4
Comitê BAS

Devido à flexibilidade do algoritmo BAS, que aceita qualquer distribuição inicial para os exemplos, diversas heurísticas podem ser implementadas com o objetivo de criar classificadores de alto desempenho através de um esquema de Boosting.

Neste capítulo, apresentamos o Comitê BAS, um algoritmo de aprendizado de máquina baseado no BAS. Este comitê diversifica a busca por boas distribuições iniciais de pesos, utilizando uma estratégia de agrupamento dos exemplos pelos seus atributos.

Dois dos passos desse algoritmo são importantes. São eles a inicialização dos pesos e a combinação dos classificadores.

4.1
Inicialização dos pesos

A inicialização dos pesos é o passo do algoritmo Comitê BAS onde são determinados os possíveis valores a serem utilizados como pesos iniciais por cada membro do comitê a ser treinado.

Para construir os membros do comitê, adotamos duas estratégias simultâneas. A primeira é perturbar o algoritmo BAS, atribuindo diferentes conjuntos de pesos a cada membro do comitê. A segunda é determinar que, em cada membro, dois exemplos tem o mesmo peso inicial se e somente se estão no mesmo grupo.

O agrupamento dos exemplos é realizado com base em seus atributos. Diferentes estratégias podem ser aplicadas dependendo da natureza dos exemplos.

4.1.1
Dados provenientes de tabelas

A primeira estratégia é apropriada para exemplos provenientes de tarefas de classificação de dados. Nesse caso, os exemplos são transformados em pontos no espaço. Normalmente, os atributos desses exemplos são numéricos ou

categóricos e, em ambos os casos, os seus valores poder ser ordenados segundo algum critério a ser explorado pelo algoritmo de agrupamento.

Por exemplo, um conjunto de dados sobre carros pode conter entre seus atributos, os atributos *preço* e *número de portas*. O preço pode ter os seguintes valores: *muito alto*, *alto*, *médio* e *baixo*. Tais valores podem ser ordenados e uma métrica de distância pode ser inferida. Já o número de portas pode ter os seus valores variando entre 2 e 5. Mesmo tais valores sendo numéricos, eles podem ser encarados da mesma forma como categorias e a ordenação e a métrica para distância determinadas.

Os exemplos são então convertidos e normalizados em pontos no espaço, onde cada dimensão corresponde a um atributo. O algoritmo de agrupamento é então aplicado utilizando uma métrica de distância padrão como por exemplo a distância euclidiana (Nis05).

A distância σ_e entre dois pontos $P^1 = (P_1^1, P_2^1, \ldots, P_n^1)$ e $P^2 = (P_1^2, P_2^2, \ldots, P_n^2)$ em um espaço euclideano de dimensão n é definida como

$$\begin{aligned} \sigma_e(P_1, P_2) &= \sqrt{(P_1^1 - P_1^2)^2 + (P_2^1 - P_2^2)^2 + \cdots + (P_n^1 - P_n^2)^2} \\ &= \sqrt{\sum_{k=1}^{n} (P_k^1 - P_k^2)^2} \end{aligned}$$

Caso algum dos atributos não possua uma ordenação inerente, como por exemplo o atributo cor, onde os valores vermelho, verde e azul não possuem uma distância calculável, simplesmente atribuímos uma distância cujo valor é zero para valores diferentes e uma distancia com valor um para valores iguais, na dimensão definida por tal atributo.

4.1.2 Dados Textuais

A segunda estratégia é utilizada em tarefas de processamento de linguagem natural, onde os exemplos são representados por sentenças. Tais sentenças possuem diversos atributos por unidade léxica. Entretanto, em nossas abordagens, o mais importante é o atributo palavra.

O objetivo aqui é criar uma métrica de distância baseada na similaridade entre sentenças. Um dos critérios de similaridade mais utilizados em tarefas de Recuperação de Informação é o peso TF-IDF.

TF-IDF

O peso *TF-IDF* (Wit99) é uma medida estatística que determina a importância de uma palavra para um documento em uma coleção de documentos. Este peso aumenta proporcionalmente ao número de vezes que uma palavra aparece em um documento, compensando pela frequência da palavra na coleção completa. Usualmente, o peso *TF-IDF* é utilizado em ferramentas de busca com o objetivo de determinar a relevância e ordenar documentos dada uma consulta.

