Big Data e Machine Learning com Hadoop e Spark

Conteúdo

CONTEÚDO PROGRAMÁTICO

- Visão geral da ciência de dados e aprendizado de máquina em escala
- Visão geral do ecossistema do Hadoop
- Instalação de um Cluster Hadoop
- Trabalhando com dados do HDFS e tabelas do Hive usando o Hue
- Visão geral do Python
- Visão geral do R
- Visão geral do Apache Spark 2
- Leitura e gravação de dados
- Inspeção da qualidade dos dados
- Limpeza e transformação de dados
- Resumindo e agrupando dados
- Combinando, dividindo e remodelando dados
- Explorando dados
- Configuração, monitoramento e solução de problemas de aplicativos Spark
- Visão geral do aprendizado de máquina no Spark MLlib
- Extraindo, transformando e selecionando recursos
- Construindo e avaliando modelos de regressão
- Construindo e avaliando modelos de classificação
- Construindo e avaliando modelos de cluster
- Validação cruzada de modelos e hiperparâmetros de ajuste
- Construção de pipelines de aprendizado de máquina
- Implantando modelos de aprendizado de máquina

MATERIAL DIDÁTICO

- Slides do treinamento em PDF
- GitHub com exercícios e códigos exemplo
- Máquinas virtuais para exercícios simulados
- Gravação das aulas disponível durante 3 meses

Métodos Bayesianos

Lembre-se do teorema de Bayes?

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

- Vamos usá-lo para aprendizagem de máquina! Eu quero um classificador de spam.
- Exemplo: como podemos expressar a probabilidade de um email ser spam se contém a palavra "free"?

$$P(Spam \mid Free) = \frac{P(Spam)P(Free \mid Spam)}{P(Free)}$$

- O numerador é a probabilidade de uma mensagem ser spam e conter a palavra "free" (isso é sutilmente diferente do que procuramos)
- O denominador é a probabilidade geral de um email contendo a palavra "free". (Equivalente a P (Free | Spam) P (Spam) + P (Free | Not Spam) P (Not Spam))
- Então, juntos - essa proporção é a % de emails com a palavra “free” que são spam.

E todas as outras palavras?

- Podemos construir P (Spam | Word) para cada palavra (significativa) que encontramos durante o treinamento
- Depois, multiplique-os juntos ao analisar um novo e-mail para obter a probabilidade de ser spam.
- Pressupõe a presença de palavras diferentes independentes uns dos outros - uma razão pela qual isso é chamado "Naïve Bayes".

Soa como um monte de trabalho.

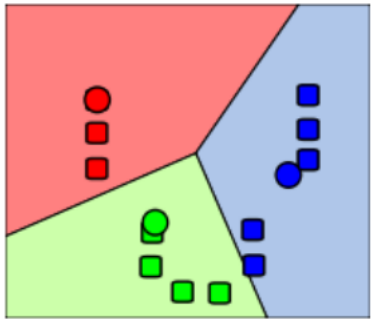
- Scikit-learn auxilia nesse trabalho
- O CountVectorizer nos permite operar em muitas palavras de uma só vez, e MultinomialNB faz todo o trabalho pesado em Naïve Bayes.
- Vamos treiná-lo em conjuntos conhecidos de spam e e-mails "ham" (sem spam)
    - Então, isso é aprendizado supervisionado!
- Vamos fazer isso

Exemplo

K-Means Clustering

K-Means clusters

- Tenta dividir os dados em grupos K que são mais próximos de K centroids
- Aprendizado não supervisionado - usa somente posições de cada ponto de dados
- Pode descobrir grupos interessantes de pessoas / coisas / comportamento
  - Exemplo: onde vivem milionários?
  - Que gêneros de música / filmes / etc. naturalmente vem dos dados?
  - Crie seus próprios estereótipos de dados demográficos



K-Means Clusters

- Funcionamento simples.
  - Escolher aleatoriamente K centróides (k-means)
  - Atribuir cada ponto de dados ao centróide mais próximo
  - Recompute os centróides com base na posição média de cada ponto centróide
  - Iterar até que os pontos parem de mudar a designação para centróides
- Se você quiser prever o cluster para novos pontos, basta encontrar o centróide que eles estão mais próximos.

