**1. Giới thiệu**

Máy vector hỗ trợ (Support Vector Machines - SVM) là một phương pháp học máy mạnh mẽ được sử dụng chủ yếu trong các bài toán phân lớp, đặc biệt là phân lớp nhị phân. Được giới thiệu bởi Vapnik vào những năm 1990, SVM tìm cách xây dựng một siêu phẳng tối ưu để tách các lớp dữ liệu, tối đa hóa khoảng cách (lề) giữa siêu phẳng và các điểm dữ liệu gần nhất. Phương pháp này không chỉ hiệu quả trong các không gian tuyến tính mà còn được mở rộng để xử lý các bài toán phi tuyến thông qua kỹ thuật nhân (kernel).

Trong báo cáo này, chúng ta sẽ khám phá các khái niệm cốt lõi của SVM, bao gồm siêu phẳng tách, SVM nguyên thủy (primal), SVM đối ngẫu (dual), kỹ thuật nhân, và các phương pháp tính toán. Ngoài ra, một số ứng dụng thực tế của SVM sẽ được trình bày để minh họa tính hiệu quả của phương pháp này trong các lĩnh vực như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, và y học.

**2. Siêu phẳng tách được**

Một siêu phẳng trong không gian RD \mathbb{R}^D RD là một tập hợp các điểm x \boldsymbol{x} x thỏa mãn phương trình:

w⊤x+b=0,\boldsymbol{w}^\top \boldsymbol{x} + b = 0,w⊤x+b=0,

trong đó w∈RD \boldsymbol{w} \in \mathbb{R}^D w∈RD là vector pháp tuyến, b∈R b \in \mathbb{R} b∈R là độ lệch (bias), và x∈RD \boldsymbol{x} \in \mathbb{R}^D x∈RD là vector đặc trưng.

Trong bài toán phân lớp nhị phân, mục tiêu là tìm một siêu phẳng sao cho dữ liệu thuộc hai lớp yi∈{−1,+1} y\_i \in \{-1, +1\} yi​∈{−1,+1} được tách biệt hoàn toàn. Giả sử tập dữ liệu huấn luyện là {(xi,yi)}i=1N \{(\boldsymbol{x}\_i, y\_i)\}\_{i=1}^N {(xi​,yi​)}i=1N​, trong đó xi∈RD \boldsymbol{x}\_i \in \mathbb{R}^D xi​∈RD và yi∈{−1,+1} y\_i \in \{-1, +1\} yi​∈{−1,+1}. Một siêu phẳng được gọi là **tách được** nếu tồn tại w \boldsymbol{w} w và b b b sao cho:

yi(w⊤xi+b)≥1,∀i=1,…,N.y\_i (\boldsymbol{w}^\top \boldsymbol{x}\_i + b) \geq 1, \quad \forall i = 1, \ldots, N.yi​(w⊤xi​+b)≥1,∀i=1,…,N.

Điều kiện này đảm bảo rằng các điểm dữ liệu của mỗi lớp nằm ở hai phía đối diện của siêu phẳng, với khoảng cách tối thiểu từ siêu phẳng đến các điểm gần nhất (gọi là **lề**) được xác định bởi 1∥w∥ \frac{1}{\|\boldsymbol{w}\|} ∥w∥1​. Để tối đa hóa lề, chúng ta cần tối thiểu hóa ∥w∥2 \|\boldsymbol{w}\|^2 ∥w∥2, dẫn đến bài toán tối ưu hóa của SVM.

**3. Máy Vector Hỗ Trợ**

**3.1. Máy vector hỗ trợ lề cứng (Hard Margin SVM)**

Trong trường hợp dữ liệu có thể tách biệt tuyến tính, SVM lề cứng tìm siêu phẳng tối ưu bằng cách giải bài toán tối ưu hóa sau:

minimize12∥w∥2,subject toyi(w⊤xi+b)≥1,∀i=1,…,N.\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|\boldsymbol{w}\|^2, \\ &\text{subject to} \quad y\_i (\boldsymbol{w}^\top \boldsymbol{x}\_i + b) \geq 1, \quad \forall i = 1, \ldots, N. \end{aligned}​minimize21​∥w∥2,subject toyi​(w⊤xi​+b)≥1,∀i=1,…,N.​

Hàm mục tiêu 12∥w∥2 \frac{1}{2} \|\boldsymbol{w}\|^2 21​∥w∥2 tương ứng với việc tối đa hóa lề 2∥w∥ \frac{2}{\|\boldsymbol{w}\|} ∥w∥2​. Các ràng buộc đảm bảo rằng tất cả các điểm dữ liệu được phân loại chính xác với khoảng cách ít nhất là 1 từ siêu phẳng (theo đơn vị chuẩn hóa).

