i = 1,$
- M-variate skew normal distribution:

$$f(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\Sigma}, \boldsymbol{\Lambda}) = 2^M \phi_M(\mathbf{z} \mid \boldsymbol{\xi}, \Omega)\Phi_M\left(\Lambda^{\mathrm{T}}\Omega^{-1}(\mathbf{z}-\boldsymbol{\xi}) \mid \boldsymbol{\Delta}\right)$$

  where $\Omega = \Sigma + \Lambda\Lambda^{\mathrm{T}}$, $\boldsymbol{\Delta} = \left(\mathbf{I}_M + \Lambda^{\mathrm{T}}\Sigma^{-1}\Lambda\right)^{-1} = \mathbf{I}_M - \Lambda^{\mathrm{T}}\boldsymbol{\Omega}^{-1}\Lambda$ and, ϕ_k and Φ_k represents pdf and cdf of a M-dimensional multivariate normal distribution, respectively. We write $\mathbf{Z} \sim SN_M(\boldsymbol{\xi}, \boldsymbol{\Sigma}, \boldsymbol{\Lambda})$
- The one of useful properties of MSN is that Z has a convenient stochastic representation which can be written as

$$\mathbf{Z} \stackrel{d}{=} \boldsymbol{\xi} + \Lambda\boldsymbol{\tau} + \mathbf{U}$$

  where $\boldsymbol{\tau} \sim HN_k(\mathbf{0}, I)$, $\mathbf{U} \sim N_k(0, \Sigma)$ and $\boldsymbol{\tau}$ and $\mathbf{U}$ are independent.
- By using the representation, we can estimate $(\boldsymbol{\xi}, \Sigma, \Lambda)$ with EM algorithm.

Estimation Part,(Cont.)

Estimated parameters by EM algorithm,(Lin (2009))

$$\pi_i^{(t+1)} = \sum_{j=1}^{n} \frac{w_{ij}^{(t)}}{n}, \qquad \hat{\xi}_i^{(k+1)} = \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \mathbf{z}_j - \hat{\Lambda}_i^{(k)} \sum_{j=1}^{n} w_{ij}^{(k)} \hat{\eta}_{ij}^{(k)} \right) / \sum_{j=1}^{n} w_{ij}^{(k)}$$

$$\hat{\Lambda}_i^{(k+1)} = \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \left(\mathbf{z}_j - \hat{\boldsymbol{\xi}}_i^{(k+1)} \right) \hat{\boldsymbol{\eta}}_{ij}^{(k)} \right) \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \hat{\Psi}_{ij}^{(k)} \right)^{-1} .$$

$$\hat{\Sigma}^{(t+1)} = \frac{1}{n} \left\{ \sum_{j=1}^{n} \left(w_j - \hat{\xi}^{(t+1)} - \hat{\Lambda}^{(t+1)} \hat{\eta}_j^{(t)} \right) \left(w_j - \hat{\xi}^{(t+1)} - \hat{\Lambda}^{(t+1)} \hat{\eta}_j^{(t)} \right)^{\top} \right.$$
$$\left. + \hat{\Lambda}^{(t+1)} \left(\hat{\Psi}_j^{(t)} - \hat{\eta}_j^{(t)} \hat{\eta}_j^{(t)\top} \right) \hat{\Lambda}^{(t+1)\top} \right\}$$

Objective function

Objective function

We minimize the objective function for finding optimal weight and bias such that

$$J(\boldsymbol{\theta}_e, \boldsymbol{\theta}_d) = \frac{1}{n} \mid\mid \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i; \boldsymbol{\theta}_e, \boldsymbol{\theta}_d \mid\mid_2^2 + \frac{\lambda_1}{n} \sum_{i=1}^{n} EN(\mathbf{z}_i).$$

- **Reconstruction error** (compression part)

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \mid\mid \boldsymbol{x} - \hat{\boldsymbol{x}} \mid\mid_2^2 .$$