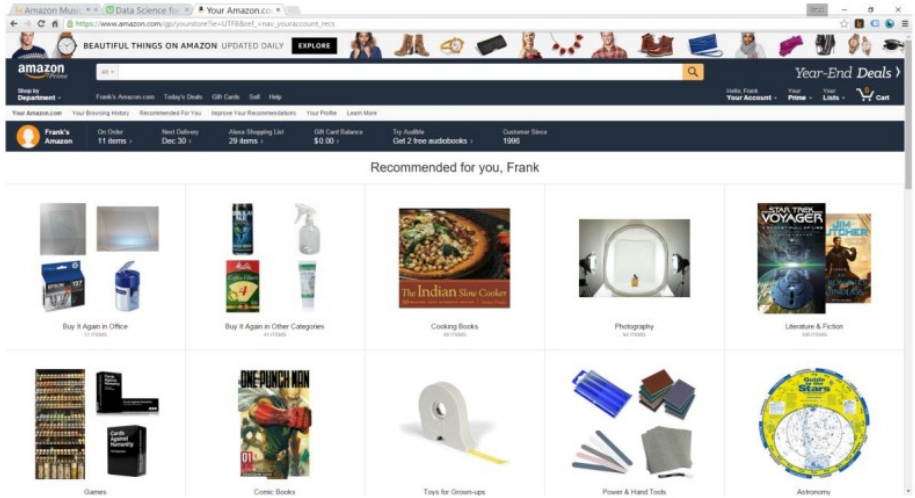
Exemplo gráfico

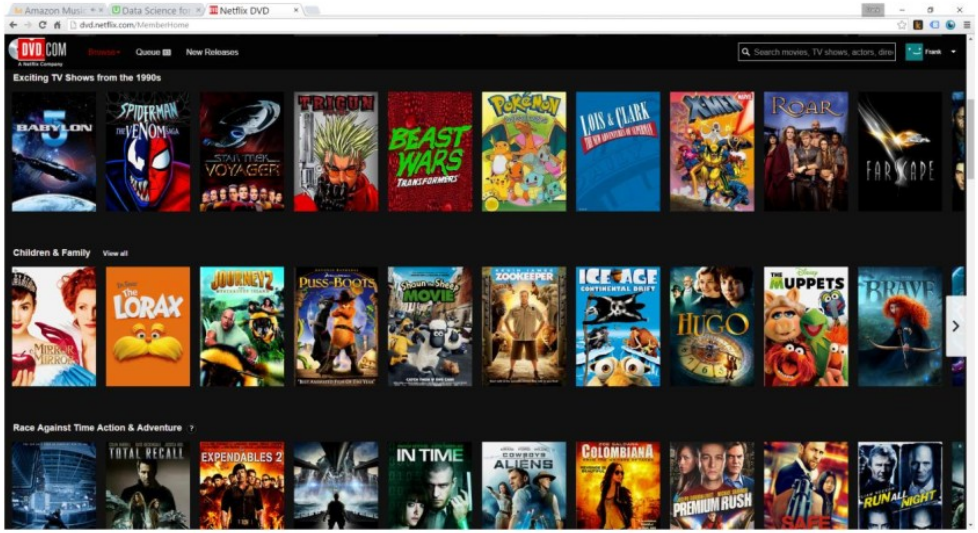
Desafios do K-Means Clustering

- Escolhendo K
  - Tente aumentar os valores de K até que você pare de obter grandes reduções no erro quadrado (distâncias de cada ponto aos seus centróides)
- Evitar mínimos locais
  - A escolha aleatória dos centróides iniciais pode produzir resultados diferentes
  - Execute algumas vezes apenas para garantir que seus resultados iniciais não sejam malucos
- Rotulando os clusters
  - O K-Means não tenta atribuir nenhum significado aos clusters encontrados
  - Cabe a você pesquisar os dados e tentar determinar

Exemplo

Sistemas de Recomendação


BEAUTIFUL THINGS ON AMAZON UPDATED DAILY
EXPLORE
amazon
Year-End Deals
Recommended for you, Frank
Buy It Again in Office
Buy It Again in Other Categories
Cooking Books
Photography
Literature & Fiction
Games
Comic Books
Toys for Grown-ups
Power & Hand Tools
Astronomy


DVD.COM
Queue
New Releases
Exciting TV Shows from the 1990s
Children & Family
Race Against Time Action & Adventure

Filtragem Colaborativa Baseada no Usuário

- Constrói uma matriz de coisas que cada usuário comprou / visualizou / avaliou
- Computa pontuações de similaridade entre usuários
- Encontra usuários semelhantes a você
- Recomenda coisas que outros compraram / visualizaram / classificaram e que você ainda não.

