Lưu ý rằng hiệu suất của một phương pháp thường được cải thiện khi càng nhiều dữ liệu có sẵn.

Hiệu suất của một phương pháp phụ thuộc vào nhiều yếu tố:

¨ Phân phối dữ liệu

¨ Quy mô đào tạo

¨ Tính đại diện của dữ liệu đào tạo trên toàn bộ không gian

**2. Some evaluation techniques** (Một số kỹ thuật đánh giá)

Hold-out  
Stratified sampling  
Repeated hold-out  
Cross-validation  
¨ K-fold  
¨ Leave-one-out  
Bootstrap sampling

* **Hold out**

Cách phổ biến nhất để thực hiện 1 mô hình là chia tập dữ liệu ban đầu mà ta thu thập được ra 2 tập D\_test & D\_train, tỉ lệ chia có thể là

* D\_test =1/3 D
* D\_train=2/3 D

Và lưu ý cách này thì sẽ thường được thực hiện trên tập dữ liệu lớn, còn tập dữ liệu nhỏ thì sẽ không phù hợp bởi vì nó có thể không chính xác do độ nhiễu khá lớn

*Lưu ý:* Không có phiên bản Dtest nào được sử dụng trong giai đoạn huấn luyện.

¨ Không có phiên bản Dtrain nào được sử dụng trong giai đoạn thử nghiệm.

* **Stratified sampling**

We randomly split each class of D into 2 parts: one is for Dtrain, and the other is for Dtest.

¨ for each class:

¡ Note that this technique cannot be applied to regression and unsupervised learning

* **Repeated hold-out**

Chúng ta có thể giữ lại nhiều lần, sau đó thực hiện kết quả trung bình.

¨ Lặp lại tạm dừng n lần. Lần thứ i sẽ biểu diễn kết quả p. Dữ liệu đào tạo cho mỗi lần tạm dừng phải khác nhau từ nhau.

¨ Lấy giá trị trung bình p = mean(p1,…, pn) làm chất lượng cuối cùng.

* **Cross-validation**

Trong trường hợp giữ lại nhiều lần: có sự chồng chéo giữa hai tập dữ liệu huấn luyện/kiểm tra. Nó có thể là dư thừa.

Xác thực chéo K-fold:

¨ Chia D thành K phần bằng nhau không trùng nhau.

¨ Chạy K (folds): ở mỗi lần chạy, một phần được sử dụng để kiểm tra và các phần còn lại được sử dụng để đào tạo.

¨ Lấy giá trị trung bình làm chất lượng cuối cùng từ K lần chạy riêng lẻ.

Lựa chọn phổ biến của K: 10 hoặc 5

Thật hữu ích khi kết hợp kỹ thuật này với phân tầng lấy mẫu.

*kỹ thuật này phù hợp với các bộ dữ liệu nhỏ/trung bình.*

* **Leave-one-out cross-validation**

Đó là xác thực chéo K-fold khi K = |D|.

¨ Mỗi bộ thử nghiệm chỉ bao gồm một phiên bản từ D.

¨ Phần còn lại dành cho đào tạo.

Vì vậy, tất cả các trường hợp quan sát được khai thác nhiều như khả thi.

Không có sự ngẫu nhiên nào xuất hiện.

Nhưng nó đắt tiền, và do đó phù hợp với quy mô bộ dữ liệu nhỏ.

* **Bootstrap sampling**

Các phương thức trước đây không cho phép lặp lại một thể hiện trong bất kỳ phần đào tạo nào.

Lấy mẫu Bootstrap:

¨ Giả sử D có n thể hiện.

¨ Xây dựng Dtrain bằng cách lấy mẫu ngẫu nhiên (có thay thế/lặp lại) n trường hợp từ D.

¨ Dtrain được sử dụng cho giai đoạn huấn luyện.

¨ Dtest = D\Dtrain được sử dụng để kiểm tra chất lượng.

¨ Lưu ý rằng 𝐷

𝑡𝑒𝑠𝑡 = {𝑧 ∈ 𝐷: 𝑧 ∉ 𝐷𝑡𝑟𝑎𝑖𝑛}

Có thể thấy rằng Dtrain chứa gần 63,2% khác nhau trường hợp của D. 36,8% của D được sử dụng để thử nghiệm.

*Kỹ thuật này phù hợp với các tập dữ liệu nhỏ*

**3.**

λ trong hồi quy Ridge; C trong SVM tuyến tính

Bộ xác thực Tvalid thường được sử dụng để tìm một cài đặt tốt.

¨ Nó thường là tập con của D.

¨ Cài đặt tốt sẽ giúp chức năng đã học dự đoán tốt trên Tvalid.

à chúng ta đang xấp xỉ lỗi tổng quát hóa trên toàn bộ dữ liệu không gian bằng cách chỉ sử dụng một bộ nhỏ Tvalid.

* Use hold out

Cho một tập dữ liệu được quan sát D, chúng ta có thể chọn một giá trị tốt

cho siêu tham số λ như sau:

¨ Chọn một tập hợp hữu hạn S chứa tất cả các giá trị tiềm năng của λ.

¨ Chọn thước đo hiệu suất P.

¨ Chia ngẫu nhiên D thành 2 tập con không trùng nhau: Dtrain và Tvalid

¨ Với mỗi λ Î S: huấn luyện hệ thống đã cho Dtrain và λ. đo lường chất lượng trên Tvalid để lấy Pλ.

¨ Chọn λ\* tốt nhất tương ứng với Pλ tốt nhất.

Việc học lại từ D với λ\* để nhận được một chức năng tốt hơn.

Giữ nguyên có thể được thay thế bằng các kỹ thuật khác, ví dụ: lấy mẫu, xác thực chéo.