Aprendizado de Máquina

Tipos de aprendizado de máquina

Supervisionado

Os algoritmos de aprendizado supervisionados fazem previsões com base em um conjunto de exemplos. Por exemplo, as cotações históricas podem ser usadas para arriscar palpites em preços futuros. Cada exemplo usado para treinamento é rotulado com o valor de seu interesse — neste caso, o preço da ação. Um algoritmo de aprendizado supervisionado procura por padrões nesses rótulos de valor. Ele pode usar qualquer informação que seja relevante — o dia da semana, a temporada, os dados financeiros da empresa, o tipo de setor, a presença de eventos geopolíticos perturbadores — e cada algoritmo procura tipos diferentes de padrões. Depois que o algoritmo tiver encontrado o melhor padrão possível, usará esse padrão para fazer previsões para dados de testes sem rótulos — os preços de amanhã.

Este é um tipo popular e útil de aprendizado de máquina. Com uma exceção, todos os módulos do Aprendizado de Máquina do Azure são algoritmos de aprendizado supervisionados. Há vários tipos específicos de aprendizado supervisionado representados no Aprendizado de Máquina do Azure: classificação, regressão e detecção de anomalias.

* **Classificação**. Quando os dados estiverem sendo usados para prever uma categoria, o aprendizado supervisionado também será chamado de classificação. Esse é o caso ao atribuir uma imagem como uma foto de um ‘gato’ ou de um ‘cachorro’. Quando há apenas duas opções, isso é chamado de **classificação binomial** ou de **duas classes**. Quando houver mais categorias, como na previsão do vencedor do torneio NCAA March Madness, esse problema é conhecido como **classificação multiclasse**.
* **Regressão**. Quando um valor estiver sendo previsto, assim como acontece com preços de cotações, o aprendizado supervisionado será chamado de regressão.
* **Detecção de anomalias**. Às vezes, o objetivo é identificar os pontos de dados que são simplesmente incomuns. Na detecção de fraudes, por exemplo, quaisquer padrões incomuns de gasto em cartão de crédito são suspeitos. As variações possíveis são tão numerosas e os exemplos de treinamento tão poucos, que não é viável aprender como seria uma atividade fraudulenta. A abordagem que usa detecção de anomalias é simplesmente aprender como seria uma atividade normal (usando transações não fraudulentas históricas) e identificar tudo que seja significativamente diferente.

Não supervisionado

No aprendizado não supervisionado, os pontos de dados não têm rótulos associados a eles. Em vez disso, a meta de um algoritmo de aprendizado sem supervisão é organizar os dados de alguma forma ou descrever sua estrutura. Isso pode significar agrupá-los em clusters ou encontrar diferentes maneiras de consultar dados complexos para que eles pareçam mais simples ou mais organizados.

Aprendizado de reforço

No aprendizado de reforço, o algoritmo escolhe uma ação em resposta a cada ponto de dados. O algoritmo de aprendizado também recebe um sinal de recompensa pouco tempo depois, indicando se a decisão foi boa. Com base nisso, o algoritmo modifica sua estratégia para alcançar a recompensa mais alta. Atualmente, não há nenhum módulo de algoritmo de reforço de aprendizado no Aprendizado de Máquina do Azure. O aprendizado de reforço é comum em robótica, em que o conjunto de leituras do sensor, em um ponto no tempo, é um ponto de dados e o algoritmo deve escolher a próxima ação do robô. Também é um ajuste natural para aplicativos da Internet das Coisas.

Considerações ao escolher um algoritmo

Precisão

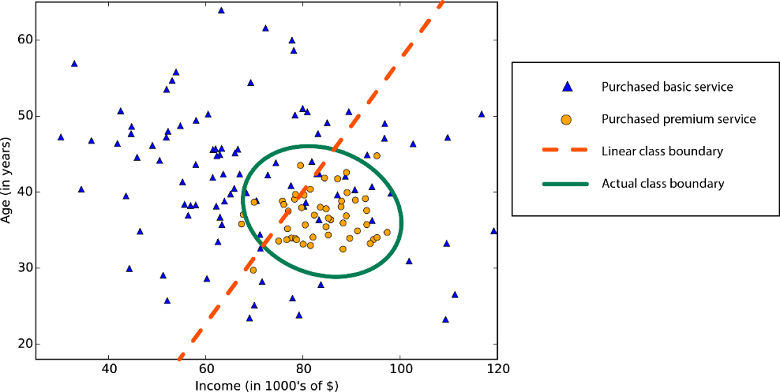
Obter a resposta mais precisa possível nem sempre será necessário. Às vezes uma aproximação será adequada, dependendo do uso que você quiser dar a ela. Se esse for o caso, talvez seja possível reduzir o tempo de processamento drasticamente usando métodos mais aproximados. Outra vantagem dos métodos mais aproximados é que eles naturalmente tendem a evitar o [superajuste](https://youtu.be/DQWI1kvmwRg).