Problemas com Filtragem Colaborativa Baseada no Usuário

- As pessoas são inconstantes; gostos mudam
- Geralmente há muito mais pessoas do que coisas
- As pessoas fazem coisas ruins

E se nós basearmos recomendações em relacionamentos entre as coisas em vez de pessoas?

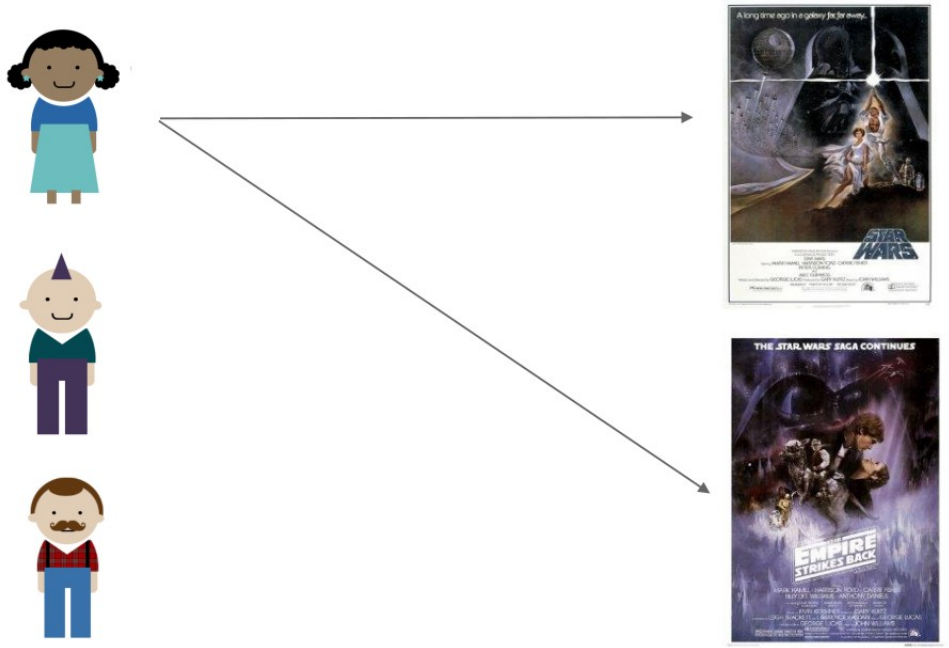
- Um filme será sempre o mesmo filme - não muda
- Geralmente há menos coisas que pessoas (menos computação para fazer)
- Difícil de enganar o sistema

Filtragem Colaborativa Baseada em Itens

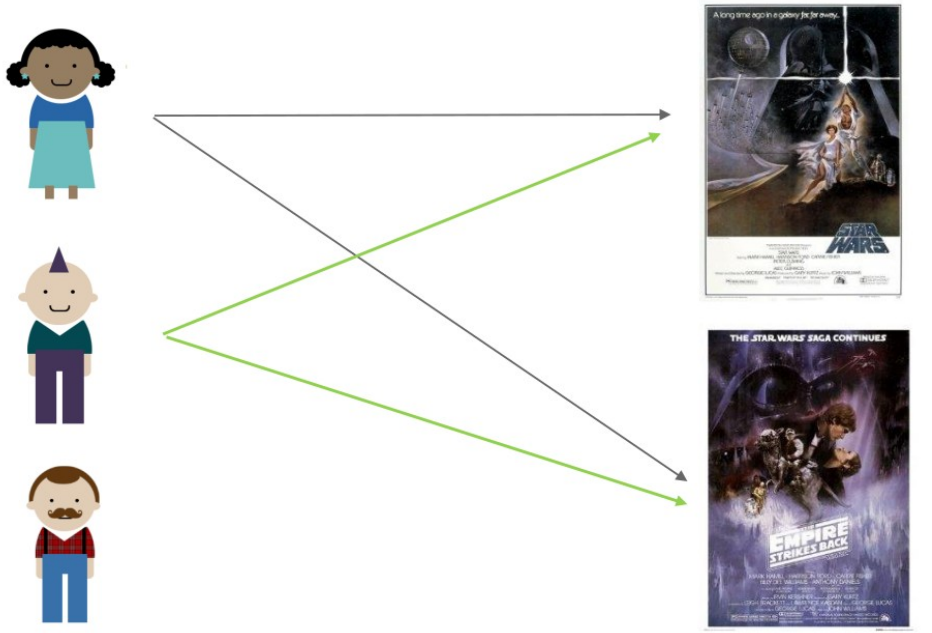
- Encontre cada par de filmes que foram assistidos pela mesma pessoa
- Avalie a similaridade de suas classificações em todos os usuários que assistiram ambos
- Classificar por filme e depois por força de similaridade

- (Esta é apenas uma maneira de fazer isso!)

