HDSCLU+
Finding Clusters of Unknown Dimension in High-dimensional Data Streams

Katelyn Gao
Stanford University

August 18, 2015
Table of Contents

1 Introduction

2 Concepts

3 Algorithm

4 Results

5 Conclusion
Streaming Data

What is streaming data?
- continually coming
- infinite
- dynamic

It's everywhere:
- E-commerce
- Finance
- Social media
- Network traffic
Introduction

Challenges

Of streaming data analysis:
- One pass
- Limited computation resources (CPU and memory)
- Concept drift

Of high-dimensional data:
- Curse of dimensionality!
HDSCLU is an online algorithm that
- clusters high-dimensional streaming data
- is robust to outliers and misassignments
- recognizes clusters in arbitrary subspaces

HDSCLU+ modifies HDSCLU to
- Learn dimensions of clusters

Applications:
- Market research
- Anomaly detection
- and many more
Table of Contents

1 Introduction
2 Concepts
3 Algorithm
4 Results
5 Conclusion
Clusters in Arbitrary Subspaces: An Example
**Subspace of cluster $k$:** Span of the $d_k$ smallest eigenvectors of its covariance matrix.

**Radius of cluster $k$:** Square root of the $d_k$ smallest eigenvalue of its covariance matrix.

**Projected distance from point $x$ to cluster $k$:** Euclidean distance between $x$ and the center of cluster $k$, after both are projected into that cluster’s subspace.

**Adjusted projected distance from point $x$ to cluster $k$:** Projected distance from point $x$ to cluster $k$ divided by the square root of the dimension of cluster $k$, $d_k$. 
# Table of Contents

1. Introduction

2. Concepts

3. Algorithm

4. Results

5. Conclusion
Algorithm Structure

- Assign to a Cluster
- Update Cluster Summaries
Assigning a Point

- Compute the adjusted projected distances from the point to each cluster, and label the cluster with the smallest distance as the closest cluster.
- If the projected distance from the point to the closest cluster is less than three times that cluster’s radius, assign it to that cluster.
- If not, the point forms a new cluster.
After a point $x$ is assigned to a cluster $k$:

- Age is set to 0.
- The weight of $x$, $w_x$, is set to $e^{-\ell \beta}$, where $\ell$ is the projected distance from $x$ to cluster $k$.
- Weight is scaled by $2^{-\lambda t}$, where $t$ is the time between the current and previous data points, and then incremented by $w_x$.
- New center and covariance matrix are computed using incremental algorithms.
- Eigendecomposition of covariance matrix are computed using rank-one update formulas.
- Dimension, subspace and radii are recomputed as described next.
Learning Cluster Dimensions

Goal: Find the number of small eigenvalues of the covariance matrix.

Modified Scree Test

- Adapted from the Scree Test of factor analysis, where the number of factors is the number of large eigenvalues of the data covariance matrix.

- Scale the computed eigenvalues so that the largest equals 1: \( \lambda_1 \geq \ldots \geq \lambda_p \geq 0 \).

- Compute the second differences
  \( \Delta_i = (\lambda_i - \lambda_{i+1}) - (\lambda_{i+1} - \lambda_{i+2}) \).

- Find the smallest \( j \) such that \( \Delta_{j+1}, \Delta_{j+2}, \ldots \) is less than \( \delta \).

- Return \( p - j \).
Modified Scree Test: Example
Updating Other Clusters

Because of the passage of time, the clusters other than the one the current point is assigned to must be updated:

- Age is incremented by $t$.
- The weight is scaled by $2^{-\lambda t}$.
- Delete a cluster if the ratio of weight to age is less than $\epsilon$. 
Table of Contents

1 Introduction

2 Concepts

3 Algorithm

4 Results

5 Conclusion
Simulation Setup

- Mixture of four 50-dimensional Gaussians of dimensions 10, 15, 20, and 25.
- Every coordinate of the means $\sim U[-3, 3]$.
- Covariance matrix of Gaussian of dimension $d$ is $QDQ^T$, where $Q$ is a random orthogonal matrix and $D$ is a diagonal matrix with $d$ coordinates chosen independently from $U[0, 0.5]$ and the rest from $U[1, 2]$.
- One percent of data comes from a Gaussian with dimension 20 - the outliers.
Results

Slow Concept Drift

Figure: $K = 150, \lambda = 0.001, \beta = 0.01, \epsilon = 0.02$
Sudden Concept Drift

Figure:  $K = 180, \lambda = 0.001, \beta = 0.01, \epsilon = 0.01$
Table of Contents

1 Introduction
2 Concepts
3 Algorithm
4 Results
5 Conclusion
Conclusion and Future Work

- We proposed an online algorithm to find clusters from streaming data, based on HDSCLU.
- It learns the dimensions of clusters.
- On simulated data with concept drift, the performance is variable; difficulties arise when the distribution changes.
- Future work: visualization of cluster evolution
Acknowledgements

This work was supported by NSF Grant DGE-114747.

Thank You For Listening!