Tempo de treinamento

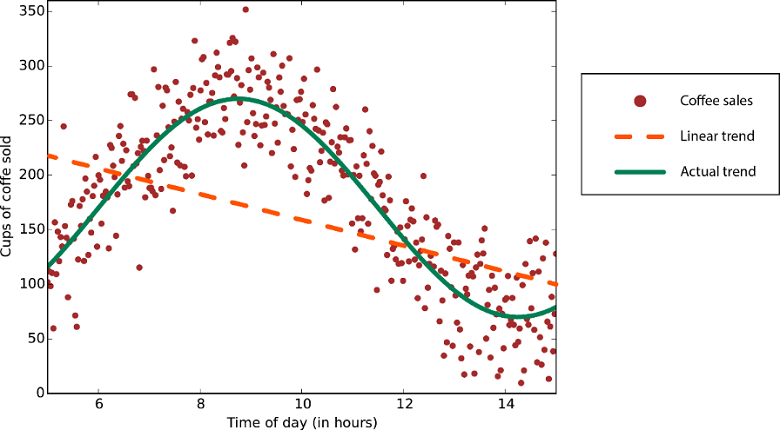
O número de minutos ou de horas necessários para treinar um modelo varia muito entre algoritmos. Em geral, o tempo de treinamento está intimamente vinculado à precisão — um normalmente acompanha o outro. Além disso, alguns algoritmos são mais sensíveis ao número de pontos de dados do que outros. Quando o tempo for limitado, ele poderá orientar a escolha do algoritmo, especialmente quando o conjunto de dados for grande.

Linearidade

Muitos algoritmos de aprendizado de máquina usam a linearidade. Os algoritmos de classificação linear supõem que as classes podem ser separadas por uma linha reta (ou seu análogo em dimensões maiores). Isso inclui a regressão logística e as máquinas de vetor de suporte (como implementado no Aprendizado de Máquina do Azure). Os algoritmos de regressão linear supõem que as tendências de dados seguem uma linha reta. Essas suposições não são ruins para alguns problemas, mas em outros elas podem reduzir a precisão.



***Limite de classe não linear*** *- contar com um algoritmo de classificação linear resultaria em baixa precisão*



***Dados com uma tendência não linear*** *- usar um método de regressão linear geraria erros muito maiores do que o necessário*

Apesar de seus riscos, os algoritmos lineares são muito populares como uma primeira linha de ataque. Eles tendem a ser algoritmicamente simples e rápidos de treinar.

Número de parâmetros

Os parâmetros são os botões que o cientista de dados precisa girar ao configurar um algoritmo. Eles são números que afetam o comportamento do algoritmo, como tolerância a erros ou o número de iterações, ou opções entre variantes de como o algoritmo se comporta. O tempo de treinamento e a precisão do algoritmo às vezes podem ser muito importantes para obter apenas as configurações corretas. Em geral, os algoritmos com um grande número de parâmetros exigem mais tentativas e erros para a localização de uma boa combinação.

Como alternativa, há um bloco de módulo de [limpeza de parâmetro](https://azure.microsoft.com/pt-br/documentation/articles/machine-learning-algorithm-parameters-optimize/) no Aprendizado de Máquina do Azure que experimenta automaticamente todas as combinações de parâmetro em qualquer granularidade que você escolher. Embora essa seja uma excelente maneira de certificar-se de que você tenha estendido o espaço de parâmetro, o tempo necessário para treinar o modelo aumenta exponencialmente com o número de parâmetros.

A vantagem é que ter muitos parâmetros geralmente indica que um algoritmo tem mais flexibilidade. Geralmente, isso pode significar uma precisão muito boa. Desde que você consiga encontrar a combinação certa de configurações de parâmetro.

Número de recursos

Para determinados tipos de dados, o número de recursos pode ser muito grande em comparação ao número de pontos de dados. Geralmente, isso acontece com dados de genética ou de texto. O grande número de recursos pode reduzir alguns algoritmos de aprendizado, fazendo com que o tempo de treinamento seja impraticável. As Máquinas de Vetor de Suporte são particularmente adequadas para esse caso (veja abaixo).