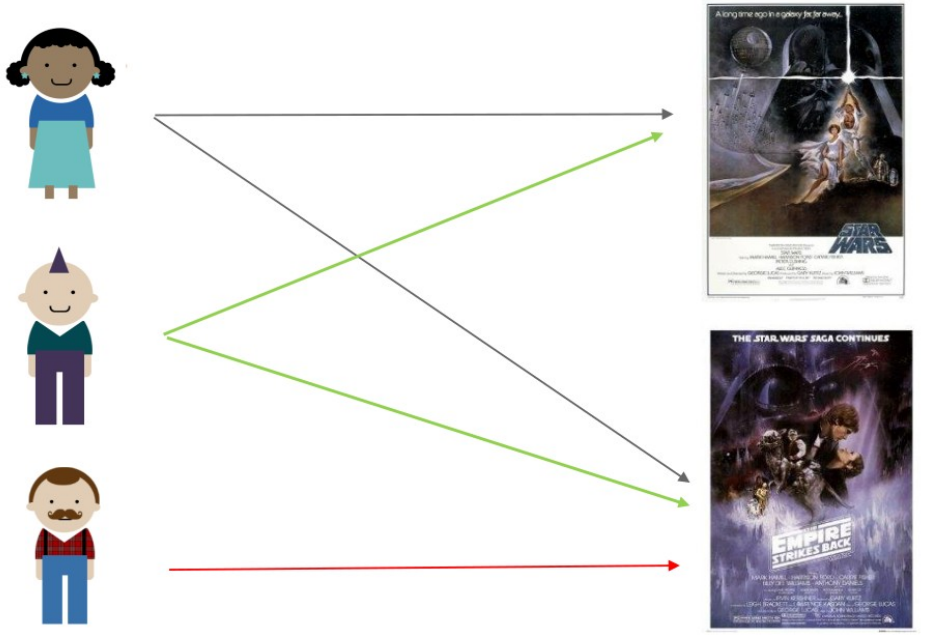
Filtragem Colaborativa Baseada em Itens



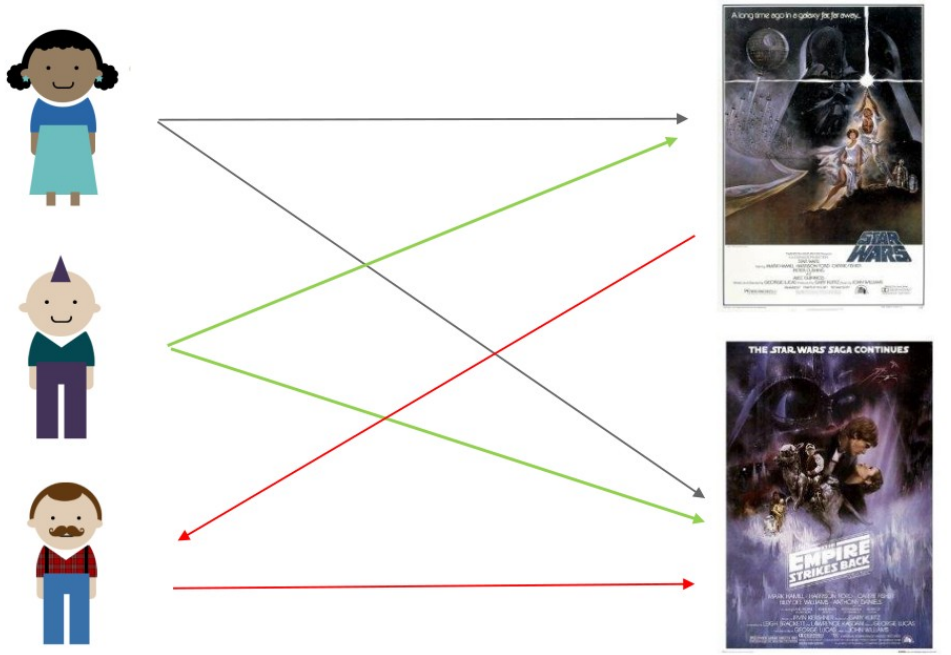
Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Filtragem Colaborativa Baseada em Itens