- **Energy function** (estimation part)

$$EN(\mathbf{z}_i) = -log(\sum_{j=1}^{k} \hat{\pi}_j f\left(\mathbf{z}_j \mid \xi_i, \Sigma_i, \Lambda_i\right)),$$

where $f(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\Sigma}, \boldsymbol{\Lambda}) = 2^p \phi_p(\mathbf{z} \mid \boldsymbol{\xi}, \Omega) \Phi_p\left(\Lambda^{\mathrm{T}} \Omega^{-1}(\mathbf{z} - \boldsymbol{\xi}) \mid \boldsymbol{\Delta}\right).$

Simulation Environments

- The compression part was constructed in the symmetric way as FC(6,12,tanh)- FC(12,4,tanh)- FC(4,1,none)-FC(4,12,tinh)- FC(12,6,none)
- The estimation part was performed with FC(3,10,tanh) -FC(10,2,softmax) where FC(α, β, f) refers to the fully connected layer consisting of the input neurons α and β and the activation function f for DAGMM
- learning rate = 0.001, regularization parameter λ_1=0.001
- Optimization by RMSprop, Initialization by HE-initialization
- Use a 90%-10% split of sampled data for training and testing, employing Monte Carlo with 50 replications.
- $\boldsymbol{y} = [\boldsymbol{X}, \boldsymbol{O}]$ where $\boldsymbol{X}$ is normal samples and $\boldsymbol{O}$ is anomaly samples. Both samples are constructed with 6- dimension.

BASELINE MODEL

Model				estimation method	
	distribution assumption	end-to-end	Autoencoder	MLN	EM algorithm
DAGMM	GMM	O	O	O	X
DAGMMEM	GMM	O	O	X	O
DAGM	Gasussian	O	O	X	X
DASKN-EM	Skew-normal	O	O	X	O
DASKNM-EM	SKNM	O	O	X	O

- MLN means multi-layer neural network ; parameter estimation.
- Two-step approach: dimensionality reduction is first conducted, and then density estimation is performed.
- The threshold is chosen the point which has the largest f1-score by each iteration

Simulation Distribution for CASE1 and Case2

- Sampling distribution for case 1 where $\boldsymbol{X}$ and $\boldsymbol{O}$ denote skewed normal samples and separately distributed anomaly samples with normal, respectively.

$$\begin{array}{ll}
\boldsymbol{X} = [X_1, X_2, X_3, X_4, X_5, X_6] & \boldsymbol{O} = [O_1, O_2, O_3, O_4, O_5, O_6] \\
X_1 \sim \text{Lognormal}(10, 10) & O_1 \sim N(20, 15) \\
X_2 \sim \text{Gamma}(0.02, 10) & O_2 \sim \text{Uniform}(20, 25) \\
X_3 \sim \text{Beta}(10, 0.02) & O_3 \sim \text{-Exponential}(3) \\
X_4 = X_1 + \log(X_2) + N(0,1) & O_4 = O_1 * O_2 + N(0,1) \\
X_5 = X_2 + \exp(X_3) + N(0,1) & O_5 = O_3^2 + \log(O_2) + N(0,1) \\
X_6 = X_2 \cdot X_1 + N(0,1) & O_6 = \exp(O_1) + O_3 + N(0,1)
\end{array}$$

- Sampling distribution for case 2 involves $\boldsymbol{X}$ representing skewed normal samples and $\boldsymbol{O}$ representing closely distributed anomaly samples with normal distribution, respectively.

$$\begin{array}{l}
\boldsymbol{O} = [O_1, O_2, O_3, O_4, O_5, O_6] \\
O_1 \sim \text{Lognormal}(17, 17), \;\; O_4 = O_1 + \log(O_2) + N(0,1) \\
O_2 \sim \text{Gamma}(1, 10), \;\; O_5 = O_2^2 + \log(O_3) + N(0,1) \\
O_3 \sim \text{Beta}(10, 1), \;\; O_6 = \exp(O_1) + O_3 + N(0,1)
\end{array}$$

Simulation Distribution for CASE3 and Case4

- Sampling distribution for case 3 involves $\boldsymbol{X}$ representing symmetric normal samples and $\boldsymbol{O}$ representing separately distributed anomalies with normal.

