A Few Useful Things to Know About Machine Learning

PEDRO DOMINGOS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING UNIVERSITY OF WASHINGTON

Summary by: William Steimel

Source

- https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

Motivation

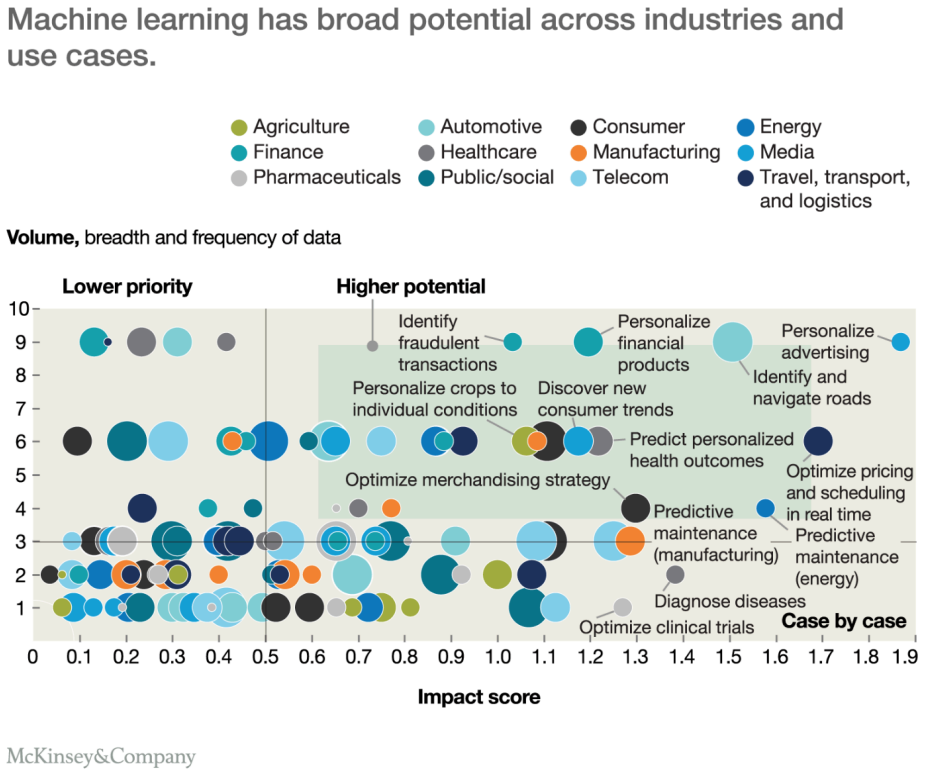
- I choose this article as it was recommended for beginners in Machine Learning
- I am a beginner and hope to improve my Machine Learning knowledge

Abstract

- Developing Machine Learning Algorithms requires some “black art” or “folk wisdom” which is often not found in textbooks.
- This Article summarizes 12 key lessons from Machine Learning researchers including pitfalls, important issues to focus on, and answers to common questions

Introduction

- Machine Learning use has recently spread rapidly throughout computer science and our everyday lives.
  - Web Search
  - Spam Filters
  - Recommender Systems
  - Ad Placement
  - Credit Scoring
  - Fraud Detection
  - Stock Trading
  - Drug Design
  - Etc.



Introduction

- Many textbooks exist to teach machine learning but the "Folk knowledge" needed to create effective models is often neglected and not available
- This article seeks to convey some of that knowledge to machine learning researchers
- This paper largely focuses on applications to classification models but can be applied across all of machine learning

Learning = Representation + Evaluation + Optimization

- Choosing which learning algorithm to use for a situation can sometimes be challenging
- When choosing a learning algorithm the key is to remember that it consists of combination of 3 components

Table 1: The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Learning = Representation + Evaluation + Optimization

- Representation- A classifier must be represented in some formal language that the computer can handle (hypothesis space)
- Evaluation – Objective functions or scoring functions are essential for distinguishing good classifiers from bad ones –
  - How good is the machine learning model?
- Optimization- Methods to search among classifiers in the language for the highest scoring one –
  - Method to search for the most optimal model

Learning = Representation + Evaluation + Optimization

- Example from Research Paper of these three components in action.

Representation – Decision Tree

Evaluation – Information Gain

Optimization – Greedy Search

Algorithm 1 LearnDT(*TrainSet*)

if all examples in *TrainSet* have the same class y_* **then**
  return MakeLeaf(y_*)
if no feature x_j has InfoGain(x_j,y) > 0 **then**
  $y_* \leftarrow$ Most frequent class in *TrainSet*
  return MakeLeaf(y_*)
$x_* \leftarrow \text{argmax}_{x_j}$ InfoGain(x_j,y)
$TS_0 \leftarrow$ Examples in *TrainSet* with $x_* = 0$
$TS_1 \leftarrow$ Examples in *TrainSet* with $x_* = 1$
return MakeNode(x_*, LearnDT(TS_0), LearnDT(TS_1))

It's Generalization that Counts

- The fundamental goal of machine learning is to generalize beyond the examples in the training set
- Many machine learning beginners make the mistake of just testing on training data with the illusion of success
- Data used to train the model must be kept separate from data used to evaluate the model

Data Alone is not enough

- Every learner must have some knowledge or assumptions beyond the data in order to generalize beyond it.
- Very General assumptions like smoothness, similar examples have similar classes, limited dependencies, or limited complexity often do well
- Induction – Turning a small amount of input knowledge into a lot of output knowledge
- Machine Learning isn't magic: Examples-
  - If we have knowledge about what makes data similar in our domain – instance based methods may be best
  - If we have knowledge about probabilistic dependencies – graphical models are a good fit
  - If we have knowledge about what kind of preconditions are required for each class, "If Then" Rules may be the best
- "Learners combine knowledge with data to grow programs"

