A Project Report

On

Frequent Pattern Mining and Association Learning

BY
Mohammad Jambughodawala
ID-2022A7PS0044H

Under the supervision of

PROF. ANEESH SRIVALLABH CHIVUKULA

CS F376 DESIGN PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI(RAJASTHAN) HYDERABAD CAMPUS

(September 2024)

Acknowledgement

I would like to express my heartfelt gratitude to Aneesh Srivallabh Chivukula Sir for his invaluable guidance, support, and mentorship throughout the course of this project. His expertise in machine learning and optimization techniques, particularly in the context of Maximum Entropy models, has been instrumental in shaping the direction and successful completion of this research.

Furthermore, I am thankful to Birla Institute of Technology and Sciences, for providing the necessary resources and facilities to carry out this research.

Abstract

This report presents a comparative analysis of various inequality constraints applied to Maximum Entropy models and explores their impact on improving model performance. The inequalities analysed include Box-type, 1-norm, 2-norm, and Gaussian MAP constraints, which were integrated into the model optimized using the Limited-Memory Variable Metric (LMVM) algorithm. Root Mean Square Error (RMSE) values were calculated for each constraint, and the ones with the lowest RMSE were selected for further evaluation. For these selected inequalities, K-fold validation was performed across multiple datasets, providing practical insights into their implementation and effectiveness.

Additionally, this work introduces a modification of Maximum Entropy models for sparse datasets by replacing the traditional Generalized Iterative Scaling (GIS) algorithm with the LMVM optimization method. While Maximum Entropy models are widely used in machine learning due to their flexibility and minimal assumptions, GIS often proves inefficient for large-scale problems due to its slow convergence. LMVM, a quasi-Newton method, addresses these limitations by using second-order derivative approximations, enabling faster convergence, improved scalability, and reduced computational time. This integration also mitigates the risk of overfitting in sparse datasets.

As part of this project, the use of SPMF and the Minimum Description Length (MDL) principle was also explored for discretizing large datasets. These methods were employed to transform continuous data into discrete intervals, enabling the application of the Maximum Entropy framework on large-scale transactional data.

Drawing insights from key papers such as *"Maximum Entropy Models with Inequality Constraints"* and *"A Comparison of Algorithms for Maximum Entropy Parameter Estimation,"* this study highlights the advantages of LMVM over GIS in Maximum Entropy frameworks. By combining efficient optimization with the strategic use of inequality constraints, the findings underscore the practical benefits and enhanced performance of the proposed approach.

Table of Contents

Introduction

This project focuses on improving the performance of Maximum Entropy (ME) models for sparse datasets by exploring the use of different optimization algorithms. Maximum Entropy models are widely utilized due to their flexibility in integrating rich feature sets while making minimal assumptions about the underlying data. In many applications, particularly associative regression, these models rely on frequent patterns—recurrent combinations of features within the dataset—to define constraints and enhance predictive accuracy.

The research paper under consideration proposes a supervised learning technique that emphasizes the role of significant frequent patterns in associative regression. These patterns are treated as key indicators that can quantify correlations within a dataset. The original framework constrains the Generalized Iterative Scaling (GIS) algorithm to converge more effectively in Maximum Entropy models by using combinations of ME parameters and GIS probabilities as discriminative weights for these frequent patterns. This approach aims to extract and prioritize the most informative patterns in the data, optimizing the model's predictive performance.

In this project, we extend the original framework by replacing GIS with Limited-Memory Variable Metric (LMVM) optimization. While GIS has traditionally been used for parameter estimation, LMVM was selected for its superior convergence properties and scalability, particularly in large-scale and sparse datasets. The optimization in LMVM is performed using the same objective function defined in the original Maximum Entropy formulation, ensuring consistency in the model's training process while aiming to improve efficiency.

This project also explores the incorporation of inequality constraints, such as Box-type, 1-norm, 2-norm, and Gaussian MAP constraints, into Maximum Entropy models to enhance their performance. These constraints serve as regularization

techniques, ensuring that the model parameters adhere to predefined bounds, thereby improving stability and preventing overfitting. By incorporating these inequalities, the models can prioritize significant frequent patterns within the dataset, effectively capturing correlations and improving predictive accuracy. The use of inequality constraints not only refines the model's training process but also provides a structured way to balance flexibility with robustness, making it particularly effective for sparse and large-scale datasets.

