Uma Abordagem Probabilística ao Machine Learning

O que a Estatística tem a ver com Machine Learning?

Nate Silver é talvez uma das figuras mais proeminente da ciência de dados hoje. Um analista quantitativo que deixou um trabalho de consultoria de gestão para desenvolver métodos de previsão de beisebol, ele ganhou fama por meio de seu site de previsão eleitoral http://www.fivethirtyeight.com. Aqui ele usou métodos quantitativos para analisar os resultados das pesquisas eleitoriais para prever os resultados das eleições presidenciais dos EUA. Na eleição de 2008, ele acertou com precisão o vencedor de 49 dos 50 estados e melhorou em 2012 para 50 em 50. Os resultados da eleição de 2016 foram um choque para quase todos.Entre os comentaristas públicos o Nate Silver era o unico que identificou uma chance substancial de Trump ganhar o colégio eleitoral enquanto perdia o voto popular. Isso realmente provou ser o cara!

Silver escreveu um excelente livro *The Signal and the Noise: Why so many predictions fail – but some don't*. Lá ele escreve sensatamente sobre a previsão de estado-de-arte (*state-of-the-art*) em vários campos, incluindo esportes, clima, previsão de terremoto e modelagem financeira. Ele descreve os princípios para modelagem eficaz, Incluindo:

- **Pense probabilisticamente:** as previsões que fazem declarações concretas são menos significativos do que aqueles que são inerentemente probabilísticos. Uma previsão de que Lula/Bolsonaro tem apenas 47,7% de chance de vencer é mais significativa do que outra que afirma categoricamente que um deles perderá. O mundo real é um lugar incerto e os modelos de sucesso reconhecem essa incerteza. Há sempre uma gama de resultados possíveis que podem ocorrer com ligeiras perturbações da realidade, e isso deve ser capturado em seu modelo. As previsões de quantidades numéricas não devem ser números únicos, mas, em vez disso, relatar distribuições de probabilidade. A especificação de um desvio padrão σ juntamente com a previsão média μ é suficiente para descrever tal distribuição, particularmente se for considerada normal. Várias das técnicas de Machine Learning que estudaremos/estudamos fornecem naturalmente respostas probabilísticas. A regressão logística fornece uma confiança junto com cada classificação que faz. Métodos que votam entre os rótulos dos k vizinhos mais próximos definem uma medida de

confiança natural, baseada na consistência dos rótulos da vizinhança. Coletar dez dos onze votos para o azul significa algo mais forte do que sete dos onze.

- **Mude sua previsão em resposta a novas informações:** os modelos vivos são muito mais interessantes do que os mortos. Um modelo está ativo se estiver atualizando continuamente as previsões em resposta a novas informações. Construir uma infraestrutura que mantém um modelo vivo é mais complicado do que uma computação única, mas muito mais valioso. Os modelos vivos são mais intelectualmente honestos do que os mortos. Novas informações devem mudar o resultado de qualquer previsão. Os cientistas devem estar abertos a mudanças de opinião em resposta a novos dados: de fato, é isso que separa os cientistas dos hacks e trolls. As previsões que mudam dinamicamente oferecem excelentes oportunidades para avaliar seu modelo. Eles finalmente convergem para a resposta correta? A incerteza diminui à medida que o evento se aproxima? Qualquer modelo ao vivo deve rastrear e exibir suas previsões ao longo do tempo, para que o visualizador possa avaliar se as alterações refletiram com precisão o impacto das novas informações.
- **Procure consenso**: uma boa previsão vem de várias fontes distintas de evidência. Os dados devem derivar de tantas fontes diferentes quanto possível. Idealmente, vários modelos devem ser construídos, cada um tentando prever a mesma coisa de maneiras diferentes. Você deve ter uma opinião sobre qual modelo é o melhor, mas se preocupe quando ele diferir substancialmente do rebanho. Frequentemente, outros grupos produzem previsões concorrentes, que você pode monitorar e comparar. Ser diferente não significa que você está errado, mas fornece uma verificação da realidade. Quem tem se saído melhor ultimamente? O que explica as diferenças nas previsões? Seu modelo pode ser melhorado? O modelo de previsão *Flu Trends* do Google previu surtos de doenças monitorando palavras-chave na pesquisa: um aumento no número de pessoas que procuram aspirina ou febre pode sugerir que a doença está se espalhando. O modelo de previsão do Google provou ser bastante consistente com as estatísticas do Centro de Controle de Doenças (CDC) sobre casos reais de gripe por vários anos, até que eles se desviaram de forma embaraçosa. O mundo muda. Entre as mudanças, a interface de pesquisa do Google começou a sugerir consultas de pesquisa em resposta ao histórico do usuário. Quando oferecida a sugestão, muito mais pessoas começaram a procurar aspirina depois de procurar febre. E o modelo antigo de repente não era mais preciso. Os pecados do Google estão em não monitorar seu desempenho e ajustar ao longo do tempo. Certos métodos de Machine Learning buscam explicitamente o consenso. Algoritmos de *Boosting* combinam um grande número de classificadores fracos para produzir um classificador forte. Os métodos de conjunto (*ensemble*) de árvore de decisão constroem muitos classificadores independentes e votam entre eles para tomar a melhor decisão. Tais métodos podem ter uma robustez que ilude modelos mais *single-track*.
- **Empregue o raciocínio Baysiano**: o teorema de Bayes tem várias interpretações, mas talvez o mais convincente forneça uma maneira de calcular como as probabilidades

