2
Fundamentos Teóricos

2.1
Aprendizado de Máquina

Aprendizado de Máquina é uma área de Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Possui um grande número de aplicações como, por exemplo, máquinas de busca, diagnóstico médico, detecção automática de fraude de cartão de crédito, análise de aplicações financeiras, classificação de seqüências de DNA e classificação automática de textos.

Um programa aprende um conjunto de tarefas *T* com uma medida de desempenho *D* a partir de uma experiência *E*, se seu desempenho de aprendizado *D* aumenta com a experiência *E* [Mitchell, 1997], ou seja, se é capaz de tomar decisões baseado em experiências acumuladas por meio da solução bem sucedida de problemas anteriores. Por exemplo, uma aplicação que necessite reconhecer pessoas através de um sensor tem como tarefa *T* associar imagens captadas pelo sensor a pessoas, medida de desempenho *D* como sendo a percentagem de pessoas corretamente reconhecidas e experiência de treinamento *E* como sendo um conjunto de pares ordenados compostos por uma imagem e o nome de uma pessoa.

Existe uma série de problemas que são de difícil solução que podem ser bem resolvidos com as técnicas de aprendizado de máquina. Por exemplo: algumas tarefas só podem ser bem definidas através de exemplos, onde não é conhecida nenhuma relação à priori entre os dados de entrada e a saída desejada. Além disso, tarefas que são sujeitas a muitas mudanças podem ser aprendidas pela máquina, uma vez que a máquina se adapta a tais mudanças. A capacidade de adaptação de sistemas de aprendizado de máquina também permite que tais sistemas sejam portados para domínios completamente diferentes sem a necessidade de recriar o sistema para suportar o novo domínio. Assim, se uma máquina é capaz de

aprender a classificar notícias de jornais em uma hierarquia de categorias, a mesma máquina também é capaz de aprender a filtrar spams ou até mesmo resolver problemas de ambigüidade em palavras no processamento de linguagem natural.

Para desenvolver um sistema de aprendizado é necessário tomar algumas decisões de projeto. Primeiramente, deve-se escolher que tipo de exemplos de treinamento será usado. Por exemplo, no problema de classificação automática de textos, o treinamento será um conjunto de pares compostos de um documento e sua respectiva classe.

Após a escolha do tipo de dados a ser usado como exemplo de treinamento, o próximo passo é definir exatamente que tipo de conhecimento será aprendido. O problema de aquisição de conhecimento pode, então, ser reduzido ao problema de se aprender uma função objetivo *f*, que associa valores de entrada (os específicos das tarefas a serem aprendidas) com valores de saída correspondentes, que auxiliam nas decisões ou ações a serem tomadas. O sistema gera uma hipótese *h* que se aproxima da função objetivo *f* conforme realizado o treinamento.

Em aprendizado de máquina existem diversas formas de aprendizagem, como por exemplo, Simbólico, Estatístico, Baseado em Exemplos, Conexionista e Evolutivo.

Os sistemas de aprendizado simbólico constroem representações simbólicas de um conceito através da análise de exemplos e contra-exemplos, na forma de alguma expressão lógica, árvore de decisão, regras ou rede semântica.

O aprendizado estatístico utiliza modelos estatísticos para encontrar uma aproximação da função objetivo. O problema a ser resolvido é, dado um conjunto de exemplos de distribuição de probabilidade não conhecida, descobrir a qual distribuição um novo exemplo pertence.

O aprendizado baseado em exemplos utiliza a idéia de que para determinar a saída da função objetivo, dado um valor de entrada não conhecido, deve-se buscar outro valor de entrada similar cuja saída é conhecida e assumir que o novo exemplo terá o mesmo valor de saída. Para isso, é necessário guardar os exemplos de treinamento em memória, sendo, por isso, chamado de sistemas de aprendizado "lazy", em oposição aos sistemas de aprendizado "eager", que utilizam os exemplos para induzir o modelo, descartando-os logo em seguida.

O aprendizado conexionista utiliza redes neurais, que são construções matemáticas não lineares altamente interconectadas inspiradas no fenômeno de aprendizado do cérebro humano. As conexões entre os neurônios artificiais possuem pesos que são obtidos através do treinamento, ou seja, o conhecimento adquirido durante o treinamento de uma rede neural fica armazenado nos pesos das ligações entre os neurônios artificiais. Muitas são as aplicações práticas das redes neurais, como por exemplo, reconhecimento de imagens, reconhecimento de genes em uma seqüência de DNA e previsão do movimento da bolsa de valores, a partir de uma série histórica.

