



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Kalman-Enhanced Streaming Machine Learning for Real-Time Land Use Classification in Satellite Imagery

LAUREA MAGISTRALE IN MATHEMATICAL ENGINEERING - INGEGNERIA MATEMATICA

Author: NICOLA FRANCESCON

Advisor: PROF. EMANUELE DELLA VALLE

Co-advisor: GIACOMO ZIFFER

Academic year: 2023-2024

1. Introduction

The rapid escalation in data generation has shifted Machine Learning toward the management of continuous, unbounded data streams, establishing the field of Streaming Machine Learning (SML) [1]. This thesis investigates the application of SML to satellite image classification, a domain in which real-time analysis is crucial for effective decision-making in areas such as environmental monitoring and disaster response. Unlike traditional batch processing, SML enables models to adapt as new data arrives, ensuring continued relevance even as data distributions evolve. To address the computational demands posed by high-dimensional satellite data, this research introduces an optimized processing pipeline that compares different SML classifiers. Additionally, advanced dimensionality reduction techniques are incorporated to reduce the computational load without sacrificing accuracy.

This work addresses the classification of temporally correlated satellite image streams, which evolve over time. The **main contributions** provided by this study are:

- Kalman filter adaptation to improve the Streaming Linear Discriminant Analysis

(SLDA) model in presence of concept drifts.

- A streaming version of UMAP, an offline dimensionality reduction technique, to handle large-scale, evolving datasets efficiently.
- A detailed performance comparison of dimensionality reduction techniques in streaming settings with an extended experimental campaign using different classification algorithms.
- Experimental validation and comparison of the new methods with state-of-the-art models to demonstrate effectiveness.

2. Related Works

2.1. Streaming Machine Learning

SML is a branch of Machine Learning that focuses on learning from continuous, potentially unbounded data streams. Unlike traditional batch learning, which relies on access to an entire dataset from the outset, SML refines its parameters incrementally with each new observation. A central challenge in SML is **Concept Drift**, a change in the data distribution over time that can impact the accuracy of models unless addressed by adaptive learning mechanisms. Currently, SML algorithms have demon-

strated efficient learning performance with low-dimensional data, such as tabular datasets with a limited number of features. However, processing high-dimensional data, such as raw images containing hundreds of thousands of features, presents additional computational challenges. To manage this, a preliminary feature extraction step is often necessary, typically performed via **Convolutional Neural Networks**. This step reduces the dimensionality of the data, making it feasible for real-time analysis. Once transformed into a more manageable feature representation, each processed sample can serve as input to an SML classifier.

Three commonly used SML classifiers are **Gaussian Naive Bayes (GNB)**, **Softmax Regression (SMR)**, and **Streaming Linear Discriminant Analysis (SLDA)** [2]. GNB is a probability-based classifier based on Bayes' theorem, assuming conditional independence among features. SMR is a multi-class classifier that estimates class probabilities, assuming linear separability among classes. SLDA builds upon the offline Linear Discriminant Analysis (LDA) classifier by maintaining a mean vector for each class and a shared covariance matrix, both updated as new data becomes available.

Each of these classifiers has unique properties and computational demands: SLDA, for instance, requires the inversion of a $d \times d$ matrix, where d represents the feature dimensionality, making it the most computationally intensive of the three.

2.2. Dimensionality Reduction Techniques

To improve the computational efficiency of streaming classifiers, dimensionality reduction techniques are commonly employed as an intermediate step. These techniques aim to retain the most informative aspects of each sample, reducing its feature vector size. The guiding principle of dimensionality reduction in classification tasks is **data similarity**: ideally, similar samples should maintain similar labels in the reduced feature space.

Sparse Random Projection (SRP) [3] is a computationally efficient method for dimensionality reduction that achieves this by employing matrix-vector multiplications. This method offers a feasible approach for high-dimensional

data, preserving the overall structure of the data while significantly reducing its dimensionality for faster processing in streaming contexts. However, SRP may suffer from approximation errors and may not adequately capture complex structures or non-linear relationships in the data. Consequently, similar points in the original space are not guaranteed to maintain their proximity in the reduced dimensionality.

Uniform Manifold Approximation and Projection (UMAP) [4] is a non-linear dimensionality reduction technique designed to preserve both local and global structures in data, with a particular emphasis on maintaining data similarity. UMAP operates through a series of steps that introduce a force-directed layout, optimizing the representation of data in a reduced-dimensional space by maintaining the neighborhood relationships from the original high-dimensional space. However, this method currently has limitations in streaming contexts: it has primarily been developed for offline use and has not yet been adapted for real-time streaming environments. Additionally, UMAP can require substantial memory resources when applied to complex datasets, posing challenges for real-time applications.

