Học Máy

(Machine Learning)

Thân Quang Khoát

khoattq@soict.hust.edu.vn

Viện Công nghệ thông tin và Truyền thông
Trường Đại học Bách Khoa Hà Nội

Năm 2015

Nội dung môn học:

- Giới thiệu chung
- Các phương pháp học không giám sát
- Các phương pháp học có giám sát
- **Đánh giá hiệu năng hệ thống học máy**

1. Đánh giá hiệu năng hệ thống học máy.

- Làm thế nào để thu được một đánh giá đáng tin cậy về hiệu năng của hệ thống?
  - Tập huấn luyện càng lớn, thì hiệu năng của hệ thống học có xu hướng càng tốt
  - Tập kiểm thử càng lớn, thì việc đánh giá càng chính xác
- Làm thế nào để lựa chọn tốt các tham số cho một phương pháp học máy?
- *Ngoài giải thuật học, hiệu năng của hệ thống còn phụ thuộc vào:*
  - Phân bố lớp (Class distribution)
  - Kích thước của tập huấn luyện (Size of the training set)
  - Khả năng đại diện của tập học (representative of the whole space)

1. Đánh giá hiệu năng hệ thống học máy..

- Đánh giá lý thuyết (theoretical evaluation): nghiên cứu các khía cạnh lý thuyết của một hệ thống mà có thể chứng minh được.
  - Tốc độ học, thời gian học,
  - Bao nhiêu ví dụ học là đủ?
  - Độ chính xác trung bình của hệ thống,
  - Khả năng chống nhiễu,…
- Đánh giá thực nghiệm (experimental evaluation): quan sát hệ thống làm việc trong thực tế, sử dụng một hoặc nhiều tập dữ liệu và các tiêu chí đánh giá. Tổng hợp đánh giá từ các quan sát đó.
- Chúng ta sẽ nghiên cứu cách đánh giá thực nghiệm.

1. Đánh giá hiệu năng hệ thống học máy...

- **Bài toán đánh giá** (model assessment): *cần đánh giá hiệu năng của phương pháp (model) học máy A, chỉ dựa trên bộ dữ liệu đã quan sát D.*
- Việc đánh giá hiệu năng của hệ thống
  - *Thực hiện một cách tự động*, sử dụng một tập các ví dụ.
  - Không cần sự tham gia (can thiệp) của người dùng.
- **Chiến lược** đánh giá (evaluation strategies)

  → Làm sao có được một đánh giá đáng tin cậy về hiệu năng của hệ thống?
- Các **tiêu chí** đánh giá (evaluation metrics)

  → Làm sao để đo hiệu năng của hệ thống?

2. Các phương pháp đánh giá

- Hold-out
- Stratified sampling
- Repeated hold-out
- Cross-validation
  - *k*-fold
  - Leave-one-out
- Bootstrap sampling

Hold-out (Splitting)

- Toàn bộ tập ví dụ D được chia thành 2 tập con **không giao nhau**
  - Tập huấn luyện D_train – để huấn luyện hệ thống
  - Tập kiểm thử D_test – để đánh giá hiệu năng của hệ thống đã học

  → $D = D\_train \cup D\_test$, và thường là $|D\_train| >> |D\_test|$
- Các yêu cầu:
  - Bất kỳ ví dụ nào thuộc vào tập kiểm thử D_test đều không được sử dụng trong quá trình huấn luyện hệ thống
  - Bất kỳ ví dụ nào được sử dụng trong giai đoạn huấn luyện hệ thống (i.e., thuộc vào D_train) đều không được sử dụng trong giai đoạn đánh giá hệ thống
  - Các ví dụ kiểm thử trong D_test cho phép một đánh giá không thiên vị đối với hiệu năng của hệ thống
- Các lựa chọn thường gặp: $|D\_train|=(2/3).|D|$, $|D\_test|=(1/3).|D|$
- Phù hợp khi ta có tập ví dụ D có kích thước lớn

Stratified sampling

- Đối với các tập ví dụ có kích thước nhỏ hoặc không cân xứng (unbalanced datasets), các ví dụ trong tập huấn luyện và thử nghiệm có thể không phải là đại diện
  - Ví dụ: Có (rất) ít ví dụ đối với một số lớp
- Mục tiêu: Phân bố lớp (class distribution) trong tập huấn luyện và tập kiểm thử phải xấp xỉ như trong tập toàn bộ các ví dụ (D)
- Lấy mẫu phân tầng (Stratified sampling)
  - Là một phương pháp để cân xứng (về phân bố lớp)
  - Đảm bảo tỷ lệ phân bố lớp (tỷ lệ các ví dụ giữa các lớp) trong tập huấn luyện và tập kiểm thử là xấp xỉ nhau
- Phương pháp lấy mẫu phân tầng không áp dụng được cho bài toán hồi quy (vì giá trị đầu ra của hệ thống là một giá trị số thực, không phải là một nhãn lớp)