Các điểm dữ liệu thỏa mãn đẳng thức yi(w⊤xi+b)=1 y\_i (\boldsymbol{w}^\top \boldsymbol{x}\_i + b) = 1 yi​(w⊤xi​+b)=1 được gọi là **vector hỗ trợ**, vì chúng nằm gần siêu phẳng nhất và xác định vị trí của siêu phẳng.

**3.2. Máy vector hỗ trợ lề mềm (Soft Margin SVM)**

Trong thực tế, dữ liệu thường không tách biệt tuyến tính hoàn hảo do nhiễu hoặc chồng lấn. SVM lề mềm đưa vào các **biến chùng** (slack variables) ξi≥0 \xi\_i \geq 0 ξi​≥0 để cho phép một số điểm dữ liệu vi phạm ràng buộc lề. Bài toán tối ưu hóa trở thành:

minimize12∥w∥2+C∑i=1N\xihanei,subject toyi(w⊤xi+b)≥1−ξi,ξi≥0,∀i=1,…,N.\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum\_{i=1}^N \xihane\_i, \\ &\text{subject to} \quad y\_i (\boldsymbol{w}^\top \boldsymbol{x}\_i + b) \geq 1 - \xi\_i, \quad \xi\_i \geq 0, \quad \forall i = 1, \ldots, N. \end{aligned}​minimize21​∥w∥2+Ci=1∑N​\xihanei​,subject toyi​(w⊤xi​+b)≥1−ξi​,ξi​≥0,∀i=1,…,N.​

Trong đó, C>0 C > 0 C>0 là tham số điều chỉnh, kiểm soát sự đánh đổi giữa việc tối đa hóa lề (nhỏ ∥w∥2 \|\boldsymbol{w}\|^2 ∥w∥2) và giảm thiểu lỗi phân loại (nhỏ ∑ξi \sum \xi\_i ∑ξi​). Giá trị C C C lớn ưu tiên phân loại chính xác, trong khi C C C nhỏ ưu tiên lề lớn hơn.

**4. Máy Vector Hỗ Trợ Đối Ngẫu**

Thay vì giải bài toán nguyên thủy (primal), SVM thường được giải trong dạng đối ngẫu (dual), vì dạng này có một số ưu điểm, đặc biệt khi kết hợp với kỹ thuật nhân.

**Bài toán đối ngẫu**

Bài toán đối ngẫu của SVM lề mềm được xây dựng bằng cách sử dụng **bộ nhân tử Lagrange** αi≥0 \alpha\_i \geq 0 αi​≥0 và μi≥0 \mu\_i \geq 0 μi​≥0 cho các ràng buộc. Hàm Lagrangian được định nghĩa như:

L(w,b,ξ,α,μ)=12∥w∥2+C∑i=1Nξi−∑i=1Nαi[yi(w⊤xi+b)−1+ξi]−∑i=1Nμiξi.\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum\_{i=1}^N \xi\_i - \sum\_{i=1}^N \alpha\_i \left[ y\_i (\boldsymbol{w}^\top \boldsymbol{x}\_i + b) - 1 + \xi\_i \right] - \sum\_{i=1}^N \mu\_i \xi\_i.L(w,b,ξ,α,μ)=21​∥w∥2+Ci=1∑N​ξi​−i=1∑N​αi​[yi​(w⊤xi​+b)−1+ξi​]−i=1∑N​μi​ξi​.