A frequência de um termo *TF* em um documento é definida como sendo o número de vezes que o termo aparece no documento. Entretanto, tal frequência deve ser normalizada de forma a prevenir um viés para documentos longos e determinar uma medida de importância do termo t_i no documento d_j. Levando isso em conta, a frequência do termo é definida como

$$tf_{i,j} = \frac{f_{i,j}}{\sum_{k=1}^{K} f_{k,j}}$$

onde $f_{i,j}$ é a quantidade de ocorrências do termo t_i no documento d_j e K é o quantidade de termos distintos.

A frequência de documento invertida *IDF* é a medida da importância geral do termo. Ela é definida como sendo o logaritmo do quociente entre o número total de documentos D e o número de documentos que contém o termo, sendo dada por

$$idf_i = \log \frac{D}{|D_{t_i}|}$$

onde D_{t_i} é o conjunto de documentos que contém o termo t_i e $|D_{t_i}|$ é o tamanho de tal conjunto.

O peso *TF-IDF* então é calculado através do produto entre os dois valores *TF* e *IDF*

$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_i$$

O peso *TF-IDF* é diretamente proporcional à frequência do termo no documento e inversamente proporcional à frequência do termo na coleção de documentos.

Utilizando tais grandezas, a medida de similaridade entre dois documentos é definida como sendo o produto entre os dois pesos *TF-IDF*, normalizada pelo tamanho do documento base utilizado.

Assim, temos

$$Sim(D_i, D_j) = \frac{\sum_{t \in D_i} tf\text{-}idf_{t,i} \times tf\text{-}idf_{t,j}}{|D_j|}$$

Uma medida de distância entre dois documentos pode ser derivada da similaridade, simplesmente assumindo que dois documentos muito similares devem possuir uma distância entre eles baixa e, dois documentos pouco similares devem possuir uma distância alta.

Levando em conta tal constatação, a distância $\sigma_{tf\text{-}idf}$ entre dois documentos D_1 e D_2 baseada no peso *TF-IDF* é definida como

$$\begin{aligned} \sigma_{tf\text{-}idf}(D_1, D_2) &= \frac{1}{Sim(D_1, D_2)} \\ &= \frac{|D_2|}{\sum_{t \in D_i} tf\text{-}idf_{t,1} \times tf\text{-}idf_{t,2}} \end{aligned}$$

Agrupando Sentenças

Com o objetivo de agrupar os exemplos, cada sentença do corpus é considerada como sendo um documento e a medida de distância utilizada é a distância *TF-IDF* dada por $\sigma_{tf\text{-}idf}$. Cabe ressaltar aqui que a relação definida por essa distância não é simétrica, ou seja,

$$\sigma_{tf\text{-}idf}(D_i, D_j) \neq \sigma_{tf\text{-}idf}(D_j, D_i),$$

pois a fórmula da distância necessita de um documento-base.

Tal característica, entretanto, não se configura um óbice pois a distância *TF-IDF* é utilizada em um algoritmo de agrupamento que calcula distâncias de pontos a pontos-base chamados centroides que são determinados pelo próprio algoritmo de agrupamento. Assim, os documentos-centroides encontrados são considerados como os documentos-base, sendo as distâncias calculadas em relação a eles.

4.1.3 Incorporando Amostras no Treinamento

Em conjuntos de treinamento pequenos, por vezes o agrupamento dos exemplos não é efetivo, levando a uma distribuição inicial que não reflete as características do conjunto que se deseja.

Nesse caso, podem ser empregadas amostras para ajudar a balancear os grupos de exemplos. O agrupamento de exemplos é então mais próximo do desejado e o desempenho do classificador BAS é melhorado.

Essa estratégia de utilização de amostras é chamada de Comitê BAS Semi-Supervisionado.

Aprendizado Semi-Supervisionado

Aprendizado supervisionado é uma abordagem de aprendizado de máquina que aprende uma função através de dados rotulados, ou exemplos.

Normalmente, os dados rotulados são divididos em conjuntos de treinamento e teste. Tal divisão é necessária para assegurar que o modelo está generalizando e não somente decorando os exemplos. Um bom desempenho no conjunto de teste é um bom indicativo que o modelo tem um bom desempenho em dados novos.

Infelizmente, a geração de dados rotulados é muito cara e depende da habilidade de um agente humano para, manualmente, rotular os exemplos.