$$
\begin{array}{ll}
\boldsymbol{X} = [X_1, X_2, X_3, X_4, X_5, X_6] & \boldsymbol{O} = [O_1, O_2, O_3, O_4, O_5, O_6] \\
X_1 \sim N(1,2) & O_1 \sim N(10,10) \\
X_2 \sim \text{Uniform}(0,2) & O_2 \sim \text{Uniform5},10) \\
X_3 \sim \text{Gamma}(5,03) & O_3 \sim \text{-Exponential}(8) \\
X_4 = X_1 + \log(X_2) + N(0,1) & O_4 = O_1 + \log(O_2) + N(0,1) \\
X_5 = X_2 + \log(X_3) + N(0,1) & O_5 = O_2 + \exp(O_3) + N(0,1) \\
X_6 = X_2 \cdot X_1 + N(0,1) & O_6 = \exp(O_2) \cdot O_1 + N(0,1)
\end{array}
$$

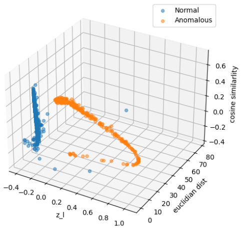
- Sampling distribution for case 4, which only differs from case 1 in terms of sampling $\boldsymbol{O}$, aims to bring normal and anomaly samples closer.

$$
\begin{array}{l}
\boldsymbol{O} = [O_1, O_2, O_3, O_4, O_5, O_6] \\
O_1 \sim N(4,3), \;\; O_4 = O_1 + \log(O_2) + N(0,1) \\
O_2 \sim \text{Uniform}(2,4), \;\; O_5 = O_2^2 + \exp(O_3) + N(0,1) \\
O_3 \sim \text{Exponetial}(2), \;\; O_6 = \exp(O_1) \cdot O_3 + N(0,1)
\end{array}
$$

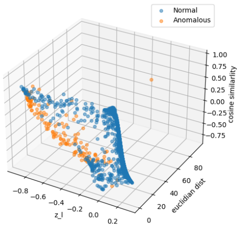
INFORMATION CRITERIA

Table 1: Information criteria for component selection in simulation

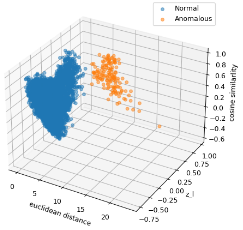
		The number of clusters			
		case 1 and case 2		case3 and cas4	
Model		2	3	2	3
GMM	AIC	**-7957.880**	-7975.055	5811.679	**5734.921**
	BIC	**-7957.056**	-7822.156	5920.585	**5901.147**
SKNM	AIC	-3295.583	**-3291.4170**	**11373.073**	12244.780
	BIC	**-3321.723.**	-3331.078	**11346.933**	12205.119



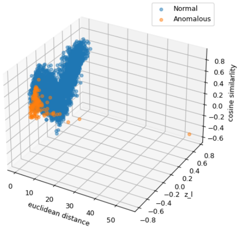
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

Figure 4: The z-space visualization

Metrics

Confusion matrix

Predicted Class

		P	**N**
Actual Class	**P**	True Positive	False Negative
	N	False Positive	True Negative

- Precision : $\frac{TP}{TP+FP}$ where positive is anomaly
- Recall : $\frac{TP}{TP+FN}$
- F1 score : $2 \times \frac{Precision \times Recall}{Precision + Recall}$

RESULT:CASE1

Table 2: The average accuracy over 50 simulations for case 1

n	Model	F1-score	Precision	Recall	FPR	FP
1500	DAGM	0.9864	0.9756	0.9976	0.0067	1.92
	DAGMM	0.9777	0.9574	0.9997	0.0122	3.48
	DAGMM-EM	0.9907	0.9824	0.9995	0.0048	1.38
	DASKN-EM	0.9831	0.9679	0.9995	0.0091	2.60
	DASKNM-EM	**0.9951**	**0.95**	**0.9994**	**0.0023**	**0.65**
3000	DAGM	0.9904	0.9836	0.9976	0.0045	2.54
	DAGMM	0.9780	0.9597	0.9979	0.0114	6.50
	DAGMM-EM	0.9913	0.9831	**0.9997**	0.0046	2.60
	DASKN-EM	0.9803	0.9615	0.966	0.0176	10.05
	DASKNM-EM	**0.9955**	**0.9918**	0.9992	**0.0044**	**1.26**

