METAM Algorithm Implementation and Analysis

as proposed in the paper “Goal-Oriented Data Discovery” [Sainyam Galhotra et al.]

Sofiya Shtetenson and Timna Smadja

GitHub

Streamlit App

1 Introduction

In modern data-driven decision-making, the availability of large and diverse datasets presents both an opportunity and a challenge. While more data can potentially improve predictive performance, integrating heterogeneous data sources in a meaningful way is nontrivial. In this work, we implement and evaluate METAM (Minimal Essential Task Augmentation Mechanism)—a goal-oriented data discovery algorithm that iteratively identifies and joins external candidate datasets to improve a downstream task’s performance (for example - classification accuracy). By selectively augmenting the core dataset with relevant features, METAM automates what data scientists often do manually—searching for beneficial external data, joining it, and then observing performance gains.

2 Background

Traditional data augmentation methods often rely on manual feature engineering or pre-defined heuristics to select supplementary datasets. However, these approaches can be inefficient and may fail to capture the true incremental value of additional data. The METAM algorithm, as described in “METAM: Goal-Oriented Data Discovery” by Sainyam Galhotra et al., introduces a feedback loop in which candidate augmentations are evaluated based on their effect on a downstream utility function. Key aspects of the approach include:

- Data Profiles: Each candidate augmentation is characterized by task-independent measures such as join key overlap, data completeness, and feature richness.
- Adaptive Querying: The algorithm alternates between evaluating individual candidates and groups (via clustering) to efficiently explore the augmentation space.
- Minimal Augmentation Set: Augmentations are only incorporated if they yield a significant improvement in the target task’s utility, ensuring that the final augmented dataset is both parsimonious and effective.

3 Methodology Overview

3.1 Problem Formulation

Given a base dataset D_{base} and a repository of candidate datasets $\{D_1, D_2, ..., D_n\}$, the goal is to select a minimal subset $T \subseteq \{D_i\}$ such that the augmented dataset $\Gamma(D_{base}, T)$ achieves a utility $u(\Gamma(Dbase, T)) \geq \theta$, where θ is a predetermined threshold (e.g., based on accuracy for classification tasks).

3.2 Algorithm Overview

The METAM algorithm proceeds as follows:

1. Candidate Generation:

   Each candidate augmentation is processed to compute a vector of data profiles that includes:

   (a) Join Key Overlap: The ratio of common join keys between the base dataset and the candidate.

   (b) Completeness: 1 minus the missing rate in the candidate's join key.

   (c) Normalized Column Count: The number of columns in the candidate dataset divided by a normalization constant.

   (d) Semantic Similarity: A measure (derived from cosine similarity of SentenceTransformer embeddings) of how semantically similar the candidate's column names are to those in the base dataset.

   An initial quality score is then assigned as the mean of these profile values.

2. Clustering:

   Candidates are clustered based on the similarity of their profile vectors. This step reduces redundant evaluations by grouping together augmentations that are expected to yield similar effects on the utility function.

3. Adaptive Querying:

   The algorithm iteratively queries the utility function by merging candidate augmentations with the base dataset. During each iteration, candidates are evaluated both individually and as part of groups. Only those augmentations that yield a measurable improvement in utility (exceeding a specified threshold □) are selected.

4. Minimality Check:

   After candidate selection, a minimality check is performed by attempting to remove each selected augmentation. If removing an augmentation does not reduce the overall utility below the threshold □, it is deemed redundant and is removed from the final augmentation set.

3.3 Implementation Details

Our implementation of METAM is in Python and integrates a Streamlit UI for interactive experimentation.

Key implementation details include:

- Utility Function:

  The utility function is computed using an 80/20 train-test split. For classification tasks, logistic regression accuracy is used as the performance metric.

  For regression tasks, Linear Regression's R² score is employed. Non-numeric features and missing values are handled via filtering and imputation.

