**Bài thực hành về nhà**

*Nguyễn Hồng Đăng*

**Nội dung:**

1. Thử nghiệm thuật toán trích xuất đặc trưng BoW với số lượng từ trong từ điển khác nhau. Vẽ biểu đồ độ chính xác trên tập test ứng với từng trường hợp.

2. Sử dụng thuật toán Grid search kết hợp với Cross Validation tìm tham số tốt nhất cho mô hình SVM.

3. Thay SVM bằng các bộ phân loại: K-Nearest Neighbors, Decision tree, Random forest. So sánh các kết quả và đưa ra nhận xét.

4. Thử nghiệm bài toán Phân loại cảm xúc FERC-2013: <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge>

**1. Thử nghiệm thuật toán trích xuất đặc trưng BoW với số lượng từ trong từ điển khác nhau. Vẽ biểu đồ độ chính xác trên tập test ứng với từng trường hợp.**

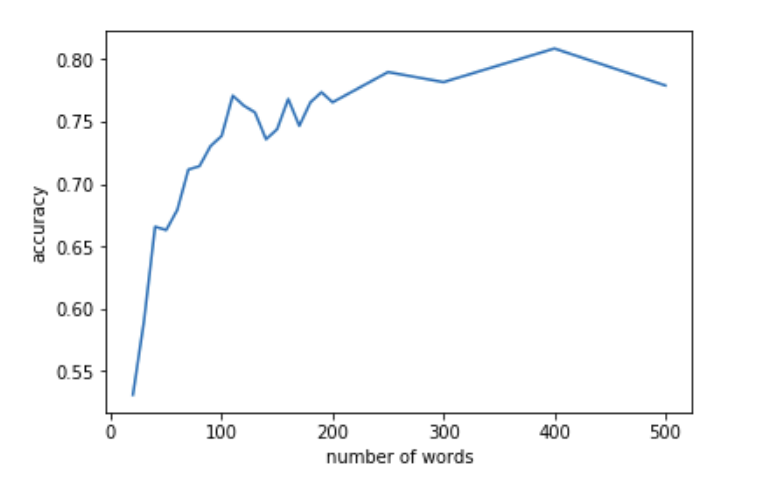
**1.1. Thử nghiệm với số lượng từ trong từ điển thuộc tập hợp:**

- Với mong muốn số chiều dữ liệu đầu vào là thấp (giảm thời gian tính toán), đồng thời phù hợp với mô hình trong đó số lớp là thấp (5 lớp) , ta kì vọng số từ trong từ điển là <200.

- Lựa chọn các giá trị number of words thử nghiệm như sau:

{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 250, 300, 400, 500}

**1.2. Kết quả:**



**1.3. Nhận xét:**

Độ chính xác biến thiên tăng giảm theo đường gấp khúc. Trong đó có 2 giá trị đáng chú ý:

* Tổng quan, độ chính xác đạt cực đại là 0.80863 tại giá trị number of words= 400.
* Trong khoảng [0,150], đạt cực đại địa phương (local maximum) là 0.77089 tại number of words= 110.

**2. Sử dụng thuật toán Grid search kết hợp với Cross Validation tìm tham số tốt nhất cho mô hình SVM.**

- Theo đánh giá phía trên, ta lựa chọnsố lượng từ trong từ điển là 110. Các tham số mà ta quan tâm đối với model SVM trong trường hợp này là: C, kernel, decision\_function\_shape.

- Sử dụng thuật toán grid search với parameter grid như sau:

param\_grid={'C':[10,30,100,300,1000,3000,10000],

'kernel':['rbf','linear','poly','sigmoid'],

'decision\_function\_shape':['ovr','ovo']}

**Kết quả:**

- Bộ tham số tối ưu:

{'C': 10, 'decision\_function\_shape': 'ovr', 'kernel': 'rbf'}

- Accuracy trên tập validation: 0.8280512474713418

- Accuracy trên tập test: 0.8280512474713418

*Như vậy, độ chính xác đã được cải thiện 1.62% so với bộ tham số trước.*

**3. Thay SVM bằng các bộ phân loại: K-Nearest Neighbors, Decision tree, Random forest. So sánh các kết quả và đưa ra nhận xét.**

Cách tiếp cận chung: sử dụng grid search để hiệu chỉnh tìm ra tham số tốt cho các mô hình nói trên, so sánh kết quả trên tập test.

**3.1. K-Nearest Neighbors**

- parameter grid sử dụng:

param\_grid={'n\_neighbors':[11,21,31,41,51,71,91,121],

'weights':['uniform','distance']}

**Kết quả:**

- Best parameter found:{'n\_neighbors': 21, 'weights': 'distance'}

- Best score on validation set: 0.466621712744437

- Accuracy on test set: 0.4636118598382749

**Nhận xét**: thuật toán đơn giản "không học gì" hoạt động rất kém đối với bài toán.

**3.2. Decision tree**

Sử dụng decision tree với tham số mặc định, thu được kết quả:

- Accuracy on test set: 0.6199460916442049

**3.3. Random forest**

- Ta quan tâm tới số lượng decision tree sử dụng trong mô hình, parameter grid sử dụng:

param\_grid={'n\_estimators':[10,30,50,70,100,150,200]}

**Kết quả:**

- Best parameter found:{'n\_estimators': 200}

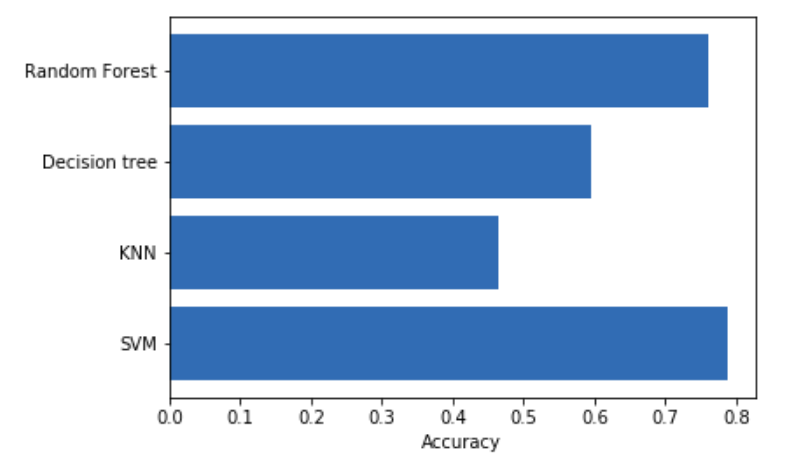
- Best score on validation set: 0.75590020229265

- Accuracy on test set: 0.7601078167115903

**Nhận xét:**

Một cách dễ hiểu, mô hình cho độ chính xác càng cao khi số lượng decision trees sử dụng trong mô hình càng cao (trong trường hợp này là 200).

**3.4. So sánh kết quả:**



- Như đã phân tích, KNN hoạt động rất kém.

- Random Forest sử dụng nhiều decision tree cho kết quả cao hơn rất nhiều Decision tree (khoảng 15%).

- SVM model hoạt động tốt nhất vì tính phù hợp của nó, và hơn nữa, số lượng từ trong từ điển là 110 cũng được tối ưu sử dụng đánh giá trên mô hình SVM.