Repeated hold-out

- Áp dụng phương pháp đánh giá Hold-out nhiều lần, để sinh ra (sử dụng) các tập huấn luyện và thử nghiệm khác nhau
  - Trong mỗi bước lặp, một tỷ lệ nhất định của tập D ***được lựa chọn ngẫu nhiên*** để tạo nên tập huấn luyện (có thể sử dụng kết hợp với phương pháp lấy mẫu phân tầng – stratified sampling)
  - Các giá trị lỗi (hoặc các giá trị đối với các tiêu chí đánh giá khác) ghi nhận được trong các bước lặp này được *lấy trung bình cộng (averaged)* để xác định giá trị lỗi tổng thể
- Phương pháp này vẫn không hoàn hảo
  - Mỗi bước lặp sử dụng một tập kiểm thử khác nhau
  - Có một số ví dụ trùng lặp (được sử dụng lại nhiều lần) trong các tập kiểm thử này

Cross-validation

- Để tránh việc trùng lặp giữa các tập kiểm thử (một số ví dụ cùng xuất hiện trong các tập kiểm thử khác nhau)
- k-fold cross-validation
  - Tập toàn bộ các ví dụ D được chia thành k tập con **không giao nhau** (gọi là "*fold*") có kích thước xấp xỉ nhau
  - Mỗi lần (trong số *k* lần) lặp, một tập con được sử dụng làm tập kiểm thử, và (k-1) tập con còn lại được dùng làm tập huấn luyện
  - k giá trị lỗi (mỗi giá trị tương ứng với một *fold*) được tính trung bình cộng để thu được giá trị lỗi tổng thể
- Các lựa chọn thông thường của k: 10, hoặc 5
- Thông thường, mỗi tập con (fold) được lấy mẫu phân tầng (xấp xỉ phân bố lớp) trước khi áp dụng quá trình đánh giá Cross-validation
- Phù hợp khi ta có tập ví dụ D vừa và nhỏ

Leave-one-out cross-validation

- Một trường hợp (kiểu) của phương pháp Cross-validation
  - Số lượng các nhóm (folds) bằng kích thước của tập dữ liệu (k=|D|)
  - Mỗi nhóm (fold) chỉ bao gồm một ví dụ
- Khai thác tối đa (triệt để) tập ví dụ ban đầu
- Không hề có bước lấy mẫu ngẫu nhiên (no random sub-sampling)
- Áp dụng lấy mẫu phân tầng (stratification) không phù hợp

  → Vì ở mỗi bước lặp, tập thử nghiệm chỉ gồm có một ví dụ
- Chi phí tính toán (rất) cao
- Phù hợp khi ta có một tập ví dụ D (rất) nhỏ

Bootstrap sampling.

- Phương pháp Cross-validation sử dụng việc lấy mẫu không lặp lại (sampling without replacement)
  → Đối với mỗi ví dụ, *một khi đã được chọn (được sử dụng), thì không thể được chọn (sử dụng) lại* cho tập huấn luyện
- Phương pháp Bootstrap sampling sử dụng việc ***lấy mẫu có lặp lại (sampling with replacement)*** để tạo nên tập huấn luyện
  - Giả sử tập toàn bộ D bao gồm n ví dụ
  - Lấy mẫu có lặp lại n lần đối với tập D, để tạo nên tập huấn luyện D_train gồm n ví dụ
    - Từ tập D, lấy ra ngẫu nhiên một ví dụ x (nhưng **không loại bỏ** x khỏi tập D)
    - Đưa ví dụ x vào trong tập huấn luyện: $D\_train = D\_train \cup x$
    - Lặp lại 2 bước trên n lần
  - Sử dụng tập D_train để huấn luyện hệ thống
  - Sử dụng tất cả các ví dụ thuộc D **nhưng không thuộc** D_train để tạo nên tập thử nghiệm: $D\_test = \{z \in D; z \notin D\_train\}$

Bootstrap sampling..

