Paper SAS2924-2018

Automated Assistants for Fraud Investigation Productivity

Michael Ames, SAS Institute Inc.

ABSTRACT

Analyst and investigator productivity is top of mind for many organizations, and they are looking to technology to boost both analyst productivity and investigation quality. The application of automation technology that allows employees to configure “automated assistants” often referred to as “robotic process automation”. Automated assistants is an emerging form of clerical process automation based on the notion of software robots performing common investigator tasks.

INTRODUCTION

One of the most often sited uses of machine learning is to combat fraud, waste, and abuse; however the machine learning and data science community has failed to embrace one of the easiest applications of machine learning! Simply applying machine learning to improve investigator productivity and effectiveness. While typical machine learning approaches focus on the surveillance side of the equation, new approaches are looking at how to automate investigator activities and provide “virtual assistants” to aide investigators in mundane tasks like fetching data, authoring comments, summarizing activity all the way to automating the investigation process.

INTELLIGENT AGENTS AND VIRTUAL ASSISTANTS

An intelligent agent can be deployed as both an autonomous and interactive agent. These agents are capable of making decisions and executing tasks based on its experience and can learn from the outcomes of those decisions. Unlike supervised learning, learning from labeled data, intelligent agents learn to navigate and adapt to unknown terrain.

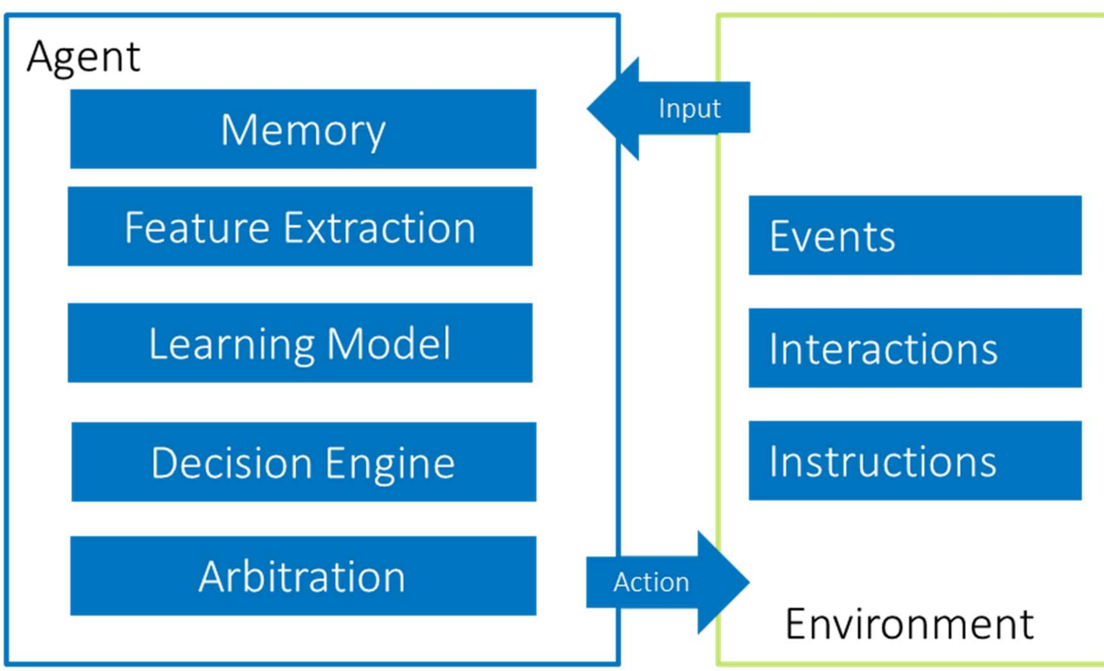

Figure 1. Intelligent Agent.

As figure 1 shows the agent is a process, that contains recent memory and perform things like feature extraction, it has a learning model, a decision engine and performs arbitration between conflicting decisions and task activities.

MEMORY

Intelligent agents require some amount of working memory. Just like a human, the agent retains recent events in “working memory”, so that it can adapt to new and emerging interactions with the environment. Typically, this means storing the last several “events of interest” that have been fed into the agent. This might be recent fraud patterns that the learning model hasn’t adapted to yet or even white / blacklists that aide the decision engine in handling high value customers. The working memory of the agent also contains instruction sets for arbitration to ensure that conflicting decisions are biased toward events of interest.