2.3. Evaluation of the Results

The models are evaluated using **prequential evaluation**, an evaluation protocol where each model predicts the class of an incoming sample and subsequently updates its parameters once the true class label is revealed, introducing a continuous evaluation process.

Two performance metrics are employed to assess the models' effectiveness in a multiclass setting. **Accuracy** provides a general measure of overall correctness by calculating the ratio of correctly predicted instances to the total number of instances. Accuracy is useful for understanding general performance but can be less informative for imbalanced data, as it may favor majority classes. **Balanced Accuracy** is introduced to account for class imbalance, providing a more precise view of model performance across all classes. Balanced accuracy gives equal weight to each class by computing the average recall across classes, mitigating the influence of dominant classes in the dataset.

3. Problem Statement

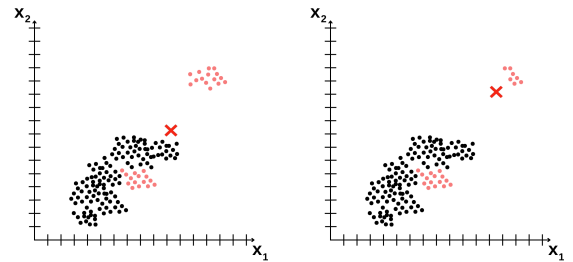
The goal of this thesis is to develop an efficient and scalable solution for satellite image classification using SML methods. The main research questions are the following:

- **RQ1:** Can the current limitations of the state-of-the-art SLDA classifier be improved to efficiently **handle concept drift** in streaming data and maintain performance on continuously updating data?
- **RQ2:** Is it possible to develop a **novel streaming dimensionality reduction approach** that incorporates data similarity, inspired by existing offline methods?
- **RQ3:** Which SML method provides the **optimal trade-off** between classification performance and computational efficiency for evolving satellite images?
- **RQ4:** How do the proposed methods perform compared to **state-of-the-art techniques** in terms of performance and scalability when applied to the context of temporally correlated satellite images?

4. Problem Solving

4.1. The Dataset

The dataset analyzed in this thesis originates from the *Functional Map of the World* (FMoW), initially released in 2018 and updated in 2021 as FMoW-Time [5]. This revised dataset presents a simplified classification task with 126,165 RGB images across 62 land-use categories. Each sample, sized at 224×224 pixels, includes three RGB color channels and metadata specifying the land use label and year of observation, although observations are unordered within each year. The 62 categories exhibit class imbalance, with observations for a specific class ranging from 77 to over 10,000 samples in the entire dataset. Categories span a variety of land uses, such as airports, agricultural fields, and urban facilities. Some methods in this thesis require an offline initialization phase. For this purpose, samples from the first six years, comprising 12,874 data points, will be merged in a single batch and used exclusively for initializing models that require it, while the remaining 113,291 samples will serve to evaluate model performance. This separation ensures consistent performance assessment across all models.



(a) The class mean of the red class, marked with a cross, remains far from converging to the new distribution (b) In Kalman-SLDA, the mean updates rapidly to follow concept drift

Figure 1: Comparison of concept drift adaptation in SLDA and Kalman-SLDA for a bidimensional dataset.

4.2. Kalman-SLDA

The original SLDA algorithm struggles to handle concept drift effectively. In a stable scenario where all classes are linearly separable, observing a new data point in a different region assigns a weight of $\frac{1}{N+1}$ to this sample, where N represents the number of previously observed samples for the same class of the new observation. This weighting helps keep the sample mean closer to the original region, providing robustness against outliers. However, when concept drift occurs and new points are observed in the new region, the sample mean is slow to shift. If N points are observed in both the initial and new regions, the sample mean will lie between the two, requiring more than N points in the new region to fully shift to the new distribution. A visual example is shown in Figure 1a. Consequently, SLDA may underperform in classification accuracy when concept drift occurs.

To address this limitation, Kalman filtering [6] is integrated into the original SLDA model to update the mean vector and covariance matrix dynamically. This integration preserves SLDA’s structure while enhancing its responsiveness to concept drift. The Kalman filter adaptively adjusts the weight of each new sample based on previous predictions and observed outcomes, refining the mean and covariance updates over time. The impact of this approach is illustrated in Figure 1b, where, after N observations in the initial region and fewer than N in the new re-

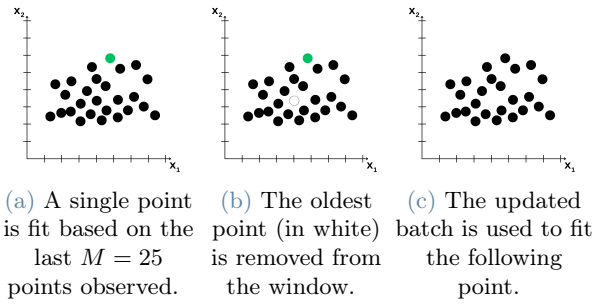


Figure 2: Streaming-UMAP implementation for a bidimensional dataset.

gion, the sample mean has shifted toward the new distribution.

