Aplicação de Modelos de Aprendizado Semissupervisionado na Classificação de Imagens de Sensoriamento Remoto

Rogério G. Negri [1]
Sidnei J. S. Sant'Anna
Luciano V. Dutra

Resumo: Nas mais diversas aplicações, a escassez de informação para o devido treinamento e utilização de métodos de Aprendizado de Máquina supervisionado é um problema persistente. Este fato motivou o desenvolvimento do paradigma de aprendizado semissupervisionado, que pode ser entendido como uma combinação de conceitos dos paradigmas supervisionado e não supervisionado. A maneira como o aprendizado é conduzido permite organizar os métodos semissupervisonados em diferentes modelos. Este trabalho apresenta um estudo comparativo entre diferentes modelos de aprendizado semissupervisionado. É também proposta uma versão semissupervisonada do método SVM, o qual alcançou melhor desempenho nas comparações realizadas.

Abstract: In many applications, the dearth of information to a proper training and use of supervised Machine Learning methods is a persistent problem. This fact led to the development of the semi-supervised learning paradigm. This paradigm can be understood as a combination of concepts of unsupervised and supervised paradigms. The way how the learning is conducted allows to organize the semi-supervised methods in different models. This paper presents a comparative study between different semi-supervised learning models. It is also proposed a semi-supervised version of SVM, which achieved better performance in the comparisons made.

1 Introdução

Ao longo dos anos, a humanidade tem-se confrontado com problemas econômicos, estratégicos, de segurança, tomada de decisão e predição de comportamentos. Em um contexto atual, estes problemas podem ser interpretados como o monitoramento do comportamento de clientes de um banco, classificação de páginas da internet de acordo com seu conteúdo, distinção de objetos em imagens, reconhecimento de fala e de escrita. A realização manual de tais tarefas, em alguns casos, torna-se inviável em função do tempo e do custo de mão de obra exigidos. Diante problemas como este, uma área da computação, denominada Reconhe-

[1]Divisão de Processamento de Imagens – Instituto Nacional de Pesquisas Espaciais (INPE)
Av. dos Astronautas, 1.758 – CEP: 12227-010 – São José dos Campos – SP – Brazil
{rogerio,sidnei,dutra}@dpi.inpe.br

cimento de Padrões, tem sido motivada quanto ao desenvolvimento de algoritmos capazes de auxiliar o homem na realização de determinadas tarefas [4].

Classificação de imagem é o nome dado à aplicação de métodos de Reconhecimento de Padrões em imagens com a finalidade de detectar os objetos que a compõe. Formalmente, a classificação de imagem consiste na estimação de uma função capaz de mapear um conjunto de padrões (i.e., os *pixels* de uma imagem) em um determinado conjunto de classes. A etapa de estimação da função de mapeamento caracteriza um processo de *aprendizagem*.

O tipo de aprendizagem é uma característica que permite distinguir os métodos de classificação de imagens. Dentre diferentes tipos existentes na literatura, os aprendizados supervisionado e não supervisionado são comumente utilizados. A principal característica que distingue estas formas de aprendizado é a maneira com são estimadas as funções para classificação dos padrões. Enquanto o aprendizado supervisionado realiza esta estimação baseado em informações fornecidas *a priori*, o aprendizado não supervisionado fundamenta-se em analogias construídas ao observar os padrões. As informações fornecidas *a priori* consistem em um conjunto de padrões "rotulados", isto é, padrões cuja classe é conhecida. Quando não existe classe associada ao padrão, o mesmo é dito "não rotulado". Um conjunto de padrões rotulados é denominado por *conjunto de treinamento*.

Quando as classes do problema são definidas de antemão, o aprendizado supervisionado é preferível [21], porém, ele é capaz de proporcionar bons resultados desde que sejam fornecidas informações suficientes para um aprendizado adequado [9]. Esta exigência se torna um ponto crítico quando existem custos relacionados à obtenção de amostras rotuladas [33]. Nestas circunstâncias, o problema de insuficiência de amostras para um treinamento adequado pode ser minimizado com o aumento do conjunto de treinamento, utilizando padrões não rotulados, os quais são abundantes na maioria dos problemas de classificação [12]. Partindo desta motivação surge um novo tipo de aprendizado, denominado *semissupervisionado*, o qual pode ser entendido como um "meio caminho" entre os aprendizados com e sem supervisão, uma vez que são utilizadas informações rotuladas e não rotuladas [11].

A maneira como os padrões não rotulados são explorados permite organizar o aprendizado semissupervisionado em diferentes modelos, como por exemplo: *gerativo*, que envolve a estimação de probabilidades, *separadores de baixa densidade*, baseado em superfícies de decisão, *co-treinamento*, que define um esquema de cooperação entre métodos para aprendizagem, e *indireto*, responsável por pré-processar e expandir o conjunto de dados de treinamento [1, 26].

Dessa forma, o objetivo deste trabalho consiste em apresentar um estudo comparativo entre diferentes modelos de aprendizado semissupervisionado aplicados no treinamento de métodos de classificação de imagens. Para isso foi realizado um experimento Monte Carlo sobre imagens simuladas visando à quantificação do desempenho dos modelos de aprendi-

zado semissupervisionado na classificação de tais imagens. O desempenho destes modelos foi comparado ao aprendizado supervisionado, verificando a existência de vantagens em determinados casos.

O presente artigo está organizado da seguinte forma: na Seção 2 são discutidos os principais conceitos dos modelos gerativo, separador de baixa densidade, co-treinamento e indireto. Uma proposta de Máquina de Vetores Suporte semissupervisionada é apresentada. Os modelos discutidos são utilizados no experimento conduzido na Seção 3. Conclusões sobre o estudo realizado são conduzidas na Seção 4.

2 Modelos de Aprendizado Semissupervisionado

Formalmente, um classificador é representado por uma função $f : \mathcal{X} \mapsto \Omega$, que associa elementos $\mathbf{x} \in \mathcal{X}$ uma determinada classe $\omega \in \Omega$. O conjunto $\mathcal{X}$ é denominado por *espaço de atributos*, que contém $\mathcal{D}$, conjunto de padrões que f classifica. Em classificação de imagens os padrões $\mathbf{x}$ representam os atributos dos *pixels* que compõem por sua vez uma dada imagem $\mathcal{D}$.

Para os classificadores de aprendizado supervisionado é requisito a existência do conjunto de padrões rotulados $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega;\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, c\}$, também denominado por *conjunto de treinamento*. As informações deste conjunto são utilizadas na estimação da função f, que em seguida é aplicada sobre os padrões de $\mathcal{D} \subset \mathcal{X}$ para determinação das respectivas classes. Após a rotulação de $\mathbf{x}_i$ a notação $(\mathbf{x}_i, \omega_j)$ especifica que o padrão $\mathbf{x}_i$ foi associado à classe ω_j. Ainda, $f(\mathcal{D})$ denota a classificação de todos elementos de $\mathcal{D}$ por f.

A estimação de um classificador f capaz de mapear adequadamente elementos de $\mathcal{D}$ às classes de Ω depende diretamente das informações contidas em $\mathcal{D}_l$. A insuficiência de informações em $\mathcal{D}_l$ conduz à estimação de uma f incapaz de realizar o mapeamento entre padrões e classes de forma satisfatória. Embora a solução se resuma em utilizar um conjunto de treinamento adequado, existem casos em que não é possível obter tal conjunto.

A limitação de informações para uma adequada estimação de f motiva o aprendizado semissupervisionado. Este tipo de aprendizado explora informações contidas em um dado conjunto de padrões não rotulados $\mathcal{D}_u = \{\mathbf{x}_i \in \mathcal{D} : i = m + 1, \ldots, m + n\}$, que de modo complementar às informações presente em $\mathcal{D}_l$, possibilite a estimação adequada de f.

Existem diferentes propostas para extração de informações de $\mathcal{D}_u$, onde cada uma delas define um *modelo de aprendizado semissupervisionado*. Gerativo, separador de baixa densidade, co-treinamento e indireto são alguns exemplos destes modelos, os quais são discutidos nas subseções seguintes.

