7
Seleção de Atributos

Diferentes abordagens são usadas para criar novos atributos que ajudem a classificar o sentimento. Porém, pouca ênfase tem sido dada a técnicas de seleção desses atributos. Wiebe et al. (21) mede a eficiência dos atributos baseado na distribuição de ocorrências de sua classe em textos subjetivos. Abbasi et al. (01) utiliza a técnica EWGA, um algoritmo genético ponderado por entropia, e mostra que essa técnica, apesar de ser computacionalmente pesada, é muito eficiente para classificação de sentimentos. A seleção dos atributos tem dois importantes benefícios, eles podem melhorar a acurácia da classificação eliminando os atributos que causam ruídos, e nos dão boas pistas de quais classes de atributos são importantes para a classificação do sentimento.

Nosso dicionário de palavras no tópico tem mais de 2.900 palavras das quais a maioria não é muito informativa para o classificador. No experimento com 1,2,3-gramas de palavras o número de atributos sobe para mais de 16.000. Esse crescente número de atributos tem dois inconvenientes. Primeiro ele deixa o experimento mais lento, pois existem muito mais atributos do que o classificador realmente precisa. O segundo problema é que eles podem realmente diminuir a acurácia do classificador, já que o SVM precisa considerar todos esses atributos no momento de criar o hiperplano.

Nesse trabalho investigamos o uso de 4 técnicas diferentes de seleção de atributos e o uso combinado de algumas delas. As técnicas de seleção incluem *Proportional Difference* (PD), método proposto por Tim O'Keefe (11) para a tarefa de classificação de sentimento; Fischer Score, um método já bastante utilizado em outras tarefas; EWGA, proposto por Abbasi com resultados positivos; e finalmente por Árvore de decisão. A combinação de PD com Fischer Score apresenta o melhor resultado como será mostrado a seguir.

7.1
Proportional Difference - PD

Introduzido por Simeon e Hilderman (17) e proposto por Tim O'Keefe e Irena Koprinska (11) para ser utilizado na tarefa de classificação de sentimento,

a PD é uma métrica que mede quão perto de serem iguais dois números estão. Isso é usado para descobrir atributos que ocorram com mais freqüência ou em textos positivos ou em textos negativos. Para tal, utilizamos a freqüência dos atributos em textos positivos e negativos segundo a equação 7-1. Enquanto Simeon e Hilderman utilizam uma equação mais genérica para um problema multi-classe, seguimos a metodologia de O'Keefe e Koprinska, com uma equação simplificada, já que nosso problema é de apenas 2 classes.

$$PD(r) = \frac{|Freq_{pos}(r) - Freq_{neg}(r)|}{Freq_{pos}(r) + Freq_{neg}(r)} \tag{7-1}$$

Para cada atributo r é calculada sua PD. Se ela aparece igualmente em texto positivos e negativos, sua PD será próxima de zero, se ela aparece predominantemente em apenas um tipo de documento, sua PD será próxima de 1. Uma PD alta significa que o atributo é muito informativo, enquanto que a PD baixa indica que o atributo não nos informa nada e por isso pode ser eliminado. Por exemplo, imagine que a palavra "ações"aparece igualmente em textos positivos e negativos, se a encontramos no exemplo novo que queremos classificar, ela não nos dará nenhuma informação e sua PD será zero. Em contrapartida, se a palavra "subiu"ocorre apenas em textos positivos, sua PD será igual a 1, e se a encontrarmos em um novo exemplo, saberemos que esse exemplo tem grande chance de ser positivo.

Para utilizar a PD como seleção de atributos basta eliminar os atributos cuja PD é inferior a um limite específico. O limite utilizado foi de $0,125$, conforme proposto por O'Keefe e Koprinska. Ou seja, nos experimentos calculamos a PD de cada atributo utilizando apenas o grupo de treino e filtramos apenas os que possuíam PD superior a 0,125. Em nossos experimentos temos uma redução de aproximadamente 77% do numero de atributos.

Na tabela abaixo podemos observar a redução do número de atributos e a acurácia de alguns experimentos.

