Contraction-Expansion Model

Saurav Jadhav

June 2025

1. Contraction-Expansion Classification Framework

1.1 Motivation and Theoretical Foundation

The original m-neighbor classification model utilizes a *contraction-based* scoring mechanism to reject observations that are likely to be outliers. This is done using a **contamination parameter** γ, which defines a class-specific rejection threshold by selecting the $(1-\gamma)$-th quantile of a deviation score g derived from stickiness and boundary properties.

However, we observed a limitation: the model only contracts the decision boundary, without the ability to **expand** it when required. This restricts the model's utility in scenarios where slightly anomalous but valid observations should be classified, not rejected.

To address this, we introduce an **expansion mechanism** to the thresholding strategy. This approach retains the original structure but modifies how thresholds are computed. It offers flexibility to either shrink (contract) or grow (expand) the classification boundaries based on data-driven deviation scores.

1.2 Deviation Scoring Function

Let x_{ij} be an observation in class C_i. Define:

- s_{ij}: mean distance to the m-nearest neighbors in C_i (empirical stickiness)
- b_{ij}: mean distance to all other points in C_i (empirical boundary)
- $S_i = \text{median}(s_{ij})$ across $x_{ij} \in C_i$
- $B_i = \text{median}(b_{ij})$ across $x_{ij} \in C_i$

The deviation score is defined as:

$$g_{ij} = \lambda \cdot \frac{s_{ij} - S_i}{S_i} + (1 - \lambda) \cdot \frac{b_{ij} - B_i}{B_i}$$

where $\lambda \in (0, 1)$ controls the relative weight of stickiness and boundary components.

1.3 Contraction vs Expansion Thresholding

To determine the threshold τ_i for class C_i:

- **Contraction:**

$$\tau_i = \text{quantile}_{1-\gamma}(g_i)$$

  where g_i is the list of all g_{ij} scores for class C_i.

- **Expansion:**

$$\tau_i = \max(g_i) + \text{quantile}_{\gamma}(g_i)$$

  This enlarges the classification region by adding a small quantile-based margin to the maximum score.

This formulation provides a flexible and robust strategy for real-world settings, where both tight rejection and boundary expansion may be desirable.

1.4 Algorithmic Implementation

Algorithm 1 Model Training: Class Characterization and Threshold Determination (with Contraction or Expansion)

Require: Training data $\mathcal{D} = \{(x_{ij}, y_j)\}$ for $j = 1, \ldots, n_i$, $i = 1, \ldots, k$; parameters m, $\lambda \in (0, 1)$, rate $\gamma \in [0, 1)$, threshold type $\in \{\text{contraction}, \text{expansion}\}$

Ensure: Class characteristics $\{S_i, B_i\}_{i=1}^{k}$, thresholds $\{\tau_i\}_{i=1}^{k}$

for each class $C_i \in \{C_1, \ldots, C_k\}$ **do**
  Let $X_i = \{x_{i1}, \ldots, x_{in_i}\}$
  Initialize lists $L_s^i \leftarrow [\,]$, $L_b^i \leftarrow [\,]$
  for each observation $x_{ij} \in X_i$ **do**
    Compute stickiness $s_{ij} = \frac{1}{m} \sum_{\ell=1}^{m} d_{ij}(\ell)$
    Append s_{ij} to L_s^i
    Compute boundary $b_{ij} = \frac{1}{n_i - 1} \sum_{\substack{x_{i\ell} \in X_i \\ \ell \neq j}} f(x_{ij}, x_{i\ell})$
    Append b_{ij} to L_b^i
  end for
  $S_i \leftarrow \text{median}(L_s^i)$
  $B_i \leftarrow \text{median}(L_b^i)$
  Initialize $L_g^i \leftarrow [\,]$
  for each $x_{ij} \in X_i$ **do**
    $g_{ij} \leftarrow \lambda \cdot \frac{s_{ij} - S_i}{S_i} + (1 - \lambda) \cdot \frac{b_{ij} - B_i}{B_i}$
    Append g_{ij} to L_g^i
  end for
  if threshold type is **contraction then**
    $\tau_i \leftarrow (1 - \gamma)$-quantile of L_g^i
  else if threshold type is **expansion then**
    $\tau_i \leftarrow \max(L_g^i) + \text{quantile}_{\gamma}(L_g^i)$
  end if
end for
return $\{S_i, B_i\}_{i=1}^{k}$, $\{\tau_i\}_{i=1}^{k}$

Algorithm 2 Prediction for a New Observation (with Contraction or Expansion)

Require: New observation x_0; trained model $\{S_i, B_i, \tau_i\}_{i=1}^k$; training data $\{X_i\}_{i=1}^k$; parameters m, λ

Ensure: Predicted label for x_0 or "Outlier"

Initialize $S\_scores \leftarrow [\,]$
for each class $C_i \in \{C_1, \ldots, C_k\}$ **do**
  Compute stickiness: $s_{i0} = \frac{1}{m} \sum_{\ell=1}^{m} d_{i0}(\ell)$
  Compute boundary: $b_{i0} = \frac{1}{n_i} \sum_{x_{i\ell} \in X_i} f(x_0, x_{i\ell})$
  $g_{i0} = \lambda \cdot \frac{s_{i0} - S_i}{S_i} + (1 - \lambda) \cdot \frac{b_{i0} - B_i}{B_i}$
  Append (g_{i0}, i) to S_scores
end for
Initialize $P \leftarrow \emptyset$
for each $(g_{m0}, m) \in S\_scores$ **do**
  if $g_{m0} \leq \tau_m$ **then**
    Add m to P
  end if
end for
if $P = \emptyset$ **then**
  return "Outlier"
else
  $p^* \leftarrow \arg\min_{m \in P} g_{m0}$
  return class C_{p^*}
end if