O aprendizado evolutivo se baseia na lei da seleção natural de Darwin. O algoritmo é iniciado com uma população de estimadores para a função objetivo. Tais estimadores competem para realizar a reprodução e com isso produzir indivíduos similares. Indivíduos que possuem desempenho fraco tendem a ser descartados da população de estimadores, enquanto indivíduos de desempenho alto tendem a proliferar. Dessa forma, conforme o algoritmo é executado, o melhor estimador da função objetivo vai sendo aprimorado.

2.2
O problema da classificação

Em geral, existem dois tipos de raciocínio. O raciocínio dedutivo e o raciocínio indutivo. O raciocínio dedutivo é uma forma de inferência na qual a conclusão tem o mesmo grau de certeza que as premissas, em oposição ao raciocínio indutivo, onde a conclusão pode ter um grau de certeza inferior às premissas.

Um exemplo de raciocínio dedutivo:

Todos os pássaros possuem asas.

Um canário é um pássaro.

Logo, um canário possui asas.

Um exemplo de raciocínio indutivo:

Todos os corvos observados são pretos.

Logo, todos os corvos são pretos.

Através do raciocínio indutivo, é possível obter conclusões genéricas sobre um conjunto particular de exemplos. Em aprendizado de máquina, o raciocínio indutivo é utilizado generalizar conceitos, para aproximar funções, para descobrir novas funções através de exemplos fornecidos.

O aprendizado indutivo deve ser aplicado com cuidado, pois se o número de exemplos for insuficiente, ou se os exemplos não forem bem escolhidos, as hipóteses obtidas podem ser inconsistentes.

Classificação e regressão são dois tipos de aplicações de aprendizado indutivo com a finalidade de predizer valores de uma função objetivo *f*, dado um valor de entrada ainda não conhecido.

A regressão consiste em aproximar uma função contínua a partir de um conjunto de exemplos composto de pontos. Tal método pode ser aplicado na previsão de preços de um produto, valores de ações no mercado de ações e prever a ocorrência de um determinado evento (regressão logística).

Na tarefa de classificação, o objetivo do algoritmo de aprendizado é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados, dado um conjunto de classes e um conjunto de exemplos de treinamento.

O aprendizado indutivo pode ser subdivido em aprendizado não supervisionado e aprendizado supervisionado.

No aprendizado supervisionado, o conjunto de treinamento consiste de pares ordenados constituídos de um objeto (tipicamente vetores) e o seu respectivo valor da função objetivo. A saída da função objetivo pode ser contínua, no caso da regressão, ou pode ser discreta, no caso da classificação, onde os valores de saídas são rótulos de categorias.

No aprendizado não supervisionado, o conjunto de treinamento consiste apenas de exemplos sem nenhum valor de função associado. Tipicamente, o problema se resume em particionar os exemplos de treinamento em agrupamentos, ou clusters. Ainda assim, pode-se considerar o problema como um caso de aprendizagem de uma função objetivo, pois o valor da função é o nome do agrupamento ao qual um objeto de entrada pertence.

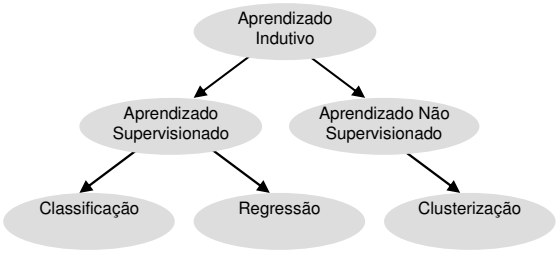

Figura 1 – Hierarquia do Aprendizado

2.3 Classificação automática de textos

A classificação de textos é a tarefa de associar textos em linguagem natural a rótulos pré-definidos. Esse problema vem sendo tratado desde os anos 60, porém só nos anos 90, a classificação de textos começou a ser amplamente aplicada, graças ao desenvolvimento de máquinas mais potentes e da facilidade de publicação de textos em forma eletrônica.

A classificação de textos é uma área que engloba conceitos de extração de informação e de aprendizado de máquina e possui muitas características em comum com outras tarefas como extração de conhecimento e mineração de textos.

A classificação de textos pode ser aplicada em uma grade variedade de contextos como, por exemplo: indexação automática de textos [Maron, 1961], identificação de autores de textos [Mosteller & Wallace, 1964], filtragem de e-mails [Graham, 2002], classificação hierárquica de páginas da internet [McCallum et al., 1998] e geração automática de metadados [Giles et al., 2003].