Literature Review

1. Maximum Entropy Based Associative Regression for Sparse Datasets

This paper introduces a supervised learning technique for associative regression that emphasizes the significance of frequent patterns in sparse datasets. The authors propose a framework that integrates the Generalized Iterative Scaling (GIS) algorithm within Maximum Entropy (ME) models. By leveraging combinations of ME parameters and GIS probabilities as discriminative weights for frequent patterns, the approach aims to enhance predictive accuracy and prioritize the most informative patterns, effectively addressing the challenges posed by data sparsity.

2. A Comparison of Algorithms for Maximum Entropy Parameter Estimation

Malouf's study compares various algorithms for estimating parameters in Maximum Entropy models, highlighting the computational inefficiencies of traditional iterative scaling methods, including GIS. He emphasizes the advantages of limited-memory variable metric methods, like LMVM, which significantly enhance convergence speed and overall efficiency.

3. LMVM-(Limited-Memory Variable Metric)

LMVM is a quasi-Newton method that approximates the Hessian matrix using information from previous iterations, allowing it to capture the curvature of the loss function without the computational burden of calculating the full Hessian. This is particularly beneficial for high-dimensional problems, as LMVM requires only a small amount of memory to store updates, making it suitable for large-scale applications.

In practice, LMVM updates the weights iteratively by using both gradient information and an approximation of the second-order information (Hessian) to guide the optimization process more effectively than first-order methods. The combination of these features allows LMVM to converge more quickly and robustly in complex optimization landscapes, providing a compelling alternative to traditional methods like GIS.

2. Maximum Entropy Models with Inequality Constraints

Kazama and Tsujii discuss the limitations of standard Maximum Entropy estimation, particularly concerning overfitting in sparse datasets. They propose the use of inequality constraints, which allow for some violation of equality constraints to promote regularization and improve generalization performance. Their empirical results show that inequality ME models outperform traditional ME estimation methods, such as GIS, while simplifying the feature selection process and mitigating the risks of overfitting.

Several types of inequality constraints have been explored, including Box-type, 1-norm, 2-norm, and Gaussian MAP. These constraints modify the standard ME optimization objective by introducing bounds on the model parameters.

The paper involves various inequality constraints:

1) **Box-type inequality**
   The maximum entropy estimation with box-type inequality constraints can be formulated as the following optimization problem:
   maximize p(y|x) $H(p)=-\sum \tilde{p}(x)\sum p(y|x)\log p(y|x)$
   Subject to:

A. Upper constraints: $E_{\tilde{p}}[f_i]-E_p[f_i]-A_i<=0$
B. Lower constraints: $E_p[f_i]- E_{\tilde{p}}[f_i]-B_i<=0$

The box-type inequality constraints are reasonable as $A_i, B_i \to 0$, the optimization problem approaches that of standard ME estimation.
For our implementation the objective function used is
$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x)$
under the constraints
$-B_i \le E_{\tilde{p}}[f_i]-E_p[f_i] \le A_i$

There are two methods for calculating the constants A_i and B_i.

1. Single Width:
   The first is to use a common width for all features, which is calculated with the following formula.
   $A_i = B_i = W \times 1/L$
   where *W* is a constant *width factor* to control the widths, and *L* is the number of training examples. This method takes the reliability of training examples as a whole into account, and is called *single*. The same width is used for upper and lower inequality.
2. Bayesian width:
   The second method for calculating the constraints is using the Bayesian method.
   $A_i=B_i=W\times\sqrt{V[E\,\tilde{p}[f_i, j]]}$.
   Where $V[E\,\tilde{p}[f_{i,j}]]$.is the variance of the empirical expectation of feature f_i

2) 1-norm penalty extension:

It is also possible to impose 1-norm penalties in the objective function. I f we impose 1-norm penalties, we obtain the following optimization problem.
Maximize:
$H(p)-C1\sum\delta_i-C2\sum\gamma_i$
subject to
$E\,\tilde{p}\,[f_i]-E_p[f_i]-A_i \le \delta_i$
and
$E_p[f_i]- E\,\tilde{p}\,[f_i]-B_i \le \gamma_i (\gamma_i>0)$
Where H(p) is the objective function.