mudam em resposta a novas evidências. Quando declarado como

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

fornece uma maneira de calcular como a probabilidade do evento A muda em resposta à nova evidência B. A aplicação do teorema de Bayes requer uma probabilidade *a priori* $P(A)$, a probabilidade do evento A *antes* de saber o status de um determinado evento B. Isso pode ser o resultado da execução de um classificador para prever o status de A a partir de outros recursos ou conhecimento prévio sobre frequências de eventos em uma população. Sem uma boa estimativa para este prior, é muito difícil saber com que seriedade levar o classificador. Suponha que A seja o evento em que a pessoa x é realmente um traficante e B seja o resultado de um classificador baseado em características que decide se x se parece com um traficante. Quando treinado/avaliado em um conjunto de dados de 1000 pessoas, metade das quais eram traficantes, o classificador alcançou uma precisão invejável de, digamos, 90%. O classificador agora diz que o Vahid parece um traficante. Qual é a probabilidade de Vahid ser realmente uma traficante? A principal percepção aqui é que a probabilidade *a priori* de "x é um traficante" é muito, muito baixa. Se houver cem traficante operando Brasil, então $P(A) = 100/200,000,000 = 0,5 \times 10^{-7}$. A probabilidade do detector de traficante dizer sim, $P(B) = 0,5$, enquanto a probabilidade do detector acertar ao dizer sim $P(B|A) = 0,9$. Multiplicar isso dá uma probabilidade ainda muito pequena de que eu seja um cara mau,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{(0,9)(0,5 \times 10^{-7})}{(0,5)}$$

embora reconhecidamente agora maior do que a de um cidadão aleatório. Fatorar probabilidades *a priori* é essencial para obter a interpretação correta desse classificador. O raciocínio bayesiano começa com a distribuição *a priori* e, em seguida, pesa mais evidências sobre a intensidade com que deve impactar a probabilidade do evento.

Classificadores Bayesiano Naive (Ingênuo)

Os classificadores *Naive Bayes* são uma família de "classificadores probabilísticos" simples baseados na aplicação do teorema de Bayes com suposições de independência fortes (ingênuas) entre os features. Eles estão entre os modelos de rede bayesiana mais simples, mas, juntamente com a estimativa de densidade do kernel, podem atingir altos níveis de precisão. Os classificadores Naive Bayes são altamente escaláveis, exigindo um número de parâmetros linearmente dependendo do número de variáveis (features/preditores) em um problema de aprendizagem.

