Supervised Machine Learning: Classification

Personal Note - Supervised Learning Sheet

1. Hồi Quy Logistic & Thước Đo Lỗi

1.1. Bài toán & mô hình

- Bài toán: phân loại nhị phân ($y \in \{0, 1\}$).
- Mô hình: xác suất lớp 1 theo hàm **Sigmoid**:

$$p(y = 1|x) = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad z = w^T x + b$$

- Ký hiệu: $x \in \mathbb{R}^d$ (vector đặc trưng), w (trọng số), b (bias), σ (sigmoid), z (logit).
- Biên quyết định (decision boundary): tập điểm $\{x : w^T x + b = 0\}$. Khi $z = 0$, $p = 0.5$.

1.2. Hàm mất mát (Binary Cross-Entropy / Log Loss)

$$\mathcal{L}(w, b) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] + \lambda ||w||_2^2$$

- Ký hiệu: $p_i = \sigma(w^T x_i + b)$, λ là hệ số L2 regularization (giảm overfitting).
- Gradient:

$$\nabla_w \mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (p_i - y_i) x_i + 2\lambda w$$

$$\partial_b \mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (p_i - y_i)$$

1.3. Hiệu chỉnh ngưỡng (threshold)

- Dự đoán nhãn $\hat{y} = 1[p \geq \tau]$.
- Mặc định $\tau = 0.5$ nhưng có thể tối ưu theo mục tiêu (F1, Recall...).

1.4. Thước đo đánh giá (Classification Metrics)

- Confusion Matrix (nhãn dương = 1): TP, FP, TN, FN.
- **Accuracy**: $\frac{TP+TN}{TP+FP+TN+FN}$ - dễ sai lệch khi mất cân bằng lớp.
- **Precision**: $\frac{TP}{TP+FP}$ - tỷ lệ dự đoán dương đúng.
- **Recall (TPR)**: $\frac{TP}{TP+FN}$ - khả năng bắt đúng dương.
- **F1**: $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$.
- **ROC-AUC**: diện tích dưới đường ROC (TPR vs FPR).
- **PR-AUC**: tốt khi dữ liệu lệch lớp.
- **Log Loss**: đánh giá xác suất, phạt nặng dự đoán tự tin nhưng sai.
- **Calibration**: xác suất mô hình có "thật" không.

1.5. Key Takeaways

- Sigmoid biến logit thành xác suất; điều chỉnh ngưỡng quan trọng.
- Dùng regularization (L2) để chống overfit.
- Với dữ liệu lệch lớp: ưu tiên Recall/F1/PR-AUC thay vì Accuracy.
- Calibration hữu ích nếu cần xác suất tin cậy (ra quyết định rủi ro).

---

2. K-Nearest Neighbors (KNN)

2.1. Ý tưởng

- "Bạn của tôi là K láng giềng gần nhất".
- Phân loại theo đa số phiếu (hoặc có trọng số khoảng cách).

2.2. Khoảng cách thường dùng

- **Euclidean (L2)**:

$$||x - x'||_2 = \sqrt{\sum_j (x_j - x'_j)^2}$$

- **Manhattan (L1)**:

$$\sum_j |x_j - x'_j|$$

- **Minkowski (p)**:
$$\left(\sum_j |x_j - x'_j|^p\right)^{1/p}$$
- **Cosine**: $1 - \frac{x \cdot x'}{||x||||x'||}$ - cho dữ liệu dạng hướng.

Chú ý: KNN nhạy cảm thang đo → cần chuẩn hóa/scale đặc trưng.

2.3. Dự đoán

- **Classification**: $\hat{y} = \text{mode}(y_{(1)}, ..., y_{(K)})$ hoặc vote có trọng số $w_i = 1/(d_i + \epsilon)$.
- **Regression**: trung bình (hoặc trung vị) của K láng giềng.

2.4. Chọn K & độ phức tạp

- K nhỏ → phương sai cao (overfit). K lớn → độ chệch cao (underfit).
- Độ phức tạp: huấn luyện rẻ (lazy), suy diễn tốn $O(n \cdot d)$ mỗi truy vấn.
- Có thể tăng tốc bằng KD-Tree/Ball Tree.

