

Resumo do artigo "A Few Useful Things to Know About Machine Learning"

Luís Carvalho, Afonso Hipólito

26 de fevereiro de 2023

Resumo

Este documento apresenta um resumo do artigo intitulado "A Few Useful Things to Know About Machine Learning" [1]. O artigo apresenta 12 lições importantes e de "*folk knowledge*" (conhecimento popular), que investigadores e profissionais, da área de aprendizagem automática aprenderam através da experiência e que são difíceis de encontrar em livros. *Machine learning* é usado em diversas áreas como pesquisas web, filtros de spam, sistemas de recomendação, *ad placement*, entre outros. O tipo mais usado de *machine learning* é classificação (*classifiers*), usado por exemplo numa aplicação de email para distinguir emails *spam* e outras categorias.

1 Learning = Representation + Evaluation + Optimization

Todos os algoritmos têm 3 componentes: *Representation*; *Evaluation*; *Optimization*. A representação ou "*hypothesis space*" (espaço de hipóteses) é o conjunto de funções que podem ser aprendidas, se não está no espaço de hipóteses, então não pode ser aprendido. A *evaluation* ou "*objective function or scoring function*" é uma avaliação, que pontua o quão bom é o modelo de aprendizagem automática. E o *optimization*, que é o método usado para procurar o modelo de aprendizagem automática superior ("*the highest-scoring one*").

2 It's Generalization that Counts

O objetivo principal da aprendizagem automática é generalizar para além dos exemplos no conjunto de treino. Por mais informação que tenhamos, é improvável que vejamos os mesmos exemplos no momento do teste. O algoritmo tem que sair bem em dados não vistos antes, ter uma boa exatidão de treino, pode também significar que o método apenas memorizou os exemplos antes vistos. Logo, é importante ter dados de testes separados para a avaliação final.

3 Data Alone is Not Enough

De acordo com a lição anterior, generalizar é o objetivo, então existe outra grande consequência, que é, dados sozinhos não são suficientes. Logo, todos os aprendizes devem combinar o conhecimento com esses dados para crescer. Muitas vezes, os alunos utilizam a dedução e indução, que é uma alavanca de conhecimentos, pois transforma uma pequena quantidade de conhecimento de “input” em grande quantidade de conhecimento “output”. Como o artigo cita: “*Farmers combine seeds with nutrients to grow crops. Learners combine knowledge with data to grow programs.*”

4 Overfitting Has Many Faces

Overfitting: Se o conhecimento e os dados que temos não são suficientes para classificar corretamente o classificador, então estamos simplesmente a codificar peculiaridades aleatórias nos dados e não a realidade. Ou seja, é o nome dado a um problema que ocorre quando a informação/data disponível não é suficiente para determinar um classificador correto. Este é um dos maiores problemas relacionados com machine learning.

Existem dois tipos de *data*: *training data* e *test data*. *Overfitting* acontece quando a *training data* tem mais exatidão do que a *test data*, no entanto nem sempre é “óbvio” detectar *overfits*.

Uma das maneiras de interpretar “*Overfitting*” é partir o erro de generalização em duas componentes:

- *Bias* - Tendência de o aluno aprender constantemente a mesma coisa errada.
- *Variance* - Tendência de o aluno aprender coisas aleatórias independentemente da realidade.

Algumas ferramentas contra o “*Overfitting*”:

- Suposições falsas fortes podem ser melhores do que as verdadeiras fracas (pois com a última o aluno precisa de mais dados para evitar o “*overfitting*”).
- A validação cruzada (“*Cross-validation*”), utilizada para escolher o melhor tamanho da árvore de decisões para aprender (No entanto se usado com muitos parâmetros pode resultar também em *overfit*);
- Prazo de regularização para avaliação da função. (“*regularization term*”), que penaliza classificadores mais estruturados, favorecendo os mais pequenos, que por sua vez tem menos margem para *overfit*;
- Aplicar um “*statistical significance test*” antes de adicionar uma nova estrutura para decidir se a distribuição da classe é diferente com e sem esta nova estrutura, útil quando a informação/data é escassa.

De notar que, apesar de haverem algumas opções para mitigar o *overfitting*, nenhuma resolve este problema a 100%.

Em suma, assim como há *overfitting* (variância) também existe *underfitting* (*bias*), que consiste no erro oposto ao primeiro. Ao evitar *overfitting*, podemos ter um problema maior de *underfitting*, e vice-versa. Para evitar os dois problemas simultaneamente seria necessário encontrar um classificador perfeito, e nenhuma única técnica o faz.