Independência de Eventos

Lembre-se de que dois eventos A e B são independentes se $p(AeB) = p(A) \cdot p(B)$. Se A é o evento em que "meu time favorito ganha hoje" e B é "o mercado de ações sobe hoje", então

presumivelmente A e B são independentes. Mas isso não é verdade em geral. Considere o caso se A for o evento "Recebi 10 em Ciência de Dados neste semestre" e B é "Recebi 10 em um curso diferente neste semestre". Existem dependências entre esses eventos: entusiasmos renovados por ambos estudar ou beber afetará o desempenho do curso de maneira correlata. No caso geral,

$$p(A \text{ e } B) = p(A) + p(B) - p(A \text{ ou } B)$$

Se tudo fosse independente, o mundo da probabilidade seria um lugar muito mais simples. O algoritmo de classificação de Bayes ingênuo cruza os dedos e assume a independência de eventos, para evitar a necessidade de calcular essas probabilidades condicionais confusas.

Formulação

Suponha que desejamos classificar o vetor $X = (x_1, ..., x_n)$ em uma das m classes $C_1, ..., C_m$. Procuramos calcular a probabilidade de cada possível classe, dado o vetor X. Então podemos atribuir ao X o rótulo da classe com maior probabilidade. Pelo teorema de Bayes,

$$p(C_i|X) = \frac{p(C_i) \cdot p(X|C_i)}{p(X)}$$

Vamos analisar esta equação. O termo $p(C_i)$ é a probabilidade anterior, a probabilidade do rótulo da classe sem nenhuma evidência específica. Eu sei que você, leitor, tem mais probabilidade de ter cabelo preto do que ruivo, porque mais pessoas no mundo têm cabelo preto do que cabelo ruivo. O termo $p(X|C_i)$ é a probabilidade de observar o vetor X dado que a classe é C_i. Por exemplo, se X é um vetor de características de um documento, então $p(X|C_i)$ é a probabilidade de observar esse documento dado que a classe é C_i. Deve ser claro que $p(X|C_i)$ geralmente será muito pequeno: existe um enorme espaço de possíveis vetores de entrada consistentes com a classe, dos quais apenas um corresponde ao item fornecido. O termo $p(C_i|X)$ é a probabilidade posterior, a probabilidade do rótulo da classe dado o vetor X. O denominador $P(X)$ dá a probabilidade de ver o dado vetor de entrada X sobre todos os vetores de entrada possíveis. Estabelecer o valor exato de $p(X)$ parece um tanto arriscado, mas felizmente geralmente é desnecessário. Observe que esse denominador é o mesmo para todas as classes. Procuramos apenas estabelecer um rótulo de classe para X, então o valor de $p(X)$ não tem efeito em nossa decisão. Selecionar a classe com maior probabilidade significa

$$C(X) = \underset{i=1,...,m}{\text{argmax}} \frac{p(C_i) \cdot p(X|C_i)}{p(X)} = \underset{i=1,...,m}{\text{argmax}} p(C_i) \cdot p(X|C_i)$$

Mas agora suponha que vivêssemos onde tudo fosse independente, ou seja, a probabilidade do evento A e do evento B fosse sempre $p(A) \cdot p(B)$. Então

$$p(X|C_i) = \prod_{j=1}^{n} p(x_j|C_i)$$

Agora, qualquer um que realmente acredite em um mundo de probabilidades independentes é bastante ingênuo, daí o nome de *ingênuo (Naive) Bayes*. Mas tal suposição realmente torna os cálculos muito mais fáceis. Juntando isso:

$$C(X) = \underset{i=1,\ldots,m}{\text{argmax}} p(C_i) \cdot p(X|C_i) = \underset{i=1,\ldots,m}{\text{argmax}} p(C_i) \prod_{j=1}^{n} p(x_j|C_i)$$

Por fim, devemos um log no operador produto ($\prod$) para transformá-lo em soma (Σ), para melhor estabilidade numérica. Os logs de probabilidades serão números negativos, mas os eventos menos prováveis são mais negativos do que os comuns. Assim, o algoritmo de Bayes ingênuo completo é dado pela seguinte fórmula:

$$C(X) = \underset{i=1,\ldots,m}{\text{argmax}} (\log p(C_i) + \sum_{j=1}^{n} \log p(x_j|C_i)).$$