- Trong mỗi bước lặp, một ví dụ có xác suất = $\left(1-\frac{1}{n}\right)$ để không được lựa chọn đưa vào tập huấn luyện
- Vì vậy, xác suất để một ví dụ (sau quá trình lấy mẫu lặp lại – bootstrap sampling) được đưa vào tập kiểm thử là:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} \approx 0.368$$

- Có nghĩa rằng:
  - Tập huấn luyện (có kích thước =n) bao gồm xấp xỉ 63.2% các ví dụ trong D (Lưu ý: Một ví dụ thuộc tập D có thể **xuất hiện nhiều lần** trong tập D_train)
  - Tập kiểm thử (có kích thước <n) bao gồm xấp xỉ 36.8% các ví dụ trong D (Lưu ý: Một ví dụ thuộc tập D chỉ có thể **xuất hiện tối đa 1 lần** trong tập D_test)
- Phù hợp khi ta có một tập dữ liệu D có kích thước (rất) nhỏ

3. Lựa chọn tham số

- Nhiều phương pháp học máy thường có (tập) tham số (hyperparameters), buộc người dùng phải đưa giá trị vào.
  - Ridge regression: λ
  - Linear SVM: C
- Làm sao để lựa chọn giá trị tốt nhất cho các tham số?
  → *model selection*
- **Model selection:** *từ một tập học D, cần lựa chọn bộ tham số (model) trong phương pháp học A sao cho hệ thống được huấn luyện tốt nhất từ D*.
- Tập tối ưu (validation set) được sử dụng để tối ưu giá trị các tham số trong giải thuật học máy (thường lấy từ tập D)

  → Đối với một tham số, giá trị tối ưu là giá trị giúp sinh ra *hiệu năng cực đại đối với tập tối ưu*

Lựa chọn tham số: sử dụng Hold-out

- Cho trước tập quan sát D, ta lựa chọn tham số λ cho phương pháp học A như sau:
  - Chọn tập hữu hạn S mà chứa các giá trị tiềm năng cho λ.
  - Chọn độ đo P để đánh giá hiệu năng.
  - Chia D thành 2 tập rời nhau: D_{train} và $T_{validation}$
  - Với mỗi giá trị $\lambda \in S$:
    - Học A từ tập học D_{train} với tham số đầu vào λ. Đo hiệu năng trên tập $T_{validation}$ → thu được P_λ
  - Chọn λ^* mà có P_λ tốt nhất.
- Có thể học lại A từ D với tham số λ^* để thu được kết quả học tốt.
- Có thể thay Hold-out bằng kỹ thuật khác (e.g., sampling, cross-validation).

4. Đánh giá và lựa chọn mô hình

- Cho trước tập quan sát D, ta cần lựa chọn tham số λ (model selection) cho phương pháp học A và đánh giá (assessment) chất lượng tổng thể của A.
  - Chọn tập hữu hạn S mà chứa các giá trị tiềm năng cho λ.
  - Chọn độ đo P để đánh giá hiệu năng.
  - Chia tập D thành 3 tập rời nhau: D_{train}, $T_{validation}$, và T_{test}
  - Với mỗi giá trị $\lambda \in S$:
    - Học A từ tập học D_{train} với tham số đầu vào λ. Đo hiệu năng trên tập $T_{validation}$ → thu được P_λ
  - Chọn λ^* mà có P_λ tốt nhất.
  - Huấn luyện A trên tập $D_{train} \cup T_{validation}$, với tham số đầu vào λ^*.
  - Đo hiệu năng của hệ thống trên tập T_{test}.
- Có thể thay Hold-out bằng kỹ thuật khác (cross-validation).

5. Các tiêu chí đánh giá.

- **Tính chính xác (Accuracy)**
  - →Mức độ dự đoán (phân lớp) chính xác của hệ thống (đã được huấn luyện) đối với các ví dụ kiểm chứng (test instances)
- Tính hiệu quả (Efficiency)
  - →Chi phí về thời gian và tài nguyên (bộ nhớ) cần thiết cho việc huấn luyện và kiểm thử hệ thống
- Khả năng xử lý nhiễu (Robustness)
  - →Khả năng xử lý (chịu được) của hệ thống đối với các ví dụ nhiễu (lỗi) hoặc thiếu giá trị