Aprendizado semi-supervisionado, em contrapartida, é uma abordagem que utiliza tanto dados rotulados quanto dados não-rotulados, ou amostras, no treinamento.

O objetivo principal do aprendizado semi-supervisionado é tirar vantagem de grandes quantidades baratas de amostras que são um sub-produto de processos ordinários. Utilizando esse grande conjunto de amostras, não é difícil inferir propriedades estatísticas do domínio que podem ser bastante úteis no desenvolvimento de esquemas de treinamento eficientes.

Dois exemplos de abordagens de aprendizado de semi-supervisionado são o Auto-treinamento (*Self-Training*) (Car03), que gera classificadores iterativamente, classificando as amostras com o classificador corrente e incorporando esses novos *pseudo-exemplos* no próximo classificador a ser treinado e o Co-treinamento (*Co-Training*) (Blu98), que é bastante similar ao Auto-treinamento, a não ser pelo fato de que utiliza dois algoritmos de aprendizado de máquina que trocam *pseudo-exemplos* com alta confiança de classificação.

Comitê BAS Semi-Supervisionado

Uma importante fase do algoritmo Comitê BAS é o agrupamento dos exemplos que determina as possíveis distribuições iniciais para os algoritmos BAS do comitê. Algoritmos de agrupamentos se encontram na classe de algoritmos não-supervisionados, ou seja, não necessitam de rótulos para os dados.

Caso estejam disponíveis amostras do mesmo domínio dos dados de treinamento, elas podem ser combinadas aos exemplos a serem agrupados.

O algoritmo de agrupamento é então aplicado neste conjunto estendido e os grupos de amostras e exemplos são determinados. Após isso, as amostras

são descartadas e os grupos são utilizados para determinar os pesos iniciais dos exemplos da mesma forma que no esquema completamente supervisionado.

Caso o algoritmo-base a ser utilizado pela estratégia BAS suporte a utilização de amostras no seu treinamento, tais amostras também podem ser utilizadas de forma a melhorar o treinamento do classificador-base. Nesse caso, as amostras não são descartadas e o peso de cada amostra será determinado pelo peso dos exemplos encontrados em seu grupo.

4.2 Combinação dos Classificadores

Outro passo importante do algoritmo Comitê BAS, bem como de qualquer algoritmo de comitê, é a forma de combinação dos classificadores obtidos pelo modelo.

Nesta seção, apresentamos as duas formas de combinação adotadas: Votação pela Melhor Maioria e Votação ETL.

4.2.1 Votação pela Melhor Maioria

Nesse esquema, escolhemos um subconjunto dos classificadores, de tamanho N' segundo um critério de otimalidade.

Cada classificador obtido no comitê é avaliado em um conjunto de validação extraído do conjunto de treinamento inicial. Por meio de uma estratégia gulosa, os N' melhores classificadores BAS, em termos de desempenho no conjunto de validação, são selecionados. Cada classificador possui um poder de voto igual.

Para conjuntos de dados de treinamento muito grandes tal esquema pode não ser eficiente pois o número total de classificadores-base gerados pode ser muito grande. Neste caso, no lugar de controlarmos o número de classificadores BAS treinados e escolhidos pelo comitê final, podemos controlar o número de classificadores-bases M, treinados pelos classificadores BAS e o número de classificadores-base M', escolhidos para formar o comitê BAS final.

4.2.2 Votação ETL

Apesar de eficiente, claramente uma votação pela maioria não é sempre a forma ideal de se combinar classificadores. Uma possibilidade interessante de combinação é a geração de regras de classificação, como no exemplo a seguir.

"Se o voto do classificador$_1$ for igual a classe$_a$ e o voto do classificador$_2$ for igual a classe$_a$ então classifique como classe$_a$."

Supondo que tal comitê possua mais do que quatro classificadores, mesmo se todos os outros votos forem para a classe$_b$, a classe escolhida seria a classe$_a$, diferente do comportamento padrão de uma votação pela maioria.

