Nanodegree Engenheiro de Machine Learning

Proposta de projeto final

Rafael Ferrari Zitto
13 de março de 2018

Histórico do assunto

O comércio eletrônico é parte integrante da vida moderna, tanto para pessoas como para empresas. Tudo pode ser encontrado online, em grande variedade de marcas, modelos e preços.

Mas esta quantidade de opções traz um desafio: como organizar tudo isto?

Tradicionalmente, lojas online utilizam uma estrutura hierárquica de categorias de produtos. Porém, apesar de parecer uma atividade trivial, classificar cada produto pode se tornar um pesadelo quando se fala em centenas de milhares deles, em centenas ou mesmo milhares de categorias[(2)].

Tudo fica ainda mais complexo quando se considera as atuais plataformas de marketplace, onde os mesmos produtos são cadastrados diversas vezes, por diferentes varejistas, muitas vezes de forma automática.

Trazendo para minha experiência pessoal, pude visualizar em clientes que estive, mesmo que de forma indireta, o quão valioso seria uma categorização refinada de produtos para um marketplace. Alguns benefícios seriam:

- Navegação simplificada.
- Visualização e comparação de produtos similares em uma mesma página.
- Oferta de produtos e serviços relacionados (cross-selling/up-selling), tais como garantia estendida e seguros.
- Geração de relatórios e dashboards segmentados.

Descrição do problema

O objetivo deste trabalho é estudar estratégias para a classificação de produtos em categorias, a partir de suas características.

Em particular, as características serão limitadas ao nome do produto e seu fabricante, e a classificação terá uma estrutura hierárquica de categorias e sub-categorias, conforme padrão de mercado.

Sugestão

- Somente o nome do produto pode tornar "difícil" pro classificador encontrar a categoria. Por exemplo, suponhamos que você treinou numa base que não possui "galaxy duos" (e nem nenhum dessa linha da samsung) e quer predizer em qual categoria "galaxy duos" se encaixa. Nesse caso, a unica informação que o classificador tem embutida no modelo é o titulo e fabricante, se não tiver nenhum celular que lembre o nome "galaxy duos" ele terá que se basear no campo fabricante o que pode tornar a classificação difícil, pois alguns fabricantes fazem todo tipo de coisa (por exemplo, a samsung faz até carros na Coreia). Qual seria a sugestão então? Além do titulo e fabricante, extraia também a descrição do produto (pode ser em texto puro, veja mais detalhes na revisão do conjuntos e entradas e descrição da solução).

Compreendi sua sugestão e faz todo sentido.
O modo mais direto que vejo para fazer isto seria adicionar a descrição na "bag of words" existente, incrementando o vetor de vocabulário.
Porém vejo uma limitação nesta abordagem, pois considero que o fato de uma palavra estar no título é mais importante do que ela estar na descrição, e esta informação se perde quando todas são agrupadas em um único 'bag' antes do treinamento.
Aproveitando o exemplo sobre celulares, imaginemos 2 produtos:

- *nome: "apple iphone 8"; descrição: "... inclui um fone de ouvido ...", categoria: "celular"*
- *nome: "fone de ouvido apple"; descrição: "... compatível com iphone 8 ...", categoria: "acessório para celular"*

É esperado que uma pesquisa por "apple iphone 8" retorne "celular" apesar de todas as palavras ocorrerem nos dois casos, pois o fato das palavras estarem no nome é mais relevante.
Acha interessante/relevante tratar isto? Ensemble/Stacking ajudaria?

Conjuntos de dados e entradas

O problema em questão será tratado como classificação por aprendizagem supervisionada, e portanto fará uso de dados pré-classificados para treinamento e validação.

Os dados serão obtidos por um web crawler, que fará a varredura de um grande site de ecommerce.