Modelo	# Atributos	# Atributos com PD
1,2,3-grams palavra	16.210	12.736
1,2,3-grams palavra + POS	30.895	23.103
1,2,3-grams palavra + Chunk	32.377	25.489
1,2,3-grams palavra + POS + Chunk	50.429	38.512

Tabela 7.1: Redução do número de atributos e acurácia com uso do seletor PD

7.2 Fisher Score

Muito parecido com Proportional Diference, Fisher Score busca medir o quanto um atributo contribui para a classificação de um determinado exemplo utilizando funções probabilísticas. J.Weston et al. (20) fez um estudo com vários métodos de seleção de features para o SVM, o de Fisher Score foi um dos propostos e se mostrou muito eficiente.

F-Score utiliza a média e a variância do atributo em textos positivos e negativos. Ele é zero quando a média nas duas classes é igual. Ou seja, não é informativa pois está igualmente dividido nas médias.

$$FS(r) = \frac{\mu_{pos}(r) - \mu_{neg}(r)}{\sigma_{pos}(r) + \sigma_{neg}(r)} \tag{7-2}$$

Diferentemente do PD, o F-Score foi desenvolvido para atributos não categóricos. Ou seja, os atributos com valores de presença (zero ou 1) são melhor avaliados pelo fitro do PD, porém, os atributos estruturais, cujos valores estão no intervalos entre zero e um, são tratados somente pelo F-Score. Por isso esse último método de seleção é mais eficiente que o primeiro.

A combinação desses dois primeiros seletores resultou na técnica de seleção de atributos mais eficiente e rápida. Primeiramente utilizamos o filtro da PD, eliminando atributos categóricos não informativos, em seguida utilizamos o filtro do F-Score, refinando a seleção de atributos categóricos e não categóricos.

Na tabela abaixo podemos observar a redução do número de atributos com a aplicação do Fischer Score e com a aplicação do PD e F Score.

Modelo	# Atributos	com FS	com PD e FS
1,2,3-grams palavra	16.210	5.799	5.100
1,2,3-grams palavra+POS	30.895	10.060	7.791
1,2,3-grams palavra+Chunk	32.377	12.312	9.568
1,2,3-grams palavra+POS+Chunk	50.429	18.671	13.708

Tabela 7.2: Redução do número de atributos com uso do seletor

7.3 Entropy Weighted Genetic Algorithm - EWGA

Abbasi, Chen e Salem (01) propuseram essa metodologia, que envolve um algoritmo genético ponderado por entropia, como forma de seleção de atributos. Algoritmos genéticos são inspirados na teoria da evolução de Darwin. O

algoritmo começa com um conjunto de indivíduos, representados por cromossomas, chamado de população. Cada cromossoma é formado por uma seqüência de genes, onde cada gene representa a instância de um determinado atributo. Indivíduos de uma população são utilizados para formar uma nova população. Isto é motivado pela esperança que a nova população será melhor do que a primeira. A "seleção natural"de indivíduos que se cruzam é determinada por uma função de adequação, quanto melhor um indivíduo, maior suas chances de reprodução. O cruzamento desses indivíduos é feita pela troca de sub-partes de cada um dos cromossomas pai. Em seguida o cromossoma formado pelo cruzamento passa por mutações aleatórias que alteram alguns genes.

No EWGA cada cromossoma é um vetor de zeros e uns indicando se um determinado atributo deve ou não entrar na solução. Por exemplo:

$$\mathrm{S} = 011001$$
$$\mathrm{T} = 101011$$

Nesse exemplo o dicionário de palavras é formado por apenas 6 atributos. Os indivíduos S e T representam soluções para a seleção desses atributos, ou seja, eles informam quais atributos devem ou não ser considerados para o experimento. Nesse caso S e T concordam que o terceiro e o último atributo são importantes e por isso devem estar presentes nos vetores de entrada do SVM. Eles concordam também que o quarto atributo não é informativo e por isso deve ser eliminado dos vetores de entrada do classificador. Quanto aos outros atributos, eles divergem, e por isso criamos populações que misturam essas divergências.

A medida de adequação para cruzar indivíduos é a acurácia daquela determinada solução. Para cada indivíduo da população é necessário re-escrever os vetores de entrada do SVM considerando apenas os atributos marcados com "1"e medir a acurácia dessa solução.