RESULT:CASE2

Table 3: The average accuracy over 50 simulations for case 2

n	Model	Accuracy				
		F1-score	Precision	Recall	FPR	FP
1500	DAGM	0.8978	0.8665	0.9347	0.0385	10.98
	DAGMM	0.9033	0.8584	**0.9557**	0.0425	12.10
	DAGMM-EM	0.8956	0.8724	0.9151	0.0373	10.62
	DASKN-EM	0.8888	0.8699	0.9111	0.0365	20.82
	DASKNM-EM	**0.9115**	**0.8948**	0.9309	**0.0294**	**8.37**
3000	DAGM	0.9043	0.8628	0.9444	0.0402	22.90
	DAGMM	0.8996	0.8574	**0.9487**	0.0430	24.28
	DAGMM-EM	0.8950	0.8728	0.9200	0.0358	20.42
	DASKN-EM	0.8924	0.8791	0.9087	0.0335	19.12
	DASKNM-EM	**0.9104**	**0.8906**	0.9324	**0.0305**	**17.33**

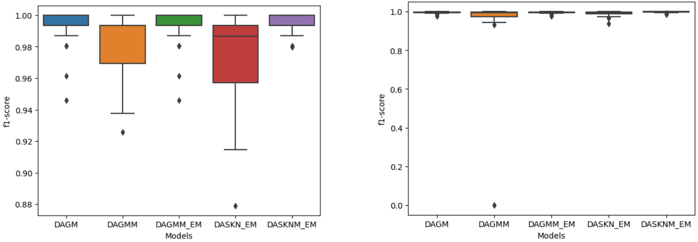


Figure 5: Box-plot of f1-score when n is 1500 (left) and when n is 3000 (right) for case 1

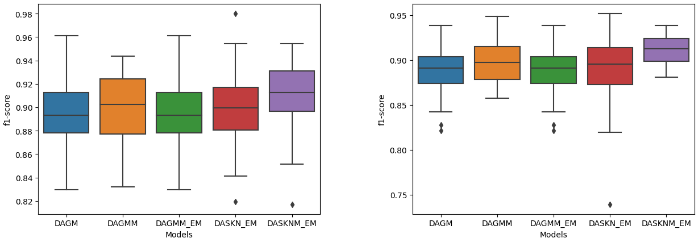


Figure 6: Box-plot of f1-score when n is 1500 (left) and when n is 3000 (right) for case 2

RESULT:CASE3

Table 4: The average accuracy over 50 simulations for case 3

		Accuracy				
n	Model	F1-score	Precision	Recall	FPR	FP
1500	DAGM	**0.9991**	**0.9982**	**1.000**	**0.0005**	**0.14**
	DAGMM	0.9971	0.9948	0.9995	0.0041	0.40
	DAGMM-EM	0.9985	0.9976	0.9995	0.0006	0.18
	DASKN-EM	0.9961	0.9947	0.9976	0.0014	0.44
	DASKNM-EM	0.9929	0.9921	0.9943	0.0023	0.64
3000	DAGM	**0.9990**	**0.9984**	**0.9995**	**0.0004**	**0.24**
	DAGMM	0.9955	0.9941	0.9971	0.0015	0.90
	DAGMM-EM	0.9985	0.9972	0.9997	0.0007	0.42
	DASKN-EM	0.9953	0.9968	0.9965	0.0008	0.48
	DASKNM-EM	0.9940	0.9921	0.9973	0.0011	0.60

Result:case4

Table 5: The average accuracy over 50 simulations for case 4

		Accuracy				
n	Model	F1-score	Precision	Recall	FPR	FP
1500	DAGM	0.8944	0.8883	0.9037	**0.0309**	**8.82**
	DAGMM	0.8770	0.8715	0.8848	0.0352	10.02
	DAGMM-EM	**0.9126**	**0.8885**	**0.9397**	0.0314	8.96
	DASKN-EM	0.8596	0.8642	0.8608	0.0368	10.48
	DASKNM-EM	0.8782	0.8813	0.804	0.0312	9.42
3000	DAGM	0.8793	0.8866	0.8744	0.0300	17.12
	DAGMM	0.8867	0.8889	0.8867	0.0294	16.78
	DAGMM-EM	**0.9114**	0.9035	**0.9207**	0.0262	14.92
	DASKN-EM	0.8558	0.8778	0.8388	0.0315	17.94
	DASKNM-EM	0.8889	**0.9154**	0.8793	**0.0261**	**14.88**