Os dados serão compostos por:

Entradas (features)

- Nome do produto
- Nome do fabricante

Classificação (label)

- Categoría (hierárquica)

Exemplo:

"Console Playstation 4 500Gb Slim", "Sony", "Entretenimento > Games > Consoles"

Por simplicidade, será assumido que a estrutura de categorias obtida é a ideal, ou seja, que será utilizada "as-is" para a classificação de produtos no futuro.

Mudança Requerida

- Verifique se o site de onde serão extraídos os dados não possui nenhum termo de privacidade que proíbe o crawling dos dados. Além disso, mencione qual será a fonte dos dados, ou seja, qual site.
- Dê ao menos uma ideia do volume de dados que será utilizado, isto é, quantos produtos pretende incluir na análise?
- As categorias e sub-categorias utilizadas serão as presentes no site? É bom já dar uma ideia de quantas vão existir.

Havia considerado inicialmente o www.pontofrio.com.br, mas realmente me parece que os termos não me permitem utilizar estes dados, mesmo que apenas para estudo. Gostaria de dados em português, mas este tipo de termo de uso parece ser a norma em todos. Assim, devo seguir com sua sugestão de utilizar o Best Buy.

Com relação ao volume de dados, pretendo trabalhar com um valor que seja prático dentro da capacidade computacional que disponho em casa.
Tinha feito um crawling inicial no Ponto Frio extraindo 100.000 produtos, distribuídos em aproximadamente 1.000 categorias.
O treinamento "flat" com SVM levou cerca de 100s.
O treinamento de um nó da hierarquia levou cerca de 15s. Considerando-se que seriam cerca de 700 nós para treinamento, o total seria de cerca de 3 horas.
Concluindo, este seria um volume factível. Considera suficiente?

Descrição da solução

Conforme já mencionado acima, o problema será tratado como classificação por aprendizagem supervisionada, utilizando-se um dos algoritmos mais indicados para este fim[3], tais como:

- SVM (Support Vector Machine)
- Decision Tree
- Random Forest
- Logistic Regression

Este problema apresenta um desafio (ou oportunidade) adicional, visto que o label tem uma estrutura hierárquica. Assim, algumas estratégias podem ser consideradas:

1. Considerar toda a estrutura hierárquica da categoria de um produto como sendo um label único, indivisível. Neste caso o conceito de hierarquia estaria sendo desconsiderado, e tem-se um treinamento único, com uma grande quantidade de classes de saída[2].

|----- "categoria 1 > subcategoria 1.1 > subcategoria 1.1.1"

|----- “categoria 1 > subcategoria 1.1 > subcategoria 1.1.2”
(nome, fabricante) => | ...
|----- “categoria 9 > subcategoria 9.1 > subcategoria 9.1.1”
|----- “categoria 9 > subcategoria 9.1 > subcategoria 9.1.2”

2. Considerar cada nível da estrutura hierárquica individualmente. Neste caso teremos um treinamento para cada nível da hierarquia, e a predição ocorrerá de forma progressiva, nível por nível[3].

|----- “categoria 1”
|----- “categoria 2”
(nome, fabricante) => | … =>
|----- “categoria 9”
|----- “categoria 10”

|----- “categoria 9.1”
|----- “categoria 9.2”
(nome, fabricante) => | … =>
|----- “categoria 9.8”
|----- “categoria 9.9”

|----- “categoria 9.1.1”
|----- “categoria 9.1.2”
(nome, fabricante) => | ...
|----- “categoria 9.1.8”
|----- “categoria 9.1.9”

Modelo de referência (benchmark)

A acurácia de modelos utilizados para este fim podem ser encontradas nas referências (1), (2) e (3). Na referência (3) ainda temos as seguintes métricas mais refinadas: F1-score, hierarchical F1-score, hierarchical precision e hierarchical recall.

Métricas de avaliação

Os modelos serão inicialmente avaliados por sua acurácia e F1-score. Estas métricas são amplamente conhecidas e de fácil cálculo e interpretação.

Devido ao aspecto hierárquico da saída, as mesmas métricas também serão aplicadas nível a nível, permitindo mensurar como a acurácia se deteriora com as progressivas segmentações dos produtos.