- Streamlit UI:

  The UI is written in python and built using Streamlit. It allows the user to select from the experiments detailed in section 4 and run the algorithms on them. The UI displays candidate augmentation details—including their computed profiles and quality scores—in a table, and presents the final results. In Addition, the user can adjust the threshold θ and see how it affects the results of the experiments.

4 Experimental Setup

1. Feature Engineering: Classification - Expensive Housing in Boston - Using Partitioned Features

   In this experiment, we adopt a novel approach by taking a single, feature□rich dataset (the Boston Housing dataset) and partitioning it into distinct feature subjects. Specifically, we split the dataset into three groups, then use a reduced version of the dataset as our base. The candidate augmentations, which are the partitioned feature groups, are then merged back into the base data using METAM.

   This experimental setting is significant for the task of feature engineering, in the feature reduction perspective.

   This "reduction" capability highlights METAM's ability to achieve a minimal yet effective augmentation set, a concept discussed in the original paper in the context of minimality of the augmentation set. While the paper primarily focuses on goal-oriented data discovery, **our experiment demonstrates that the same framework can be leveraged to reduce redundancy and perform implicit feature selection**

   - Base Set: 'RM' (average number of rooms per dwelling) and 'LSTAT' (percentage of lower status of the population)—along with a dummy join key "Id" and a binary target "expensive" (1 if MEDV is at or above the median, 0 otherwise).
   - Task: predicting the likelihood of housing to be expensive (greater than the median)
   - Augementation candidates: The remaining features were grouped into three candidate subjects:

  - Crime/Industrial Features (CRIM, INDUS, NOX, AGE, DIS.)
  - Zoning/Tax Features) (ZN, RAD, TAX, PTRATIO.)
  - Structural/Demographic Features (CHAS,B)
- Results: Selected augmentation: Zoning/Tax Features
- Explanation:

  Initial Base Utility: The base model using only ’RM’ and ’LSTAT’ achieved an accuracy of 81.37%. Iteration 1: Merging Candidate 1 resulted in a utility of 84.31% (gain of 2.94%). In the grouping step, Candidate 2 from the identified group yielded a higher utility of 85.29%. The algorithm selected Candidate 2 (Zoning/Tax Features), which improved the base utility from 81.37% to 85.29%. A minimality check confirmed that removing this augmentation dropped the utility back to 81.37%, verifying its contribution.

  Final Performance: 85.29% accuracy, with "Zoning/Tax Features" selected as the effective augmentation.
- Conclusion: This experiment demonstrates that when a rich dataset is partitioned into distinct feature groups, METAM can be used for feature reduction—effectively identifying which groups add incremental predictive power and which are redundant. In this case, even though all candidate profiles showed high overlap and data completeness, the algorithm selected the "Zoning/Tax Features" augmentation because it provided a measurable improvement in accuracy. This result aligns with the goal-oriented nature of METAM as described in the paper, emphasizing the minimality of the augmentation set: augmentations are incorporated only when they significantly enhance the utility of the model.

2. The augmentation set is minimal: Expensive Housing Classification in Seattle

   - Base set: seattle_housing_prices
   - Augementation candidates: seattle_incomes, seattle_pet_licenses, seattle_crime_rates,
   - Task: predicting the likelihood of housing to be expensive (greater than the medain) based on zip codes
   - Result: no augmentation was selected
   - Explanation: The base model already achieved a very high predictive utility (98.02% accuracy).

     We computed candidate profiles that measured the overlap of join keys, data completeness, and relative feature richness. For instance, the crime rate candidate had a high overlap (0.93) and a quality score of 0.65. Despite these favorable attributes, when we merged the candidate into the base dataset, the observed gain in utility was negligible.

     As we can see, because the base dataset already yielded near-optimal performance, the additional features did not increase the utility sufficiently. Therefore, the algorithm correctly concluded that further augmentation was unnecessary and did not select any candidate.