Considere um conjunto de documentos $D = \{d_1, d_2,..., d_{|D|}\}$ e um conjunto de categorias $C = \{c_1, c_2,..., c_{|C|}\}$. O problema da classificação automática de textos consiste em estimar uma função objetivo $\Phi : D \times C \rightarrow \{0,1\}$, que associa um valor booleano a cada par $(d_j, c_i) \in D \times C$. Um valor 1 associado ao par (d_j, c_i) indica que o documento d_j pertence à categoria c_i, enquanto que um valor 0 indica que o documento d_j não pertence à categoria c_i.

2.4
Classificação unirótulo e multirótulo

Na classificação de textos podem existir restrições relacionadas à quantidade de categorias do conjunto *C* e a quantidade de categorias associadas a cada documento d_j.

Considerando a restrição de quantidade de categorias no conjunto *C*, pode-se subdividir o problema da classificação em classificação binária, onde cada documento $d_j \in D$ está associado a uma categoria c_i ou ao seu complemento $\overline{c_i}$, e na classificação multicategoria, onde o conjunto de categorias *C* possui mais de duas categorias.

Considerando a restrição de quantidade de categorias associadas a cada documento, a classificação unirótulo corresponde ao caso em que um documento $d_j \in D$ está associado a exatamente uma categoria. Nesse caso, as categorias são mutuamente exclusivas.

A classificação multirótulo corresponde ao caso em que um documento $d_j \in D$ está associado a zero ou mais categorias, onde tais categorias não são mutuamente exclusivas.

Pode-se verificar que a classificação do tipo binária é um caso particular da classificação unirótulo.

2.5
Conjuntos de treinamento, validação e teste

A classificação de textos baseada nas técnicas de máquina de aprendizado necessita de um conjunto de documentos (corpus) $\Omega = \{d_1,...,d_{|\Omega|}\} \subset D$ pré-classificados nas categorias do conjunto $C = \{c_1,...,c_{|C|}\}$. Para a construção de um classificador divide-se o conjunto Ω em três subconjuntos disjuntos: um conjunto de treinamento $T_r = \{d_1,...,d_{|Tr|}\}$, um conjunto de validação $T_v = \{d_1,...,d_{|T_v|}\}$ e um conjunto de teste $T_e = \{d_1,...,d_{|T_e|}\}$. Em alguns casos, o conjunto de validação T_v é vazio.

O classificador é construído através do conjunto de treinamento T_r, e os parâmetros são ajustados através de repetidos testes realizados no conjunto de

validação T_v. Então, através do conjunto de teste T_e, são realizados testes para medir a eficiência do algoritmo de classificação.

Os subconjuntos são disjuntos para assegurar que os resultados experimentais obtidos, através do conjunto de teste, sejam de um conjunto diferente do usado para realizar o aprendizado, tornando os resultados estatisticamente válidos.

2.6 Representação de documentos

A forma mais simples de representar documentos é associar ao documento d_j um vetor de pesos $\vec{d}_j = \{w_{1j}, ..., w_{|V|j}\}$ onde V é o conjunto de termos que ocorrem em pelo menos um documento de T_r e o peso w_{kj} que representa, grosso modo, quanto o termo t_k contribui para a semântica do documento d_j.

Existem diversas abordagens que diferem na definição do que significa o termo t_k e como calcular os pesos w_{kj}.

A representação mais abordada na literatura de classificação de textos é conhecida como "bag of words". Nessa abordagem, cada termo corresponde a uma única palavra no conjunto de palavras do conjunto de treinamento.

Lewis, [1992], mostrou que representações de documentos mais sofisticadas, como frases, resultaram em um pior desempenho em experimentos rodados na base de notícias da Reuters. Além disso, [Scott & Matwin, 1999] acrescentaram informação semântica à tarefa de classificação de textos e não obtiveram resultados satisfatórios.

Em contraste, outros trabalhos apresentaram melhor desempenho na utilização de frases [Mladenić & Grobelnik, 1998] e reconhecimento de nomes próprios [Basili, 2000], comparados à representação tradicional "bag of words". Desta forma, a eficácia de modelos mais complexos ainda necessita de mais estudos.

Com relação ao cálculo dos pesos w_{kj}, a abordagem mais conhecida e muito comum em sistemas de recuperação de informação é a combinação de duas medidas, "Term Frequencie" (*tf*) e "Inverse Document Frequencie" (*idf*) definida como:

$$tfidf(t_k, d_j) = tf(t_k, d_j) idf(t_k, d_j)$$
$$tf(t_k, d_j) = \#(t_k, d_j) \qquad (1)$$
$$idf(t_k, d_j) = \log \frac{|Tr|}{\#Tr(t_k)}$$

Onde $\#(t_k, d_j)$ é a freqüência do termo t_k no documento d_j, $\#Tr(t_k)$ é o número de documentos do conjunto T_r que possuem pelo menos uma ocorrência do termo t_k.