Exemplo (fictício):

- Acurácia para predição do nível 1: 99%
- Acurácia para predição dos níveis 1 e 2: 95%
- Acurácia para predição dos níveis 1, 2 e 3: 90%
- …

Design do projeto

1. Coleta dos dados
    a. Criação de web crawler para extração dos dados para treinamento.
2. Pré-processamento dos dados
    a. Exclusão (prunning) de ramos de categorias com poucos exemplos.
    b. Retirada de caracteres especiais, acentuação e conversão para caixa-baixa.
3. Treinamento
    a. Extração de features dos textos, utilizando “Bag of Words” e “Term Frequency times Inverse Document Frequency”[4]
    b. Divisão dos dados em treinamento (80%) e validação (20%)
    c. Treinamento preliminar utilizando apenas o primeiro nível da estrutura de categorias. Este treinamento é mais simples e rápido, e será utilizado para pré-seleção do algoritmo de aprendizado. Serão considerados os seguintes algoritmos:
        i. SVM (Support Vector Machine)
        ii. Decision Tree
        iii. Random Forest
        iv. Logistic Regression
    d. Otimização de parâmetros do algoritmo escolhido utilizando-se “Grid Search”.
    e. Treinamento utilizando-se as categorias completas, sem quebra por nível (opção 1 descrita em “Descrição da solução”).
    f. Implementação de algoritmo de aprendizagem hierárquico.
        i. Fit (treinamento): percorrer cada nó na árvore de categorias, treinando uma instância do algoritmo aprendiz selecionado acima (c), com o respectivo sub-conjunto de dados/labels.
        ii. Predict (predição): iniciando no primeiro nível, utilizar a respectiva instância do algoritmo aprendiz para prever a categoria. Repetir com as subcategorias da categoria prevista, e assim por diante, até o último nível. Uma maneira mais sofisticada que será considerada é utilizar-se um algoritmo de busca, como o “A* search”[5]
    g. Treinamento completo, utilizando o algoritmo implementado (opção 2 descrita em “Descrição da solução”).

Sugestão

- Com relação a utilizar o A star. Qual heurística seria utilizada para estimar a distância? Isto é, o h(n) da f(n)=g(n)+h(n). Quem sabe você possa começar com uma busca em largura ao invés de usar diretamente o A star?

  Havia pensado em simplesmente h(n) = 0, ou seja, sempre seguiria onde g(n) é mínimo.
  No caso colocaria g(n) como o oposto da probabilidade acumulada do nó:

  $g(n) = 1- p(c_n)$, onde c_n é uma sub-categoria no nível n

  Isto implica que teria que utilizar um algoritmo que além da classificação, retorne a probabilidade, como o SVM.

  Entendo que a busca em largura implica em executar a predição em todos os nós da hierarquia, e o A star economizaria processamento, mesmo com h(n) = 0. Está correto?

4. Avaliação
   a. Cálculo das métricas de avaliação.
   b. Conclusões

Referências

(1) Ankush Bhalotia. Implementing a Machine-Learning Based eCommerce Product Classification System. Acessado em 06/03/2018.
https://blog.dataweave.com/implementing-a-machine-learning-based-ecommerce-product-classification-system-f846d894148b

(2) Amadeus Magrabi. Boosting Product Categorization with Machine Learning. Acessado em 06/03/2018.
https://techblog.commercetools.com/boosting-product-categorization-with-machine-learning-ad4dbd30b0e8

(3) Mikael Karlsson, Anton Karlstedt. Product Classification - A Hierarchical Approach. Acessado em 06/03/2018.
http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8889613&fileOId=8889614

(4) Scikit-learn. Working With Text Data. Acessado em 06/03/2018.
http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

(5) A* search algorithm. Acessado em 13/03/2018.
https://en.wikipedia.org/wiki/A*_search_algorithm

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.302&rep=rep1&type=pdf