Identifying outliers in the data by using Variational Autoencoders

G Sai Keerthi - CS22BTECH11024
Nalavolu Chetana - CS22BTECH11042
Rishitha Surineni - CS22BTECH150

1 Problem Statement

Given a dataset with 10 numerical features, the goal is to identify anomalous or outlier data points. Rather than applying traditional clustering or distance-based outlier detection techniques directly in the original feature space, we aim to:

1. Learn a lower-dimensional latent representation using a Variational Autoencoder (VAE).
2. Perform K-means clustering on the means of the distributions learned in the latent space.
3. Identify outliers using a dual strategy:
   - Points lying far from cluster centers.
   - Points that belong to clusters with significantly fewer members (small clusters).

2 Dataset Description

The dataset, stored in `data.csv`, comprises 1190 samples, each with 10 continuous numerical features: `cov1` to `cov7`, `sal_pur_rat`, `igst_itc_tot_itc_rat`, and `lib_igst_itc_rat`. These features represent various metrics, though their specific semantic meanings are not provided, and they are treated equally in this unsupervised learning context.

2.1 Basic Statistics

The dataset includes:

- **Samples**: 1190
- **Features**: 10
- **Feature Types**: Continuous numerical values with varying scales (e.g., `cov1` ranges from 0.1262 to 1.0, `sal_pur_rat` includes extreme values like 34.367).

Initial inspection confirms no missing values or categorical variables, making it suitable for VAE and clustering-based outlier detection.

2.2 Preprocessing

The data was standardized using `StandardScaler` from scikit-learn, transforming each feature to have zero mean and unit variance. This step ensures that features with different scales (e.g., `sal_pur_rat` vs. `cov1`) contribute equally to the VAE's latent representation and the subsequent distance-based clustering and outlier detection.

3 Methodology

3.1 Dimensionality Reduction with VAE

A Variational Autoencoder (VAE) reduces the 10-dimensional data into a lower-dimensional latent space. The VAE consists of:

- **Encoder**: Maps input x to a latent distribution $q(z|x) = \mathcal{N}(\mu, \sigma^2)$, outputting mean μ (z_μ) and log-variance $\log(\sigma^2)$ (z_{var}).
- **Decoder**: Reconstructs the input from a sampled latent vector z.

The loss function combines reconstruction loss and KL divergence:

$$\mathcal{L} = \text{Reconstruction Loss} + \text{KL Divergence}$$

where:

- **Reconstruction Loss**: Mean Squared Error (MSE) between input x and reconstructed output $\hat{x}$:
$$\text{Reconstruction Loss} = \sum (x - \hat{x})^2$$
- **KL Divergence**: Regularizes the latent distribution to approximate $\mathcal{N}(0, 1)$:
$$\text{KL} = \frac{1}{2} \sum (\exp(z_{\text{var}}) + z_\mu^2 - 1 - z_{\text{var}})$$

The optimal latent dimension was determined by testing dimensions 2, 3, and 4, selecting the one with the lowest total loss.

3.2 K-Means Clustering on Latent Centers

The latent distribution centers (z_μ) were extracted and clustered using K-means. The elbow method determined the optimal number of clusters k by plotting the within-cluster sum of squares (WCSS) against k, identifying the ”elbow” point where WCSS reduction slows.

3.3 Outlier Detection

Outliers were identified using two complementary methods:

1. **Boundary Outliers via Mahalanobis Distance**: For each cluster, the Mahalanobis distance was computed between each point and its cluster centroid, accounting for the covariance structure:

   $$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}$$

   where x is a latent point, μ is the cluster centroid, and S is the covariance matrix. Points with distances exceeding a threshold (95th percentile of the chi-squared distribution with degrees of freedom equal to the latent dimension) were flagged as boundary outliers.

2. **Small Cluster Outliers**: Clusters with fewer than 5% of the total points (i.e., ¡ 59.5 points for 1190 samples) were considered outliers, representing sparse regions in the latent space.

The union of these sets formed the final outlier list.

3.4 Visualization

The 3D latent space (optimal dimension = 3) was visualized directly using a 3D scatter plot, with clusters in distinct colors and outliers highlighted in red. Additionally, a 2D PCA projection was generated for easier interpretation.

4 Implementation Details

The implementation was executed in a Jupyter Notebook using Python, with key libraries:

- `PyTorch`: VAE model definition and training.
- `scikit-learn`: K-means clustering, PCA, and Mahalanobis distance computation.
- `Matplotlib`: Visualization of clusters and outliers.
- `NumPy` and `Pandas`: Data manipulation.

4.1 VAE Architecture and Training

The VAE architecture included:

- **Encoder**: Linear layers (10 → 16 → latent_dim) with ReLU activation.
- **Decoder**: Linear layers (latent_dim → 16 → 10) with ReLU activation.
- **Parameters**:
  - Input dimension: 10
  - Hidden dimension: 16
  - Latent dimensions tested: 2, 3, 4

  - Batch size: 32
  - Learning rate: 1×10^{-3}
  - Optimizer: Adam
  - Scheduler: `ReduceLROnPlateau`
  - Early stopping: Patience=20, Tolerance=1×10^{-4}, Min epochs=500, Max epochs=5000

The loss function was implemented as:

```
def loss_function(x, x_reconstructed, z_mu, z_var):
    recon_loss = F.mse_loss(x_reconstructed, x, reduction='sum')
    kl_loss = 0.5 * torch.sum(torch.exp(z_var) + z_mu**2 - 1.0 -
       z_var)
    return recon_loss + kl_loss
```

4.2 Latent Dimension Selection

Latent dimensions were evaluated:

- Latent Dim 2: Loss = 9660.03
- Latent Dim 3: Loss = 9462.92 (selected)
- Latent Dim 4: Loss = 11425.79

Dimension 3 was chosen for its lowest loss, indicating optimal balance between reconstruction fidelity and regularization.