FEATURE EXTRACTION

As the agent is learning and adapting, it will generate new features from events as they are fed through the system. Think of the agent’s feature extraction as “automated feature engineering”, as time goes on new features are generated, older features are aged off, the result is a changing feature space that the learning model adapts to as it learns behaviors of interest.

LEARNING MODEL

A number of machine learning methods that can be employed as the learning model; however, in practice supervised models tend to train and adapt best. There are a number of machine learning methods that have been applied, however my experience has shown that support vector machines often work extremely well with sparse data.

DECISION ENGINE & ARBITRATION

The outcome from the learning model is a “probably of task”, or score for each predicted task the agent is potentially going to make on the analysts behalf, this task is then turned into an instruction set. Depending on the data presented to the agent, the agent simply predicts which tasks it should likely perform. For example if an analyst interactively asks a question like “show me similar accounts” the agent then predicts the likelihood of tasks to answer the question. Tasks can be simple instruction sets like execute query_1 against the ware house, or it might be generate near neighbors. Sometimes agent tasks can be in conflict with what the agent’s learning model says to do. In those cases results are sent to arbitration. This is where the agent can be biased toward what the learning model performs.

For example. The “show me similar accounts” could predict “taskA” and “taskC” which are in total conflict with one another. Say taskA “closes customer’s account” and taskC “increases their available credit” with similar probability, which task should the agent choose? Arbitration rules allow for biasing of prediction results to avoid conflicting tasks.

ADAPTIVE LEARNING PROCESS

The learning process is not too different from a traditional machine learning process, it is just completely automated.

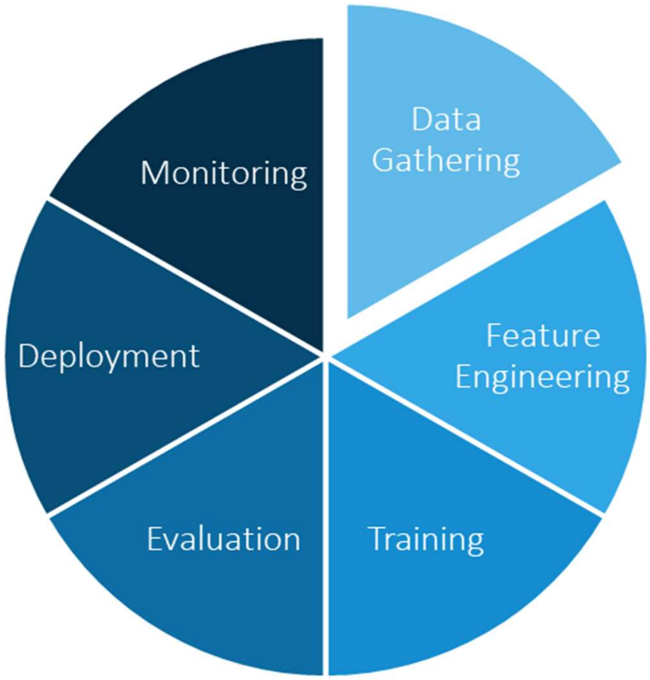

Figure 2. End to End Process

As figure 2 shows, the learning process is not really different from a manual machine learning model training process, it’s just automated the phases. Here is an overview of how it works.

DATA GATHERING

One of the things that separates “bots” and “automated assistants” from other applications of machine learning is that data gathering can be interactive, in fact the data gathering, feature engineering and training phases are often collapsed into a single pipeline.

FEATURE ENGINEERING PHASE

Given a data set, typically with unstructured or semi-structured, the system parses the data and using various text pre-processing methods features are engineered. The feature space with a textual assistant can be quite wide but using dimension reduction techniques like PCA(Principle Component Analysis), SVD(Singular Value Decomposition), or Matrix Factorization and others we can dramatically reduce the number dimensions to deal with.

TRAINING PHASE

Given a data set with engineered features and with labeled data, the intelligent agent learns patterns of the tasks within the data producing a “model of its environment:”

EVALUATION PHASE

The agent evaluates its expected performance versus historic performance within the environment, determines the critical factors and tailors the “instruction sets” & “interactions” to the agent’s tasks ans objectives.