Casos especiais

Alguns algoritmos de aprendizado fazem suposições específicas sobre a estrutura de dados ou os resultados desejados. Se você conseguir encontrar um que atenda às suas necessidades, ele oferecerá resultados mais úteis, previsões mais exatas ou tempos de treinamento menores.

| **Algoritmo** | **Precisão** | **Tempo de treinamento** | **Linearidade** | **Parâmetros** | **Observações** |
| --- | --- | --- | --- | --- | --- |
| **Classificação de duas classes** |  |  |  |  |  |
| [regressão logística](https://msdn.microsoft.com/library/azure/dn905994.aspx) |  | ● | ● | 5 |  |
| [floresta de decisão](https://msdn.microsoft.com/library/azure/dn906008.aspx) | ● | ○ |  | 6 |  |
| [selva de decisão](https://msdn.microsoft.com/library/azure/dn905976.aspx) | ● | ○ |  | 6 | Volume de memória insuficiente |
| [árvore de decisão aumentada](https://msdn.microsoft.com/library/azure/dn906025.aspx) | ● | ○ |  | 6 | Grande volume de memória |
| [rede neural](https://msdn.microsoft.com/library/azure/dn905947.aspx) | ● |  |  | 9 | [A personalização adicional é possível](http://go.microsoft.com/fwlink/?LinkId=402867) |
| [perceptron médio](https://msdn.microsoft.com/library/azure/dn906036.aspx) | ○ | ○ | ● | 4 |  |
| [máquina de vetor de suporte](https://msdn.microsoft.com/library/azure/dn905835.aspx) |  | ○ | ● | 5 | Bom para conjuntos de recursos grandes |
| [máquina de vetor de suporte localmente profunda](https://msdn.microsoft.com/library/azure/dn913070.aspx) | ○ |  |  | 8 | Bom para conjuntos de recursos grandes |
| [computador do ponto de Bayes](https://msdn.microsoft.com/library/azure/dn905930.aspx) |  | ○ | ● | 3 |  |
| **Classificação multiclasse** |  |  |  |  |  |
| [regressão logística](https://msdn.microsoft.com/library/azure/dn905853.aspx) |  | ● | ● | 5 |  |
| [floresta de decisão](https://msdn.microsoft.com/library/azure/dn906015.aspx) | ● | ○ |  | 6 |  |
| [selva de decisão](https://msdn.microsoft.com/library/azure/dn905963.aspx) | ● | ○ |  | 6 | Volume de memória insuficiente |
| [rede neural](https://msdn.microsoft.com/library/azure/dn906030.aspx) | ● |  |  | 9 | [A personalização adicional é possível](http://go.microsoft.com/fwlink/?LinkId=402867) |
| [um contra todos](https://msdn.microsoft.com/library/azure/dn905887.aspx) | - | - | - | - | Consulte as propriedades do método de duas classes selecionadas |
| **Regressão** |  |  |  |  |  |
| [linear](https://msdn.microsoft.com/library/azure/dn905978.aspx) |  | ● | ● | 4 |  |
| [Linear Bayesiano](https://msdn.microsoft.com/library/azure/dn906022.aspx) |  | ○ | ● | 2 |  |
| [floresta de decisão](https://msdn.microsoft.com/library/azure/dn905862.aspx) | ● | ○ |  | 6 |  |
| [árvore de decisão aumentada](https://msdn.microsoft.com/library/azure/dn905801.aspx) | ● | ○ |  | 5 | Grande volume de memória |
| [quantil rápido de floresta](https://msdn.microsoft.com/library/azure/dn913093.aspx) | ● | ○ |  | 9 | Distribuições em vez de previsões de ponto |
| [rede neural](https://msdn.microsoft.com/library/azure/dn905924.aspx) | ● |  |  | 9 | [A personalização adicional é possível](http://go.microsoft.com/fwlink/?LinkId=402867) |
| [Poisson](https://msdn.microsoft.com/library/azure/dn905988.aspx) |  |  | ● | 5 | Tecnicamente linear em log. Para previsão de contagem |
| [ordinal](https://msdn.microsoft.com/library/azure/dn906029.aspx) |  |  |  | 0 | Para previsão de ordem de classificação |
| **Detecção de anomalias** |  |  |  |  |  |
| [computador de vetor de suporte](https://msdn.microsoft.com/library/azure/dn913103.aspx) | ○ | ○ |  | 2 | Especialmente bom para conjuntos de recursos grandes |
| [Detecção de anomalias baseada em PCA](https://msdn.microsoft.com/library/azure/dn913102.aspx) |  | ○ | ● | 3 |  |
| [K-means](https://msdn.microsoft.com/library/azure/5049a09b-bd90-4c4e-9b46-7c87e3a36810/) |  | ○ | ● | 4 | Um algoritmo de clustering |