The filter’s ability to assign appropriate weights based on past prediction accuracy refines classification performance, remaining sensitive to changes in the data distribution and improving the robustness in dynamic environments.

4.3. Streaming-UMAP and Batch-UMAP

UMAP is a dimensionality reduction technique that preserves both local and global structures in high-dimensional data, but it was initially designed to operate offline, requiring access to the entire dataset at once. This design limits its suitability for real-time applications, where data arrives continuously, necessitating an adaptation of UMAP to handle streaming data.

Streaming-UMAP manages continuous data by using a sliding window to update the embedding incrementally as new data arrives. Rather than embedding the entire dataset at once, it

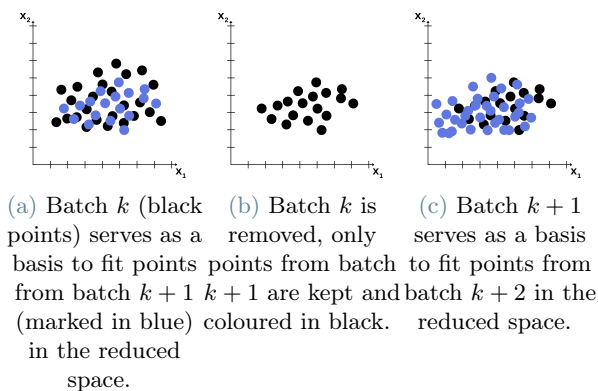


Figure 3: Iterative fitting of Batch-UMAP for a bidimensional dataset.

processes only the most recent M observations. As each new data point enters the window, the oldest one exits, shifting the window forward over time. Figure 2 illustrates this method.

Batch-UMAP is suited for data arriving in intervals or batches. Rather than updating the embedding with each new data point, Batch-UMAP processes data in groups, refining the embedding after each batch. Data from batch k are embedded using batch $k - 1$ ’s embedding as a reference. Once batch k is processed, batch $k - 1$ is discarded, ensuring embedding consistency while minimizing computational load. Figure 3 presents this approach.

Batch-UMAP is ideal for structured interval data, while Streaming-UMAP is suitable in real-time contexts. Together, these methods extend UMAP’s applicability in dynamic environments, enabling flexible dimensionality reduction for both streaming and batch processing scenarios.

5. Experiments and Results

5.1. Experimental Setup

All simulations are conducted on a virtual machine with 100 GB of storage, 16 GB of RAM, and an 8-core processor to enable parallelization. The experimental pipeline is fully replicable, with random seeds specified at each step to ensure consistency.¹

¹<https://github.com/NicolaFrancescon/Thesis>

Classifier	Time per Sample (seconds)
Kalman-SLDA	0.07
SLDA	0.07
GNB	0.05
SMR	0.04

Table 1: Time required to classify the dataset using different SML classifiers.

Classifier	Accuracy	Balanced Accuracy
Kalman-SLDA	0.30	0.28
SLDA	0.25	0.27
GNB	0.21	0.22
SMR	0.25	0.13

Table 2: Average classification performance for different SML classifiers.

5.2. Results Analysis

The analysis of results focuses on the trade-off between execution time and classification performance across the various classifiers and dimensionality reduction techniques.

Initially, models are assessed without dimensionality reduction. As shown in **Table 1**, GNB and SMR are faster than SLDA and Kalman-SLDA, with Kalman-SLDA demonstrating comparable execution time to the original SLDA. **Table 2** presents the classification performance metrics. Kalman-SLDA outperforms the other models, demonstrating to be a consistent method for satellite image classification. Notably, SMR shows a discrepancy between the metrics, indicating a focus on the most frequent classes instead of a more balanced performance.

Dimensionality reduction is introduced as an intermediate step to reduce computational overhead on large feature vectors, aiming to maintain the most informative features while discarding others, thus reducing execution time. **Table 3** shows execution times (including dimensionality reduction time) for each classifier when using SRP to halve feature dimensions. The pipeline now executes more quickly, although with a performance reduction in most cases, as shown in **Table 4**. SMR continues to struggle with class imbalance, while GNB demonstrates

Classifier	Time per Sample (seconds)
Kalman-SLDA	0.04
SLDA	0.04
GNB	0.03
SMR	0.03

Table 3: Time required to classify the dataset using different SML classifiers with half-sized features through SRP.