2.1 Gerativo

O modelo gerativo tem objetivo de estimar as probabilidades condicionais $p(\omega_j|\mathbf{x})$ das classes que compõem o problema de classificação, denominadas *probabilidade a posteriori*. Quando estas probabilidades são conhecidas, o padrão $\mathbf{x}$ deve pertencer à classe ω_j cujo valor de $p(\omega_j|\mathbf{x})$ é máximo. No entanto, tais probabilidades são geralmente desconhecidas. Uma forma conveniente de estimar $p(\omega_j|\mathbf{x})$ é fazendo uso do Teorema de Bayes:

$$p(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)\, p(\omega_j)}{p(\mathbf{x})} \tag{1}$$

onde $p(\omega_j)$ é denominada *probabilidade a priori*, que especifica a proporção de cada classe $\omega_j \in \Omega$, $p(\mathbf{x}|\omega_j)$ é denominada *probabilidade classe-condicional*, que representa a distribuição estatística dos valores $\mathbf{x}$ com relação a classe ω_j e $p(\mathbf{x})$ representa a distribuição marginal de $p(\omega, \mathbf{x})$, isto é, $p(\mathbf{x}) = \sum_{j=1}^{c} p(\omega_j)\, p(\mathbf{x}|\omega_j)$.

A probabilidade marginal $p(\mathbf{x})$ é responsável pela modelagem de $\mathbf{x}$, independentemente da classe com que este padrão está associado, ao contrário da probabilidade $p(\mathbf{x}|\omega_j)$, que modela o comportamento de $\mathbf{x}$ segundo uma dada classe ω_j. No contexto do aprendizado semissupervisionado, a probabilidade marginal torna-se importante, uma vez que exerce influência no cálculo de $p(\omega_j|\mathbf{x})$ e é responsável pela modelagem dos dados não rotulados, considerados como fonte auxiliar de informação [18].

Uma vez que o conhecimento no modelo gerativo é representado por $p(\omega_j|\mathbf{x})$, o processo de aprendizagem consiste na modelagem desta probabilidade. A modelagem de funções que descrevem probabilidades é realizada a partir de um conjunto de parâmetros Θ. Logo, pode-se concluir que o processo de aprendizado nos modelos gerativos consiste em determinar os parâmetros de Θ responsáveis pela modelagem de $p(\omega_j|\mathbf{x})$. Por conveniência, as probabilidades $p(\omega_j)$, $p(\mathbf{x})$, $p(\omega_j|\mathbf{x})$ e $p(\mathbf{x}|\omega_j)$ passam a ser denotadas, respectivamente, por $p(\omega_j;\Theta)$, $p(\mathbf{x};\Theta)$, $p(\omega_j|\mathbf{x};\Theta)$ e $p(\mathbf{x}|\omega_j;\Theta)$, especificando assim o conjunto de parâmetros que as modelam.

No aprendizado supervisionado, os parâmetros de Θ são obtidos a partir de informações conhecidas (i.e., extraídas de $\mathcal{D}_l$). Com a disponibilidade de $\mathcal{D}_l$, os parâmetros $\hat{\Theta}$ ótimos que modelam $p(\mathbf{x}|\omega_j;\Theta)$ podem ser obtidos pelo estimador de máxima verossimilhança:

$$\begin{aligned}\hat{\Theta} = \Theta\arg\max\, p(\mathbf{x}, \omega_j;\Theta) = \Theta\arg\max \textstyle\prod_{i=i}^{m} p(\mathbf{x}_i, \omega_j;\Theta) \equiv \\ \equiv \Theta\arg\max \textstyle\sum_{i=i}^{m} \log p(\mathbf{x}_i;\Theta)\, p(\mathbf{x}_i|\omega_j;\Theta)\end{aligned} \tag{2}$$

uma vez que $p(\mathbf{x}_i, \omega_j;\Theta) = p(\mathbf{x}_i;\Theta)\, p(\mathbf{x}_i|\omega_j;\Theta)$. Além disso, $\hat{\Theta}$ e $\log\hat{\Theta}$ são funções com o mesmo máximo, logo, a utilização de $\log\hat{\Theta}$ torna-se preferível devido às facilidades em sua manipulação [34].

No desenvolvimento conduzido em (2) são utilizadas apenas informações de $\mathcal{D}_l$. Entretanto, informações não rotuladas (i.e., pertencentes a $\mathcal{D}_u$) devem ser consideradas quando o tipo de aprendizado é semissupervisionado. Assim, (2) é redefinida por:

$$\begin{aligned}\hat{\Theta} &= \Theta \arg\max \log\left(\prod_{i=1}^{m} p\left(\mathbf{x}_i, \omega_j; \Theta\right) \cdot \prod_{i=m+1}^{m+n} p\left(\mathbf{x}_i; \Theta\right)\right) \equiv \\ &\equiv \Theta \arg\max \left(\sum_{i=1}^{m} \log p\left(\mathbf{x}_i, \omega_j; \Theta\right) + \sum_{i=m+1}^{m+n} \log p\left(\mathbf{x}_i; \Theta\right)\right)\end{aligned} \tag{3}$$

Com a inclusão de dados não rotulados ao conjunto de treinamento, a estimação dos parâmetros de Θ por (3) perde a capacidade de resolução analítica, tornando necessário o uso de técnicas iterativas, como por exemplo, o algoritmo EM (*Expectation Maximization*), proposto em [13]. Este método considera que os dados são provenientes de um modelo de mistura, isto é, um conjunto de componentes independentes que totalizam um único modelo. Por questão de simplicidade, é comum admitir que as componentes do modelo de mistura são distribuídas de forma Gaussiana.

O Algoritmo 1 define detalhadamente os procedimentos da estimação dos parâmetros Θ de acordo com o método EM semissupervisionado (SemiEM), supondo que os dados são descritos por um modelo de mistura de Gaussianas. Neste algoritmo a convergência é alcançada quando $|\pi^{(t)} - \pi^{(t+1)}| \leq \epsilon$, sendo ϵ um parâmetro definido.

Este algoritmo gera $\hat{\Theta}$ como resultado, o qual é utilizado na classificação dos elementos de $\mathcal{D}$ a partir da seguinte regra de classificação:

$$(\mathbf{x}_i, \omega_j) \Leftrightarrow j = 1, \ldots, c \arg\max p(\omega_j | \mathbf{x}_i; \Theta);\ \forall \mathbf{x}_i \in \mathcal{D} \tag{4}$$

Supondo que $p(\omega_j|\mathbf{x}_i; \Theta)$ seja uma função densidade de probabilidade, cujos parâmetros são estimados a partir de $\mathcal{D}_l$, a mesma regra de classificação expressa em (4) define o método Classificação por Máxima Verossimilhança (*Maximum Likelihood Classification* - MLC). A classificação por Máxima Verossimilhança é uma da técnica utilizada em diferentes aplicações relacionadas à classificação de imagens devido sua simplicidade, baixo custo computacional e capacidade de gerar bons resultados [23].

2.2 Co-treinamento

O co-treinamento é um modelo de aprendizado fundamentado no conceito de cooperação. Nos métodos de classificação de imagens o aprendizado por co-treinamento acontece com a utilização de dois classificadores, f_1 e f_2, em cooperação mútua. Um conceito introduzido por este modelo é a *multi-visão*, o qual consiste em observar o problema de classificação em diferentes fragmentos. Nas diferentes visões, os exemplos de uma mesma classe devem possuir o mesmo rótulo. Além disso, as visões devem ser independentes, isto é, são capazes de induzir o aprendizado de um classificador sem depender de outra visão [24].