Um algoritmo muito utilizado para criação de regras de classificação é o TBL, descrito na Seção 5.2, que gera regras a partir de padrões de correção de erros chamados gabaritos. O TBL lida com a determinação da melhor sequência de combinação de atributos a serem aplicados de forma a corrigir um classificação incorreta. Infelizmente, o TBL depende de um bom conjunto de gabaritos de forma a garantir um bom desempenho na correção dos erros encontrados. O TBL, então, não pode ser diretamente aplicado para construir um comitê porque o processo de treinamento teria que ser "interrompido" para que um especialista determine um conjunto de gabaritos que melhor combine os classificadores.

O ETL (Mil08a), descrito na Seção 5.4, vem solucionar tal problema, determinando automaticamente um bom conjunto de gabaritos utilizando árvores de decisão aplicadas ao conjunto de treinamento. Como o ETL necessita de um classificador de forma a corrigir os seus erros, a Votação pela Maioria é utilizada como classificador inicial.

Utilizando tal configuração, o ETL determina os melhores conjuntos de gabaritos e regras utilizando como atributos os votos, {voto$_{maioria}$, voto$_1$, voto$_2$, ..., voto$_N$}. Tal procedimento evita a necessidade do parâmetro N', ou M', pois combinações mais complexas são geradas para os votos dos classificadores.

Por exemplo, nos experimentos conduzidos com o conjunto de dados *Contraceptive Method Choice* (CMC) descrito na Seção 6.3.1, o ETL determina os conjuntos de gabaritos e regras mostrados na Tabela 4.1.

Tabela 4.1: Conjuntos de gabaritos e regras para o conjunto CMC.

Gabaritos	**Regras**
voto$_2$	voto$_2$=usar voto$_7$=usar voto$_6$=não-usar voto$_4$=usar → não-usar
voto$_2$ voto$_7$	voto$_2$=usar voto$_7$=usar voto$_6$=não-usar voto$_4$=não-usar → usar
voto$_2$ voto$_7$ voto$_6$	voto$_3$=não-usar → não-usar
voto$_2$ voto$_7$ voto$_6$ voto$_4$	voto$_2$=usar voto$_7$=não-usar voto$_4$=usar → usar
voto$_2$ voto$_7$ voto$_4$	voto$_2$=não-usar voto$_7$=não-usar voto$_6$=usar voto$_4$=usar → não-usar
voto$_2$ voto$_3$	voto$_7$=não-usar voto$_6$=usar voto$_4$=não-usar → usar
voto$_7$	
voto$_7$ voto$_6$	
voto$_7$ voto$_6$ voto$_4$	
voto$_7$ voto$_4$	
voto$_3$	

Nesse exemplo, cabe ressaltar que o ETL somente utiliza os votos do segundo, terceiro, quarto, sexto e sétimo membros de forma a corrigir o voto da maioria. Também, caso todas as condições não sejam satisfeitas no conjunto de regras gerado, o voto da maioria é mantido como classificação final.