**Propriedades do algoritmo:**

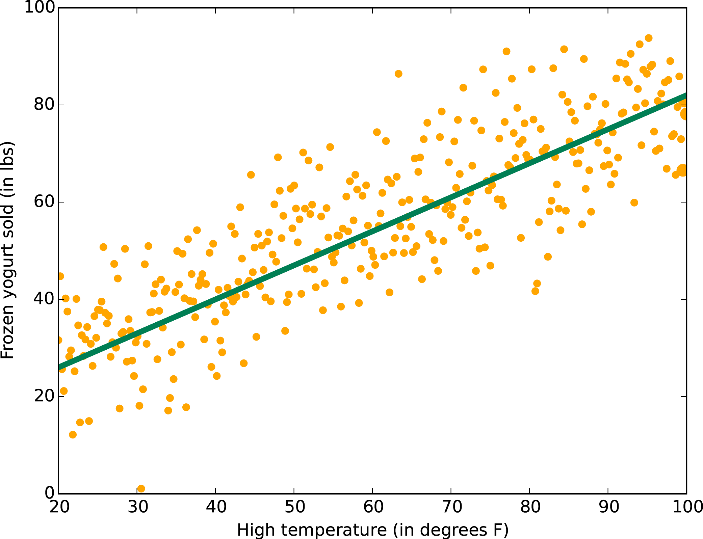
**●** - mostra excelente precisão, tempos de treinamento pequenos e o uso de linearidade

**○** - mostra boa precisão e tempos de treinamento moderados

Anotações sobre algoritmo

Regressão linear

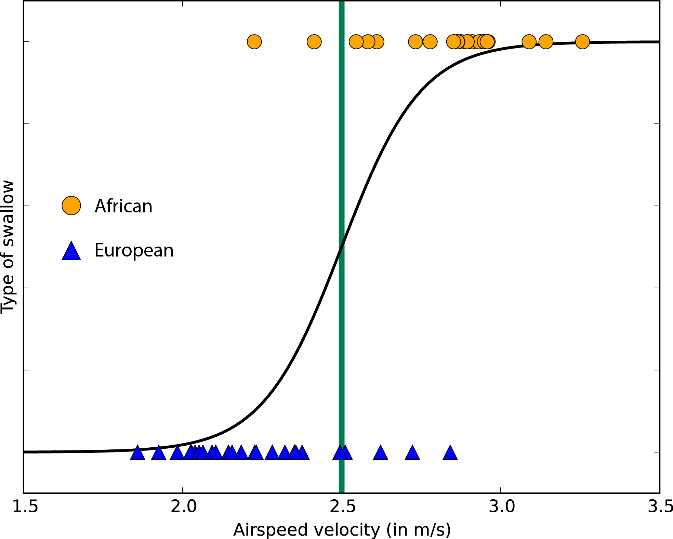
Como mencionado anteriormente, a [regressão linear](https://msdn.microsoft.com/library/azure/dn905978.aspx) ajusta uma linha (ou plano ou hiperplano) ao conjunto de dados. Ela é uma força de trabalho, simples e rápida, mas pode ser muito simplista para alguns problemas. Veja aqui um [tutorial de regressão linear](https://azure.microsoft.com/pt-br/documentation/articles/machine-learning-linear-regression-in-azure/).



***Dados com uma tendência linear***

Regressão logística

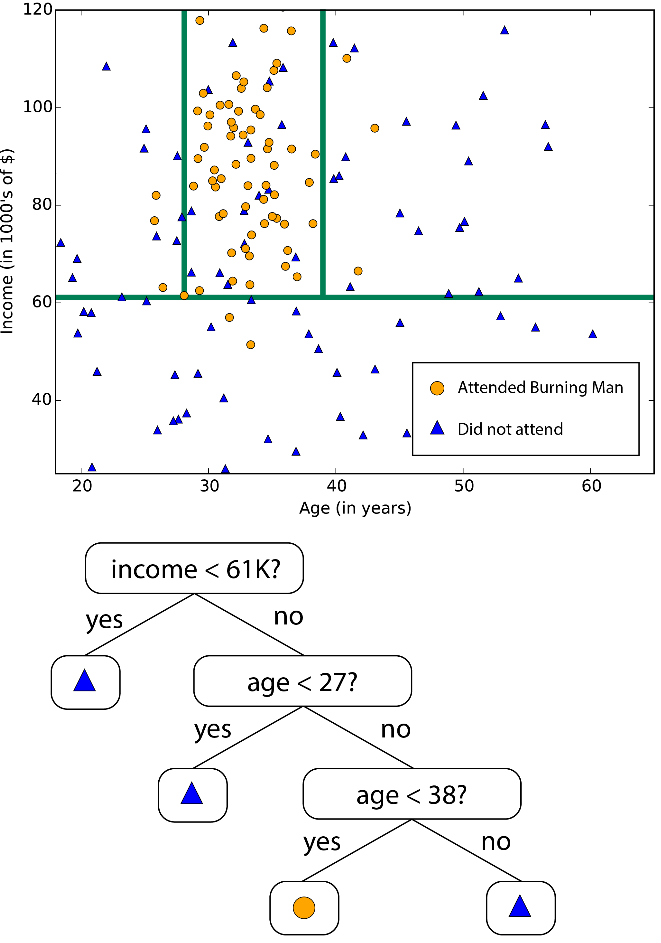
Embora inclua erroneamente ‘regressão’ no nome, a regressão logística é, na verdade, uma poderosa ferramenta para a classificação de [duas classes](https://msdn.microsoft.com/library/azure/dn905994.aspx) e [multiclasse](https://msdn.microsoft.com/library/azure/dn905853.aspx). É rápida e simples. O fato de que ela usa uma curva em forma de ‘S’ em vez de uma linha reta faz com que ela seja uma opção natural para a divisão de dados em grupos. A regressão logística oferece limites de classe lineares e, portanto, quando for usá-la, verifique se uma aproximação linear serve para você.