Algorithm 1 EM semissupervisionado para modelo de mistura de Gaussianas

Entrada: $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega;\ i = 1, 2, \ldots, m;\ j = 1, \ldots, c\}$
$\mathcal{D}_u = \{\mathbf{x}_i;\ i = m+1, m+2, \ldots, m+n\}$
$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \ldots, \mathbf{x}_{m+n}\}$
$\Omega = \{\omega_1, \ldots, \omega_c\}$
$t = 0$
Inicializar[1] $\Theta^{(t)} = \{\pi_j^{(t)}, \mu_j^{(t)}, \Sigma_j^{(t)}; j = 1, \ldots, c\}$ a partir de $\mathcal{D}_l$
Enquanto $\Theta^{(t)}$ não convergir:
$\forall \mathbf{x}_i \in \mathcal{D}_u \wedge \forall \omega_j \in \Omega$:

$$\gamma_{ij} = p\left(\omega_j | \mathbf{x}_i; \Theta^{(t)}\right) = \frac{\pi_j^{(t)} \mathcal{N}\left(\mathbf{x}_i; \mu_j^{(t)}, \Sigma_j^{(t)}\right)}{\sum_{k=1}^{c} \pi_k^{(t)} \mathcal{N}\left(\mathbf{x}_i; \mu_k^{(t)}, \Sigma_k^{(t)}\right)}$$

$\forall (\mathbf{x}_i, \omega_j) \in \mathcal{D}_l$:
Se $\mathbf{x}_i = \omega_j$ **então** $\gamma_{ij} = 1$,
Senão $\gamma_{ij} = 0$
Para $\forall \mathbf{x}_i \in \mathcal{D}$:
$t = t + 1$
$s_j = \sum_{i=1}^{m+n} \gamma_{ij}$
$\mu_j^{(t)} = \frac{1}{s_j} \sum_{i=1}^{m+n} \gamma_{ij} \mathbf{x}_i$
$\Sigma_j^{(t)} = \frac{1}{s_j} \sum_{i=1}^{m+n} \gamma_{ij} \left(\mathbf{x}_i - \mu_j^{(t)}\right) \left(\mathbf{x}_i - \mu_j^{(t)}\right)^T$
$\pi_j^{(t)} = \frac{s_j}{m+n}$
Saída: $\Theta^{(t)}$

No processo de aprendizado por co-treinamento, inicialmente os classificadores f_1 e f_2 são treinados a partir dos conjuntos $\mathcal{D}_{l1}$ e $\mathcal{D}_{l2}$, que contém as mesmas informações. Em seguida estes classificadores são aplicados na classificação de visões distintas, isto é, dois conjuntos de padrões não rotulados extraídos do problema de classificação, denotados por $\mathcal{D}_{u1}$ e $\mathcal{D}_{u2}$. Os padrões classificados por f_1 com *alto grau de confiança* são integrados ao conjunto de treinamento de f_2 e *vice-versa*.

O processo de aprendizado descrito é realizado iterativamente enquanto houver padrões classificados de forma confiável, isto é, acima de determinado nível de confiança, em ambas as visões, permitindo assim a troca de padrões rotulados para o treinamento dos classificadores. Ao fim do aprendizado, os conjuntos de treinamento obtidos em cada visão são fundidos e utilizados no treinamento de um novo classificador, aplicado na classificação de todo o conjunto de padrões. O Algoritmo 2 descreve o processo de aprendizado por co-

[1] π_j representa a proporção de padrões na classe j. $\mathcal{N}(\mathbf{x}_i; \mu_j, \Sigma_j)$ representa a probabilidade de $\mathbf{x}_i$ segundo uma distribuição Gaussiana Multivariada da classe j, com vetor média μ_j e matriz de covariância Σ_j

treinamento. Este algoritmo apresenta de forma genérica o aprendizado por co-treinamento, independente do método de classificação. Neste estudo, este modelo será aplicado no aprendizado do método MLC, cujo resultado desta junção será denominada CoMLC. A interpretação do conceito de classificação com "alto grau de confiança" presente neste modelo está vinculada ao método de classificação adotado. No método CoMLC o nível de confiança está associado a probabilidade com que os padrões são classificados. Isso torna necessária a definição de um parâmetro que atue como um limiar mínimo, cujos padrões classificados com valores de probabilidade acima deste limiar são caracterizados como de "alta confiança".

Algorithm 2 Co-treinamento

Entrada: $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega; i = 1, 2, \ldots, m;\ j = 1, \ldots, c\}$
$\mathcal{D}_u = \{\mathbf{x}_i; i = m+1, \mathbf{x}_i; i = m+2, \ldots, m+n\}$
Inicializar: $\mathcal{D}_{l1} = \mathcal{D}_l$ e $\mathcal{D}_{l2} = \mathcal{D}_l$, conjuntos de padrões rotulados para as *visões* 1 e 2
$\mathcal{D}_{u1} \bigcup \mathcal{D}_{u2} = \mathcal{D}_u$, com $\mathcal{D}_{u1} \bigcap \mathcal{D}_{u2} = \emptyset$
Repita
Treine f_1 com $\mathcal{D}_{l1}$ e f_2 com $\mathcal{D}_{l2}$
Classifique $\mathcal{D}_{u1}$ com f_1 e $\mathcal{D}_{u2}$ com f_2
Se $\nexists \mathbf{x}_{i1} \in \mathcal{D}_{u1}$ e $\nexists \mathbf{x}_{i2} \in \mathcal{D}_{u2}$ classificados com *alta confiança* **então** FIM
Senão:
Para $\forall \mathbf{x}_{i1} \in \mathcal{D}_{u1}$ e $\forall \mathbf{x}_{i2} \in \mathcal{D}_{u2}$ classificados com *alta confiança*:
$\mathcal{D}_{l1} = \mathcal{D}_{l1} \bigcup \{(\mathbf{x}_{i2}, \omega_{j2})\}$
$\mathcal{D}_{l2} = \mathcal{D}_{l2} \bigcup \{(\mathbf{x}_{i1}, \omega_{i1})\}$
Remova $\mathbf{x}_{i1}$ e $\mathbf{x}_{j2}$ de $\mathcal{D}_{u1}$ e $\mathcal{D}_{u2}$, respectivamente
$\mathcal{D}_l = \mathcal{D}_{l1} \bigcup \mathcal{D}_{l2}$
Treine f_1 (ou f_2) com $\mathcal{D}_l$
Classifique $\mathcal{D} = \mathcal{D}_l \bigcup \mathcal{D}_u$ com f_1 (ou f_2)
Saída: $f_1(\mathcal{D})$ (ou $f_2(\mathcal{D})$)

2.3 Separadores de baixa densidade

Separadores de baixa densidade consistem nos métodos cuja regra de classificação é determinada por superfícies de separação definidas em regiões do espaço de atributos onde a concentração de padrões é baixa. Dentre diferentes métodos, o Perceptron de Múltiplas Camadas [19], o Discriminantes de Fisher [30] e a Máquina de Vetores Suporte [31] são alguns exemplos de separadores de baixa densidade.

Introduzido por Vladmir Vapnik, a Máquina de Vetores Suporte (*Support Vector Machine* - SVM), é um recente método de Reconhecimento de Padrões que tem recebido grande atenção nas pesquisas devido sua sólida fundamentação teórica e determinadas características atrativas, como a independência de modelos de distribuição estatística, arquitetura simples,

complexidade computacional moderada, excelente capacidade de generalização e maior robustez diante ao fenômeno de Hughes [6]. O fenômeno de Hughes refere-se a necessidade de uma quantidade cada vez maior de padrões de treinamento, que a maioria dos métodos de Reconhecimento de Padrões apresentam, a medida que a dimensão do espaço de atributos aumenta. Segundo [25], o método SVM é menos sensível a este fenômeno uma vez que sua formulação não envolve a estimativa de densidades, mas sim a busca de uma superfície de decisão entre classe baseado no comportamento geométrico dos padrões no espaço de atributos.

Dado um conjunto de treinamento $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega; i = 1, 2, \ldots, m;\ j = 1, 2\}$, o método SVM determina a separação entre as classes ω_1 e ω_2 a partir do seguinte hiperplano de separação ótimo:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0 \tag{5}$$

sendo $\mathbf{w}^T$ o transposto do vetor ortogonal ao hiperplano de separação e b um escalar real tal que $\frac{|b|}{\|\mathbf{w}\|}$ representa a distância do hiperplano à origem do espaço de atributos.