O termo $tf(t_k, d_j)$ indica a importância do termo t_k no documento d_j. Dessa forma, se um termo é muito freqüente no documento d_j, então ele deve ser importante para representação do documento. Porém, um termo que seja freqüente em muitos documentos do conjunto T_r não é um termo representativo para o documento d_j. Desta forma, calcula-se a medida *idf*, que é inversamente proporcional à quantidade de documentos em que o termo t_k ocorre.

Com o propósito de igualar o tamanho dos vetores que representam os documentos, o valor da medida *tfidf* é normalizado:

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|V|} (tfidf(t_s, d_j))^2}} \qquad (2)$$

Outra abordagem muito usada em classificadores probabilísticos é representar um documento através de pesos binários, onde um valor 0 representa a ausência e o valor 1 representa a presença do termo no documento [Robertson & Sparck Jones, 1976].

A abordagem utilizada nesse trabalho representa um documento através de um vetor de freqüências, onde cada componente corresponde à freqüência do termo no documento.

2.7 Pré-processamento de documentos

A preparação dos textos é a primeira fase do processo de criação indutiva de classificadores de texto. Esta fase envolve a seleção de termos que melhor expressam o conteúdo de textos, ou seja, toda a informação que não refletir nenhuma idéia considerada importante é desconsiderada.

Desta forma, a seleção de termos reduz a quantidade de termos e, por conseguinte, a dimensão dos vetores que representam os documentos. Uma

redução da dimensão dos vetores implica em uma menor quantidade de memória utilizada e em menor processamento. Além disso, reduz a possibilidade de "overfitting", fenômeno que ocorre quando o classificador é ajustado de forma muito específica para o conjunto de treinamento, implicando em uma baixa perfomance na classificação de documentos não conhecidos pelo classificador [Sebastiani, 1999].

2.7.1 Stopwords

"Stopwords" são palavras consideradas não relevantes para a análise de textos. Na maioria das vezes são palavras auxiliares ou conectivas, não fornecendo nenhuma informação discriminativa na expressão do conteúdo do texto. Palavras como pronomes, artigos, preposições e conjunções podem ser consideraradas "stopwords".

Para cada língua existe um conjunto de "stopwords" (também conhecido como "stoplist"). Uma vez definida a lista de "stopwords", para cada documento, são retiradas as ocorrências no texto de todas as "stopwords".

2.7.2 Stemming

A tarefa de "stemming" consiste em agrupar palavras que possuem a mesma raiz morfológica.

Algumas técnicas de "stemming" serão apresentadas a seguir, com o objetivo de elucidar as diferentes abordagens utilizadas pelos algoritmos existentes.

2.7.2.1. Método de Stemmer S

O método mais simples de "stemming" é o stemmer S [Harman, 1991], no qual apenas alguns finais de palavras são removidos. Por exemplo, os finais de palavras da língua inglesa "ies", "es" e "s" (com exceções). Embora o stemmer S não descubra muitas fusões, alguns sistemas o usam, pois ele é conservador e raramente surpreende o usuário.

2.7.2.2. Método de Porter

O método de "stemming" de Porter [Porter, 1980] consiste na identificação das diferentes inflexões de uma mesma palavra e sua substituição por um radical comum. O algoritmo remove 60 sufixos diferentes em uma abordagem composta de cinco fases.

Termos com um "stem" comum muitas vezes possuem significados similares, por exemplo:

ESTUDO

ESTUDOS

ESTUDAR

ESTUDADO

2.7.2.3. Método de Lovins

O método de Lovins [Lovins, 1968] é composto de um único passo, é sensível ao contexto e usa um algoritmo de combinação mais longa para extrair em torno de 250 sufixos distintos. Tal método retira no máximo um sufixo por palavra, removendo o sufixo mais longo. Comparado aos outros dois métodos apresentados, este método é o mais agressivo.

2.7.3 Freqüência de Documentos (DF)

Freqüência de documentos (DF) é o número de documentos no qual um termo ocorre. A idéia desse método é calcular a freqüência DF de cada termo e remover o termo do vocabulário do corpus, se a freqüência for menor que um determinado limiar. A suposição básica é a de que termos raros são não-informativos para predizer a categoria ou não influenciam o desempenho global.