***Uma regressão logística para dados de duas classes com apenas um recurso***\* - o limite de classe é o ponto no qual a curva logística é mais próxima de ambas as classes\*

Árvores, florestas e selvas

Florestas de decisão ([regressão](https://msdn.microsoft.com/library/azure/dn905862.aspx)[duas classes](https://msdn.microsoft.com/library/azure/dn906008.aspx) e [multiclasse](https://msdn.microsoft.com/library/azure/dn906015.aspx)), selvas de decisão ([duas classes](https://msdn.microsoft.com/library/azure/dn905976.aspx) e[multiclasse](https://msdn.microsoft.com/library/azure/dn905963.aspx)) e árvores de decisão aumentadas ([regressão](https://msdn.microsoft.com/library/azure/dn905801.aspx) e [duas classes](https://msdn.microsoft.com/library/azure/dn906025.aspx)) são todos baseados em árvores de decisão, um conceito fundamental de aprendizado de máquina. Há muitas variantes de árvores de decisão, mas todas fazem a mesma coisa — subdividir o espaço de recursos em regiões com basicamente o mesmo rótulo. Elas podem ser regiões de categoria consistente ou de valor constante, dependendo se você estiver fazendo classificação ou regressão.



***Uma árvore de decisão subdivide um espaço de recursos em regiões de valores aproximadamente uniformes***

Como um espaço de recurso pode ser subdividido em pequenas regiões de forma arbitrária, fica fácil imaginar a divisão minuciosa o suficiente para ter um ponto de dados por região – um exemplo extremo de superajuste. Para evitar isso, um grande conjunto de árvores é criado com cuidado matemático especial para que as árvores não estejam correlacionadas. A média dessa "floresta de decisão" é uma árvore que evita o superajuste. As florestas de decisão podem usar muita memória. As selvas de decisão são uma variante que consome menos memória às custas de um tempo de treinamento um pouco mais longo.

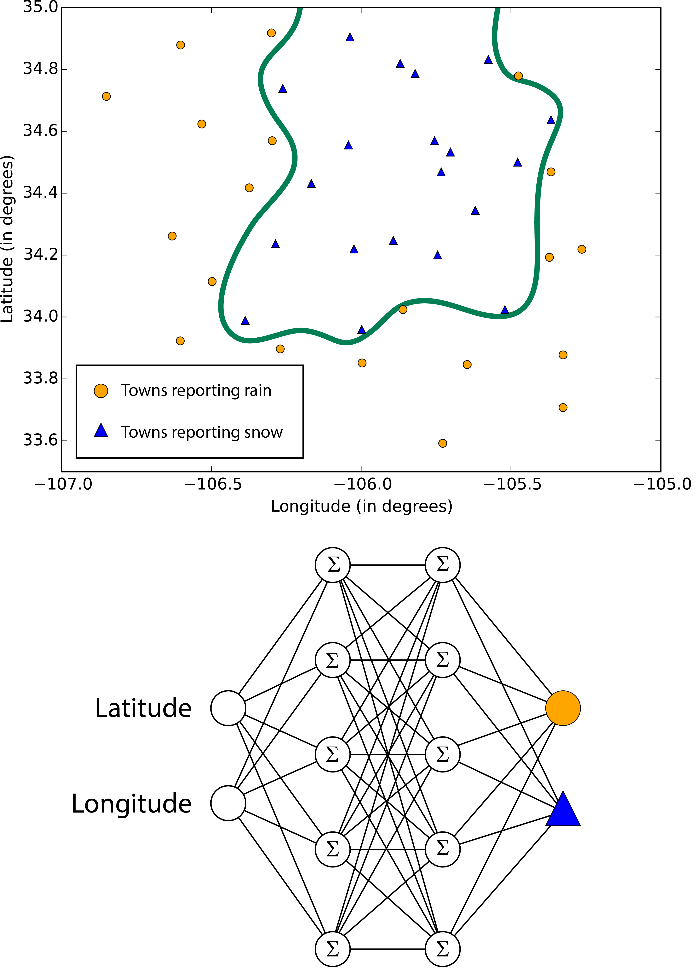
As árvores de decisão aumentadas evitam o superajuste ao limitarem o número de vezes que podem se subdividir e como poucos pontos de dados são permitidos em cada região. O algoritmo constrói uma sequência de árvores, e cada uma delas aprende a compensar o erro deixado pela árvore anterior. O resultado é um aprendiz muito preciso que tende a usar muita memória. Para obter a descrição técnica completa, confira o [documento original de Friedman](http://www-stat.stanford.edu/~jhf/ftp/trebst.pdf).