Vamos fazer isso

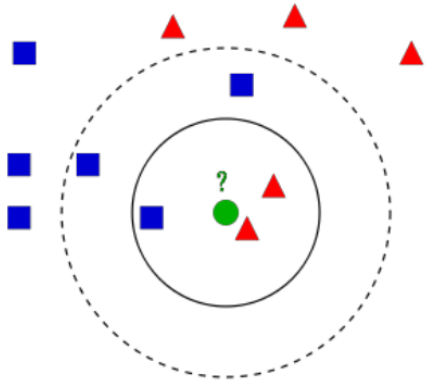
- Em seguida, usaremos o Python para criar "semelhanças de filme" reais usando o Conjunto de dados MovieLens.
  - Além de ser importante para a filtragem colaborativa baseada em itens, esses os resultados são valiosos em si mesmos - pense que "as pessoas que gostaram do X também gostaram de Y"
- São dados do mundo real e encontraremos problemas do mundo real
- Então, usaremos esses resultados para criar recomendações de filmes para indivíduos

Exemplo

K-Nearest Neighbor

K-Nearest Neighbor

Usado para classificar novos pontos de dados com base em "distância" para dados conhecidos
Encontre os K vizinhos mais próximos, com base na sua métrica de distância
Deixe que todos votem na classificação



É realmente simples

- Embora seja um dos modelos de aprendizado de máquina mais simples que existe, ainda se qualifica como “aprendizado supervisionado”.
- Mas vamos fazer algo mais complexo com isso
- Semelhanças cinematográficas baseadas apenas em metadados!

Customers Who Watched This Item Also Watched

Exemplo

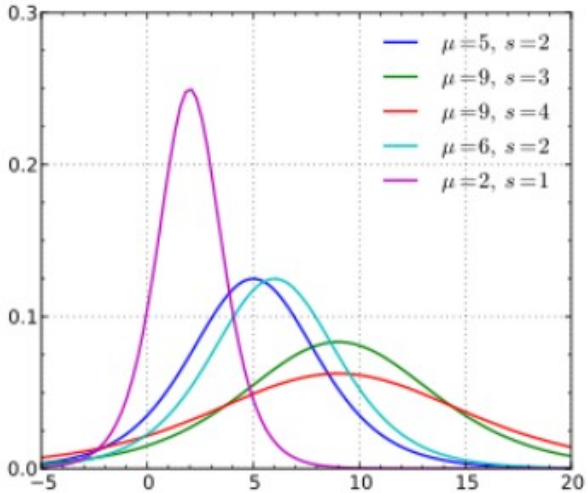
Modelos de Escolha Discreta

Modelos de Escolha Discreta

- Prevêm algumas escolhas entre alternativas discretas
  - Eu pego o trem, ônibus ou carro para trabalhar hoje? (Escolha Multinomial)
  - Para qual faculdade eu vou? (Multinomial)
  - Vou trair meu cônjuge? (Binário)
- As alternativas devem ser finitas, exaustivas e mutuamente exclusivas

Modelos de Escolha Discreta

- Usa algum tipo de regressão nos atributos relevantes
  - Atributos das pessoas
  - Variáveis das alternativas
- Geralmente usa modelos Logit ou Probit
  - Regressão Logística, Modelo Probit
  - Baseado em alguma função de utilidade você define
  - Similar - um usa a distribuição logística, o Probit usa distribuição normal. Logística parece muito com normal, mas com caudas mais gordas (maior curtose)



Limpando seus dados

Limpando seus dados

- Muito do seu tempo como um cientista de dados será gasto preparando e "limpando" seus dados
- Outliers
- Dados ausentes
- Dados maliciosos
- Dados errados
- Dados irrelevantes
- Dados inconsistentes
- Formatação

Garbage In, Garbage Out

- Olhe seus dados! Examine!
- Questione seus resultados!
  - Sempre faça isso - não apenas quando você não tenha um resultado que você goste!

Vamos analisar alguns dados de log da web.

- O que eu quero: as páginas mais populares no meu site de notícias sem fins lucrativos.
- Quão difícil isso pode ser?

Exemplo

Apache Spark
Introdução à MLLib

Alguns recursos do MLLib

- Extração de características
  - Freqüência a termo / Freqüência de documento inversa útil para pesquisa

- Estatísticas básicas
  - Qui-quadrado, correlação de Pearson ou Spearman, min, max, média, variância

- Regressão Linear, Regressão Logística
- Máquinas de vetores de suporte
- Classificador Naïve Bayes
- Árvores de decisão
- K-significa clusters
- Análise de componentes principais, decomposição de valores singulares
- Recomendações usando Mínimos Quadrados Alternados

Exemplo

Obrigado!!!

Nos vemos amanhã!!!

Bom descanso!