Freqüência de documentos é a técnica mais simples de redução de termos. Ela é facilmente escalável para conjuntos de muitos documentos, com uma complexidade computacional praticamente linear em relação à quantidade de documentos.

2.7.4
Ganho de Informação (IG)

Ganho de informação é amplamente empregado como um critério de importância de termos no campo de aprendizado de máquina. Através dessa técnica, é medida a "quantidade de informação" relativa à predição da categoria, pela presença ou ausência de um termo em um documento. Dado um conjunto de categorias $C = \{c_1, c_2,..., c_{|C|}\}$, o ganho de informação de um termo t_k é definido como:

$$G(t_k) = -\sum_{i=1}^{|C|} P(c_i)\log P(c_i)$$
$$+P(t_k)\sum_{i=1}^{|C|} P(c_i \mid t_k)\log P(c_i \mid t_k) \quad (3)$$
$$+P(\overline{t_k})\sum_{i=1}^{|C|} P(c_i \mid \overline{t_k})\log P(c_i \mid \overline{t_k})$$

Dado um conjunto de documentos de treinamento, para cada termo, é calculado o ganho de informação e são removidos do corpus os termos que possuírem um ganho de informação inferior a um limiar pré-determinado.

2.7.5
Informação Mútua (MI)

Informação mútua é um critério normalmente usado em modelagem estatística da linguagem em associações de palavras. Dado um termo t_k e uma categoria c_i, considere A o número de vezes que t_k e c_i co-ocorrem, B o número de vezes que t_k ocorre sem c_i, C o número de vezes que c_i ocorre sem t_k e N o número total de documentos de treinamento. A medida de informação mútua é definida como:

$$I(t_k, c_i) = \frac{P(t_k \wedge c_i)}{P(t_k)P(c_i)} \quad (4)$$

Estima-se a medida de informação mútua usando-se:

$$I(t_k, c_i) \approx \frac{\log A \times N}{(A+C)(A+B)} \quad (5)$$

A medida $I(t_k, c_i)$ tem o valor de zero caso t_k e c_i forem independentes. Para medir a importância de um termo globalmente, combinam-se as pontuações de um termo específicas para cada categoria em duas formas alternativas:

$$I_{avg}(t_k) = \sum_{i=1}^{|C|} P(c_i) I(t_k, c_i) \quad (6)$$

$$I_{\max}(t_k) = \max_{i=1}^{|C|} \{ I(t_k, c_i) \} \quad (7)$$

Desta forma, para cada termo, é calculada a informação mútua e são removidos do corpus os termos que possuírem um valor inferior a um limiar pré-determinado.

2.7.6 Estatística χ^2 (CHI)

A estatística χ^2 mede o grau de dependência entre um termo t_k e uma categoria c_i. Considerando A o número de vezes que t_k e c_i co-ocorrem, B o número de vezes que t_k ocorre sem c_i, C o número de vezes que c_i ocorre sem t_k e D o número de vezes que nem c_i nem t_k ocorrem, e N o número total de documentos de treinamento, a medida é definida por:

$$\chi^2(t_k, c_i) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad (8)$$

Assim como a medida de informação mútua, calcula-se o grau de importância global de um termo t_k de duas formas:

$$\chi^2(t_k) = \sum_{i=1}^{|C|} P(c_i) \chi^2(t_k, c_i) \quad (9)$$

$$\chi^2_{\max}(t_k) = \max_{i=1}^{|C|} \{ \chi^2(t_k, c_i) \} \quad (10)$$

2.8 Classificadores probabilísticos

Classificadores probabilísticos utilizam probabilidades para aproximar a função objetivo do problema da classificação de textos.

Desta forma, dado um conjunto de categorias $C = \{c_1, ..., c_{|C|}\}$ e um documento $\vec{d}_j = \{w_{1j}, ..., w_{|V|j}\}$ não conhecido, o algoritmo estima a probabilidade do documento pertencer a cada uma das categorias do conjunto C, representada

por $P(c_i \mid \vec{d}_j)$, e classifica o documento $\vec{d}_j$ na categoria com maior probabilidade estimada.