Como calculamos o $p(x_j|C_i)$, a probabilidade de observação x_j dada rótulo de classe i? Isso é fácil a partir dos dados de treinamento, principalmente se x_j for uma variável categórica, como "tem cabelo ruivo". Podemos simplesmente selecionar todas as instâncias de classe i no conjunto de treinamento e calcular a fração delas que possui a propriedade x_j. Esta fração define uma estimativa razoável de $p(x_j|Ci)$. Um pouco mais de imaginação é necessário quando x_j é uma variável numérica, como "idade = 18" ou "a palavra cachorro ocorreu seis vezes no documento fornecido", mas, em princípio, é calculado pela frequência com que esse valor é observado no conjunto de treinamento.

A Figure 1 ilustra o procedimento ingênuo de Bayes. À esquerda, apresenta uma tabela com dez observações das condições meteorológicas, e se cada observação provou ser um dia para ir à praia ou ficar em casa. Esta tabela foi discriminadas à direita, para produzir probabilidades condicional da condição do tempo, dada a atividade. A partir dessas probabilidades, podemos usar o teorema de Bayes para calcular:

P (Beach|(Sunny,Mild,High)) = (P (Sunny|Beach) × P (Mild|Beach) × P (High|Beach) × P (Beach) = (3/4) × (1/4) × (2/4) × (4/10) = 0.0375

P (No Beach|(Sunny,Mild,High)) = (P (Sunny|No) × P (Mild|No) × P (High|No)) × P (No) = (1/6) × (2/6) × (2/6) × (6/10) = 0.0111

Como 0,0375 > 0,0111, o Naive Bayes está nos dizendo para ir à praia. Observe que é irrelevante que essa combinação específica de (Sunny,Mild,High) tenha aparecido nos dados de treinamento. Estamos baseando a nossa decisão nas probabilidades agregadas, não em uma única linha como na classificação do vizinho mais próximo.

Day	Outlook	Temp	Humidity	Beach?
1	Sunny	High	High	Yes
2	Sunny	High	Normal	Yes
3	Sunny	Low	Normal	No
4	Sunny	Mild	High	Yes
5	Rain	Mild	Normal	No
6	Rain	High	High	No
7	Rain	Low	Normal	No
8	Cloudy	High	High	No
9	Cloudy	High	Normal	Yes
10	Cloudy	Mild	Normal	No

P(X|Class)	Probability in Class	
Outlook	Beach	No Beach
Sunny	3/4	1/6
Rain	0/4	3/6
Cloudy	1/4	2/6
Temperature	Beach	No Beach
High	3/4	2/6
Mild	1/4	2/6
Low	0/4	2/6
Humidity	Beach	No Beach
High	2/4	2/6
Normal	2/4	4/6
P(Beach Day)	4/10	6/10

Figure 1: Probabilidades para apoiar um cálculo ingênuo de Bayes sobre se hoje é um bom dia para ir à praia: eventos tabulados (à esquerda) com distribuições de probabilidade marginal (à direita)

Lidando com Contagens Zero (Descontos)

Há um problema sutil, mas importante, de preparação de features, particularmente associado com o algoritmo Naive Bayes. As contagens observadas não capturam com precisão o frequência de eventos raros, para os quais normalmente há uma cauda longa. A questão foi levantada pela primeira vez pelo matemático Laplace, que perguntou: Qual é a probabilidade de o sol nascer amanhã? Pode estar perto de um, mas não é exatamente 1,0. Embora o sol tenha nascido como um relógio todas as manhãs durante os 36,5 milhões de manhãs desde que o homem começou a perceber essas coisas, isso não acontecerá para sempre. Chegará a hora em que a terra ou o sol explodirá e, portanto, há uma chance pequena, mas diferente de zero, de que esta noite seja a noite. Sempre pode haver eventos que ainda não foram vistos em nenhum conjunto de dados finito. Você pode muito bem ter registros de uma centena de pessoas, nenhuma das quais tem cabelo ruivo. Concluir que a probabilidade de ser ruivo é $0/100 = 0$ é potencialmente desastroso quando nos pedem para classificar alguém com cabelo ruivo, pois a probabilidade de estar em toda e qualquer classe será zero. Pior ainda seria se houvesse exatamente um ruivo em todo o conjunto de treinamento, digamos rotulado como classe C_2 . Nosso ingênuo classificador de Bayes decidiria que toda futura ruiva só tinha que estar na classe C_2 , independentemente de outras evidências.

