

Universidade Autónoma de Lisboa

Relatório do Projeto - Inteligência Artificial

Professor: Gonçalo Valadão

Curso de Engenharia Informática

Constituintes:

Nuno Cartaxo: 30007214

Frederico Bobone: 30007775

Duarte Cabrita: 30007375

Daniel Cardoso: 30007243

Introdução

Neste relatório, apresentamos os resultados de um projecto em que implementamos os algoritmos Naive Bayes, perceptrão e AC-3 em Python. Os algoritmos Naive Bayes e perceptrão foram utilizados para desenvolver filtros de spam, enquanto que o algoritmo AC-3 foi utilizado para resolver (certos) puzzles de sudoku.

O algoritmo multinomial naive Bayes é um método de aprendizagem amplamente utilizado para tarefas de classificação. Baseia-se na ideia de utilizar o teorema de Bayes para estimar a probabilidade de um evento com base nas probabilidades de várias características que estão relacionadas com o evento. No caso do nosso filtro de spam, o evento é a classificação de um e-mail como spam ou ham, e as características são as palavras que aparecem no e-mail.

O algoritmo perceptrão é outro algoritmo de aprendizagem que pode ser utilizado para tarefas de classificação. Baseia-se na ideia de treinar um classificador linear, ajustando os pesos das características com base nos erros cometidos durante o processo de classificação. No nosso filtro de spam, as características são novamente as palavras que aparecem no e-mail, e os pesos são ajustados com base nos erros de classificação cometidos pelo perceptrão.

O algoritmo AC-3 é um algoritmo de satisfação de restrições que pode ser utilizado para resolver problemas em que as variáveis são condicionadas pelas relações entre as variáveis. Na nossa implementação, utilizámos o algoritmo AC-3 para resolver puzzles sudoku impondo restrições sobre os valores possíveis que podem ser atribuídos a cada célula do puzzle, com base nos valores das células circundantes.

A todos os datasets que foram utilizados (treino, teste e validação) removemos caracteres que não consideramos relevantes (pontos, vírgulas, pontos de exclamação, etc).

Naive Bayes

No contexto da filtragem do spam, as características são tipicamente as palavras que aparecem no email. O algoritmo calcula primeiro a probabilidade de cada palavra ocorrer nos emails de spam e de ham com base na frequência da palavra nos dados de formação. Em seguida, utiliza estas probabilidades, juntamente com o teorema de Bayes, para estimar a probabilidade de um dado email ser spam ou ham.

Para calcular a probabilidade de um e-mail ser spam ou ham, o algoritmo calcula primeiro a probabilidade de cada palavra do e-mail ocorrer em e-mails de spam ou ham. Em seguida, combina estas probabilidades utilizando a hipótese de independência, que afirma que a ocorrência de uma palavra num e-mail é independente da ocorrência de qualquer outra palavra.

Finalmente, o algoritmo utiliza o teorema de Bayes para calcular a probabilidade de um e-mail ser spam ou ham com base nas probabilidades das palavras individuais no e-mail. Especificamente, calcula a proporção entre a probabilidade de o e-mail ser spam e a probabilidade de o e-mail ser ham, e classifica o e-mail como spam se esta proporção for superior a 1 e como ham se for inferior a 1.

O "naive" em naive Bayes refere-se à presunção de independência que é feita pelo algoritmo. Especificamente, o algoritmo Naive Bayes assume que a presença ou ausência de uma determinada característica (por exemplo, uma palavra num e-mail) num conjunto de dados é independente da presença ou ausência de qualquer outra característica. Esta suposição não é frequentemente verdadeira em conjuntos de dados do mundo real, e pode levar a um mau desempenho do algoritmo em alguns casos.

Por exemplo, considere um conjunto de dados de e-mails que são classificados como spam ou ham. O Naive Bayes pode estimar a probabilidade de um e-mail ser spam com base nas frequências das palavras individuais no e-mail, tais como "viagra" ou "grátis". Contudo, a presença da palavra "viagra" num e-mail não é provavelmente independente da presença da palavra "livre", uma vez que os e-mails que contêm a palavra "viagra" são susceptíveis de conter também a palavra "livre". Se o algoritmo tratasse estas palavras como independentes, subestimaria a probabilidade de uma mensagem de correio electrónico contendo ambas as palavras ser spam.

Perceptrão

Para treinar o algoritmo para a filtragem de spam, o algoritmo é primeiro fornecido com um conjunto de dados que são rotulados como spam ou ham. O algoritmo inicializa então os pesos das características a valores aleatórios e inicia o processo de formação.

Antes da fase de treino tratamos os datasets de modo a que a primeira coluna que originalmente era uma string ("spam" ou "ham") fosse -1 ou 1 (respetivamente), e que a segunda coluna passasse a ser a contagem de cada palavra nesse email.

Durante o treino, o algoritmo itera por cada email do conjunto de dados e classifica o email com base nos pesos atuais das características. Se o email for corretamente classificado, os pesos não são ajustados. Se o email for mal classificado, os pesos são ajustados de uma maneira que aumentará a probabilidade de classificar o email corretamente no futuro.

Os pesos são ajustados com base no erro cometido durante a classificação, que é calculado como a diferença entre a verdadeira etiqueta do email (1 para ham e -1 para spam) e a etiqueta prevista.