Desta forma, a função que aproxima a função objetivo da tarefa de classificação pode ser definida como:

$$\breve{\Phi}(\vec{d}_j, c_k) = 1 \leftrightarrow c^*(\vec{d}_j) = c_k$$
$$\breve{\Phi}(\vec{d}_j, c_k) = 0 \leftrightarrow c^*(\vec{d}_j) \neq c_k \quad (11)$$

$$\begin{aligned} c^*(\vec{d}_j) &= \arg\max_{c_i}\{P(c_i \mid \vec{d}_j)\} \\ &= \arg\max_{c_i}\{\log P(c_i \mid \vec{d}_j)\} \\ &= \arg\max_{c_i}\{\log(P(c_i)P(\vec{d}_j \mid c_i))\} \end{aligned} \quad (12)$$

Para calcular a probabilidade $P(c_i \mid \vec{d}_j)$, o algoritmo faz uso do teorema de Bayes:

$$P(c_i \mid \vec{d}_j) = \frac{P(c_i)P(\vec{d}_j \mid c_i)}{P(\vec{d}_j)} \quad (13)$$

Onde $P(c_i)$ é a probabilidade de um documento escolhido aleatoriamente pertencer à categoria c_i, $P(\vec{d}_j)$ é a probabilidade de um documento ser representado pelo vetor $\vec{d}_j$ e $P(\vec{d}_j \mid c_i)$ é a probabilidade de um documento ser representado pelo vetor $\vec{d}_j$, dado que ele pertence à categoria c_i.

2.8.1 Naive Bayes

O classificador naive Bayes assume que existe independência entre os termos de um documento. Tal hipótese simplificadora é muito criticada por não representar a realidade, porém Domingos & Pazzani, [1997] mostraram teoricamente que a suposição de independência de palavras na maioria dos casos não prejudica a eficiência do classificador.

Basicamente, existem dois tipos de modelos estatísticos para os classificadores naive Bayes que serão apresentados nas duas seções a seguir.

2.8.1.1. Modelo binário

O modelo binário representa um documento através de um vetor binário, onde um valor 0 na posição *k* significa que o documento não possui nenhuma ocorrência do termo t_k e um valor 1 significa que o documento possui pelo menos uma ocorrência do termo t_k. Desta forma, calcula-se:

$$P(\vec{d}_j \mid c_i) = \prod_{k=1}^{|V|} P(w_{kj} \mid c_i) \quad (14)$$

$$P(w_{kj} \mid c_i) = P(t_k \mid c_i)^{w_{kj}} (1 - P(t_k \mid c_i))^{1-w_{kj}} \quad (15)$$

$$P(t_k \mid c_i) = \frac{1 + \sum_{x=1}^{|Tr|} w_{kx} \Phi(d_x, c_i)}{2 + \sum_{s=1}^{|V|} \sum_{x=1}^{|Tr|} w_{sx} \Phi(d_x, c_i)} \quad (16)$$

Com o propósito de evitar que a probabilidade $P(\vec{d}_j \mid c_i)$ seja 0 simplesmente porque uma palavra do documento d_j não ocorre em nenhum documento da categoria c_i, o valor 1 é somado ao numerador e o valor 2 é somado ao denominador. Esta técnica é chamada de amortização de Laplace, sendo muito utilizada na literatura.

Substituindo-se a equação (15) na equação (14) tem-se:

$$\begin{aligned} P(\vec{d}_j \mid c_i) &= \prod_{k=1}^{|V|} P(t_k \mid c_i)^{w_{kj}} (1 - P(t_k \mid c_i))^{1-w_{kj}} \\ &= \left(\frac{P(t_k \mid c_i)}{1 - P(t_k \mid c_i)} \right)^{w_{kj}} (1 - P(t_k \mid c_i) \end{aligned} \quad (17)$$

Desta forma, através da equação (12), pode-se derivar:

$$c^*(\vec{d}_j) = \arg\max_{c_i} \left\{ \begin{array}{l} \log P(c_i) + \sum_{k=1}^{|V|} w_{kj} \log \frac{P(t_k \mid c_i)}{1 - P(t_k \mid c_i)} + \\ \sum_{k=1}^{|V|} w_{kj} \log(1 - P(t_k \mid c_i)) \end{array} \right\} \quad (18)$$

Para calcular $P(c_i)$:

$$P(c_i) = \frac{\sum_{k=1}^{|Tr|} \Phi(d_k, c_i)}{|Tr|} \quad (19)$$

2.8.1.2. Modelo multinomial

Já o modelo multinomial representa um documento através de um vetor de freqüências, onde o peso w_{kj} representa a freqüência do termo t_k no documento d_j.