[Regressão rápida de quantil de floresta](https://msdn.microsoft.com/library/azure/dn913093.aspx) é uma variação de árvores de decisão para casos especiais onde você deseja conhecer não apenas o valor típico (médio) dos dados em uma região, mas também sua distribuição na forma de quantis.

Redes neurais e perceptrons

As redes neurais são algoritmos de aprendizado inspirados no cérebro que cobrem problemas[multiclasse](https://msdn.microsoft.com/library/azure/dn906030.aspx), de [duas classes](https://msdn.microsoft.com/library/azure/dn905947.aspx) e de [regressão](https://msdn.microsoft.com/library/azure/dn905924.aspx). Elas vêm em uma variedade infinita, mas as redes neurais do Aprendizado de Máquina do Azure estão todas na forma de gráficos acíclicos direcionados. Isso significa que os recursos de entrada são passados para a frente (nunca para trás) por meio de uma sequência de camadas transformadas em saídas. Em cada camada, as entradas são ponderadas em várias combinações, somadas e passadas para a próxima camada. Essa combinação de cálculos simples resulta na capacidade de aprender limites de classe e tendências de dados sofisticados, aparentemente mágicos. As redes de várias camadas desse tipo executam o “aprendizado profundo” que alimenta tantos relatórios técnicos e a ficção científica.

No entanto, esse alto desempenho tem um preço. As redes neurais podem ter um treinamento muito longo, particularmente para grandes conjuntos de dados com muitos recursos. Elas também têm mais parâmetros do que a maioria dos algoritmos, o que significa que a limpeza de parâmetros aumenta muito o tempo de treinamento. E para as pessoas brilhantes que desejam [especificar sua própria estrutura de rede](http://go.microsoft.com/fwlink/?LinkId=402867), as possibilidades são inesgotáveis.

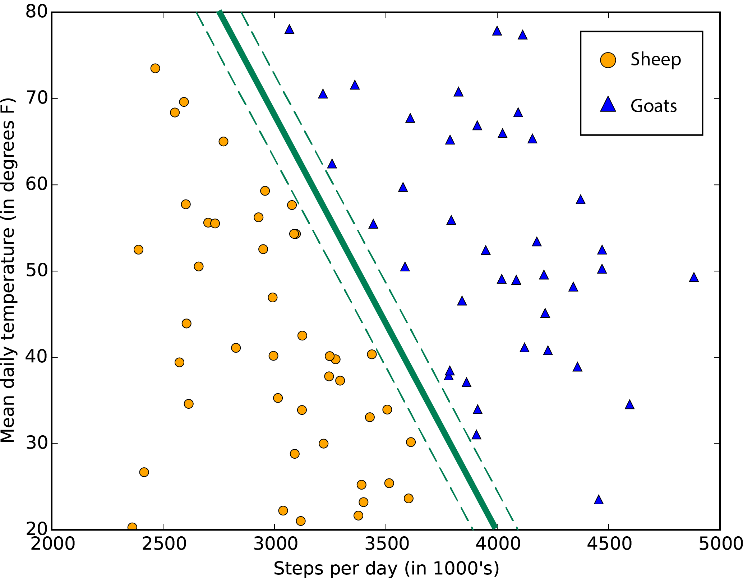


***Os limites aprendidos por redes neurais podem ser complexos e irregulares***

O [perceptron médio de duas classes](https://msdn.microsoft.com/library/azure/dn906036.aspx) é a resposta das redes neurais para os tempos de treinamento estratosféricos. Ele usa uma estrutura de rede que fornece os limites de classe linear. É quase primitivo pelos padrões de hoje, mas tem uma longa história de trabalho robusto e é pequeno o suficiente para aprender rapidamente.