Determinar os parâmetros de (5) equivale a resolver o seguinte problema de otimização [30]:

$$\begin{array}{l} \lambda\max\ L_D = \sum_{i=1}^{m} \lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i\lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{sujeito a: } \left\{ \begin{array}{l} 0 \leq \lambda_i \leq C, i = 1, \ldots, m \\ \sum_{i=1}^{m} \lambda_i y_i = 0 \end{array} \right. \end{array} \tag{6}$$

onde y_i, denominado *indicador de classe*, assume valor +1 se $\mathbf{x}_i \in \omega_1$ ou -1 se $\mathbf{x}_i \in \omega_2$, C é um parâmetro de regularização introduzido para ajuste do hiperplano, λ_i são *Multiplicadores de Lagrange* e $K(\mathbf{x}_i, \mathbf{x}_j)$ é uma função simétrica que atende as condições de Mercer, denominada *kernel*. Esta função realiza um mapeamento implícito dos padrões para outro espaço de atributos onde a separabilidade é maior. A função $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}}$, com $\sigma \in \mathbb{R}$, é denominada *Kernel RBF*, empregada em diversas aplicações.

Os parâmetros $\mathbf{w}$ e b que fornecem o hiperplano ótimo (5) são obtidos a partir dos *Multiplicadores de Lagrange* (λ_i) resultantes de (6) através das relações:

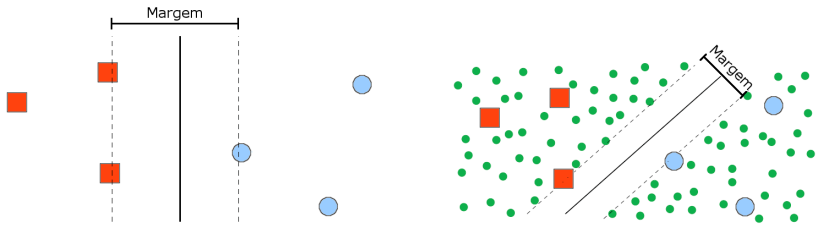
$$\mathbf{w} = \sum_{\lambda_i \neq 0} \lambda_i y_i \mathbf{x}_i; \tag{7}$$

$$b = \frac{1 - y_i[\mathbf{w}\mathbf{x}_i]}{\lambda_i};\ \forall \lambda_i \neq 0 \tag{8}$$

Ao observar (7) e (8), nota-se que o hiperplano (5) é definido em função de padrões (i.e., vetores) de treinamento cujos *Multiplicadores de Lagrange* são não nulos. Estes padrões são denominados *vetores suporte*.

A formulação do método SVM permite apenas a separação entre duas classes, o que não atende a maioria dos problemas reais. Para contornar esta limitação são empregadas Estratégias Multiclasse. De acordo com [22] tais estratégias podem consistir na decomposição do problema multiclasse em subproblemas binários ou até mesmo na reformulação do método SVM. Alguns exemplos de Estratégia Multiclasse são abordados em [29, 32, 30]. Um-Contra-Todos (*One-Against-All* - OAA) é uma das estratégias baseada em decomposições binárias mais adotadas devido sua simplicidade. Em um problema de separação dos dados em c classes, são definidos c classificadores binários, onde cada um é responsável pela separação de uma classe específica com relação as demais.

Partindo da formulação original do SVM, a transformação deste método para o aprendizado semissupervisionado consiste em determinar hiperplanos de separação em regiões do espaço de atributos onde a densidade de padrões é menor. A Figura 1 ilustra a construção de hiperplanos em regiões de baixa densidade, após a inclusão de padrões não rotulados.



(a) Margem de separação pelo SVM original.

(b) Margem de separação pelo SVM semissupervisionado.

Figura 1. Motivação do método SVM semissupervisionado: Definir o hiperplano de separação em regiões de baixa densidade. FONTE: Adaptado de [34].

Existem diferentes propostas que tornam semissupervisionado o aprendizado do método SVM. Em [6] é proposto uma versão que consiste na reformulação do problema de otimização (6), tornando-o capaz de tratar dados não rotulados.

Neste trabalho é proposta uma nova versão semissupervisionada de SVM, denominado por SemiSVM. Esta proposta é baseada no fato que o hiperplano de separação ótimo é definido pelo conjunto de vetores suporte. Assim, quando selecionados os padrões não rotulados próximos a um dado hiperplano inicial, os mesmos tendem a ser utilizados como vetores suporte na definição de um novo hiperplano, ajustado à região de baixa densidade. Um ajuste mais preciso é alcançado adotando um processo iterativo, onde novos hiperplanos são obtidos em função da seleção de até k padrões não rotulados, classificados pelo hiperplano atual. A seleção dos k padrões não rotulados torna-se mais criteriosa ao longo das

iterações em função de um incremento constante (α) ao parâmetro C. O processo de ajuste a região de baixa densidade é encerrado quando C atinge um valor máximo (L_C) definido inicialmente. O Algoritmo 3 descreve o método SemiSVM.

Algorithm 3 SemiSVM: Nova proposta de SVM semissupervisionado

Entrada: $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega; i = 1, 2, \ldots, m;\ j = 1, \ldots, c\}$
$\mathcal{D}_u = \{\mathbf{x}_i; i = m+1, m+2, \ldots, m+n\}$
Definir: k, máximo de candidados selecionados por iteração
C, penalidade inicial
α, incremento de penalidade
L_C, limite superior de penalidade
Enquanto $C \leq L_C$:
Para $j = 1, \ldots, c$:
Treine[2] f_j com $\mathcal{D}_l$
Classifique $\mathcal{D}_u$ com f_j
Selecione k padrões de $\mathcal{D}_u$ tal que $|f_j(\mathbf{x}) - \epsilon| \leq 1$ seja mínimo
$\mathcal{D}_l = \mathcal{D}_l \bigcup \{(\mathbf{x}_i, \omega_j) : \min 0 \leq |f_j(\mathbf{x})|;\ i = 1, \ldots, k\}$
$C = C + \alpha$
Treine[3] f com $\mathcal{D}_l$ e classifique $\mathcal{D} = \mathcal{D}_l \bigcup \mathcal{D}_u$
Saída: $f(\mathcal{D})$

2.3.1 Considerações sobre o método SemiSVM Existem diferentes propostas na literatura que tornam semissupervisionado o aprendizado do método SVM. Dentre elas podem ser citadas [31], [20], [5], [17] e [16].

Em [20] é apresentada uma versão de SVM com aprendizado transdutivo (*Transductive Support Vector Machine* - TSVM), originalmente formalizada em [31]. A idéia geral do TSVM consiste em treinar o classificador a partir de dois conjuntos de dados, um de treinamento e outro de predição, compostos respectivamente por dados rotulados e não rotulados. Com a classificação do conjunto de predição, novos dados rotulados tornam-se disponíveis para o treinamento do método SVM, podendo proporcionar melhor desempenho na classificação. As principais diferenças entre o TSVM e o SVM consiste em modificações no problema de otimização (6), na expressão do hiperplano (5) e por ser um método iterativo.

Posteriormente [5] propõe um melhoramento ao método TSVM para aplicações em classificação de imagens de sensoriamento remoto. Nesta nova versão transdutiva são propostas modificações no processo de seleção das amostras não rotuladas e ajustes iterativos na

[2] f_j representa um hiperplano de separação responsável por separar padrões da classe j com relação as demais classes. Este processo de separação é baseado na estratégia multiclasse OAA.
[3] Nesta etapa f representa uma SVM multiclasse.

penalidade, proporcionando melhor estabilidade no treinamento com poucas amostras rotuladas. Ainda, é proposta uma metodologia para lidar com o problema multiclasse baseada na estratégia OAA. A Máquina de Vetores Suporte Laplaciana (*Laplacian Support Vector Machine* - LapSVM) [17] é outra versão semissupervisionada de SVM que incorpora conceitos de classificação baseada em grafo. Para isso é definido um novo problema de otimização, o qual é composto pela combinação dos problemas de definir hiperplanos de margem máxima e regularizar um grafo construído a partir da informação de padrões rotulados e não rotulados.

Em [16] o aprendizado semissupervisionado em SVM é induzido com o uso de Matriz de *Kernel*[5]. Esta Matriz de *Kernel* é construída a partir de padrões provenientes de dois agrupamentos distintos, definidos pelo algoritmo EM. Após este processo, a Matriz de *Kernel* é utilizada pelo SVM.