Để tìm điểm tối ưu, chúng ta lấy đạo hàm của L \mathcal{L} L theo w,b,ξi \boldsymbol{w}, b, \xi\_i w,b,ξi​ và đặt bằng 0:

∂L∂w=w−∑i=1Nαiyixi=0  ⟹  w=∑i=1Nαiyixi,\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum\_{i=1}^N \alpha\_i y\_i \boldsymbol{x}\_i = 0 \implies \boldsymbol{w} = \sum\_{i=1}^N \alpha\_i y\_i \boldsymbol{x}\_i,∂w∂L​=w−i=1∑N​αi​yi​xi​=0⟹w=i=1∑N​αi​yi​xi​, ∂L∂b=−∑i=1Nαiyi=0  ⟹  ∑i=1Nαiyi=0,\frac{\partial \mathcal{L}}{\partial b} = -\sum\_{i=1}^N \alpha\_i y\_i = 0 \implies \sum\_{i=1}^N \alpha\_i y\_i = 0,∂b∂L​=−i=1∑N​αi​yi​=0⟹i=1∑N​αi​yi​=0, ∂L∂ξi=C−αi−μi=0  ⟹  αi+μi=C.\frac{\partial \mathcal{L}}{\partial \xi\_i} = C - \alpha\_i - \mu\_i = 0 \implies \alpha\_i + \mu\_i = C.∂ξi​∂L​=C−αi​−μi​=0⟹αi​+μi​=C.

Thay các biểu thức này vào L \mathcal{L} L, ta thu được bài toán đối ngẫu:

maximize∑i=1Nαi−12∑i=1N∑j=1Nαiαjyiyjxi⊤xj,subject to∑i=1Nαiyi=0,0≤αi≤C,∀i=1,…,N.\begin{aligned} &\text{maximize} \quad \sum\_{i=1}^N \alpha\_i - \frac{1}{2} \sum\_{i=1}^N \sum\_{j=1}^N \alpha\_i \alpha\_j y\_i y\_j \boldsymbol{x}\_i^\top \boldsymbol{x}\_j, \\ &\text{subject to} \quad \sum\_{i=1}^N \alpha\_i y\_i = 0, \quad 0 \leq \alpha\_i \leq C, \quad \forall i = 1, \ldots, N. \end{aligned}​maximizei=1∑N​αi​−21​i=1∑N​j=1∑N​αi​αj​yi​yj​xi⊤​xj​,subject toi=1∑N​αi​yi​=0,0≤αi​≤C,∀i=1,…,N.​

**Ý nghĩa của dạng đối ngẫu**

* **Vector hỗ trợ**: Các αi>0 \alpha\_i > 0 αi​>0 tương ứng với các vector hỗ trợ. Những điểm này quyết định siêu phẳng phân tách.
* **Tính thưa**: Chỉ một số ít αi \alpha\_i αi​ khác 0, giúp giảm chi phí tính toán khi dự đoán.
* **Tích vô hướng**: Hàm mục tiêu đối ngẫu chỉ phụ thuộc vào tích vô hướng xi⊤xj \boldsymbol{x}\_i^\top \boldsymbol{x}\_j xi⊤​xj​, mở ra khả năng sử dụng kỹ thuật nhân để xử lý dữ liệu phi tuyến.

Hàm dự đoán của SVM được biểu diễn như:

f(x)=sign(∑i=1Nαiyixi⊤x+b).f(\boldsymbol{x}) = \text{sign} \left( \sum\_{i=1}^N \alpha\_i y\_i \boldsymbol{x}\_i^\top \boldsymbol{x} + b \right).f(x)=sign(i=1∑N​αi​yi​xi⊤​x+b).

**5. Nhân (Kernel)**

Trong các bài toán mà dữ liệu không tách biệt tuyến tính trong không gian gốc, SVM sử dụng **nhân** (kernel) để ánh xạ dữ liệu vào một không gian đặc trưng có chiều cao hơn, nơi dữ liệu có thể tách biệt tuyến tính.

**Định nghĩa nhân**

Một hàm nhân K(xi,xj) K(\boldsymbol{x}\_i, \boldsymbol{x}\_j) K(xi​,xj​) thay thế tích vô hướng xi⊤xj \boldsymbol{x}\_i^\top \boldsymbol{x}\_j xi⊤​xj​ trong bài toán đối ngẫu. Theo **định lý Mercer**, một hàm K K K là nhân hợp lệ nếu nó đối xứng và **tích cực bán xác định** (positive semi-definite). Một cách biểu diễn nhân là:

K(xi,xj)=ϕ(xi)⊤ϕ(xj),K(\boldsymbol{x}\_i, \boldsymbol{x}\_j) = \phi(\boldsymbol{x}\_i)^\top \phi(\boldsymbol{x}\_j),K(xi​,xj​)=ϕ(xi​)⊤ϕ(xj​),

trong đó ϕ:RD→H \phi: \mathbb{R}^D \to \mathcal{H} ϕ:RD→H là ánh xạ vào không gian đặc trưng H \mathcal{H} H.