The parametric form and dual objective function for this optimization problem is identical to those of the inequality ME model, except that the parameters are also upper-bounded as.
For our case the objective function is:
$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x) - \sum_i (|\gamma_i| + |\delta_i|)/ 2C_1$

Under the constraints:

$-B_i - \gamma_i \le E\,\tilde{p}\,[f_i]-E_p[f_i] \le\le A_i + \delta_i$
The values of γ and δ are taken randomly between 0 and 0.1. For C1 we chose the value as 1.

3) **2-norm penalization:**

Our 2-norm extension to the inequality ME model is formulated as follows.
Maximize:
$H(p)-C1\sum\delta_i^2-C2\sum\gamma_i^2$
subject to
$E_{\tilde{p}}[f_i]-E_p[f_i]-A_i \leq \delta_i$
and
$E_p[f_i]- E_{\tilde{p}}[f_i]-B_i \leq \gamma_i (\gamma_i>0)$
Where H(p) is the objective function.
For our case the objective function is:

$$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x) - \sum_i (\gamma_i^2+ \delta_i^2)/ 2C_1$$

Under the constraints:
C1 is any constant.

$$-B_i - \gamma_i \leq E_{\tilde{p}}[f_i]-E_p[f_i] \leq \leq A_i + \delta_i$$

4) **Gaussian MAP estimation:**

In Gaussian Maximum a Posteriori(MAP) estimation, the goal is to maximize the posterior probability of parameters given the training data instead of directly maximizing the likelihood of the training data.
General Objective function in Gaussian MAP is given by:
$LL(\lambda)=\log \prod_{x,y} p\,\lambda\,(y|x)^{\tilde{p}(x)\,\tilde{p}(y|x)}$
Objective function:
$LL(\lambda) - \frac{1}{2\sigma^2}\sum\lambda^2$

For our case the objective function is:
$H(p)=- \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x) - \frac{1}{2\sigma^2}\sum\lambda^2$
For λ we use the weights which we got from the frequent patterns.

Here are the objective functions used for various inequalities.

Model Type	Objective Function	Constraints
Standard Maximum Entropy (ME)	$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x)$	$E_{\tilde{p}}[f_i] - E_p[f_i] = 0$
Box-type Inequality ME	$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x)$	$-B_i \leq E_{\tilde{p}}[f_i] - E_p[f_i] \leq A_i$
2-norm Penalized Inequality ME	$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x) - \sum_i (\gamma_i^2 + \delta_i^2) / 2C_1$	$-B_i - \gamma_i \leq E_{\tilde{p}}[f_i] - E_p[f_i] \leq A_i + \delta_i$
1-norm Penalized Inequality ME	$H(P(c, x)) = - \sum_{(c,x) \in (C,X)} P(c, x) \log P(c, x) - \sum_i (\lvert\gamma_i\rvert + \lvert\delta_i\rvert) / 2C_1$	$-B_i - \gamma_i \leq E_{\tilde{p}}[f_i] - E_p[f_i] \leq \leq A_i + \delta_i$

Methodology

This methodology outlines the approach taken to enhance Maximum Entropy models for sparse datasets, focusing on the integration of frequent patterns and optimization techniques.

The process begins by loading the dataset from a user-specified CSV file and identifying the dependent variable, which is then separated from the feature columns for analysis. A second CSV file containing frequent itemsets is also loaded, and these itemsets are transformed into a one-hot encoded format to represent the presence or absence of each item.

Missing values in the feature columns are addressed by imputing numerical columns with their mean and categorical columns with the most frequent values. Numerical features are subsequently converted into binary format using quantile binning, allowing for clear categorization based on specified intervals.

The dependent variable is then analysed to create intervals that categorize its values into discrete groups, including scaling to a target range and dynamically computing bin edges. The Jaccard similarity is calculated between the binary representation of the dataset and the one-hot encoded itemsets to measure the similarity of patterns across the dataset, and the results of these calculations are transformed into a binary matrix based on a specified threshold.

Notably, the implementation closely follows the Generalized Iterative Scaling (GIS) approach, with the primary distinction being the use of Limited-Memory Variable Metric (LMVM) optimization, which enhances convergence speed and efficiency while preserving the core principles of the original GIS framework.

After preprocessing the dataset and transforming the Jaccard similarity results into a binary matrix, the next phase involves optimizing the weights of the Maximum Entropy model using the Limited-Memory Variable Metric (LMVM) optimization technique.