2.5. Key Takeaways

- Hiệu quả khi biên quyết định phức tạp, dữ liệu không tuyến tính.
- Scale features & chọn K bằng CV; cân nhắc weight theo khoảng cách.
- Với dữ liệu lớn, xem xét cấu trúc chỉ mục hoặc mô hình khác.

---

3. Support Vector Machines (SVM) với Gaussian (RBF) Kernel

3.1. Max-margin & Soft-margin

- Mục tiêu: tìm siêu phẳng phân tách với biên lớn nhất.
- Soft-margin dùng tham số C cân bằng giữa biên rộng và lỗi huấn luyện.

3.2. Dạng dual & hàm quyết định

- Hàm quyết định:
$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b$$
- Ký hiệu: $\alpha_i \geq 0$ là trọng số trong bài toán dual; chỉ support vectors có $\alpha_i > 0$, K là hàm kernel.

3.3. Gaussian (RBF) Kernel

$$K(x, x') = \exp(-\gamma||x - x'||_2^2)$$

- γ (gamma) lớn $\rightarrow$ vùng ảnh hưởng nhỏ $\rightarrow$ biên phức tạp (overfit). γ nhỏ $\rightarrow$ biên mượt hơn.
- C lớn $\rightarrow$ phạt lỗi mạnh $\rightarrow$ ít lỗi huấn luyện nhưng dễ overfit. C nhỏ $\rightarrow$ biên rộng hơn.

3.4. Xác suất & chuẩn hóa

- SVM không sinh xác suất tự nhiên; có thể calibrate (Platt scaling / isotonic).
- Như KNN, chuẩn hóa đặc trưng là bắt buộc khi dùng RBF.

3.5. Key Takeaways

- C và γ cần tìm bằng CV (grid/random search).
- Mạnh với biên phi tuyến; nhạy với scale và tham số.
- Ít bị ảnh hưởng bởi nhiễu ngoại lai nếu chọn tham số hợp lý.

---

4. Cây Quyết Định (Decision Trees)

4.1. Ý tưởng

- Chia để trị: tách không gian đặc trưng thành các vùng đồng nhất bằng các điều kiện if-then ($x_j \leq t, ...$).

4.2. Độ đo nhiễm bẩn (Impurity)

- **Gini**: $G = 1 - \sum_k p_k^2$.
- **Entropy**: $H = -\sum_k p_k \log p_k$.
- **Information Gain (IG)** tại ngưỡng t trên thuộc tính j:

$$IG = H(\text{node}) - \left(\frac{n_L}{n}H(L) + \frac{n_R}{n}H(R)\right)$$

- Ký hiệu: p_k là tần suất lớp k trong node; L, R là node trái/phải với kích thước n_L, n_R.

4.3. Dừng & cắt tỉa

- Tham số: max_depth, min_samples_split, min_samples_leaf....
- Pruning (cost-complexity) để giảm overfit.

4.4. Ưu nhược

- **Ưu**: dễ hiểu, giải thích, xử lý feature số/lịch sử, không cần scale.
- **Nhược**: dễ overfit nếu không kiểm soát độ sâu; biên quyết định bậc thang.

4.5. Key Takeaways

- Chọn độ sâu hợp lý + pruning.
- Dùng Gini hoặc Entropy; theo dõi IG để hiểu quyết định tách.

---

5. Học Tổ Hợp (Ensembles)

5.1. Bagging

- Huấn luyện nhiều mô hình độc lập trên các bootstrap samples.
- Dự đoán bằng trung bình/đa số $\rightarrow$ giảm phương sai.

5.2. Random Forest (RF)

- Bagging cây + chọn ngẫu nhiên feature tại mỗi split.
- Giảm tương quan giữa cây $\rightarrow$ hiệu quả hơn bagging thuần.
- Feature importance có sẵn (impurity-based; nên kiểm tra thêm permutation).

5.3. Boosting (AdaBoost, Gradient Boosting)

- Xây mô hình tuần tự, mỗi mô hình mới tập trung vào lỗi còn lại.
- **AdaBoost**:
  - Sai số base learner: ϵ_t.
  - Trọng số mô hình:
$$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$$
  - Cập nhật trọng số mẫu: $w_i \leftarrow w_i \exp(-\alpha_t y_i h_t(x_i))$, rồi chuẩn hóa.
- **Gradient Boosting**: khớp residuals theo hướng giảm gradient của loss (ví dụ log-loss, MSE).
- Tham số: learning_rate, n_estimators, max_depth (của cây yếu).