Em comparação às propostas supracitadas, o método SemiSVM não faz uso de funções *Kernel* ou grafos para conduzir o aprendizado semissupervisionado. Além disso, não é proposto como em [5] um procedimento para ajuste iterativo da penalidade (C) durante o processo de aprendizado, mas sim um aumento linear gradativo desse parâmetro. Cabe também ressaltar que a partir de simples modificações na etapa de seleção de padrões não rotulados, o método SemiSVM pode fazer uso de outras estratégias multiclasse baseadas em decomposições binárias, como por exemplo, as estratégias Um-Contra-Um (*One-Against-One* - OAO) [32] e Grafos Acíclicos Diretos (*Direct Acyclic Graph* - DAG) [28]. Por fim, no método proposto não são realizadas modificações no problema de otimização (6), possibilitando o uso de heurísticas amplamente utilizadas para sua solução, como por exemplo, LibSVM [10], SMO (*Sequential Minimal Optimization*) [27] e SVM^{Light} [20].

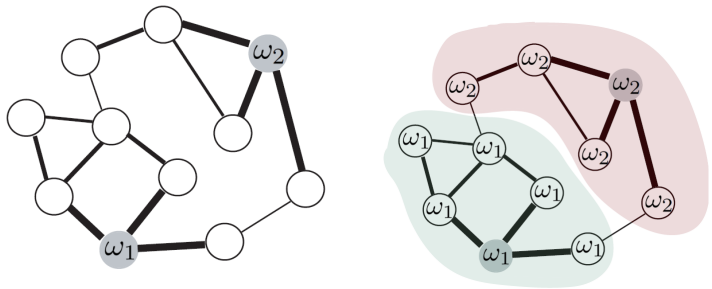
2.4 Baseado em grafo

Outro modelo de aprendizado semissupervisionado refere-se ao aprendizado baseado em grafo. Este modelo consiste inicialmente na construção de uma matriz de afinidade G, a qual é uma representação numérica de um grafo. Nesta matriz são representadas as similaridades entre padrões, sejam rotulados ou não. Formalmente, $\mathbf{x}_r, \mathbf{x}_s \in \mathcal{D}_l \bigcup \mathcal{D}_u$ representam dois vértices no grafo, cujo valor (peso) associado a aresta entre tais vértices corresponde a medida de similaridade g_{rs}, a qual é um elemento de G. Os padrões $\mathbf{x}_r$ e $\mathbf{x}_s$ tendem a estar associados a mesma classe a medida que o valor de g_{rs} aumenta.

Segundo [34], os métodos baseados em grafo são fundamentados na hipótese de "suavidade dos rótulos", ou seja, a mudança da classe associada ao padrões, nesse caso os vértices, varia de forma suave sobre o grafo. O processo de associação de uma classe a um vértice (padrão) não rotulado depende da similaridade apresentada com relação aos demais

[5]uma representação para funções *Kernel*, porém na forma matricial, onde inicialmente são computados os valores da função *Kernel* para possíveis pares de padrões entrada

vértices do grafo. A Figura 2 reproduz um exemplo de classificação baseada em grafo, onde inicialmente apenas dois vértices são rotulados e as medidas de afinidade são conhecida, representada nesta ilustração pela grossura das arestas (Figura 2(a)). A classificação dos vértices não rotulados acontece em função da maior similaridade, e não apenas no caminho de menor distância, como mostra a Figura 2(b). Cabe observar na Figura 2 que os métodos baseados em grafo não retornam uma função de decisão, utilizada para determinar a classe dos padrões não rotulados, mas sim um mapeamento entre os vértices e as classes definidas no problema.



(a) O grafo, com poucos vértices rotulados, e as similaridades entre vértices.

(b) Classificação dos vértices em função dos vértices rotulados e da similaridade entre vértices

Figura 2. Classificação baseada em grafo: Os padrões são os vértices do grafo e a classe é determinada pela similaridade entre vértices. FONE: Adaptado de [8].

Dentre diferentes propostas apresentadas na literatura, em [8] é apresentado um método semissupervisionada baseado em grafo que faz uso de função *Kernel*. Para isso, seja $\mathcal{D} = \mathcal{D}_l \bigcup \mathcal{D}_u$ um conjunto composto por m padrões rotulados e n não rotulados, a matriz de afinidade G é determinada por:

$$G_{(m+n)\times(m+n)} \; : \; g_{rs} = e^{\frac{\|\mathbf{x}_r - \mathbf{x}_s\|^2}{\sigma^2}}; \;\; r, s = 1, \ldots, m+n \tag{9}$$

A medida de similaridade entre $\mathbf{x}_r$ e $\mathbf{x}_s$, representada pelo elemento g_{rs} de G, é determinada pela aplicação da função *Kernel* RBF, previamente apresentada na Subseção 2.3. Em seguida, sobre os valores de similaridade de G é aplicada a seguinte normalização simétrica:

$$S = \sqrt{Q^{-1}} G \sqrt{Q^{-1}} \tag{10}$$

onde Q é uma matriz diagonal, denominada Matriz Grau, tal que $q_{rr} = \sum_r g_{rs}$.

Com relação aos padrões de $\mathcal{D}$ é determinada a matriz de rótulos Y, definida por:

$$Y_{(m+n)\times c} \; : \; y_{rj} = \begin{cases} 1 \text{ se } (\mathbf{x}_r, \omega_j) \\ 0 \text{ caso contrário} \end{cases} \qquad r = 1, \ldots, m+n; \;\; j = 1, \ldots, c \tag{11}$$

Cabe ressaltar que Y possui o número de linhas equivalente ao número de padrões envolvidos no problema de classificação, enquanto o número de colunas refere-se a quantidade de classes do problema. Observa-se que as linhas de Y referentes aos padrões não rotulados são nulas.

Por fim, a partir das matrizes S e Y é determinada a matriz U:

$$U = (I - \alpha S)^{-1} Y \tag{12}$$

sendo I é a matriz identidade e $\alpha \in (0, 1)$ é um parâmetro de regularização.

Para classificação dos padrões não rotulados é utilizada a seguinte regra:

$$(\mathbf{x}_i, \omega_j) \Leftrightarrow j = 1, \ldots, c \arg\max (u_{ij}) \tag{13}$$

onde u_{ij} é elemento de $U_{(m+n)\times c}$. O Algoritmo 4 descreve o método de classificação baseado em grafo proposto em [8], referenciado neste trabalho por GB.

Algorithm 4 Classificação semissupervisionada baseada em grafo

Entrada: $\mathcal{D}_l = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega; \; i = 1, 2, \ldots, m; \; j = 1, \ldots, c\}$
$\mathcal{D}_u = \{\mathbf{x}_i; \; i = m+1, m+2, \ldots, m+n\}$
$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \ldots, \mathbf{x}_{m+n}\}$
$\Omega = \{\omega_1, \ldots, \omega_c\}$
Inicializar[4] $G = Q = O_{(m+n)\times(m+n)}$
$Y = U = O_{(m+n)\times c}$
Para $\forall \mathbf{x}_r, \mathbf{x}_s \in \mathcal{D}$ **faça**:
$g_{rs} = e^{-\frac{\|\mathbf{x}_r - \mathbf{x}_s\|^2}{\sigma^2}}$
Para $\forall \mathbf{x}_r \in \mathcal{D}$ **faça**:
$d_{rr} = \sum_{s=1}^{m+n} g_{rs}$
$S = \sqrt{Q^{-1}} G \sqrt{Q^{-1}}$
Para $\forall \mathbf{x}_r \in \mathcal{D}$ **faça**:
Se $(\mathbf{x}_r, \omega_j)$ **então** $y_{rj} = 1$
$U = (I - \alpha S)^{-1} Y$
Saída: U

2.5 Modelos indiretos

Nas Seções 2.1, 2.2, 2.3 e 2.4 foram apresentados modelos de aprendizado semissupervisionado *diretos*. Nestes modelos os padrões rotulados e não rotulados são utilizados simultaneamente durante o processo de aprendizado.

Outra classe de modelos de aprendizado semissupervisionado é denominada *indireto*. De acordo com [26], os modelos de aprendizado indireto são divididos em duas etapas. O objetivo da primeira etapa consiste em melhorar (aumentar) a informação contida no conjunto de padrões rotulados a partir da rotulação e seleção de padrões inicialmente não rotulados. Em seguida, o conjunto de padrões derivado da primeira etapa é utilizado no aprendizado de um método supervisionado ou semissupervisionado direto.