O modelo se baseia na distribuição multinomial e calcula $P(\vec{d}_j \mid c_i)$ da seguinte forma:

$$P(\vec{d}_j \mid c_i) = P(\mid d_j \mid) \mid d_j \mid ! \prod_{k=1}^{|V|} \frac{P(t_k \mid c_i)^{w_{kj}}}{w_{kj}!} \quad (20)$$

$$P(t_k \mid c_i) = \frac{1 + \sum_{x=1}^{|Tr|} w_{kx} \Phi(d_x, c_i)}{|V| + \sum_{s=1}^{|V|} \sum_{x=1}^{|Tr|} w_{sx} \Phi(d_x, c_i)} \quad (21)$$

Substituindo-se a formula (20) na fórmula (12), tem-se:

$$c^*(d_j) = \arg\max_{c_i} \left(\log P(c_i) + \sum_{k=1}^{|V|} w_{kj} \log P(t_k \mid c_i) \right) \quad (22)$$

2.9 Medidas de desempenho

Nesta seção serão apresentadas as principais medidas de eficiência para classificadores e inclusive as medidas de eficiência utilizadas nos experimentos realizados sobre os dois algoritmos propostos.

2.9.1 Matriz de contingência

A matriz de contingência de um classificador oferece uma medida efetiva do modelo de classificação, uma vez que apresenta o número de classificações corretas versus as classificações preditas pelo algoritmo para cada classe. Dado um conjunto de teste T_e e um conjunto de categorias $C = \{c_1, c_2, ..., c_{|C|}\}$, os resultados são totalizados em duas dimensões: classificação verdadeira e classificação predita, onde :

$$M(c_i, c_j) = \sum_{\{d_k \in T_e : \Phi(d_k, c_i) = 1\}} \| \breve{\Phi}(d_k, c_j) = 1 \| \quad (23)$$

Classe	Predita c_1	Predita c_2	...	Predita $c_{\|C\|}$
Verdadeira c_1	$M(c_1, c_1)$	$M(c_1, c_2)$	...	$M(c_1, c_{\|C\|})$
Verdadeira c_2	$M(c_2, c_1)$	$M(c_2, c_2)$	...	$M(c_2, c_{\|C\|})$
⋮	⋮	⋮	⋮	⋮
Verdadeira $c_{\|C\|}$	$M(c_2, c_1)$	$M(c_2, c_1)$	$M(c_2, c_1)$	$M(c_{\|C\|}, c_{\|C\|})$

Tabela 1 – Matriz de contingência de um classificador.

O número de acertos para cada classe, se localiza na diagonal principal da matriz. Os demais elementos representam erros na classificação. A matriz de contingência de um classificador ideal possui todos esses elementos iguais à zero.

Por simplicidade, considere um problema de classificação binária de uma classe c_i. O problema deve classificar documentos de teste em c_i ou $\overline{c_i}$. Desta forma, tem-se a matriz de contingência:

Classe	Predita c_i	Predita $\overline{c_i}$
Verdadeira c_i	Verdadeiros positivos TP_i	Falsos negativos FN_i
Verdadeira $\overline{c_i}$	Falsos positivos FP_i	Verdadeiros negativos TN_i

Tabela 2 – Matriz de contingência de um classificador binário.

2.9.2 Precision e Recall

A eficiência da tarefa de classificação normalmente é calculada através de medidas clássicas de aquisição de informação, chamadas de recall (p) e de precision (π). Dada uma categoria c_i, recall e precision associadas à categoria c_i são definidas como:

$$p_i = \frac{TP_i}{TP_i + FN_i} \quad (24)$$

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \quad (25)$$

Além das duas medidas associadas a cada categoria, também são usadas medidas globais:

Micro recall:

$$\mu\rho = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \quad (26)$$

Micro precision:

$$\mu\pi = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \quad (27)$$

Macro recall:

$$M\rho = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|} \quad (28)$$

Macro precision:

$$M\pi = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \quad (29)$$

Essas duas medidas globais normalmente apresentam resultados um pouco diferentes, principalmente devido à distribuição de documentos de treinamento em relação às categorias. Caso a distribuição de documentos seja disforme, um classificador com um bom desempenho deve dar mais ênfase para as medidas de macro recall e macro precision e menos ênfase para as medidas de micro recall e micro precision.

Para um classificador ter uma performance boa, não basta ter somente a medida de recall alta, ou a medida de precision alta, isoladamente.

Considere o exemplo de um classificador que classifica um documento em todas as classes. Tal classificador possuirá a medida recall muito alta, uma vez que a quantidade de falsos negativos é nula, porém uma medida precision baixa.