4.3 K-Means Clustering

K-means was applied to the 1190 latent centers (z_μ) with $n\_clusters = 6$, determined by the elbow method

4.4 Outlier Detection

- **Mahalanobis Distance**: Computed using `scipy.spatial.distance.mahalanobis`, with a threshold based on the 95th percentile of the chi-squared distribution (df=3).
- **Small Clusters**: Clusters with fewer than 59.5 points were flagged.

The code identified 63 outliers, detailed in the output section.

4.5 Visualization

Two visualizations were generated:

- **3D Scatter Plot**: Direct visualization of the 3D latent space.
- **2D PCA Projection**: Reduced to 2D for interpretability (see Figure **??**).

5 Justification of Design Choices and Parameter Selection

This implementation combines a Variational Autoencoder (VAE) with KMeans clustering and Mahalanobis distance-based outlier detection. Below, we outline the rationale for each major design choice and the corresponding parameter selections.

5.1 Data Normalization using `StandardScaler`

Normalization ensures that each feature contributes equally during training. Without it, features with larger numerical ranges would dominate the reconstruction loss.

5.2 VAE Architecture and Parameters

- **Input Dimension (`INPUT_DIM = 10`):** Reflects the number of features in the dataset (`data.csv`).
- **Hidden Dimension (`HIDDEN_DIM = 16`):** Provides a balance between expressiveness and overfitting risk.
- **Latent Dimensions (`latent_dims = [2, 3, 4]`):** Optimal dimension is chosen based on total loss (reconstruction + KL divergence). As K-means can't deal with higher dimensions we are restricting our possible k values to 2, 3 and 4.
- **Learning Rate (`lr = 1e-3`) and Optimizer (Adam):** Chosen as a stable and widely used optimizer for deep learning. Used `ReduceLROnPlateau` scheduler for dynamic adaptation of learning rate.
- **Early Stopping:**
  - `PATIENCE = 20`, `TOLERANCE = 1e-4` To halt training upon convergence.
  - `MIN_EPOCHS = 500`, `MAX_EPOCHS = 5000` provide sufficient training time without excess.

5.3 Latent Space Clustering using KMeans

Optimal Cluster Selection: The elbow method is used to ind the optimal number of clusters. The selected number of clusters captures distinct data modes effectively.

5.4 Outlier Detection Strategy

- **Boundary Outliers using Mahalanobis Distance:**
  - Leverages the covariance matrix of each cluster to detect statistical anomalies.
  - Threshold set at the 95th percentile—top 5% farthest points are flagged.
  - Unlike Euclidean distance, it considers feature correlations for a more accurate anomaly measure.

- **Small Cluster Outliers:**
  - Clusters with fewer than 5% of total points are flagged as anomalous.
  - Such clusters likely represent noise or rare patterns.
- **Final Outliers:** Defined as the union of Mahalanobis and small-cluster-based anomalies to capture both boundary and isolated anomalies.

5.5 Visualization and Interpretability

- **3D Scatter Plot (for `LATENT_DIM = 3`) and 2D PCA Projections:**
  - Enable visual validation of clustering quality.
  - Ellipse overlays represent cluster boundaries and aid in interpretability.

6 Results

The methodology identified 63 outliers from 1190 samples (approximately 5.3% of the dataset). Key findings include:

- **Latent Space Structure**: The 3D latent space, clustered into 6 groups, showed distinct separation, validated by the low loss of 9462.92.
- **Outlier Characteristics**: Outliers included points with extreme feature values (e.g., `sal_pur_rat` = 34.367 at index 591) and those in sparse regions.
- **Visualization**: The 2D PCA projection and 3D plot confirmed cluster separation, with outliers either on boundaries (high Mahalanobis distance) or in small clusters.

6.1 Outlier Data Points

The following link consists of table with 63 outlier data points represented in their original dimensions.

Click here to access the spreadsheet: Google Spreadsheet Link

7 Conclusion and Summary

This study successfully identified 63 outliers in a 10-feature dataset of 1190 samples by projecting the data into a 3-dimensional latent space using a VAE, clustering the latent centers with K-means ($k = 6$), and detecting outliers via Mahalanobis distance and cluster size thresholds. The latent dimension of 3 was justified by its superior loss performance (9462.92), balancing reconstruction and regularization, while $k = 6$ was supported by the elbow method, ensuring meaningful cluster separation.

The use of Mahalanobis distance for boundary detection proved effective, capturing multivariate outliers by considering covariance, unlike simpler methods like IQR. The small cluster criterion (5%) complemented this by identifying sparse regions, yielding a comprehensive outlier set. The visualization confirmed the approach's ability identify outliers which often exhibiting extreme feature values or isolation in the latent space.