O cruzamento de indivíduos normalmente consistem em encontrar um ponto no cromossoma e trocar as sub-partes dos indivíduos pais. De um cruzamento é sempre gerado 2 filhos, o primeiro com a primeira sub-parte do pai 1 e a segunda sub-parte do pai 2, e o segundo filho com a primeira sub-parte do pai 2 e a segunda sub-parte do pai 1.

$$\mathrm{S} = 011|001$$
$$\mathrm{T} = 101|011$$

$$\Downarrow$$

S1 = 011011
T1 = 101001

No exemplo acima, o ponto de corte foi exatamente o meio. O filho S1 ficou com a primeira metade de S e a segunda metade de T, quanto que o filho T1 ficou com a primeira metade de T e a segunda metade de S.

ALGORITMO EWGA:

Ganho de Informação (IG) do atributo A:

$$IG(C, A) = H(C) - H(C|A) \tag{7-3}$$

Onde:

Entropia das classes de sentimento C (Classe Positiva ou Negativa):

$$H(C) = -\sum_{i=1}^{2} p(C = i) \log_2 p(C = i) \tag{7-4}$$

Entropia da classe C dado o atributo A:

$$H(C|A) = -\sum_{i=1}^{2} p(C = i|A) \log_2 p(C = i|A) \tag{7-5}$$

$$P(Pos|A) = \frac{[\frac{Freq_{pos}(A)}{Num_{pos}}]}{[\frac{Freq_{pos}(A)}{Num_{pos}} + \frac{Freq_{neg}(A))}{Num_{neg}}]} \tag{7-6}$$

População inicial:

O ganho de informação do atributo A varia entre 0 e 1, onde quanto maior o valor, mais informativo é o atributo. O primeiro indivíduo da população inicial é formado utilizando o calculo do ganho de informação. Todos os atributos com IG maior que 0,0025 são selecionados, enquanto que os outros são descartados. O limite de 0.0025 é proposto por Abbasi. Os outros indivíduos da população inicial são gerados aleatoriamente.

Seleção dos pais:

Para selecionar os indivíduos mais aptos para o cruzamento, fazemos uma seleção por roleta levando em consideração a acurácia de cada indivíduo. Nesse processo, quanto maior a acurácia, maior a chance do indivíduo ser selecionado. Cada indivíduo é mapeado para uma porção na roleta, onde o tamanho de cada porção é proporcional a sua acurácia. Em seguida um número é sorteado aleatoriamente dentro da roleta.

(1) S = Soma de todas as acurácias dos indivíduos

(2) n = Sorteio de um número entre 0 e S

(3) Soma as acurácia indivíduo por indivíduo, a cada soma verifica se ela é maior que n, caso seja maior, retorna o ultimo indivíduo somado.

Cruzamento:

A cada geração, 30% dos melhores pais são automaticamente copiados para a geração seguinte. Dessa maneira garantimos que uma boa solução não será perdida. Para os outros 70%, selecionamos 35% dos pais através da seleção por roleta para que eles se cruzem e gerem a cada cruzamento 2 novos filhos.

O ponto de cruzamento dos pais é escolhido baseado no ganho de informação para que a qualidade dos novos indivíduos seja melhor que a dos pais. Dado um par de indivíduos, buscamos o ponto em que maximiza a diferença do somatório do ganho de informação de cada atributo. Essa metodologia proposta por Abbasi busca diversificar a população, alguns indivíduos com alta concentração de atributos de IG alto e outros com maior concentração de atributos de IG baixo. Essa diversificação é importante para que o algoritmo não converja prematuramente para um máximo local. O ponto de cruzamento pode ser formulado da seguinte maneira:

$$max(\sum_{A=1}^{x} IG(C,A)(Sa - Ta) + \sum_{A=x}^{m} IG(C,A)(Ta - Sa)) \qquad (7\text{-}7)$$

Onde:

IG(C, A) - Ganho de informação do atributo A;

$S_A - A^{esimo}$ caracter da solução S;

$T_A - A^{esimo}$ caracter da solução T;

M - número total de atributos;

X - ponto de cruzamento do par S e T, onde 1 <X >m.