SVMs

As máquinas de vetor de suporte (SVMs) encontram o limite que separa as classes pela maior margem possível. Quando as duas classes não puderem ser claramente separadas, os algoritmos encontrarão o melhor limite possível. Como escrito no Aprendizado de Máquina do Azure, a [SVM de duas classes](https://msdn.microsoft.com/library/azure/dn905835.aspx) faz isso com apenas uma linha reta. (No jargão da SVM, usa um kernel linear). Como ela faz essa aproximação linear, é capaz de ser executada de maneira consideravelmente rápida. Mas o seu verdadeiro ponto forte são os dados de uso intenso de recursos, como texto ou genoma. Nesses casos, as SVMs são capazes de separar classes mais rapidamente e com menos superajuste do que a maioria dos outros algoritmos, além de exigir apenas uma pequena quantidade de memória.



***Um limite de classe suporte de computador de vetor de suporte típico maximiza a margem que separa duas classes***

Outro produto da Microsoft Research, a [SVM localmente profunda de duas classes](https://msdn.microsoft.com/library/azure/dn913070.aspx) é uma variante não linear de SVM que mantém a maior parte da eficiência de velocidade e de memória da versão linear. Ela é ideal para casos em que a abordagem linear não oferece respostas exatas o suficiente. Os desenvolvedores a mantiveram rápida ao dividir o problema em pequenos problemas lineares da SVM. Leia a [descrição completa](http://research.microsoft.com/um/people/manik/pubs/Jose13.pdf) para obter detalhes sobre como eles realizaram esse truque.

Usando uma extensão inteligente de SVMs não lineares, a [SVM de uma classe](https://msdn.microsoft.com/library/azure/dn913103.aspx) cria um limite que descreve totalmente o conjunto de dados inteiro. Ela é útil para a detecção de anomalias. Qualquer ponto de dados que ficar de fora desse limite será incomum o suficiente para ser notado.

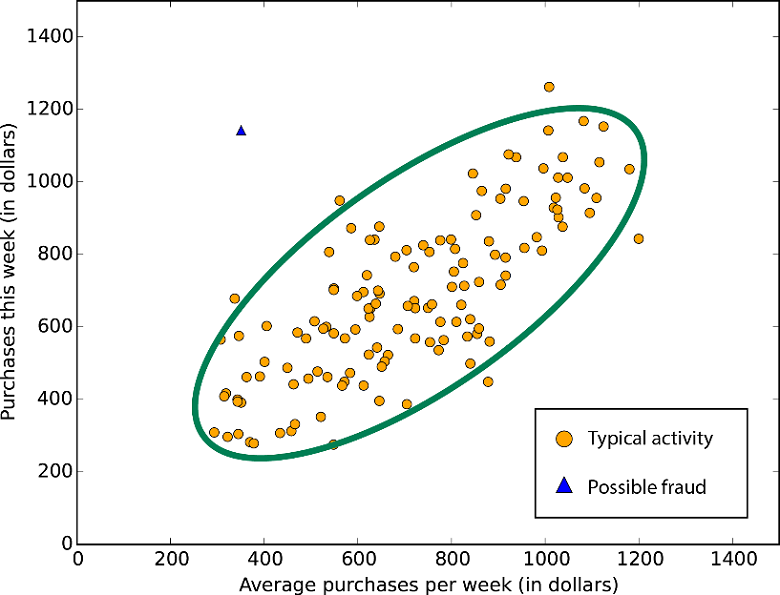
Métodos Bayesianos

Os métodos Bayesianos possuem uma qualidade altamente desejável: eles evitam o superajuste. Eles fazem isso fazendo algumas suposições com antecedência sobre a distribuição provável da resposta. Outro subproduto dessa abordagem é que eles têm muito poucos parâmetros. O Aprendizado de Máquina do Azure tem dois algoritmos Bayesianos para classificação ([Computador do ponto de Bayes de duas classes](https://msdn.microsoft.com/library/azure/dn905930.aspx)) e regressão ([regressão linear Bayesiana](https://msdn.microsoft.com/library/azure/dn906022.aspx)). Observe que eles supõem que os dados podem ser divididos ou ajustados com uma linha reta.