O desconto é uma técnica estatística para ajustar as contagens de eventos ainda não vistos, deixando explicitamente a massa de probabilidade disponível para eles. A técnica mais simples e popular é o desconto *add-one*, em que adicionamos *um* à frequência de todos os resultados, inclusive os não vistos. Por exemplo, suponha que estivéssemos tirando bolas de uma urna. Depois de ver cinco vermelhos e três verdes, qual é a probabilidade de vermos uma nova cor no próximo sorteio? Se empregarmos um desconto adicional,

P (vermelho) = (5 + 1)/((5 + 1) + (3 + 1) + (0 + 1)) = 6/11, e

P (verde) = (3 + 1)/((5 + 1) + (3 + 1) + (0 + 1)) = 4/11,

deixando a nova cor uma massa de probabilidade de

P (nov cor) = 1/((5 + 1) + (3 + 1) + (0 + 1)) = 1/11.

Para pequenos números de amostras ou grandes números de classes conhecidas, o desconto causa um amortecimento não trivial das probabilidades. Nossa estimativa para a probabilidade de ver uma bola vermelha muda de $5/8 = 0{,}625$ para $6/11 = 0{,}545$ quando empregamos o desconto *add-one*. Mas esta é uma estimativa mais segura e honesta, e as diferenças desaparecerão no nada depois de termos visto amostras suficientes.

Você deve estar ciente de que outros métodos de desconto foram desenvolvidos e adicionar *um* pode não ser o melhor estimador possível em todas as situações. Dito isso, não descontar contagens é pedir problemas e ninguém será demitido por usar o método *add-one*.

O desconto torna-se particularmente importante no processamento de linguagem natural, onde a representação tradicional do saco-de-palavras (bag of words) modela um documento como um vetor de contagem de frequência de palavras em todo o vocabulário do idioma, digamos 100.000 palavras. Como a frequência de uso das palavras é governada por uma lei de potência (lei de Zipf), as palavras na cauda são bastante raras. Você já viu a palavra em inglês *defenestrate* antes? Pior ainda, documentos com menos de um livro são muito curtos para conter 100.000 palavras, então estamos condenados a ver zeros onde quer que olhemos. O desconto de *add-one* transforma esses vetores de contagem em vetores de probabilidade sensatos, com probabilidades diferentes de zero de ver palavras raras e até agora não encontradas.

Abordagens Frequentistas vs. Bayesianas no Machine Learning

Sempre houve um debate entre inferência estatística bayesiana e frequentista. Frequentistas dominaram a prática estatística durante o século 20. Muitos algoritmos comuns de Machine Learning, como regressão linear e regressão logística, usam métodos frequentistas para realizar inferência estatística. Enquanto os Bayesianos dominaram a prática estatística antes do século 20, nos últimos anos muitos algoritmos nas escolas Bayesianas como Expectation-Maximization, Bayesian Neural Networks e Markov Chain Monte Carlo ganharam popularidade no mundo de Machine Learning.

Aqui vamos falar sobre as suas diferenças e conexões no contexto do Machine Learning. Também usaremos dois algoritmos para ilustração: regressão linear e regressão linear bayesiana.

Suposições

Para simplificar, usaremos θ para denotar o(s) parâmetro(s) do modelo ao longo desta seção.

Os métodos frequentistas assumem que os dados observados são amostrados de alguma distribuição. Chamamos essa distribuição de dados de *likelihood*: $P(X|\theta)$, onde \$ é tratado como constante e o objetivo é encontrar o θ que maximizaria o *likelihood*. Por exemplo, na regressão logística, presume-se que os dados sejam amostrados da distribuição de Bernoulli e, na regressão linear, os dados sejam amostrados da distribuição Gaussiana.