5. Các tiêu chí đánh giá..

- Khả năng mở rộng (Scalability)
  - →Hiệu năng của hệ thống (vd: tốc độ học/phân loại) thay đổi như thế nào đối với kích thước của tập dữ liệu
- Khả năng diễn giải (Interpretability)
  - →Mức độ dễ hiểu (đối với người sử dụng) của các kết quả và hoạt động của hệ thống
- Mức độ phức tạp (Complexity)
  - →Mức độ phức tạp của mô hình hệ thống (hàm mục tiêu) học được

Tính chính xác (Accuracy)

- Đối với bài toán phân loại

  → Giá trị (kết quả) đầu ra của hệ thống là một giá trị định danh

$$Accuracy = \frac{1}{|D\_test|}\sum_{x \in D\_test} Identical(o(x), c(x)); \qquad Identical(a,b) = \begin{cases} 1, \text{if} & (a=b) \\ 0, \text{if otherwise} \end{cases}$$

  - x: Một ví dụ trong tập thử nghiệm D_test
  - $o(x)$: Giá trị đầu ra (phân lớp) bởi hệ thống đối với ví dụ x
  - $c(x)$: Phân lớp thực sự (đúng) đối với ví dụ x

- Đối với bài toán hồi quy (dự đoán)

  →Giá trị (kết quả) đầu ra của hệ thống là một giá trị số

$$Error = \frac{1}{|D\_test|}\sum_{x \in D\_test} Error(x); \qquad Error(x) = |d(x) - o(x)|$$

  - $o(x)$: Giá trị đầu ra (dự đoán) bởi hệ thống đối với ví dụ x
  - $d(x)$: Giá trị đầu ra thực sự (đúng) đối với ví dụ x
  - *Accuracy* là một hàm đảo (inverse function) đối với *Error*

Ma trận nhầm lẫn (Confusion matrix)

- Còn được gọi là Contingency Table
- **Chỉ được sử dụng đối với bài toán phân loại**
  - *Không thể áp dụng cho bài toán hồi quy (dự đoán)*

- ***TP$_i$***: Số lượng các ví dụ thuộc lớp c_i được phân loại chính xác vào lớp c_i
- ***FP$_i$***: Số lượng các ví dụ không thuộc lớp c_i bị phân loại nhầm vào lớp c_i
- ***TN$_i$***: Số lượng các ví dụ không thuộc lớp c_i được phân loại (chính xác)
- ***FN$_i$***: Số lượng các ví dụ thuộc lớp c_i- bị phân loại nhầm (vào các lớp khác c_i)

Lớp c_i		Được phân lớp **bởi hệ thống**	
		Thuộc	Ko thuộc
Phân lớp **thực sự (đúng)**	Thuộc	**TP$_i$**	**FN$_i$**
	Ko thuộc	**FP$_i$**	**TN$_i$**

Precision and Recall (1)

- Rất hay được sử dụng để đánh giá các hệ thống phân loại văn bản
- **Precision** đối với lớp c_i
  - → Tổng số các ví dụ thuộc lớp c_i được phân loại chính xác chia cho tổng số các ví dụ được phân loại vào lớp c_i

$$Precision(c_i) = \frac{TP_i}{TP_i + FP_i}$$

- **Recall** đối với lớp c_i
  - → Tổng số các ví dụ thuộc lớp c_i được phân loại chính xác chia cho tổng số các ví dụ thuộc lớp c_i

$$Recall(c_i) = \frac{TP_i}{TP_i + FN_i}$$

Precision and Recall (2)

- Làm thế nào để tính toán được giá trị Precision và Recall (một cách tổng thể) cho toàn bộ các lớp $C=\{c_i\}$?

- Trung bình vi mô (Micro-averaging)

$$Precision = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \qquad Recall = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

- Trung bình vĩ mô (Macro-averaging)

$$Precision = \frac{\sum_{i=1}^{|C|} Precision(c_i)}{|C|} \qquad Recall = \frac{\sum_{i=1}^{|C|} Recall(c_i)}{|C|}$$

F_1

- Tiêu chí đánh giá F_1 là sự kết hợp của 2 tiêu chí đánh giá *Precision* và *Recall*

$$F_1 = \frac{2.Precision.Recall}{Precision + Recall} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

- F_1 là một ***trung bình điều hòa (harmonic mean)*** của các tiêu chí *Precision* và *Recall*
  - F_1 có xu hướng lấy giá trị gần với giá trị nào nhỏ hơn giữa 2 giá trị Precision và Recall
  - F_1 có giá trị lớn nếu cả 2 giá trị Precision và Recall đều lớn