5 Intuition Fails in High Dimensions

Depois do “*Overfitting*”, um dos problemas na aprendizagem automática é a maldição da dimensionalidade (“*curse of dimensionality*”), ou seja, refere que muitos algoritmos que funcionam bem em baixas dimensões se tornam inacessíveis em altas dimensões, o mesmo acontece com a generalização, torna-se mais difícil à medida que o número de características dos exemplos aumenta. As nossas intuições, vêm de um mundo tridimensional, muitas vezes não se aplicam em mundos de alta dimensionalidade. Existe um método que neutraliza esta “maldição”, denominado de “*blessing of nonuniformity*”.

6 Theoretical Guarantees Are Not What They Seem

A aprendizagem automática está cheia de garantias teóricas. O tipo mais comum é um limite para o número de exemplos necessários para garantir uma boa generalização do modelo. Além disso, também diz que dada uma grande quantidade de dados de treinamento, o nosso algoritmo retornaria uma boa hipótese com alta probabilidade ou não encontrar uma hipótese consistente, ou seja, não nos diz sobre como selecionar um bom espaço de hipóteses. Outro tipo é o limite assintótico (“*asymptotic*”), que dado um número infinito de dados, o algoritmo garante produzir um classificador correto, sendo que na prática não temos dados infinitos. Em suma, as garantias teóricas não devem ser usadas como único critério para selecionar um algoritmo.

7 Feature Engineering Is The Key

A aprendizagem automática é um processo iterativo de executar o modelo, analisar os resultados, modificar os dados e/ou o modelo. E repetir. A engenharia de características é um passo crucial na aprendizagem automática, pois é um fator importante para decidir quais os projetos que têm sucesso. Ou seja, ter o tipo certo de características (independentes que se correlacionam bem com a classe) torna a aprendizagem mais fácil. No entanto, a engenharia de características também é difícil, uma vez que requer conhecimento específico do

domínio, que se estende para além dos dados que temos. Características que parecem irrelevantes isoladas, podem ser relevante em combinação.

8 More Data Beats a Cleverer Algorithm

Como regra geral, um algoritmo simples/menos inteligente com muitos dados é mais eficaz do que um algoritmo inteligente com uma quantidade modesta de dados. Porém, quanto mais dados, maior é a quantidade de problemas. São usados classificadores mais simples, porque os complexos levam muito tempo para aprender. Parte da solução é encontrar maneiras rápidas de aprender classificadores complexos (*“simpler classifiers wind up being used”*). Os alunos com tamanho fixo (*“parametric ones”*) podem aproveitar os dados apenas até certo ponto, pois se adicionar mais dados não melhora os resultados. Os alunos com tamanho variável (*“nonparametric learners”*) podem, teoricamente, aprender qualquer função com quantidade suficiente de dados, apesar de também serem limitados pelas limitações do algoritmo.

9 Learn Many Models, Not Just One

Hoje em dia, o foco está em combinar as diversas variantes de diferentes algoritmos, de modo a obter os melhores resultados (*“we combine many variations, the results are better, often much better”*). Técnicas de ensemblagem de modelos (*“model ensembles”*): *bagging*, *boosting* e *stacking*.

10 Simplicity Does Not Imply Accuracy

Na aprendizagem automática, de acordo com a declaração de *Occam’s*, dois classificadores com o mesmo erro de treino, o mais simples dos dois provavelmente terá o menor erro de teste. Mas, na verdade, há muitos contraexemplos para isso, como, por exemplo, o teorema *“no free lunch”*. A conclusão é que hipóteses mais simples devem ser preferidas porque a simplicidade é uma virtude em si mesma, não por causa de uma conexão hipotética com a precisão.

11 Representable Does Not Imply Learnable

Só porque uma função pode ser representada, não significa que a função possa realmente ser aprendida. Restrições impostas pelos dados, tempo e memória limitam as funções que podem ser efetivamente aprendidas. Encontrar métodos para aprender representações mais profundas é uma das principais fronteiras de pesquisa em aprendizagem de máquina.

12 Correlation Does Not Imply Causation

Na última secção do artigo é referido que a relação causa-relação não se aplica à *machine learning*.

Referências

- [1] Domingos, P. - A Few Useful Things to Know About Machine Learning. Tapping into the “folk knowledge” needed to advance machine learning applications. United States: Andrew A. Chien. 0001-0782. Volume 55 2012, 78-87.