Manifold
Model-agnostic visual debugging tool for ML
Lezhi Li, Yunfeng Bai, Yang Wang
OpML’19  May 20, 2019
Manifold: model-agnostic visual debugging tool for ML

Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models

Jaewi Zhang, Yang Wang, Piern Molino, Lashi Li and David S. Ebert, Fellow, IEEE

Fig. 1: (a) Manifold consists of two iterative stages: (1) first provides a visual comparison between model predictions, and model output design. A heat map detection (2) proceeds to illustrate the decision boundary in the feature space. The brighter the color, the more significant features for classification. This visual representation can also be used to compare more diverse features among different models, i.e., based on the selected subset in red (b) or in a specific class such as c (c). (d) Manifold has been validated on the IMDB dataset and the results demonstrate high accuracy on misclassified images.

Abstract: Interpretation and diagnosis of machine learning models have gained renewed interest in recent years with breakthroughs in new approaches. We present Manifold, a framework that utilizes visual analysis techniques to support interpretation, debugging, and comparison of machine-learning models in a more transparent and intuitive manner. Compared to existing techniques usually focused on visualizing the internal logic of specific model types (e.g., neural network), visualizing the ability to predict in a more complex scenario where different model types are compared. This study has demonstrated the power of Manifold in improving model accuracy and reducing errors. In addition, Manifold can help users to identify and rectify problematic areas in the models, thereby improving the overall performance of the models.

1 INTRODUCTION

Recent technical breakthroughs in the machine learning field have led to highly improved accuracy and efficiency in many scenarios, including medical or public recognition tasks [5, 7]. However, these technical advances pose new major challenges. First, the complexity of the models being deployed and adopted has significantly increased to the point that it is difficult for model developers to explain why and how the model works. Second, model developers often lack solid understanding of the models and are unable to debug them effectively due to the limitations of the models, making the training process time-consuming and error-prone. Both of these challenges require more effective approaches to model interpretation and explanation of machine learning processes [11, 14].

Visual and interactive models have proven to be effective in terms of visual and interactive models have proven to be effective in terms of interpretability. However, the current methods for interpreting and diagnosing complex models (e.g., [15, 16]) require a deep understanding of the models. Therefore, we aim to enhance the interpretability and understanding of the models by providing a visual representation of the decision process in the feature space. The brighter the color in the heat map, the more significant features for classification. This visual representation can also be used to compare more diverse features among different models, i.e., based on the selected subset in red.
Agenda

01 Motivation
02 Workflow
03 Integration

Manifold
https://eng.uber.com/manifold

Michelangelo
https://eng.uber.com/scaling-michelangelo
Importance of Model Debugging

- 20% building initial models
- 80% improving model performance
Inadequacy of performance metrics

ROC

TPR

FPR

AUC: 0.81

Great! but what’s Next?

not actionable...
Model Interpretability?

Interpretable Machine Learning
Internal Structure
Intermediate State
Mimic Learning
Interactive Hyperparameter tuning
Model Performance Visualization

Focus on single (family of) model(s)
Too much love for deep models :/
Model-agnostic?

Interpretable Machine Learning
Internal Structure
Intermediate State
Mimic Learning
Interactive Hyperparameter tuning
Model Performance Visualization
...

Model-specific interpretability
will not scale
from the operational perspective

Machine Learning at Uber
Predicting Supply & Demand
One-click Chat
Restaurant Recommendation
Sensor Intelligence and Location
Autonomous Vehicles
Support Ticket Routing
Incentive Optimizing
Fraud Detection
Financial Planning
...
$y = f^* (x) + \varepsilon$
Model space => Data space

What went wrong with the model?
Which data **subset** did the **model(s)** make mistake?

Why the model made such mistake?
Which **feature** has contributed to the mistakes?
Manifold workflow

1. Compare

Given a dataset with the output from one or more ML models, Manifold compares and highlights performance differences across models or data subsets.

2. Slice

Users can select data subsets of interest based on model performance for further inspection.

3. Attribute

Manifold then highlights feature distribution differences between the selected data subsets, helping users find the reasons behind the performance outcomes.

City A

lower delivery radius

Other cities

faster delivery

late lunch and dinner
Manifold workflow

1. Compare
Given a dataset with the output from one or more ML models, Manifold compares and highlights performance differences across models or data subsets.

2. Slice
Users can select data subsets of interest based on model performance for further inspection.

3. Attribute
Manifold then highlights feature distribution differences between the selected data subsets, helping users find the reasons behind the performance outcomes.
Challenges

01 Interactivity √
02 Scalability ?

Operational
Computational

=> Michelangelo
Integration with Michelangelo
Integration with Michelangelo

- Train models
- Prepare scored dataset
- Analyze with Manifold
- Gained insights
Integration with Michelangelo

Case Study:
Identify useful features

- A team at Uber is evaluating the value of an extra feature.
- Adding new features did not change model’s overall performance.
- Are the new features still worth adding?
- Yes, improvement on hard-to-predict data.
Architecture

- Michelangelo UI
- Manifold
- Michelangelo API services
- Workflow execution
- Dataset storage

- Fetch data for display
- Start dataset preparation
- Start workflow
- Report status
- Store scored datasets
Manifold Workflow

- Expressing Manifold comparison dataset generation with MA physical workflow
Thank You!

Questions?
Consistent Sampling

How to consistently downsample rows across Dataframes?

- No native support for consistent sampling in Spark
- Filter based on hash value