Os métodos Bayesianos assumem as probabilidades tanto para os dados quanto para as hipóteses (parâmetros que especificam a distribuição dos dados). Em bayesianos, θ é uma variável, e as suposições incluem uma distribuição prévia das hipóteses $P(\theta)$ e um *likelihood* dos dados $P(X|\theta)$. A principal crítica da inferência bayesiana é a subjetividade de *a priori*, pois diferentes *a priori*s podem chegar a diferentes posteriores e conclusões.

Aprendizagem de Parâmetros

Frequentistas usam *maximum likelihood estimation* (MLE) para obter uma estimativa pontual dos parâmetros θ. A *log-likelihood* é expressa como:

$$L(\theta) = \log P(X|\theta) = \log \prod_{i=1}^{n} P(x_i|\theta) = \sum_{i=1}^{n} \log P(x_i|\theta)$$

Os parâmetros θ são estimados maximizando a *likelihood* logarítmica ou minimizando a *likelihood* logarítmica negativa (função de perda):

$$\theta_{MLE} = \arg\max_{\theta} L(\theta) = \arg\min_{\theta} -L(\theta)$$

Em vez de uma estimativa pontual, os **bayesianos** estimam uma distribuição posterior completa dos parâmetros usando a fórmula de Bayes:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} = \frac{P(X|\theta)P(\theta)}{\int P(X|\theta)P(\theta)d\theta}$$

Você deve ter notado que o cálculo do denominador pode ser *NP-hard* porque tem uma integral (ou soma no caso de classificação) sobre todos os valores possíveis de θ. Você também pode se perguntar se podemos ter uma estimativa pontual de θ, assim como o MLE faz. É aí que entra em jogo a estimativa *Maximum A Posteriori (MAP)*. O MAP contorna o complicado cálculo da distribuição posterior e, em vez disso, tenta encontrar as estimativas pontuais de θ que maximizam a distribuição posterior.

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} P(\theta|X) \\ &= \arg\max_{\theta} \frac{P(X|\theta)P(\theta)}{P(X)} \\ &= \arg\max_{\theta} P(X|\theta)P(\theta)\end{aligned}$$

Como as funções logarítmicas são **monotônicas**, podemos reescrever a equação acima no espaço logarítmico e decompô-la em 2 partes: maximizando a likelihood e maximizando a distribuição a priori:

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} \log P(X|\theta) + \log P(\theta) \\ &= \arg\max_{\theta} \log P(X|\theta) + \log P(\theta|X) \\ &= \arg\max_{\theta} \log P(X|\theta) + \log \frac{P(\theta|X)}{P(X)} \\ &= \arg\max_{\theta} \log P(X|\theta) + \log P(\theta|X) - \log P(X)\end{aligned}$$

Isso não se parece com o MLE?

Na verdade, a conexão entre esses dois é que o MAP pode ser tratado como executando o MLE em uma função de perda regularizada em que o anterior corresponde ao termo de regularização. Por exemplo, se assumirmos que a distribuição a priori é Gaussiana, MAP é igual a MLE com regularização L2; se assumirmos que a distribuição a priori é Laplace, MAP é igual a MLE com regularização L1.

Existe outro método para obter uma estimativa pontual da distribuição posterior: Estimativa *Expected A posteriori (EAP)*. A diferença entre MAP e EAP é que MAP obtém o modo (máximo) da distribuição posterior, enquanto EAP obtém o valor esperado da distribuição posterior.

Incerteza

A principal diferença entre as abordagens frequentista e bayesiana é a forma como medem a incerteza na estimativa de parâmetros.

Como mencionamos anteriormente, os frequentadores usam o MLE para obter estimativas pontuais de parâmetros desconhecidos e não atribuem probabilidades a possíveis valores de parâmetros. Portanto, para medir a incerteza, os Frequentistas confiam na hipótese nula e nos intervalos de confiança. No entanto, é importante ressaltar que os intervalos de confiança não

se traduzem diretamente em probabilidades de hipóteses. Por exemplo, com um intervalo de confiança de 95%, isso significa apenas que 95% dos intervalos de confiança gerados cobrirão a estimativa real, mas é incorreto dizer que cobre a estimativa real com uma probabilidade de 95% .