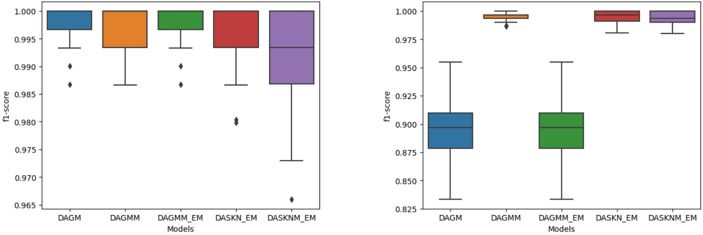


Figure 7: Box-plot of f1-score when n is 1500 (left) and when n is 3000 (right) for case 3

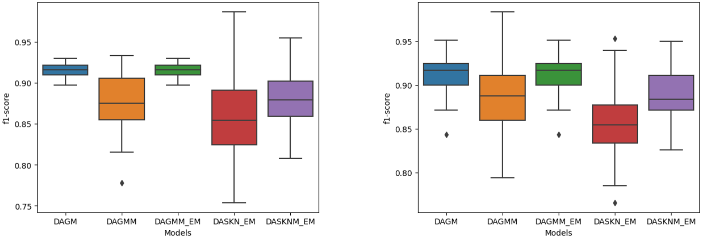


Figure 8: Box-plot of f1-score when n is 1500 (left) and when n is 3000 (right) for case 4

Real data analysis setting

	#Dimensions	#Instances	Anomality
Credit card	29	5,492	4.69%
Satellite	36	5025	1.49%

Table 6: Anomaly Dataset

	Layer	Learning rate	Regularization param	Threshold
Credit card	[25,20,10]	0.001	0.01	$\geq 80\%$
Satellite	[30,20,10]	0.001	0.01	$\geq 60\%$

Table 7: Analysis Setting

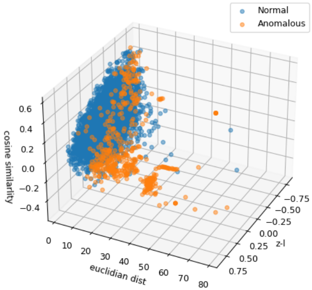
- Use a 90%-10% split of sampled data for training and testing, employing Monte Carlo with 20 replications.
- Optimization by RMSprop, Initialization by HE-initialization

Information criteria

Table 8: Information criteria for component selection in real data

		The number of clusters			
		Credit Card		Satellite	
Model		3	4	2	3
GMM	AIC	**21202.444**	21023.146	15277.540	**13027.137**
	BIC	**21388.387**	21273.208	15399.458	**13213.222**
SKNM	AIC	**35845.222**	35849.598	**24871.989**	26054.794
	BIC	**35805.562**	35796.417	**24859.370**	26028.653

Credit Card



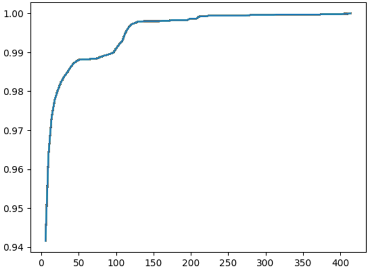


Figure 9: z-space (left) and CDF of energy value (right) of credit card data

CREDIT CARD(Cont.)