5.4. Stacking

- Học meta-model (level-2) trên out-of-fold predictions của nhiều base models (level-1).

5.5. Key Takeaways

- **RF**: mạnh, ít tuning, baseline tốt.
- **Boosting**: rất mạnh nhưng cần tuning (learning rate, n_estimators, depth). Dễ overfit nếu quá nhiều vòng.
- **Stacking**: hợp nhất điểm mạnh của nhiều mô hình, cần pipeline CV cẩn thận để tránh leakage.

---

6. Khả Năng Giải Thích Mô Hình (Model Interpretability)

6.1. Toàn cục vs Cục bộ

- **Global**: tầm ảnh hưởng chung của feature (coefficients, feature importance, PDP).
- **Local**: giải thích cho 1 quan sát (ICE, SHAP/LIME, local surrogate).

6.2. Kỹ thuật

- **Linear/Logistic**: hệ số đã chuẩn hóa → dấu & độ lớn = hướng & cường độ ảnh hưởng.
- **Trees/RF/GB**:
  - Impurity importance (nhanh, có bias).
  - Permutation importance (đáng tin hơn).
- **PDP** (Partial Dependence Plot): $\mathbb{E}_{x_{-j}}[\hat{f}(x_j, x_{-j})]$.
- **ICE**: đường ảnh hưởng theo từng cá thể.
- **SHAP**: phân rã dự đoán thành đóng góp từng feature dựa trên giá trị Shapley.
- **Calibration & Threshold**: đường reliability để kiểm tra xác suất.

6.3. Key Takeaways

- Chọn công cụ phù hợp loại mô hình & mục tiêu (global vs local).
- Permutation + PDP/ICE là combo trực quan & đáng tin.
- Với quyết định rủi ro: kiểm tra calibration.

---

7. Mô Hình Hóa Lớp Mất Cân Bằng (Modeling Unbalanced Classes)

7.1. Vấn đề

- Lớp dương hiếm (ví dụ 1%). Accuracy dễ đánh lừa.

7.2. Chiến lược

- **Đánh giá**: dùng Precision, Recall, F1, PR-AUC, confusion matrix, cost-sensitive.
- **Resampling**:
  - Downsampling lớp lớn.
  - Upsampling lớp hiếm (Random Over-Sampling) hoặc SMOTE/ADASYN (tạo mẫu tổng hợp).
- **Class weights**: tăng phạt lỗi cho lớp hiếm (weight $\propto 1/\text{freq}$).
- **Threshold moving**: tối ưu τ theo metric mong muốn (ví dụ maximize F1).
- **Stratified CV**: giữ tỉ lệ lớp trong các fold.

7.3. Key Takeaways

- Tránh chỉ báo cáo Accuracy; theo dõi PR-AUC/F1/Recall.
- Bắt đầu bằng class weights và stratified CV (đơn giản, an toàn).
- Cân nhắc SMOTE + tuning threshold để tối ưu mục tiêu nghiệp vụ.

Phụ lục: Bảng ký hiệu nhanh

- x: vector đặc trưng; x_j: đặc trưng thứ j.
- w: vector trọng số; b: bias.
- σ: sigmoid; z: logit.
- TP/FP/TN/FN: các phần tử ma trận nhầm lẫn.
- C: tham số phạt SVM; γ: tham số kernel RBF.
- IG: Information Gain; PDP/ICE/SHAP: công cụ diễn giải.

Key Takeaways toàn bộ khóa

- Bài toán phân loại cần metric phù hợp với mục tiêu và phân phối lớp.
- Regularization/Resampling/Threshold là ba đòn bẩy quan trọng.
- RF/Boosting thường cho baseline mạnh; SVM-RBF tốt khi dữ liệu vừa phải & phi tuyến; KNN đơn giản nhưng nhạy với scale.
- Giải thích mô hình không chỉ để trình bày mà còn để debug và đưa ra quyết định.