DEPLOYMENT PHASE

The agent is then deployed as an api that constantly gathers data, predicts tasks and evaluates its performance. The agent typically contains an input vector (working memory), the learning model and a default task generating rules based on the instructions from the learning phase. The agent takes in new events and questions, predicts tasks, and decides which tasks to automatically perform and which the analysts can manually initiate. Analyst's activity and manual task execution are fed back into the learning model to improve and adapt as new data is presented to the system.

LEARNING PHASE

The agent combines "events of interest" and "normal" events from the system and folds it into working memory producing "training data". The learning agent takes the data and learns to identify patterns of events of interest. As the agent learns from the events and tasks execution, it develops an instruction set to identify new and emerging task predictive patterns. By combining automated tasks, events and manual task execution, the agent is able to adapt to new and unseen patterns within the data.

MONITORING

Autonomous agents require ongoing system and activity performance monitoring. Generally agents are self-sufficient, biasing toward supervised task selection based on the data and activities within the system. System performance measures like how long it takes to make a decision (decision latency), how much resources it is using during the learning phase (resource consumption) and how much data it consumes, all need to be monitored over time. Task activity and accuracy performance is also key. It is important to track both the task accuracy and match that to analyst productivity rates over time to make sure the agent is actually doing something valuable. Finally these agents aren't a panacea, overtime tasks and patterns can change dramatically and the agent may need to be retired either for poor performance or be superseded by a better agent that is trained on different data.

SUPERVISED LEARNING

Multiclass Support Vector Machines (SMVs) are a popular choice because they handle both limited and sparse data as well as support multiclass targets, in this case multiple tasks that are predicted. There are a number of other supervised machine learning methods that can be used including neural networks, and decision trees, however the limited and sparse nature of the data make as well as support for multiclass targets make SVMs especially well-suited for agent use.

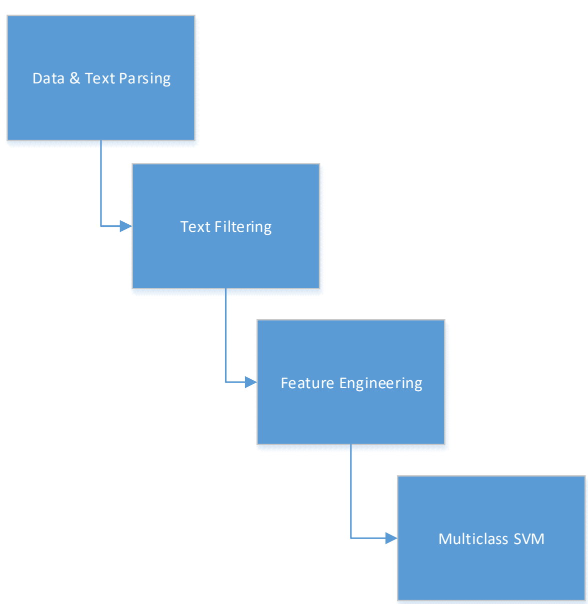

Figure 3. Typical training flow.

Data is gathered, parsed and filtered then engineered using PCA or SVD, resulting in a training dataset like table 1.

FEATURE_	FEATURE_	FEATURE_	FEATURE_	FEATURE_5	FEATURE_6	FEATURE_7	TASK
1.167	0.502	(0.067)	2.262	0.429	0.089	0.241	Task_1
(0.523)	1.010	0.276	1.475	(0.707)	0.355	1.560	Task_2
(0.715)	0.515	1.822	0.616	0.849	(0.112)	1.506	Task_3
(1.793)	1.855	0.980	1.112	(0.206)	(0.200)	0.617	Task_4
(0.848)	1.043	1.267	1.136	0.021	0.337	0.398	Task_5
(0.773)	(4.146)	(0.932)	0.027	(1.698)	0.460	0.737	Task_2
1.107	0.216	0.538	1.476	(0.252)	(0.341)	0.154	Task_3
(0.655)	0.410	1.289	(0.325)	0.546	(0.350)	0.648	Task_1
1.196	0.259	0.636	0.641	(0.395)	(0.695)	0.034	Task_1
(1.277)	(0.904)	1.477	(2.583)	0.636	(0.867)	1.138	Task_1

Table 1. Predicting tasks based on engineered features.