Classifier	Accuracy	Balanced Accuracy
Kalman-SLDA	0.20	0.20
SLDA	0.16	0.18
GNB	0.21	0.22
SMR	0.19	0.08

Table 4: Average classification performance for different SML classifiers with half-sized features through SRP.

Classifier	Time per Sample (seconds)
Kalman-SLDA	0.05
SLDA	0.05
GNB	0.06
SMR	0.05

Table 5: Time required to classify the dataset using different SML classifiers with half-sized features through Batch-UMAP.

Classifier	Accuracy	Balanced Accuracy
Kalman-SLDA	0.16	0.15
SLDA	0.06	0.06
GNB	0.16	0.16
SMR	0.18	0.09

Table 6: Average classification performance for different SML classifiers with half-sized features through Batch-UMAP.

the best results on this features; however, it is not able to reach the performance of Kalman-SLDA on unreduced features.

Table 5 reports execution times (including dimensionality reduction time) for each classifier when using Batch-UMAP to halve feature dimensions. Execution times are generally higher than those for SRP (Table 3), and in some cases exceed the times observed without dimensionality reduction (Table 1). **Table 6** shows average performance after Batch-UMAP reduction. All models experience a performance decline compared to both unreduced features (Table 2) and SRP-reduced features (Table 4). SLDA suffers particularly significant performance loss, suggesting that Batch-UMAP may introduce concept drift not present in the original dataset. Despite an overall decline, Kalman-SLDA retains stability, supporting its efficacy in dynamic settings. Batch-UMAP current implementation requires further refinement. Although it effectively captures data similarity, it does not match the results of SRP, remaining a promising alternative that needs optimization.

All results presented in this section were validated using feature extraction from two distinct CNNs, specifically MobileNet v3 Small and ResNet18. Different CNNs were selected to assess the robustness of the models to varying extracted feature sets. Despite slight differences

in sizes, each CNN yielded comparable results across classifiers and dimensionality reduction methods, underscoring the adaptability of the proposed pipeline. This consistency suggests that the models are enough versatile to perform well under different feature representations.

6. Conclusions and Future Work

The primary objective of this study was to develop an optimal pipeline for classifying satellite image streams, focusing on both accuracy and computational performance. Key findings from the experimental campaign are summarized as follows:

- **RQ1:** Kalman-SLDA effectively overcomes the limitations of SLDA in handling concept drift without increasing computational requirements. This enhancement establishes Kalman-SLDA as a more efficient model for high-dimensional classification tasks with high label fragmentation, particularly suitable for evolving data streams where concept drift needs to be managed.
- **RQ2:** Batch-UMAP is presented as an innovative dimensionality reduction approach for SML; however, it underperformed relative to the state-of-the-art technique of SRP, despite integrating data similarity.
- **RQ3:** Kalman-SLDA emerges as the optimal choice among the evaluated SML classifiers. Dimensionality reduction improves the computational efficiency of classifiers in terms of speed, yet it also leads to a decline in overall classification performance.
- **RQ4:** Kalman-SLDA demonstrates strong relevance as an SML classifier, showing improvements over SLDA without increasing computational demands. Batch-UMAP does not meet expectations in comparison to SRP, suggesting that it needs refinement to be competitive. Furthermore, SMR reveals limitations in handling class imbalance with a large number of classes, typically favoring majority classes while neglecting minority ones.

Future exploration may involve significant improvements in the scalability, efficiency, and accuracy of dimensionality reduction and classification algorithms for dynamic and high-dimensional datasets.

In the current experimental setup, attempts to apply Streaming-UMAP faced computational constraints. Specialized hardware configurations capable of parallelizing critical stages in UMAP’s algorithm, such as graph construction and neighbor-finding, may help address this. Another future challenge involves optimizing UMAP for evolving data distributions, specifically through Batch-UMAP hyperparameter tuning. Expanding the scope of UMAP’s adaptability to evolving environments could open up new applications in fields requiring continuous data integration. Additionally, future work could investigate the impact of merging labels with similar land-use characteristics to enhance model performance, a modification that can benefit classifiers like SMR.

References

- [1] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama. Machine learning for streaming data: state of the art, challenges, and opportunities. *SIGKDD Explor.*, 21(2):6–22, 2019.
- [2] T. L. Hayes and C. Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *CVPR Workshops*, pages 887–896. Computer Vision Foundation / IEEE, 2020.
- [3] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD*, pages 245–250. ACM, 2001.
- [4] L. McInnes and J. Healy. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018.
- [5] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. Koh, and C. Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *NeurIPS*, 2022.
- [6] G. Welch. An introduction to the kalman filter. In *International Conference on Computer Graphics and Interactive Techniques*, 1995.