Overfitting Has Many Faces

- Overfitting – "A modeling error which occurs when a function is too closely fit to a limited set of data points."
- Overfitting comes in many forms – Sometimes it is not easy to see
- One way to understand overfitting is by breaking generalization error into Bias and Variance
  - Bias- a learner's tendency to consistently learn the same thing wrong (Distance from the center)
  - Variance- the tendency to learn random things irrespective of the real signal (More Scattered Points)

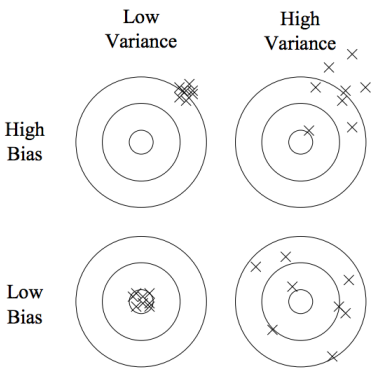


Figure 1: Bias and variance in dart-throwing.

Overfitting Has Many Faces

- Overfitting can be combatted with many methods
  - Cross-validation
  - Adding a regularization term to the evaluation function
  - Adding chi-square before adding new structure (statistical significance testing)
- Machine Learning Engineers should be skeptical
  - There is no single technique that will do this best
  - Overusing techniques can lead to overfitting and under fitting

Intuition Fails in High Dimensions

- After overfitting, the biggest problem in machine learning is the curse of dimensionality
- Generalizing becomes more difficult as dimensionality (number of features) increases
- The curse of dimensionality may outweigh the benefits of having more features in the model

Theoretical Guarantees are not what they seem

- The main role of Theoretical Guarantees in machine learning are as a source of understanding and driving force for algorithm design
- Learning is complex and just because a learner has theoretical justification and it works in practice does not mean it aligns completely

Feature Engineering Is the Key

- Why do some Machine Learning Projects Succeed or Fail ?
  - The most important factor is features used
- Feature engineering is more difficult as it is often domain specific
- Many beginner machine learning engineers are surprised as much of a their time is spent on gathering data, integrating it, cleaning it, and pre-processing it
- Machine learning is an iterative process – running the learner, analyzing the results, modifying the data or learner – repeat.

Feature Engineering Is the Key

- Automation of the feature engineering process is also often used recently
  - Ex: Selecting features based on their information gain in regards to class
  - Although these methods can help, machine learning engineers must be weary of the impact features can have on each other – Features in isolation may be different in combination

More Data Beats a Cleverer Algorithm

- "A dumb algorithm with lots and lots of data beats a clever one with modest amounts of it."
- Scalability is an issue
- In machine learning there are three main limited resources
  - Time
  - Memory
  - Training Data
- Although enormous mountains of data exist in this age it often takes too much time to process

More Data Beats a Cleverer Algorithm

- It is recommended to try the simpler learners first
  - Naïve bayes before logistic regression
  - K-nearest neighbor before support vector machines
- More sophisticated learners can get good results but often require more tuning to get desired results
- In research papers, most algorithms are typically compared by accuracy and computational cost
- Human Effort Saved and Insight gained are hard to measure but also very important factors

Learn Many Models, Not Just One

- Combination of many variations of models, often returns better results for little extra effort
- Model Ensembles are now standard practice
  - Bagging – generate random variations of the training set by resampling, learn a classifier on each, and combine the results by voting
  - Boosting – training examples have weights, and these are varied so that each new classifier focuses on the examples the previous got wrong
  - Stacking – the outputs of individual classifiers become the inputs of a "higher level" learner that figures out how to best combine them
- Many other techniques exist but the trend is toward larger ensembles

Learn Many Models, Not Just One

- Netflix Contest Example: Netflixprize.com
- Teams from all around the world competed to build the best video recommender system
- Teams found that they built the best recommender systems by combining their learners with other teams
- The winning team and runner up both use stacked ensembles of about 100 learners-
  - Combining these two also improved the results

Simplicity Does Not Imply Accuracy

- Occam's razor is an often quoted paper that claims entities should not be multiplied beyond necessity.
  - In machine learning this is often interpreted as simpler models being the most effective
  - On the contrary – there is no real connection between number of parameters of a model and overfitting tendency
- Another view equates complexity with the size of the hypothesis space
  - However- A learner with a large hypothesis space that tries fewer hypotheses is likely to over fit more than a learner that tries more hypothesis from a smaller space.
- Domingos states in his other paper based on surveys that simpler hypothesis should be preferred as simplicity is a virtue, not because of the hypothetical connection with accuracy.

Representable Does Not Imply Learnable

- Just because a function can be represented does not mean it can be learned
  - Ex: standard decision tree learners cannot learn trees with more leaves than the training data
- Limitations on data, memory, time, limit the functions that can be learnt in a practical manner
- The key question is not "Can It be represented?" but "Can it be learned?"

Correlation Does Not Imply Causation

- The goal of predictive models is often to use them as guides to action.
- Correlation may be related to a cause and effect relation but this needs to be investigated through experimentation.
- One must not take a correlation to be cause at its face value

Conclusion

- Machine learning has a lot of "folk wisdom" which many gain through experience and application
- This article summarized many useful knowledge applications related to improving model performance in machine learning
- I plan to keep these factors in mind during future model construction