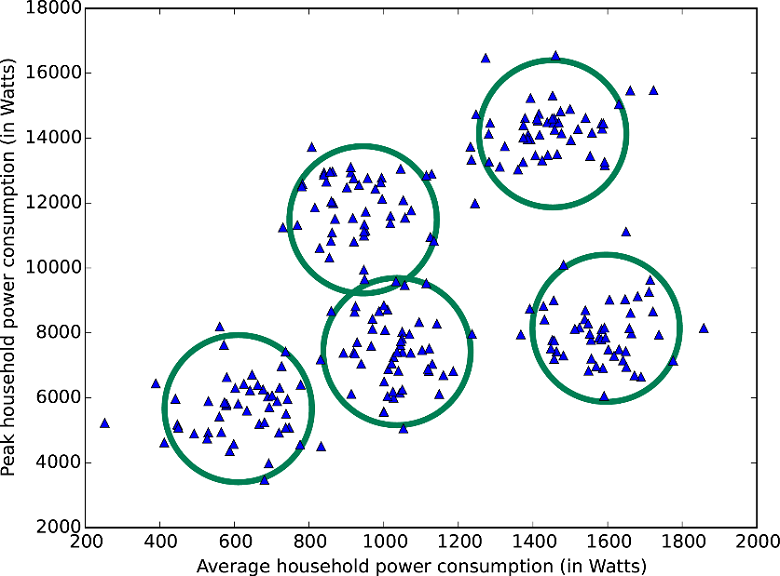
Em uma nota histórica, os computadores de ponto de Bayes foram desenvolvidos na Microsoft Research. Eles têm algum trabalho teórico excepcionalmente belo por trás deles. O aluno interessado será direcionado para o [artigo original no JMLR](http://jmlr.org/papers/volume1/herbrich01a/herbrich01a.pdf) e para um [blog perspicaz de Chris Bishop](http://blogs.technet.com/b/machinelearning/archive/2014/10/30/embracing-uncertainty-probabilistic-inference.aspx).

Algoritmos especializados

Se você tiver um objetivo muito específico, este talvez seja o seu dia de sorte. Dentro da coleção do Aprendizado de Máquina do Azure, existem algoritmos especializados em previsão de classificação ([regressão ordinal](https://msdn.microsoft.com/library/azure/dn906029.aspx)), previsão de contagem ([regressão Poisson](https://msdn.microsoft.com/library/azure/dn905988.aspx)) e detecção de anomalias (com base na [análise dos principais componentes](https://msdn.microsoft.com/library/azure/dn913102.aspx) um baseado em [máquinas de vetor de suporte](https://msdn.microsoft.com/library/azure/dn913103.aspx)). E há também um algoritmo de clustering solitário ([K-means](https://msdn.microsoft.com/library/azure/5049a09b-bd90-4c4e-9b46-7c87e3a36810/)).

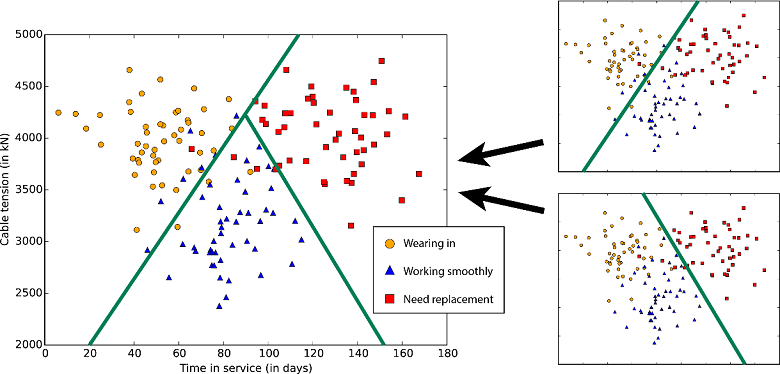


***Detecção de anomalias baseada em PCA***\* - a grande maioria dos dados se encaixa em uma distribuição estereotipada; os pontos que desviarem drasticamente dessa distribuição são suspeitos\*



***Um conjunto de dados é agrupado em 5 clusters usando K-means***

Há também um conjunto [classificador multiclasse um contra todos](https://msdn.microsoft.com/library/azure/dn905887.aspx), que divide o problema de classificação de classe N em dois problemas de classificação de classe N-1. A precisão, o tempo de treinamento e as propriedades de linearidade são determinados pelos classificadores de duas classes usados.



***Um par de classificadores de duas classes combinados para formar um classificador de três classes***

O Aprendizado de Máquina do Azure também inclui acesso a uma poderosa estrutura de aprendizado de máquina sob o título de [Vowpal Wabbit](https://msdn.microsoft.com/library/azure/8383eb49-c0a3-45db-95c8-eb56a1fef5bf). O VW desafia a categorização aqui, já que pode aprender problemas de classificação e de regressão e pode até aprender de dados parcialmente sem rótulo. Você pode configurá-lo para usar qualquer um de vários algoritmos, funções de perda e algoritmos de otimização de aprendizado. Ele foi projetado desde o início para ser eficiente, paralelo e extremamente rápido. Ele lida com conjuntos de recursos absurdamente grandes com pouco esforço aparente. Iniciado e liderado pelo próprio John Langford, da Microsoft Research, o VW é uma entrada de Fórmula 1 em um campo de algoritmos de stock cars. Nem todo problema se ajustará ao VW, mas se ele se ajustar, poderá valer a pena seguir a curva de aprendizado em sua interface. Ele também está disponível como[código-fonte aberto autônomo](https://github.com/JohnLangford/vowpal_wabbit) em várias linguagens.