4.3 Descrição do algoritmo

Agora, formalizamos o algoritmo Comitê BAS ilustrado na Figura 4.1

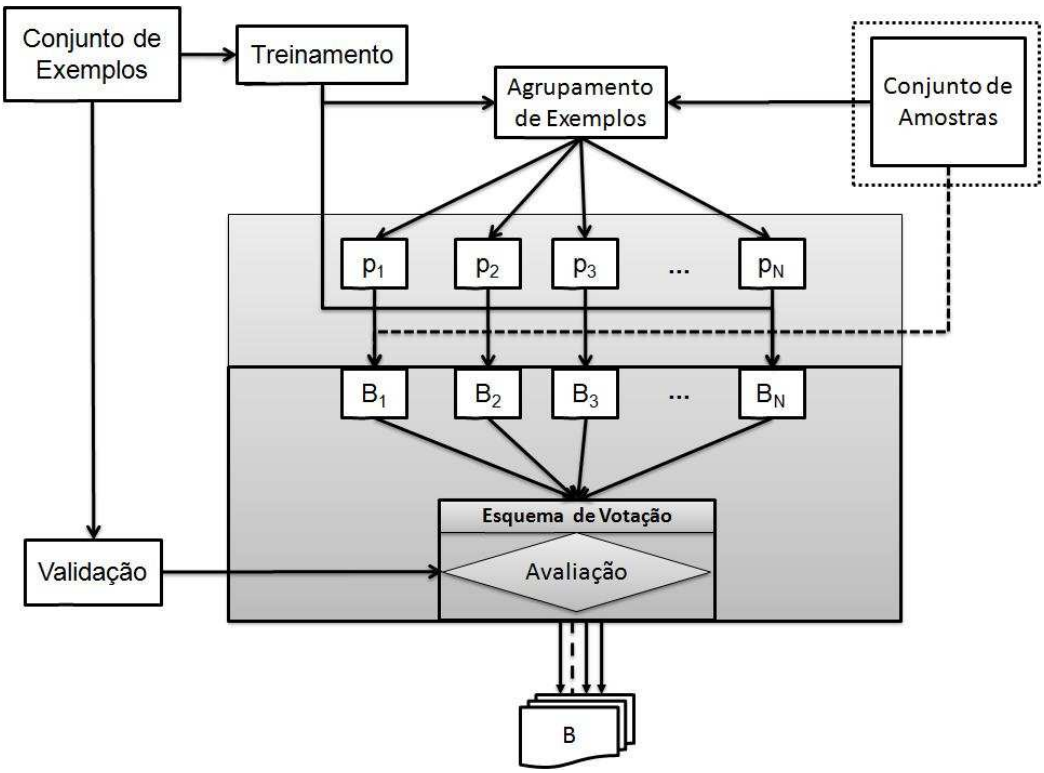

Figura 4.1: Esquema de aprendizado do Comitê BAS.

Primeiro, dividimos o conjunto de exemplos em dois subconjuntos, um menor, chamado conjunto de validação e outro maior, chamado conjunto de treinamento.

A seguir, os exemplos do conjunto de treinamento e as amostras são agrupados em k grupos. Tal agrupamento é realizado baseando-se exclusivamente em seus atributos. O algoritmo de agrupamento utilizado pode ser qualquer um.

Duas possíveis escolhas para ser o algoritmo responsável pelo agrupamento são o algoritmo K-Médias (*K-Means*) (Mac67) e o algoritmo Gás Neural Crescente (*Growing Neural Gas - GNG*) proposto por Fritzke (Fri95).

O algoritmo K-Médias tem a vantagem de exigir poucos parâmetros de entrada quando comparado com o GNG. Sua desvantagem é que o número de grupos é fixo e deve ser informado. Diferentemente, o GNG apenas exige um valor inicial para a quantidade de grupos. Após isso, o próprio GNG realiza um ajuste conforme a estrutura do conjunto.

Como próximo passo, escolhemos uma entre várias possíveis famílias de distribuição para fornecer os pesos iniciais. Tal escolha pode ser derivada de qualquer distribuição.

Finalmente, uma permutação dessa distribuição é escolhida aleatoriamente e tais pesos são aplicados aos grupos. O objetivo aqui é "perturbar" o algoritmo BAS, por meio de diferentes distribuições iniciais para cada membro, atribuindo diferentes pesos a exemplos encontrados em diferentes grupos e mesmos pesos a exemplos encontrados no mesmo grupo.

Este processo é repetido por N iterações e, após isso, um esquema de votação é aplicado, utilizando o conjunto de validação, para determinar como os membros são combinados em um classificador final.

O pseudo-código do algoritmo Comitê BAS é mostrado no Algoritmo 4.1.

Algoritmo 4.1 Comitê BAS.

Entrada: conjunto de exemplos: CE
conjunto de amostras: CA
algoritmo de agrupamento: AA
número de grupos: k
número de famílias de distribuições de pesos: f
famílias de distribuições de pesos: $P_0, ..., P_{f-1}$
número de classificadores BAS gerados: N
esquema de votação final: V
Tr, Va = SubConjuntos(CE) // Divida o conjunto de exemplos em um de treinamento e outro de validação
$Gr = AA(Tr + CA, k)$ // Aplique o algoritmo de agrupamento ao conjunto de treinamento e obtenha a estrutura de grupos associada
para $t = 1$ **to** N **faça**
    $f' = (t-1)\%f$ // Determine a família de distribuição de pesos a ser utilizada
    $P' = P_{f'}$ // Determine os possíveis valores para os pesos
    Aleatorize(P') // Determine a distribuição de pesos a ser utilizada
    p = Aplique(Gr, P') // Determine os pesos iniciais
    B_t = BAS(Tr, CA, p) // Treine o classificador BAS
fim para
$CF = V(\{B\}, Va)$ // Aplique o esquema de Votação Final
Saída: Comitê Final CF