**Các loại nhân phổ biến**

1. **Nhân tuyến tính**:

K(xi,xj)=xi⊤xj.K(\boldsymbol{x}\_i, \boldsymbol{x}\_j) = \boldsymbol{x}\_i^\top \boldsymbol{x}\_j.K(xi​,xj​)=xi⊤​xj​.

Dùng cho dữ liệu tách biệt tuyến tính.

1. **Nhân đa thức**:

K(xi,xj)=(xi⊤xj+c)d,K(\boldsymbol{x}\_i, \boldsymbol{x}\_j) = (\boldsymbol{x}\_i^\top \boldsymbol{x}\_j + c)^d,K(xi​,xj​)=(xi⊤​xj​+c)d,

trong đó d d d là bậc của đa thức và c≥0 c \geq 0 c≥0 là hằng số.

1. **Nhân Gaussian (RBF)**:

K(xi,xj)=exp⁡(−∥xi−xj∥22σ2).K(\boldsymbol{x}\_i, \boldsymbol{x}\_j) = \exp \left( -\frac{\|\boldsymbol{x}\_i - \boldsymbol{x}\_j\|^2}{2\sigma^2} \right).K(xi​,xj​)=exp(−2σ2∥xi​−xj​∥2​).

Phù hợp với dữ liệu phi tuyến phức tạp.

**Bài toán đối ngẫu với nhân**

Khi sử dụng nhân, bài toán đối ngẫu trở thành:

maximize∑i=1Nαi−12∑i=1N∑j=1NαiαjyiyjK(xi,xj),subject to∑i=1Nαiyi=0,0≤αi≤C,∀i=1,…,N.\begin{aligned} &\text{maximize} \quad \sum\_{i=1}^N \alpha\_i - \frac{1}{2} \sum\_{i=1}^N \sum\_{j=1}^N \alpha\_i \alpha\_j y\_i y\_j K(\boldsymbol{x}\_i, \boldsymbol{x}\_j), \\ &\text{subject to} \quad \sum\_{i=1}^N \alpha\_i y\_i = 0, \quad 0 \leq \alpha\_i \leq C, \quad \forall i = 1, \ldots, N. \end{aligned}​maximizei=1∑N​αi​−21​i=1∑N​j=1∑N​αi​αj​yi​yj​K(xi​,xj​),subject toi=1∑N​αi​yi​=0,0≤αi​≤C,∀i=1,…,N.​

Hàm dự đoán được viết lại là:

f(x)=sign(∑i=1NαiyiK(xi,x)+b).f(\boldsymbol{x}) = \text{sign} \left( \sum\_{i=1}^N \alpha\_i y\_i K(\boldsymbol{x}\_i, \boldsymbol{x}) + b \right).f(x)=sign(i=1∑N​αi​yi​K(xi​,x)+b).

**Ưu điểm của nhân**: Không cần tính toán trực tiếp ánh xạ ϕ \phi ϕ, giúp giảm chi phí tính toán trong không gian chiều cao.

**6. Tính Toán Máy Vector Hỗ Trợ**

**Giải bài toán đối ngẫu**

Bài toán đối ngẫu là một bài toán tối ưu hóa lồi với các ràng buộc tuyến tính, có thể được giải bằng các phương pháp tối ưu hóa số, chẳng hạn như:

* **Phương pháp gradient descent**: Không hiệu quả do số lượng lớn các biến αi \alpha\_i αi​.
* **Phương pháp phân rã (decomposition methods)**: Chia bài toán thành các bài toán con nhỏ hơn, ví dụ, **Sequential Minimal Optimization (SMO)**.