Table 9: Average accuracy of simulation 20 times for the credit card data

n	Model	Accuracy				
		F1-score	Precision	Recall	FPR	FP
5000	DAGM	0.8762	0.9136	0.8424	0.079	39.05
	DAGMM	0.8626	0.9122	0.8187	**0.0391**	39.10
	DAGMM-EM	0.8138	0.7787	0.8603	0.2532	126.60
	DASKN-EM	0.8795	0.9148	0.8477	0.0785	39.25
	DASKNM-EM	**0.9042**	**0.9292**	**0.8806**	0.0661	**33.05**

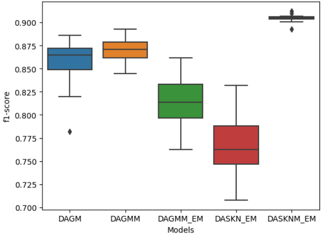


Figure 10: f1-score box plots of credit card for 20 times

2.SATELLITE

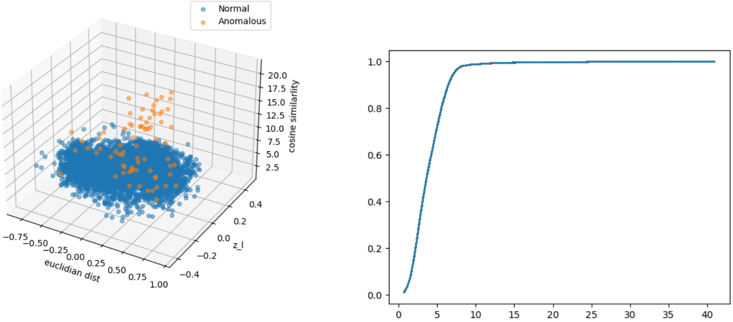


Figure 11: z-space (left) and CDF of energy value (right) of satellite d data

2.SATELLITE(Cont.)

Table 10: Average accuracy of simulation 20 times for the satellite data

Model	Accuracy				
	F1-score	Precision	Recall	FPR	FP
DAGM	0.6641	0.8318	0.5573	0.0180	9.05
DAGMM	0.5602	0.7613	0.4613	0.0260	13.10
DAGMM-EM	**0.7271**	0.8188	**0.6620**	0.2455	12.35
DASKN-EM	0.6692	0.8101	0.5780	0.0224	11.30
DASKNM-EM	0.6871	**0.8827**	0.5666	**0.0122**	**6.15**

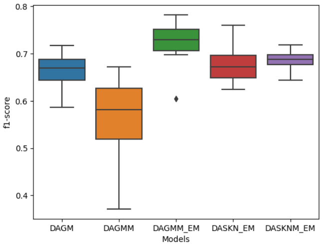


Figure 12: f1-score box plots of credit card for 20 times

Conclustion

- DASKNM-EM performs well in anomaly detection within skewed and heavy-tailed distributions.
- Furthermore, in symmetric observation cases, DASKNM-EM demonstrates competitive proficiency compared to baseline models.
- Using DASKNM-EM for detection ensures robust and reliable performance across diverse scenarios.

FURTHER STUDY AND LIMITATION

- Depending on the occasion, if you need precise detection for setting certain distribution, then this model may not be suitable.

- To make better anomaly detection, automating hyper-parameters adjustments and improve speed can help it perform better in real-world situations.

REFERENCES I

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12(2):171–178.

Azzalini, A. and Valle, A. D. (1996). The multivariate skew-normal distribution. *Biometrika*, 83(4):715–726.

Lin, T. I. (2009). Maximum likelihood estimation for multivariate skew normal mixture models. *Journal of Multivariate Analysis*, 100(2):257–265.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.

APPENDIX I

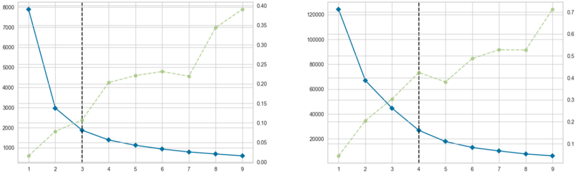


Figure 13: scree plot for K group (sat,credit)

Appendix II

E-step,(Lin (2009))

$$E\left(W_{ij} \mid \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right) = \hat{w}_{ij}^{(k)} = \frac{\hat{\pi}_i^{(k)} f\left(\mathbf{z}_j \mid \hat{\xi}_i^{(k)}, \hat{\mathbf{\Sigma}}_i^{(k)}, \hat{\Lambda}_i^{(k)}\right)}{\sum_{m=1}^{g} \hat{\pi}_m^{(k)} f\left(\mathbf{z}_j \mid \hat{\xi}_m^{(k)}, \hat{\mathbf{\Sigma}}_m^{(k)}, \hat{\Lambda}_m^{(k)}\right)},$$