Segundo exposto, os modelos indiretos visam aumentar o número de exemplos rotulados para indução de classificadores melhores. O aumento do número de exemplos rotulados pode ser conduzido por métodos de agrupamentos inicializáveis, como por exemplo, SEEDED-k-Médias, CONSTRAINED-k-Médias [2] e C-Médias Nebuloso semissupervisionado (*semi-supervised Fuzzy C-Means* - ssFCM) [3].

Ao definir as classes do problema e inicializá-las, ao fim de um processo de agrupamento é possível avaliar o *grau de pertinência* de cada padrão a seu respectivo agrupamento. É razoável a utilização de amostras rotuladas com alto grau de pertinência à classe. Nestas condições, o método ssFCM torna-se conveniente, uma vez que seu resultado expressa probabilidades sobre a pertinência de cada padrão com relação às diferentes classes. Ainda, o conceito de *grau de pertinência* possui a mesma interpretação que *grau de confiança*, introduzido pelo aprendizado por co-treinamento (Subseção 2.2). Isso faz necessário a adoção de um parâmetro que define um limiar para caracterizar classificações com alto grau de pertinência. O modelo indireto será empregado no treinamento do método MLC, cuja associação é denominada por IndMLC.

3 Experimentos e Resultados

Com objetivo de comparar os modelos de aprendizado semissupervisionado discutidos na seção anterior, foi conduzido um experimento Monte Carlo envolvendo a classificação de imagens simuladas. O uso de imagens simuladas permite a realização de experimentos controlados, isto é, experimentos cujo comportamento dos dados e os resultados esperados são conhecidos *a priori*, além de permitir a realização de avaliações sem a influência de uma imagem particular.

Na Subseção 3.1 são descritos os procedimentos realizados para geração do conjunto

[4] $O_{o_1 \times o_2}$ representa uma matriz nula de ordem $o_1 \times o_2$ empregada na inicialização das matrizes G, E, Y e U.

de imagens simuladas, em seguida, a Subseção 3.2 descreve as configurações do experimento, cujos resultados são discutidos na Subseção 3.3

3.1 Simulação de imagens

Nos estudos envolvendo a avaliação de métodos de classificação, imagens simuladas permitem a construção de experimentos controlados. A definição de funções capazes de simular imagens com diferentes canais (bandas), compostas por alvos com variados níveis de separabilidade espectral, pode ser não trivial. Uma maneira conveniente de realizar este tipo de tarefa é considerar a distribuição estatística de alvos observados em imagens reais obtidas por satélites de sensoriamento remoto, como discutido em [14] e [15].

Nos sensores remotos ópticos, a luz refletida pelos alvos é geralmente quantificada em diferentes intervalos espectrais. A informação adquirida nestes intervalos determinam as bandas destas imagens. Nestas condições, um mesmo alvo pode apresentar comportamentos distintos em cada banda, com uma dada estrutura de correlação destas bandas entre si. Ainda, devido a interferências existentes no processo de aquisição e processamento destas imagens, os *pixels* de um mesmo alvo geralmente apresentam variações distribuídas de forma Gaussiana. Partindo destas considerações, a simulação espectral de alvos é dada pela seguinte expressão:

$$\widetilde{\mathbf{p}}_j = \left(R_j^{-1} \nu L_j\right) \psi_j + (\mu_j \zeta_j) \tag{14}$$

onde μ_j refere-se ao vetor média estimado a partir de amostras de um alvo específico, indexado por $j = 1, 2, \ldots, c$. Considerando Σ_j a matriz de covariância, também estimada a partir de amostras de um alvo específico, R_j^{-1} é o inverso da matriz de autovetores de Σ_j, L_j é a matriz diagonal composta pela raiz quadrada dos autovalores de Σ_j, ν é um vetor aleatório, de mesma dimensão de μ_j, gerado por uma distribuição Gaussiana Multivariada padrão, ψ_j e ζ_j são escalares gerados por uma distribuição Uniforme entre $0,75$ e $1,25$, e por fim, $\widetilde{\mathbf{p}}_j$ representa a simulação de um único *pixel* referente ao alvo j. Cabe ressaltar que as colunas da matriz R_j estão organizadas de acordo com a ordem crescente dos respectivos autovalores, assim como os elementos da diagonal de L_j, também dispostos em ordem crescente.

Em (14), μ_j é o vetor que determina o valor médio, em cada banda espectral, dos *pixels* que serão simulados baseado nas características do j-ésimo alvo observado na imagem real. A matriz R_j representa o comportamento da estrutura de correlação entre bandas, enquanto a informação sobre a variância em cada banda é dada por L_j. Tais matrizes também são determinadas com base nas características do j-ésimo alvo observado, uma vez que são derivadas de Σ_j. Nestas condições, nota-se que a diferença entre dois *pixels* simulados é dado em função de ν, ψ_j e ζ_j. O vetor ν é introduzido para simular as variações Gaussianas entre os *pixels* que compõe um mesmo alvo. Por outro lado, ψ_j e ζ_j são empregados, respectivamente, para produzir variações na estrutura de covariância (i.e. $\left(R_j^{-1} \nu L_j\right)$) e na média original do j-ésimo alvo.

Cabe ressaltar que para geração dos *pixels* que simularam o comportamento do j-ésimo alvo, de uma dada imagem simulada, foram associados aleatoriamente valores aos escalares ψ_j e ζ_j. Em outra simulação, os valores associados a estes escalares, também para geração dos *pixels* do j-ésimo alvo, podem diferir daqueles adotados na simulação da imagem anterior. Esta característica do processo de simulação faz com que uma mesma classe, observada em simulações distintas, apresente variações entre si, proporcionando por sua vez imagens simuladas com particularidades distintas.

Para simulação dos alvos foi empregada uma imagem do satélite LANDSAT-5 TM, considerando as quatro de bandas que abrangem o intervalo espectral do visível e infravermelho próximo. Esta imagem foi adquirida em 26 de setembro de 2010, e refere-se a uma região localizada ao longo da rodovia BR-163, próxima à Floresta Nacional do Tapajós, no Estado do Pará. A seleção desta região é justificada pelo conhecimento de alvos (tipos de cobertura da terra) existentes no local, devido a um levantamento de campo conduzido no mesmo período da aquisição da imagem. Os tipos de alvos são: *floresta primária*, *pasto limpo*, *regeneração antiga*, *agricultura*, *regeneração nova* e *pasto sujo*. A Figura 3 ilustra a imagem LANDSAT-5 TM adotada e as respectivas amostras de alvos (tipos de cobertura da terra).

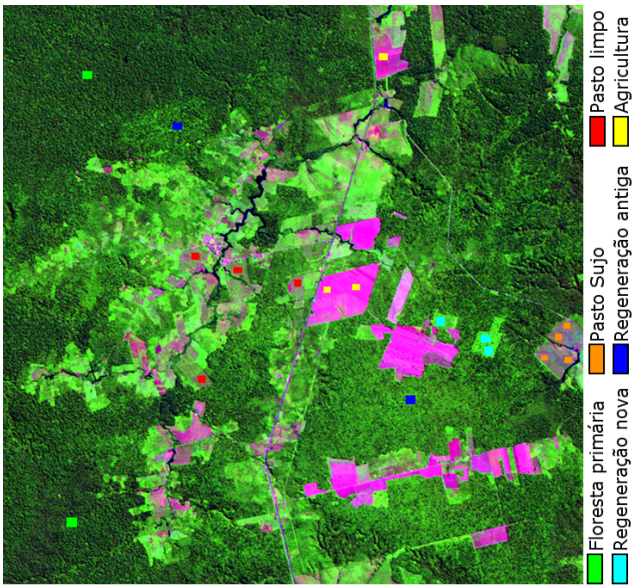
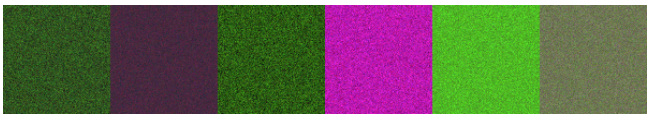


Figura 3. Imagem LANDSAT-5 TM em composição colorida e amostras de diferente tipos cobertura da terra.