SMO chia bài toán thành các cặp (αi,αj) (\alpha\_i, \alpha\_j) (αi​,αj​) và tối ưu hóa chúng trong khi giữ các αk \alpha\_k αk​ khác cố định. Thuật toán SMO có các bước chính:

1. Chọn một cặp (αi,αj) (\alpha\_i, \alpha\_j) (αi​,αj​) vi phạm điều kiện KKT (Karush-Kuhn-Tucker).
2. Tối ưu hóa cặp này bằng cách giải một bài toán tối ưu hóa bậc hai đơn giản.
3. Cập nhật b b b và lặp lại cho đến khi hội tụ.

**Tính toán tham số b b b**

Sau khi tìm được α \boldsymbol{\alpha} α, tham số b b b được tính dựa trên các vector hỗ trợ (αi>0 \alpha\_i > 0 αi​>0):

b=1Ns∑i∈S(yi−∑j∈SαjyjK(xj,xi)),b = \frac{1}{N\_s} \sum\_{i \in S} \left( y\_i - \sum\_{j \in S} \alpha\_j y\_j K(\boldsymbol{x}\_j, \boldsymbol{x}\_i) \right),b=Ns​1​i∈S∑​​yi​−j∈S∑​αj​yj​K(xj​,xi​)​,

trong đó S S S là tập các chỉ số của vector hỗ trợ, và Ns N\_s Ns​ là số lượng vector hỗ trợ.

**Thư viện tính toán**

Trong thực tế, các thư viện như **LIBSVM** và **LIBLINEAR** được sử dụng để triển khai SVM. Chúng cung cấp các công cụ hiệu quả để huấn luyện và dự đoán, hỗ trợ nhiều loại nhân và tối ưu hóa nhanh.

**7. Ứng dụng của Máy Vector Hỗ Trợ**

SVM đã được áp dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng xử lý cả dữ liệu tuyến tính và phi tuyến. Dưới đây là một số ứng dụng tiêu biểu:

1. **Nhận dạng chữ viết tay**:
   * SVM với nhân RBF được sử dụng để phân loại các chữ số viết tay trong tập dữ liệu MNIST. Các vector đặc trưng được trích xuất từ hình ảnh, và SVM phân loại các chữ số từ 0 đến 9 với độ chính xác cao.
2. **Phân loại văn bản**:
   * Trong xử lý ngôn ngữ tự nhiên, SVM được dùng để phân loại cảm xúc (sentiment analysis) hoặc lọc spam email. Dữ liệu văn bản được biểu diễn dưới dạng vector TF-IDF, và nhân tuyến tính hoặc RBF được sử dụng.
3. **Chẩn đoán y học**:
   * SVM được áp dụng để phân loại bệnh nhân dựa trên dữ liệu y tế, chẳng hạn như phát hiện ung thư vú từ hình ảnh chụp X-quang. Nhân Gaussian giúp xử lý các đặc trưng phức tạp trong dữ liệu y tế.
4. **Nhận diện khuôn mặt**:
   * SVM kết hợp với các kỹ thuật trích xuất đặc trưng như HOG (Histogram of Oriented Gradients) được sử dụng để nhận diện khuôn mặt trong các hệ thống bảo mật.
5. **Dự đoán tài chính**:
   * SVM được dùng để dự đoán xu hướng giá cổ phiếu hoặc phân loại rủi ro tín dụng dựa trên dữ liệu lịch sử.

**8. Kết luận**

Máy vector hỗ trợ là một công cụ mạnh mẽ trong học máy, đặc biệt phù hợp với các bài toán phân lớp nhị phân. Bằng cách tối đa hóa lề giữa các lớp, SVM đảm bảo khả năng tổng quát hóa tốt trên dữ liệu mới. Dạng đối ngẫu và kỹ thuật nhân cho phép SVM xử lý hiệu quả các bài toán phi tuyến, trong khi các phương pháp tính toán như SMO giúp triển khai thuật toán trên dữ liệu lớn.

Tuy nhiên, SVM cũng có một số hạn chế, chẳng hạn như độ phức tạp tính toán cao với dữ liệu rất lớn và sự nhạy cảm với việc chọn tham số C C C và nhân. Trong tương lai, các cải tiến của SVM, như kết hợp với học sâu hoặc áp dụng trong các không gian không chuẩn, có thể mở rộng thêm phạm vi ứng dụng của phương pháp này.

**9. Tài liệu tham khảo**

1. Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.
2. Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.
3. Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.
4. Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27:1–27:27.