$$E\left(W_{ij}\boldsymbol{\tau}_j \mid \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right) = \hat{w}_{ij}^{(k)} \hat{\boldsymbol{\eta}}_{ij}^{(k)} = E\left(W_{ij} \mid \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right) \mathrm{E}\left(\boldsymbol{\tau}_j \mid W_{ij} = 1, \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right),$$

and

$$\mathrm{E}\left(W_{ij}\boldsymbol{\pi}_j\boldsymbol{\tau}_j^{\mathrm{T}} \mid \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right) = \hat{w}_{ij}^{(k)} \hat{\Psi}_{ij}^{(k)} = \mathrm{E}\left(W_{ij} \mid \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right) \mathrm{E}\left(\boldsymbol{\tau}_j\boldsymbol{\tau}_j^{\mathrm{T}} \mid W_{ij} = 1, \mathbf{z}_j, \hat{\mathbf{\Theta}}^{(k)}\right)$$

where $E\left(\boldsymbol{\tau}_j \mid \mathbf{y}_j, Z_{ij} = 1\right) = \eta_{ij}$ and $E\left(\boldsymbol{\tau}_j\boldsymbol{\tau}_j^{\mathrm{T}} \mid \mathbf{y}_j, Z_{ij} = 1\right) = \Psi_{ij}$,

Appendix III

E-step,(Lin (2009))

Therefore, the Q-function can be written by

$$Q\left(\boldsymbol{\Theta} \mid \hat{\boldsymbol{\Theta}}^{(k)}\right) = \sum_{i=1}^{g} \sum_{j=1}^{n} \hat{z}_{ij}^{(k)} \left\{ \log\left(w_i\right) + \frac{1}{2} \log\left|\boldsymbol{\Sigma}_i^{-1}\right| - \frac{1}{2} \left(\mathbf{z}_j - \boldsymbol{\xi}_i - \boldsymbol{\Lambda}_i \hat{\boldsymbol{\eta}}_{ij}^{(k)}\right)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} \right.$$
$$\left. \times \left(\mathbf{z}_j - \xi_i - \Lambda_i \hat{\eta}_{ij}^{(k)}\right) - \frac{1}{2} \operatorname{tr}\left(\Sigma_i^{-1} \Lambda_i \left(\hat{\Psi}_{ij}^{(k)} - \hat{\eta}_{ij}^{(k)} \hat{\eta}_{ij}^{(k)}\right) \Lambda_i^{\mathrm{T}}\right) \right\}.$$

Appendix IV

M-step,(Lin (2009))

$$\pi_i^{(t+1)} = \sum_{j=1}^{n} \frac{w_{ij}^{(t)}}{n}, \qquad \hat{\xi}_i^{(k+1)} = \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \mathbf{z}_j - \hat{\Lambda}_i^{(k)} \sum_{j=1}^{n} w_{ij}^{(k)} \hat{\eta}_{ij}^{(k)} \right) / \sum_{j=1}^{n} w_{ij}^{(k)}$$

$$\hat{\Lambda}_i^{(k+1)} = \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \left(\mathbf{z}_j - \hat{\boldsymbol{\xi}}_i^{(k+1)} \right) \hat{\boldsymbol{\eta}}_{ij}^{(k)} \right) \left(\sum_{j=1}^{n} \hat{w}_{ij}^{(k)} \hat{\Psi}_{ij}^{(k)} \right)^{-1}.$$

$$\hat{\Sigma}^{(t+1)} = \frac{1}{n} \left\{ \sum_{j=1}^{n} \left(w_j - \hat{\xi}^{(t+1)} - \hat{\Lambda}^{(t+1)} \hat{\eta}_j^{(t)} \right) \left(w_j - \hat{\xi}^{(t+1)} - \hat{\Lambda}^{(t+1)} \hat{\eta}_j^{(t)} \right)^{\top} \right.$$

$$\left. + \hat{\Lambda}^{(t+1)} \left(\hat{\Psi}_j^{(t)} - \hat{\eta}_j^{(t)} \hat{\eta}_j^{(t)\top} \right) \hat{\Lambda}^{(t+1)\top} \right\}$$