This behavior is consistent with the discussion in the paper on the monotonicity of the utility function and the minimality of the augmentation set: when the base data already approaches optimal predictive performance, additional augmentations may be redundant.

3. Goal-oriented nature of METAM - High Cat Ratio Classification

   - Base set: seattle_pet_licenses (containing only “zipcode” and a weak predictor, “total_pets”)
   - Augementation candidates: seattle_incomes, seattle_housing_prices, seattle_crime_rates
   - Task: predicting the likelihood of having a high proportion of cats per zipcode
   - Result: selected augmentation: seattle_incomes
   - Explanation:

     In this experiment we used exactly the same dataset as in the previous example, but for a different task. The base dataset produced a modest utility of 0.6216. By incorporating the seattle_incomes dataset as a candidate augmentation METAM detected a measurable gain, improving the utility to 0.6410 and thus selected the augmentation.

   These two examples contrast highlights the goal-oriented nature of METAM: when the base dataset is already strong, additional features are unnecessary, but when the base is weak, even modest improvements from candidate augmentations are recognized and selected. This behavior aligns with the paper’s discussion on optimizing query efficiency and achieving a minimal yet effective augmentation set.

4. Goal-oriented nature of METAM: Housing Price in Seattle Regression

   - Base set: seattle_housing_prices
   - Augementation candidates: seattle_incomes, seattle_pet_licenses, seattle_crime_rates
   - Task: predicting the actual housing price (a continuous variable) based on zip codes
   - Result: selected augmentation: Crime Rate Data
   - Explanation: The Crime Rate Data showed a moderate overall quality score (≈0.515), which was the lowest of all candidates, yet was the only candidate to yield a measurable improvement, raising the utility from 0.6562 to 0.6595.

5 Conclusion

This report demonstrates the versatility and effectiveness of METAM's goal-oriented data augmentation approach across diverse scenarios. Our experiments reveal several important insights:

- Selective Augmentation: In cases where the base dataset is already highly predictive, such as in experiment 2, METAM correctly identifies that additional augmentation would be redundant. The algorithm's minimality check ensures that only augmentations yielding a measurable improvement are integrated into the final dataset. This not only prevents overfitting by avoiding unnecessary complexity but also reinforces the notion that a well-engineered base dataset can sometimes be sufficient for achieving high performance.

- Sensitivity to Modest Gains: In scenarios with a relatively weak base dataset, such as in experiment 3, even modest improvements are recognized and exploited. This demonstrates METAM's sensitivity to incremental benefits, highlighting its ability to fine-tune the feature set to boost performance. The adaptive querying mechanism dynamically balances exploration (evaluating multiple candidate augmentations) with exploitation (selecting the augmentation that most improves utility), aligning with the paper's claims about efficient query optimization.

- Application to Regression Tasks: The Housing Price Regression experiment illustrates METAM's capacity to extend beyond binary classification. By integrating external datasets (despite some candidates having lower initial quality scores), METAM was able to extract non-redundant, valuable information—evidenced by a measurable increase in the utility score. This confirms that METAM can be effectively applied to more complex, continuous prediction tasks, further broadening its applicability in real-world scenarios.

- Theoretical Consistency: Across all experiments, the behavior of METAM aligns with the theoretical foundations presented in the original paper. The algorithm's reliance on data profiles—including semantic similarity—ensures that candidate augmentations are evaluated on multiple relevant dimensions. Moreover, the clustering and minimality checks guarantee that only the most informative augmentations are retained. This selective process not only reduces computational overhead but also mitigates the risk of incorporating noisy or redundant data.

In summary, our implementation and experiments underscore the strength of METAM's adaptive, minimal, and goal-oriented approach to data augmentation. The framework successfully balances the trade-off between adding extra features and maintaining a parsimonious model, ultimately leading to enhanced predictive performance. These findings suggest that METAM can serve as a valuable tool in a variety of data-driven decision-making applications, from classification to regression tasks, by automating and optimizing the process of data augmentation.