Utilizando as amostras de cobertura da terra identificadas na Figura 3, foram calculadas os respectivos vetores média e matriz de covariância para simulação das imagens a partir de (14). A Figura 4 ilustra dois exemplos de imagens simuladas segundo a metodologia apresentada. Estas simulações são obtidas pela concatenação de regiões de 100×100 *pixels*, onde cada uma dessas regiões refere-se a uma classe específica. Ao comparar a mesma classe nas simulações apresentadas nas Figuras 4(a) e 4(b) é possível observar as variações introduzidas pelos escalares ψ_j e ζ_j.

(a)



(b)

Figura 4. Imagems simuladas segundo a metodologia apresentada baseada nas amostra identificadas na Figura 3. Ordenados da esquerda para a direita são simulados alvos referentes às classes *floresta primária*, *pasto limpo*, *regeneração antiga*, *agricultura*, *regeneração nova* e *pasto sujo*. A composição colorida é a mesma adotada na Figura 3

De acordo com [7], nos estudos envolvendo experimentos com dados simulados, é necessário realizar milhares de repetições, visando a obtenção de resultados com níveis de qualidade aceitáveis. Sendo assim, para este estudo foi gerado um conjunto de 1000 imagens simuladas, o qual encontra-se disponível em `http://www.rgnegri.com/data/SSL`.

3.2 Configurações do experimento

Como discutido, o emprego do aprendizado semissupervisionado é motivado em situações onde a quantidade de amostras de treinamento é escassa. Neste contexto, uma das características do experimento foi realizar classificações a partir de conjuntos de treinamento relativamente pequenos. Assim, foram definidos conjuntos compostos por 10, 15, 25 e 40 *pixels* em cada classe, selecionados aleatoriamente sobre as respectivas regiões. Dentre os *pixels* não selecionados para compor o conjunto de treinamento, são escolhidos aleatoriamente 5% para determinar o conjunto de padrões não rotulados.

A partir das 1000 imagens simuladas e dos diferentes conjuntos de treinamento, serão avaliados o desempenho de métodos SemiEM, CoMLC, SemiSVM, TSVM, GB e IndMLC.

Tabela 1. Parametros definidos para o experimento.

Método	**Parâmetros**
SemiEM	Convergência (ϵ): 0,005
CoMLC	Limiar de confiança: 0,98
SemiSVM	Penalidades inicial/final/incremento (C/L_C/α): 100/200/50 Num. máx. padrões selecionados por iteraçao (por hiperplano)(k):5 σ (*Kernel* RBF): 2,0
TSVM	Penalidade (C): 200 σ (*Kernel* RBF): 2,0
GB	Regularizaçao (α): 0,5 σ (*Kernel* RBF): 2,0
IndMLC	Limiar de confiança: 0,98
ML	Não possui parâmetros
SVM	Penalidade (C): 200 σ (*Kernel* RBF): 2,0

Além destes métodos semissupervisionados, serão observados os desempenhos dos métodos MLC e SVM, tomando-os como referência na discussão sobre o potencial do aprendizado semissupervisionado na classificação de imagens. A Tabela 1 apresenta os parâmetros necessários aos métodos analisados. Tais parâmetros foram ajustados manualmente a partir de experimentos preliminares.

Para avaliação dos resultados, o índice de acerto global é adotado na mensuração da qualidade das classificações. Este índice informa a porcentagem de *pixels* classificados corretamente.

Na execução dos experimentos foi utilizado um computador com processador Intel *Core* $i5$, 4 GB de memória RAM e sistema operacional Linux/Ubuntu versão 10.2. As implementações realizadas utilizaram linguagem de programação IDL (*Iteractive Data Language*) versão 7.1, com exceção do método empregado na otimização do problema (6) e para aplicação do método TSVM (SVM^{Light}, [20]).

3.3 Resultados

Os resultados do experimento conduzido são apresentados no gráfico da Figura 5. Neste gráfico é quantificado o desempenho médio obtido com a classificação das 1000 imagens simuladas, segundo os conjuntos de treinamento de diferentes dimensões. Uma estima-

tiva média sobre o tempo computacional despendido na classificação de uma imagem, por cada um dos diferentes métodos analisados, é apresentada na Tabela 2.

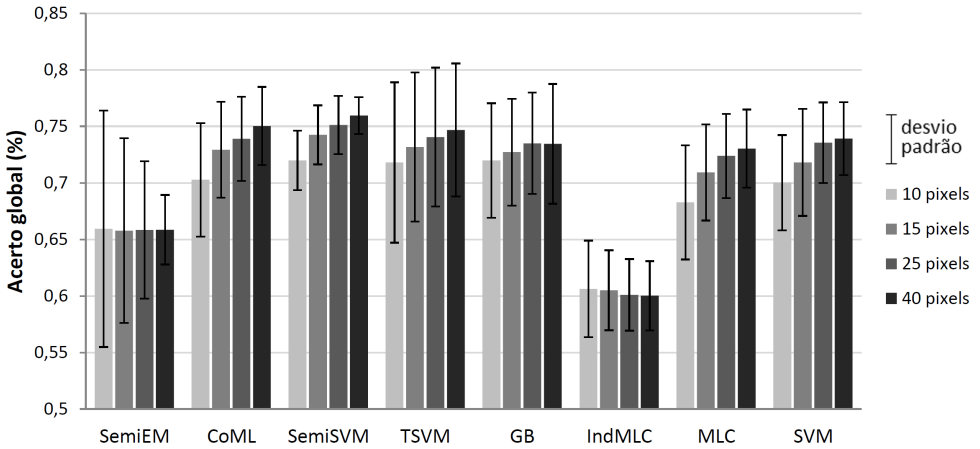


Figura 5. Desempenho dos diferentes métodos analisados.

Tabela 2. Tempo de computacional médio, em segundos, exigido pelo métodos analisados.

Pixels	**SemiEM**	**CoML**	**SemiSVM**	**TSVM**	**GB**	**IndMLC**	**MLC**	**SVM**
10	103,41	36,24	12,12	34,45	70,19	16,51	4,01	5,40
15	101,89	36,05	12,20	47,63	100,03	15,88	3,99	5,42
25	100,29	36,11	12,32	49,16	302,48	15,74	3,99	5,44
40	98,73	36,43	12,48	74,46	590,30	15,46	4,03	5,50

O método IndMLC, de aprendizado semissupervisionado indireto, obteve os mais baixos níveis de acurácia comparado aos demais métodos. Independente da dimensão do conjunto de treinamento, os índices médios de acurácia obtidos, assim como o desvio padrão destes índices, foram semelhantes. Verifica-se também que o modelo indireto prejudicou o aprendizado do método MLC, uma vez que os resultados alcançados por este método são evidentemente superiores aos resultados atingidos pelo método IndMLC. Esta queda de acurácia deriva do método ssFCM, responsável por produzir conjuntos de treinamento com informações inadequadas para o devido aprendizado do método MLC. A semelhança entre os resultados, independente da dimensão do conjunto de treinamento inicial (i.e., conjuntos com 10, 15, 25 e 40 *pixels*), também ocorre em função do método ssFCM, que embora tenha sido

inicializado por diferentes conjuntos de treinamento, convergiu para resultados semelhantes, proporcionando conjuntos de treinamento semelhantes.

O método SemiEM foi superior apenas ao método IndMLC, porém com tempo computacional aproximadamente $6,5$ vezes maior. Embora o desempenho médio seja semelhante diante os conjuntos de treinamento de diferentes dimensões, o aumento da dimensão de tais conjuntos provocou redução no desvio padrão dos índices de acerto global. Isso possibilita concluir que o método SemiEM não depende da dimensão dos conjuntos de treinamento, mas sim da qualidade da informação contida nestes conjuntos.

Ao comparar CoMLC e MLC, verifica-se uma tendência gradual de aumento nos índices de acerto global, assim como o desvio padrão destes índices, semelhante em ambos os métodos. Isso mostra que o aprendizado por co-treinamento foi capaz de proporcionar melhoras no desempenho e manter o comportamento do método de classificação que aprende por este modelo, ou seja, o método MLC.