Desta forma, é necessária uma medida que combine as duas medidas. Muitas medidas de combinação de recall e precision, sendo que as mais utilizadas são:

- Eleven-point average precision: os parâmetros do classificador, (por exemplo, limiares), são ajustados com o propósito da medida p variar de 0.0, 0.1,..., 0.9, 1.0. Além disso, a medida π é calculada para as 11 iterações e é calculada a média dos 11 valores de π.
- Breakeven point: são realizadas diversas iterações, variando-se os parâmetros do classificador e é gerado gráfico das medidas p e π. O

breakeven point é o ponto onde as medidas p e π possuem o mesmo valor.

- A função F_β, que é uma combinação das medidas p e π, e é definida como: $F_\beta = \frac{(\beta^2+1)\pi\rho}{\beta^2\pi+\rho}$ (30). Normalmente, um valor β =1 é usado, dando igual importância para as medidas p e π, e é chamada de F_1. A medida F_1 foi utilizada nesse trabalho para medir o desempenho dos algoritmos propostos.

2.10 Métodos de avaliação de classificadores

Uma vez construído um classificador de textos, deve-se mostrar sua eficiência através de metodologias de avaliação de desempenho que permitam que seus resultados sejam comparáveis com outros classificadores.

Nesta seção serão apresentados os principais métodos de avaliação de desempenho de classificadores.

2.10.1 Resubstituição

O método de resubstituição consiste em construir e testar o classificador no mesmo conjunto de documentos, ou seja, o conjunto de documentos de teste é exatamente igual ao conjunto de documentos de treinamento. Tal método fornece uma estimativa altamente otimista da eficácia do algoritmo. Porém, este método não garante que o bom desempenho no conjunto de treinamento se estenda para conjuntos independentes de teste. Desta forma, diversos métodos de reamostragem foram propostos, os quais são descritos a seguir. Todos estão baseados no mesmo princípio: não deve haver documentos em comum entre os conjuntos de treinamento, validação e de teste.

2.10.2 Holdout

O método holdout divide os documentos em uma porcentagem fixa de documentos p para treinamento e $(1 - p)$ para teste, considerando normalmente p

> 1/2. Valores típicos são $p = 2/3$ e $(1 - p) = 1/3$, embora não existam fundamentos teóricos sobre estes valores. Uma vez realizada a divisão, é induzido um classificador a partir do conjunto de treinamento e são calculadas as medidas de eficiência sobre os documentos de teste. A desvantagem desse método é que ele possui apenas uma iteração, dependendo muito da qualidade da partição escolhida.

2.10.3 Amostragem aleatória

Na amostragem aleatória, são criadas *K* partições do conjunto de todos os documentos. Cada partição é criada selecionando de forma aleatória e sem reposição um número fixo de documentos pra treinamento. São realizados *K* experimentos e a medida de eficiência é a média das medidas de eficiência obtidas em cada experimento. A amostragem aleatória pode produzir melhores estimativas de erro que o método holdout.

2.10.4 K-Fold Cross Validation

Nesse método, os documentos são aleatoriamente divididos em *K* partições mutuamente exclusivas ("folds") de tamanho aproximadamente igual a n/K, onde *n* é o tamanho do conjunto de documentos. Então, são realizados *K* experimentos, onde, em cada experimento, uma partição diferente é escolhida para o teste e as $K - 1$ partições restantes são escolhidas para o treinamento. A medida de eficiência é a média das medidas de eficiência calculadas para cada uma das partições.

A grande vantagem desse método comparado ao anterior, é que todos os documentos são usados tanto para treinamento quanto para teste.

2.10.5 Leave-One-Out

O método leave-one-out é um caso especial de cross-validation. Possui uma complexidade computacional elevada e, portanto, é mais usado em amostras pequenas. Para um conjunto de *n* documentos, um exemplo é usado para teste e *n*

– 1 documentos são usados para treinamento. Este processo é repetido *n* vezes, cada vez escolhendo um exemplo diferente para teste.

A medida de eficiência deste método é a média das medidas de eficiência dos *n* experimentos realizados.

2.10.6 Bootstrap

No método bootstrap, o conjunto de treinamento possui o mesmo tamanho do conjunto de todos os documentos e é constituído de documentos selecionados aleatoriamente com reposição a partir de tal conjunto.

Desta forma, para um mesmo conjunto de treinamento, alguns documentos podem não estar incluídos, enquanto outros podem aparecer mais de uma vez. Os documentos que não aparecem no conjunto de treinamento são usados como conjunto de teste.

Geralmente, o processo de bootstrap é repetido inúmeras vezes, sendo que a medida de eficiência estimada é a média das medidas de eficiência obtidas em cada experimento.