The transformed binary matrix is first converted into a sparse matrix format to facilitate efficient computation. The objective function is defined to compute the loss based on the predicted probabilities, which are obtained by applying the softmax function to the weighted features. This loss is calculated using the

negative log-likelihood, incorporating a small constant to prevent logarithm of zero.

Additionally, the gradient of the objective function is computed to guide the optimization process. The gradient reflects how changes in the weights affect the predicted probabilities, allowing for more efficient convergence towards optimal weights.

The optimization begins with randomly initialized weights drawn from a normal distribution. The weights are updated iteratively using the gradient descent method, described by the formula:

$w_{new} = w_{old} - \eta \cdot \nabla L(w_{old})$

The minimize function from the scipy.optimize library is then employed to find the optimal weights by minimizing the objective function. This function utilizes the L-BFGS-B method, which is well-suited for large-scale problems, and employs the previously defined gradient function to accelerate convergence. The optimized weights are returned as the final output, ready to be used for making predictions with the Maximum Entropy model.

Following the optimization of weights, the next step involves calculating the weights for each category in the dataset. For each unique category, the Jaccard similarity matrix is computed based on the binary representation of the data and the one-hot encoded frequent itemsets. The transformed similarity matrix is then created, which is passed to the optimization function to derive the optimal weights for that specific category. The optimized weights are stored in a dictionary, associating each category with its corresponding weights.

Once the weights are determined, the next phase involves calculating the predicted probabilities P(x|c)for each instance in the dataset. This is achieved by computing the Jaccard similarity again and reshaping the results to facilitate matrix multiplication with the corresponding weights for each category. The resulting probabilities are stored in a DataFrame for further analysis.

To evaluate the model's performance, the method calculates the probabilities of each category within the dataset. The category counts are derived, and their probabilities are computed relative to the total number of records. A constant value representing the record probability is set, and the ratio of this record probability to the category probabilities is calculated for each category, facilitating the determination of the maximum posterior category.

The model then predicts the most likely category for each instance based on the computed probabilities. For each entry in the dataset, the posterior probabilities are computed, and the category with the highest probability is assigned as the predicted category.

To assess the accuracy of these predictions, the Root Mean Squared Error (RMSE) is calculated by comparing the predicted categories with the actual categories from the test set. This involves evaluating the squared differences between the predicted maximum values and the actual maximum values associated with each category, providing a quantitative measure of the model's performance.

The methodology for applying various inequality constraints:

Selection of Inequality Constraints: Various constraints are used, including:

- Box-type constraints: Bound the feature functions within upper and lower limits.
- 1-norm and 2-norm penalties: Regularize the weights by penalizing deviations from expected values.
- Gaussian MAP: Introduces a Gaussian prior to regularize the weights.

Objective Function: The selected constraints are integrated into the ME objective function. For example, 1-norm and 2-norm penalties add terms that penalize large deviations in feature expectations, while Gaussian MAP adds a regularization term.

Optimization Process: The gradient of the objective function is computed, and the Limited-Memory Variable Metric (LMVM) optimization method is used to minimize the objective, ensuring efficient convergence, particularly for large-scale datasets.

Constraint Enforcement: During optimization, constraints are dynamically enforced to keep feature functions within the defined bounds, promoting regularization.

Final Model and Predictions: The optimized weights are used to compute predicted probabilities, and performance is evaluated using Root Mean Squared Error (RMSE). The inclusion of inequality constraints leads to improved model performance, reducing overfitting and enhancing generalization.

The methodology remains the same only extra inequality constraints are added to the objective function based on the inequality used.

A large dataset(concatenated_SWaT_Dataset.csv) was discretized using two approaches: SPMF and Minimum Description Length (MDL). Using SPMF, continuous attributes were binned based on frequent itemsets to align discretization with significant patterns in the data, ensuring relevance for subsequent analysis. The MDL principle was applied to segment numeric features into intervals that maximized information gain while balancing model complexity and data representation efficiency. Both methods transformed the dataset into a binary or categorical format, facilitating feature integration into the Maximum Entropy mode.

Results

We have obtained results for several datasets from the UCIML repository without applying any inequality constraints.