Mutação:

Após o cruzamento dos pais, os novos indivíduos sofrem uma mutação em alguns de seus genes. Essa mutação também é feita levando-se em consideração o IG de cada atributo. Fazemos isso para aumentar a probabilidade de incluir um atributo com IG alto e ao mesmo tempo diminuir a probabilidade de inclusão de um atributo com IG baixo. A fórmula de calculo para a probabilidade de mutação de um atributo é feito para mutar um atributo 0 para 1, dessa maneira, se um atributo tem valor 1 sua probabilidade de mutação é 1 - fórmula dada.

$$P_m(A) = \begin{cases} B * [IG(C,A)] & \text{se} \quad S_A = 0 \\ B * [1 - IG(C,A)] & \text{se} \quad S_A = 1 \end{cases} \qquad (7\text{-}8)$$

Onde:
$P_m(A)$ - Probabilidade de mutação do atributo A;
IG(C, A) - Ganho de Informação do atributo A;
$S_A - A^{esimo}$ caracter na solução S;
B - Constante igual a 0.1;

Esse seletor de atributos diminue em $8,5\%$ o número de atributos original.

7.4 Árvore de decisão

Para esse trabalho é proposto ainda uma última maneira de selecionar atributos: por árvore de decisão. Como já foi citado no capítulo de atributos criados pelos caminhamentos na árvore de decisão, esse algoritmo é uma representação de uma tabela de decisão sob a forma de uma árvore. Para gerar a árvore usamos o algoritmo já pronto do C4.5. Cada nó da árvore é um atributo e as arestas é a presença ou ausência do mesmo. Porém, nesse capitulo falaremos de selecionar atributos com a informação da árvore ao invés de criá-los. Para a tarefa de classificação de sentimento, seleção de atributos por árvore de decisão não foi testada antes.

A figura 7.1 mostra uma árvore de decisão simplificada para o problema de classificação de sentimentos de notícias do mercado financeiro. Para cada notícia selecionamos apenas os atributos que aparecem na árvore mais os atributos criados pelos caminhamentos na mesma.

Para esse exemplo selecionaríamos apenas os atributos descritos na tabela 7.4:

Subiram
Ações
Gastos
Caíram
Subiram(Existe)+Ações(Existe)
Subiram(Existe)+Ações(NãoExiste)
Subiram(Existe)+Ações(NãoExiste)+Gastos(Existe)
Subiram(NãoExiste)+Cairam(Existe)
Subiram(NãoExiste)+Cairam(Existe)+Ações(Existe)
Subiram(NãoExiste)+Cairam(Existe)+Ações(NãoExiste)+Gastos(Existe)

Tabela 7.3: Atributos selecionados pela Árvore de Decisão

Esse seletor de atributos diminui em 97% a quantidade de atributos original, apesar dele criar novos atributos derivados dos caminhamentos.

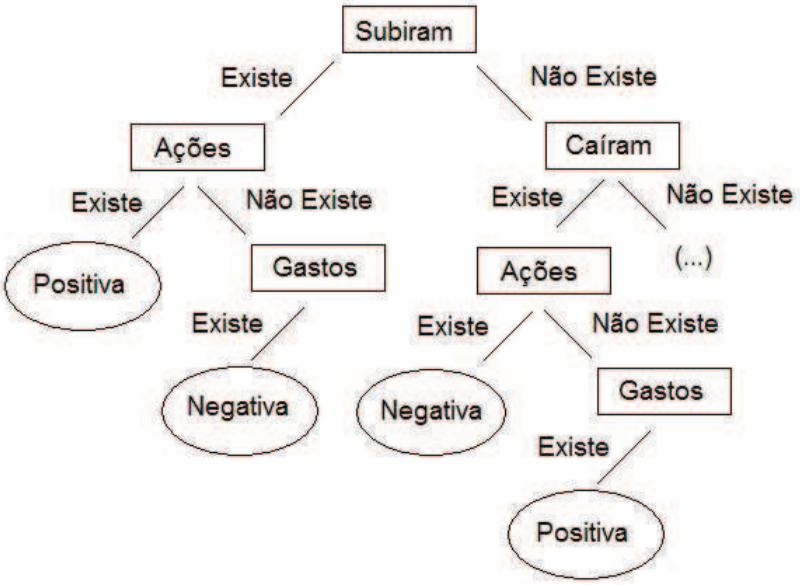

Figura 7.1: Exemplo de Árvore de decisão