Bayesianos, por outro lado, têm uma distribuição posterior completa sobre os possíveis valores de parâmetros e isso permite que eles obtenham incerteza da estimativa integrando a distribuição posterior completa.

Computação

Bayesianos são geralmente mais computacionalmente intensivos do que frequentistas devido à integração de muitos parâmetros. Existem algumas abordagens para reduzir a intensidade computacional usando prioris conjugados ou aproximando a distribuição posterior usando métodos de amostragem ou inferência variacional.

Exemplos

Nesta seção, veremos como treinar e fazer previsões com dois algoritmos: regressão linear e regressão linear bayesiana.

Regressão Linear (frequentista)

Assumimos a forma abaixo de um modelo de regressão linear onde o intercepto é incorporado no parâmetro θ:

$$y = \theta x$$

Supõe-se que os dados sejam distribuídos de acordo com a distribuição gaussiana:

$$y|x; \theta \sim \mathcal{N}(\theta x, \sigma^2)$$

Usando MLE para maximizar o log-likelihood, podemos obter a estimativa pontual de θ conforme mostrado abaixo:

$$\begin{aligned}
\theta &= \text{argmax}_\theta L(\theta) \\
&= \text{argmax}_\theta \log P(y|x;\theta) \\
&= \text{argmax}_\theta \log(\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y-\theta x)^2}{2\sigma^2}))
\end{aligned}$$

Depois de aprender os parâmetros θ dos dados de treinamento, podemos usá-los diretamente para fazer previsões com novos dados:

$$\text{predição} : y^* = \theta x^*$$

Regressão Linear Bayesiana

Como mencionado anteriormente, a maneira bayesiana é fazer suposições tanto para *a priori* quanto para o likelihood.

$$\text{a priori} : \theta \sim \mathcal{N}(0, 1)$$
$$\text{likelihood} : y|x; \theta \sim \mathcal{N}(\theta x, \sigma^2)$$

Usando essas suposições e a fórmula de Bayes, podemos obter a distribuição a posteriori:

$$\begin{aligned}
\text{posterior} : P(\theta|X) &= \frac{P(X|\theta)P(\theta)}{\int P(X|\theta)P(\theta)d\theta} \\
&= \frac{P(y|x,\theta)P(\theta)}{\int P(y|x,\theta)P(\theta)d\theta} \\
&= \frac{(\frac{1}{\sqrt{2\pi\sigma^2}} \exp{(-\frac{(y-\theta x)^2}{2\sigma^2})})(\frac{1}{\sqrt{2\pi}} \exp{(-\frac{\theta^2}{2})})}{\int (\frac{1}{\sqrt{2\pi\sigma^2}} \exp{(-\frac{(y-\theta x)^2}{2\sigma^2})})(\frac{1}{\sqrt{2\pi}} \exp{(-\frac{\theta^2}{2})}d\theta}
\end{aligned}$$

No momento da previsão, usamos a distribuição posterior e o likelihood para calcular a distribuição preditiva posterior:

$$P(y^*|x^*, X) = \int P(y^*|x^*, \theta)P(\theta|X)d\theta = E_{\theta \sim P(\theta|X)}[P(y^*|x^*, \theta)]$$

Observe que a estimativa para os parâmetros e as previsões são distribuições completas. Obviamente, se precisarmos apenas de uma estimativa pontual, sempre podemos usar MAP ou EAP.

Conclusões

O principal objetivo do Machine Learning é fazer previsões usando os parâmetros aprendidos com os dados de treinamento. Se devemos atingir o objetivo usando a abordagem frequentista ou Bayesiana depende de:

1. O tipo de previsões que queremos: uma estimativa pontual ou uma probabilidade de possiveis valores.
2. Se temos conhecimento prévio que pode ser incorporado ao processo de modelagem.

Referências

1. Frequentist vs. Bayesian Approaches in Machine Learning
2. Bayesian and frequentist reasoning in plain English
3. Machine Learning CS229
4. The Data Science Data Manual