Dataset	**RMSE values**
Bolts.csv	0.4837
Socmob.csv	0.71
Concrete.csv	0.42055
Bodyfat.csv	0.4595

Here are the results after applying various inequality constraints:

I. After using box-type inequalities: with single-width

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.6005	0.4598
Socmob.csv	0.1304	0.0986
Concrete.csv	0.4205	0.3870
Bodyfat.csv	0.4710	0.4655

Using Bayesian width estimation:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.4564	0.5457
Socmob.csv	0.08104	0.0877
Concrete.csv	0.4144	0.4299
Bodyfat.csv	0.5712	0.3996

II. Using 1-norm penalized inequality

With single-width extension:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.5327	0.546
Socmob.csv	0.8689	0.1397
Concrete.csv	0.4066	0.4680
Bodyfat.csv	0.4165	0.5141

With Bayesian estimation:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.5857	0.4987
Socmob.csv	0.8689	0.8734
Concrete.csv	0.4205	0.4528
Bodyfat.csv	0.4795	0.4709

III. Using 2-norm penalized inequality

With single-width extension:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.1357	0.2466
Socmob.csv	0.8846	0.0829
Concrete.csv	0.3838	0.3255
Bodyfat.csv	0.5579	0.5684

Using Bayesian estimation:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.5048	0.1806
Socmob.csv	0.1264	0.8756
Concrete.csv	0.4810	0.3930
Bodyfat.csv	0.4448	0.5083

IV. Gaussian MAP estimation:

Dataset	Rmse values	K-fold rmse
Bolts.csv	0.6163	0.7689
Socmob.csv	0.0684	0.2447
Concrete.csv	0.4766	0.4867
Bodyfat.csv	0.5771	0.5575

The table shows the inequalities for which the datasets had the least rmse values.

Datasets	**Rmse-values**	**Inequalities with least error**
Bolts.csv	0.1357	2-norm penalty using constants from single-width method
Socmob.csv	0.0684	Gaussian MAP estimation
Concrete.csv	0.3838	2-norm penalty using constants from single-width method
Bodyfat.csv	0.4165	1-norm penalty using constants from single-width method

We applied k-fold cross-validation with k=10 to each dataset. This method averages RMSE across multiple folds, reducing the variance and providing a more reliable performance measure.

Datasets	**K-fold Rmse-values**	**Inequalities with least error**
Bolts.csv	0.1806	2-norm penalty using constants from Bayesian estimation method
Socmob.csv	0.0829	2-norm penalty using constants from single-width method
Concrete.csv	0.3255	2-norm penalty using constants from single-width method
Bodyfat.csv	0.3996	Using constants calculated using Bayesian estimation without any penalization

Considering that k-fold cross-validation offers a more accurate measure of the model's performance, we select the best inequality constraints for each dataset on the basis of k-fold Rmse values. The given inequalities give the best Rmse-values.

Conclusion

In this project, we explored the implementation of Maximum Entropy models using various optimization algorithms, focusing on Limited-Memory Variable Metric (LMVM) optimization. While Generalized Iterative Scaling (GIS) provides a straightforward approach to parameter estimation, our findings highlight that LMVM significantly enhances efficiency and performance, especially for large datasets. Additionally, we incorporated inequality constraints into the objective function to improve generalization and robustness in sparse data scenarios.

We evaluated the performance of various inequality constraints across datasets, aiming to identify the best constraints for each dataset. To ensure robust and reliable results, k-fold cross-validation was employed. By comparing RMSE values across folds, we determined the optimal inequality constraints for each dataset. The findings are as follows:

- **Bolts.csv**: 2-norm penalty with constants calculated using the Bayesian estimation method.
- **Socmob.csv**: 2-norm penalty with constants calculated using the single-width method.
- **Concrete.csv**: 2-norm penalty with constants calculated using the single-width method.
- **Bodyfat.csv**: Bayesian estimation without any penalization.

These results underscore the flexibility of inequality constraints in adapting to the unique characteristics of different datasets while improving the performance of Maximum Entropy models.

Future Works

In future work, we plan to focus on discovering frequent patterns within large discretized datasets. This will involve applying advanced pattern-mining techniques to extract meaningful associations.

References

Papers:

1. Maximum Entropy based Associative Regression for Sparse Datasets
2. Maximum entropy models with inequality constraints† A case study on text categorization
3. A comparison of algorithms for maximum entropy parameter estimation