PREDICTING TASKS

At this point predicting tasks is just like any other multiclass problem. Data is prepared and scored by the decision engine producing a task prediction vector. Here is an example of predicting tasks 1-5 based on the feature vectors

FEATURE_	FEATURE_	FEATURE_	FEATURE_	FEATURE_5	FEATURE_6	FEATURE_7	Task_1	Task_2	Task_3	Task_4	Task_5
1.167	0.502	(0.067)	2.262	0.429	0.089	0.241	0.326	0.498	0.173	0.683	0.547
(0.523)	1.010	0.276	1.475	(0.707)	0.355	1.560	0.976	0.413	0.499	0.354	0.795
(0.715)	0.515	1.822	0.616	0.849	(0.112)	1.506	0.132	0.186	0.521	0.857	0.597
(1.793)	1.855	0.980	1.112	(0.206)	(0.200)	0.617	0.993	0.962	0.826	0.318	0.511
(0.848)	1.043	1.267	1.136	0.021	0.337	0.398	0.686	0.46	0.358	0.794	0.952

You can see that several tasks score high based on the features, which is totally expected.

AGENT DECISION ENGINE

Given a predicted set of tasks, which tasks should be executed? Depending on the agent, often business rules are used to determine which tasks can be performed in an autonomous fashion and which tasks are presented interactively to an analysts.

An example of the decision rules might be something like this:

IF task_1_score > 0.9 THEN execution = “Automatic”

IF task_1_score BETWEEN 0.5 and 0.9 THEN execution = “Interactive”

EXAMPLE INTERACTIVE AGENT

In this example from SAS Visual Investigator, the agent is deployed as a chat bot. here is a an example of the dialogue between the agent(#) and the investigator(>>).

```
Investigator >> what time is it

Agent # “15.38%, you want to know the time. The local time is 15:26:48”

Investigator >> show me similar records to Account ABC-1235

Agent # “26.67%, you want to Near Neighbors. The nearest neighbors to
Account ABC-1235 are”:
```

Last_Name	First_Name	Credentials	Gender	Street_Address_1	City_Name	Zip_Code	State_Code	Provider_Type	Num_of_Medicare	Total_Charge_Amount
TANNER	DAVID	MD	M	4310 JOHNS CREEK PKWY	SUWANEE	300246091	GA	Allergy/Immunology	233	142851
SEIDU	LUQMAN	M.D.	M	5445 MERIDIAN MARKS RD NE	ATLANTA	303424763	GA	Allergy/Immunology	160	100738
VANGALA	RAHUL	M.D.	M	3964 ELNORA DR	MACON	312101825	GA	Allergy/Immunology	132	118747
HENSON	MICHELE	MD	F	143 CANAL STREET	POOLER	31322	GA	Allergy/Immunology	142	81459
HWANG	LILY	MD	F	2110 POWERS FERRY RD SE	ATLANTA	303395048	GA	Allergy/Immunology	84	38726.3

CONCLUSION

The opportunity to improve investigator productivity and effectiveness through the deployment of machine learning and intelligent agents exists, it just isn't being exploited today. Simply automating the data gathering and mundane tasks of investigations frees investigators to perform the assessment and analysis activities resulting in more productive and quality investigations. The use of machine learning to predict and automate analyst tasks is one of the few low effort & low risk machine learning projects resulting in high value opportunities that still exist for many organizations.

REFERENCES

Hersh, W., Buckley, C., Leone, T., & Hickam, D. (1997). OHSUMED: An interactive retrieval evaluation and new large test collection for research. In

Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR-97) (pp. 192–201).

Sokolova, Marina; Lapalme, Guy A systematic analysis of performance measures for classification tasks" Information Processing and Management. 2008. Available at http://atour.iro.umontreal.ca/rali/sites/default/files/publis/SokolovaLapalme-JIPM09.pdf.

ACKNOWLEDGMENTS

I would like to thank and acknowledge Dr. Yue Qi, Jeff Miller, Tom Mutdoch, Iain Jackson, Rory MacKensie and Gordon Robinson for their support and efforts to develop the SAS Adaptive Learning and Intelligent Agent System.

RECOMMENDED READING

- SAS Visual Data Mining and Machine Learning 8.2: Procedures

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Michael Ames
SAS Institutue
michael.ames@sas.com