De forma semelhante à apresentada por MLC e CoMLC, comportam-se os resultados gerados por SVM, TSVM e SemiSVM. O aumento da dimensão do conjunto de treinamento provocou aumento nos índices de acurácia, sendo SemiSVM superior nos diferentes casos, apresentando ainda resultados com desvio padrão semelhantes. Os resultados fornecidos pelo método TSVM são pouco inferiores aos obtidos por SemiSVM, porém com desvios padrão consideravelmente maiores. Observa-se também que o método TSVM apresenta maior tempo computacional comparado ao SemiSVM, aumentando gradativamente com o aumento do conjunto de treinamento.

Bons resultados foram alcançados pelo método GB, principalmente quando seu treinamento é realizado por conjuntos com 10 e 15 *pixels* rotulados por classe, onde seus resultados são semelhantes aos obtidos por TSVM. No entanto, cabe observar o elevado tempo computacional exigido, o qual está relacionado principalmente as frequentes inversões de matriz exigidas no processo de normalização simétrica (10).

4 Conclusões

O objetivo deste estudo foi analisar diferentes modelos de aprendizado semissupervisionado em classificação de imagens. Para isso foi conduzido um experimento Monte Carlo que consistiu na aplicação dos modelos analisados na classificação de um conjunto de imagens simuladas. Ainda, para o treinamento de tais métodos, foram utilizados conjuntos de treinamento de diferentes dimensões, visando verificar situações onde há escassez de informações para o treinamento.

Neste experimento, os modelos de aprendizado semissupervisionado por co-treinamento (CoMLC), separação de baixa densidade (SemiSVM e TSVM) e baseado em grafo (GB) pro-

porcionaram resultados superiores em comparação com os métodos supervisionados analisados (MLC e SVM).

Como perspectiva para trabalhos futuros sugere-se analisar o uso do aprendizado indireto por outros métodos supervisionados que não fazem exigência sobre distribuições estatística, avaliar o método SemiSVM, proposto neste estudo, em aplicações relacionadas à classificação de outros tipos de imagens e em outros problemas que fazem uso de aprendizado semissupervisionado, assim como avaliar o modelo de aprendizado por co-treinamento por outros métodos de classificação, como por exemplo, o SVM.

Referências

[1] BANDOS, T. V., ZHOU, D., AND CAMPS-VALLS, G. Semi-supervised hyperspectral image classification with graphs. Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on, pp. 3883 – 3886.

[2] BASU, S., BANERJEE, A., AND MOONEY, R. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002)* (2002).

[3] BENSAID, A. M., HALL, L. O., BEZDEK, J. C., AND CLARKE, L. P. Partially supervised clustering for image segmentation. *Pattern Recognition 29*, 5 (1996), 859–871.

[4] BISHOP, C. M. *Pattern Recognition and Machine Learning*, 1 ed. 2007.

[5] BRUZZONE, L., CHI, M., AND MARCONCINI, M. A novel transductive svm for semisupervised classification of remote-sensing images. *Geoscience and Remote Sensing, IEEE Transactions on 44*, 11 (nov. 2006), 3363 –3373.

[6] BRUZZONE, L., AND PERSELLO, C. A novel context-sensitive semisupervised svm classifier robust to mislabeled training samples. *IEEE Transactions on Geoscience and Remote Sensing 47* (2009), 2142–2154.

[7] BUSTOS, O. H., AND FRERY, A. C. *Simulação estocástica: teoria e algoritmos*. IMPA, 1992.

[8] CAMPS-VALLS, G., BANDOS MARSHEVA, T., AND ZHOU, D. Semi-supervised graph-based hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on 45*, 10 (2007), 3044 –3054.

[9] CAMPS-VALLS, G., MEMBER, S., B, T. V., AND ZHOU, D. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing 45* (2007), 2044–3054.

[10] CHANG, C.-C., AND LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27.

[11] CHAPELLE, O., SCHÖLKOPF, B., AND ZIEN, A., Eds. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[12] CIRELO, M. C., AND COZMAN, F. G. Aprendizado de semi-supervisionado de classificadores bayesianos utilizando testes de independência. IV Encontro Nacional de Inteligência Artificial.

[13] DEMPSTER, A. P., LAIRD, N. M., AND RDIN, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 39*, 1 (1977), 1–38.

[14] DUTRA, L. V., CORREIA, A. H., MURA, J. C., SANTOS, J. R., ELMIRO, M. T., AND FREITAS, C. C. Avaliação das imagens polarimétricas da simulação mapsar para classificação de uso/ocupação do solo na região da floresta nacional do tapajós. In *Anais...* (São José dos Campos, 2007), Simpósio Brasileiro de Sensoriamento Remoto, 13. (SBSR), Instituto Nacional de Pesquisas Espaciais (INPE), pp. 7051–7056.

[15] GAO, Y., AND MAS, J. F. A comparison of the performance of pixel-based and object-based classification over images with various spatial resolutions. *Imaging 2*, 8701 (2008), 27–35.

[16] GOMEZ-CHOVA, L., BRUZZONE, L., CAMPS-VALLS, G., AND CALPE-MARAVILLA, J. Semi-supervised remote sensing image classification based on clustering and the mean map kernel. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International* (july 2008), vol. 4, pp. IV –391 –IV –394.

[17] GOMEZ-CHOVA, L., CAMPS-VALLS, G., MUNOZ-MARI, J., AND CALPE, J. Semisupervised image classification with laplacian support vector machines. *Geoscience and Remote Sensing Letters, IEEE 5*, 3 (july 2008), 336 –340.

[18] GUTIÉRREZ, V. A. L. Classificação semi-supervisionada baseada em desacordo por similaridade. Dissertação de mestrado, ICMC-USP, São Carlos - SP, 2010.

[19] HAYKIN, S. *Neural Networks and Learning Machines, year = 2008*, 3 ed. Prentice Hall.

[20] JOACHIMS, T. Advances in kernel methods. MIT Press, Cambridge, MA, USA, 1999, ch. Making large-scale support vector machine learning practical, pp. 169–184.

[21] KIYASU, S., YAMADA, Y., AND MIYAHARA, S. Semi-supervised land cover classification of remotely sensed data using two different types of classifiers. ICCAS-SICE, pp. 4874–4877.

[22] LORENA, A. C., AND CARVALHO, A. C. P. L. F. Uma introdução às support vector machines. *RITA 14*, 2 (2007), 43–67.

[23] MATHER, P. M. *Computer Processing of Remotely-Sensed Images : An Introduction*. John Wiley & Sons, June 2004.

[24] MATSUBARA, E. T., MONARD, M. C., AND BATISTA, G. E. Utilizando algoritmos de aprendizado semi-supervisionados multi-visão como rotuladores de texto. In *Anais do Workshop em Tecnologia da Informação de da Linguagem Humana (TIL2005)* (2005), pp. 2108–2117.

[25] MELGANI, F., AND BRUZZONE, L. Classification of hyperspectral remote sensing images with support vector machines. *Geoscience and Remote Sensing, IEEE Transactions on 42*, 8 (2004), 1778–1790.

[26] OLIVEIRA, C. S. Classificadores baseados em vetores suporte gerados a partir de dados rotulados e não-rotulados. Dissertação de mestrado, PUC-USP, São Paulo - SP, 2006.

[27] PLATT, J. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.

[28] PLATT, J. C., CRISTIANINI, N., AND SHAWE-TAYLOR, J. Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems 12* (2000), pp. 547–553.

[29] QI, H.-N., YANG, J.-G., ZHONG, Y.-W., AND DENG, C. Multi-class svm based remote sensing image classification and its semi-supervised improvement scheme. No. 5, Machine Learning and Cybernetics, International Conference on, pp. 3146 – 3151.

[30] THEODORIDIS, S., AND KOUTROUMBAS, K. *Pattern Recognition, Fourth Edition*, 4th ed. Academic Press, 2008.

[31] VAPNIK, V. N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[32] WEBB, A. R. *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.

[33] ZHU, X. Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2008.

[34] ZHU, X., AND GOLDBERG, A. B. *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, 2009.