Após o treino, o perceptrão pode ser usado para classificar novos emails calculando o produto interno entre os pesos e as características do email e classificando o email como spam se o produto interno for negativo e como ham se for positivo.

AC-3

O algoritmo AC-3 é um algoritmo de busca que pode ser usado para resolver problemas de sudoku sem recursão (backtracking). Ele é baseado no método de consistência arc, que é um método para garantir que nenhuma restrição do problema seja violada durante a resolução.

Para aplicar o algoritmo AC-3 a resolução de sudoku, o algoritmo é inicialmente fornecido com uma grade de sudoku incompleta e a partir deste cria um CSP(Constraint Satisfaction Problem) com estas 3 variáveis:

- Variáveis – nome de cada variável (cada célula do nosso problema; ex. V0,0)
- Domínio – Valores possíveis para cada variável. Se este valor for fornecido no puzzle inicial então será apenas esse valor. Se não, inicialmente será de 1 a 9
- Vizinhos – Lista de vizinhos do jogo sudoku (mesma linha, mesma coluna e mesmo quadrado). Estas são as restrições do sudoku.

O algoritmo começa por iterar pelas células da grade e verifica se cada uma delas está em conflito com qualquer uma das restrições. Se uma célula estiver em conflito, o algoritmo remove o número da célula e verifica se há algum outro número que possa ser colocado na célula de maneira consistente com as restrições. Se não houver nenhum outro número possível, o algoritmo volta para a última célula modificada e tenta outro número. Este processo é repetido até que a grade esteja completamente preenchida de maneira consistente com as restrições.

Um dos principais benefícios do algoritmo AC-3 é que ele é capaz de resolver problemas de sudoku muito rapidamente, geralmente em questão de segundos. Isso é possível devido à sua capacidade de remover rapidamente opções inviáveis durante o processo de resolução, o que significa que o algoritmo não precisa explorar todas as opções possíveis como a recursão faria.

O algoritmo AC-3 é um método eficiente para resolver problemas de sudoku sem recursão e tem sido amplamente usado na prática devido à sua capacidade de resolver problemas rapidamente e de forma consistente com as restrições do problema.

Métricas de classificação

Podemos observar que no output do Naive Bayes os resultados não coorespondem à realidade, isto deve se a um erro no código que não foi possível mitigar.

Temos razão para acreditar que o erro tem a haver com o threshold mas não conseguimos identificar e corrigir.

```
Naive Bayes:

O Algoritmo do Naive Bayes percorreu 834 emails, sendo observado o seguinte:

Dos 112 emails de spam, avaliou corretamente 1 emails e 111 emails foram avaliados de forma incorreta.
Dos 722 emails de ham, avaliou corretamente 722 emails e 0 emails foram avaliados de forma incorreta.
A taxa de sucesso geral é de 86.57074340527578 % e uma taxa de insucesso geral de 13.42925659472422 %

Métricas de Classificação:

A Accuracy calculada é: 0.8657074340527577
A Error rate calculada é: 0.1342925659472422
A Sensivity calculada é: 0.008849557522123894
A Specificity calculada é: 0.9986168741355463
A Precision calculada é: 0.5
O Recall calculado é: 0.0013831258644536654
A F-Measure calculada é: 0.0027586206896551726
O Geometric-mean calculado é: 26.870057685088806
```

Estas são as métricas do perceptrão e são as esperadas para o dataset que temos.

```
Perceptron:

O Algoritmo do perceptron percorreu 834 emails, sendo observado o seguinte:

Dos 112 emails de spam, avaliou corretamente 102 emails e 10 emails foram avaliados de forma incorreta.
Dos 722 emails de ham, avaliou corretamente 721 emails e 1 emails foram avaliados de forma incorreta.
A taxa de sucesso geral é de 98.6810551558753 % e uma taxa de insucesso geral de 1.3189448441247003 %

Métricas de Classificação:

A Accuracy calculada é: 0.986810551558753
A Error rate calculada é: 0.013189448441247002
A Sensivity calculada é: 0.9107142857142857
A Specificity calculada é: 0.9986149584487535
A Precision calculada é: 0.9902912621359223
O Recall calculado é: 0.12393681652490887
A F-Measure calculada é: 0.22030237580993522
O Geometric-mean calculado é: 271.18628283893713
```

Conclusões

No geral, o projeto de implementação dos algoritmos multinomial naive bayes, perceptron e AC-3 em Python foi bem-sucedido. Os algoritmos foram implementados de forma eficiente e testados com sucesso em aplicações práticas de filtragem de spam e resolução de sudoku.

O algoritmo multinomial naive bayes mostrou-se um método razoável para filtragem de spam, com um desempenho médio em conjuntos de dados de teste. O algoritmo perceptron foi bem-sucedido na filtragem de spam, com uma implementação simples e um bom desempenho.

O algoritmo AC-3, por sua vez, foi eficiente na resolução de problemas de sudoku sem recursão, mostrando-se uma opção rápida e consistente para este tipo de problema.

Em resumo, a implementação dos algoritmos multinomial naive bayes, perceptron e AC-3 em Python foi bem-sucedida e estes algoritmos são ferramentas valiosas para as respectivas aplicações de filtragem de spam e resolução de sudoku.