

# Máy học

## Đề án vấn đáp: Phân lớp ảnh chữ số viết tay bằng SVM

---

### 1 Nội dung đề án

#### 1.1 Tìm hiểu về lý thuyết mô hình SVM (Support Vector Machine)

##### Các tài liệu

- [Các video bài giảng](#): 14 – SVM (mình đã giảng ở buổi học bù), 15 – Kernel Methods, 16 – RBFs (trong đó, video 14 và 15 là hai video chính về SVM; video 16 xem thêm để hiểu hơn về Gaussian/RBF kernel)
- Tài liệu (dễ đọc) về việc chuyển từ “primal form” sang “dual form”: Mục 5 – “Lagrange duality” trong file “Lagrange.pdf” đính kèm

##### Các ý chính cần nắm

- Phân 2 lớp
  - Dữ liệu khả tách tuyến tính
    - Tập hypothesis của SVM?
    - Thuật toán học của SVM? (SVM muốn tìm “siêu phẳng” phân lớp như thế nào?)
    - “Support vector” là gì? “Support vector” liên quan như thế nào đến khả năng tổng quát hóa của SVM?
  - Dữ liệu không khả tách tuyến tính
    - SVM dùng soft-margin và kernel để giải quyết trường hợp dữ liệu không khả tách tuyến tính như thế nào?
    - Siêu tham số  $C$  trong soft-margin ảnh hưởng như thế nào đến việc học?
    - Siêu tham số  $\gamma$  trong Gaussian/RBF kernel ảnh hưởng như thế nào đến việc học?
- Phân  $K$  lớp ( $K > 2$ )
  - Từ SVM phân 2 lớp, làm thế nào để phân được  $K$  lớp? Gợi ý: “one-against-one” là phương pháp thường được sử dụng trong SVM ([xem thêm](#))

#### 1.2 Huấn luyện SVM để phân lớp ảnh chữ số viết tay

##### Mô tả dữ liệu

Bộ dữ liệu được sử dụng là bộ MNIST. Mỗi mẫu (example) trong bộ MNIST gồm: input là ảnh chữ số viết tay grayscale có kích thước  $28 \times 28$  (như vậy, véc-tơ input sẽ có số chiều là  $28 \times 28 = 784$ ), “correct output”  $\in \{0, 1, \dots, 9\}$  cho biết chữ số tương ứng của ảnh (như vậy, sẽ có tất cả 10 lớp). Dưới đây là một số mẫu trong bộ MNIST:



Bạn download file dữ liệu “mnist.pkl.gz” đính kèm. Trong file dữ liệu này, các giá trị pixel đã được scale về  $[0, 1]$  bằng cách chia cho 255. Hơn nữa, người ta cũng đã chia cho bạn 3 tập:

- Tập training: gồm 50.000 mẫu
- Tập validation: gồm 10.000 mẫu
- Tập test: gồm 10.000 mẫu

Bạn xem đoạn code đọc file dữ liệu này trong file “ReadMNIST.ipynb” đính kèm.

### Cài đặt SVM

Với level hiện tại, bạn không nên cài đặt SVM từ A đến Z. Bạn sẽ sử dụng SVM đã được cài đặt sẵn trong thư viện [scikit-learn](#) (bạn đọc document để xem cách sử dụng; rất dễ). Thư viện này đã được cài đặt cho bạn khi bạn cài đặt gói Anaconda.

### Huấn luyện SVM

- Dùng linear kernel (hay nói cách khác là không dùng kernel)
  - Thử nghiệm với các giá trị khác nhau của siêu tham số  $C$ ; với mỗi giá trị  $C$ , ghi nhận lại: độ lỗi trên tập training, độ lỗi trên tập validation, thời gian huấn luyện
  - Bình luận về kết quả
- Dùng Gaussian/RBF kernel
  - Thử nghiệm với các giá trị khác nhau của siêu tham số  $C$  và  $\gamma$ ; với mỗi giá trị  $C$  và  $\gamma$ , ghi nhận lại: độ lỗi trên tập training, độ lỗi trên tập validation, thời gian huấn luyện
  - Bình luận về kết quả.
- Chọn hàm dự đoán có độ lỗi nhỏ nhất trên tập validation là hàm dự đoán cuối cùng.

### Đánh giá SVM

Với hàm dự đoán cuối cùng ở trên, bạn đánh giá hàm dự đoán này bằng cách đo độ lỗi trên tập test. Bạn có thể xem các kết quả của người ta [ở đây](#).

## 2 Vấn đáp và nộp bài trên moodle

### 2.1 Vấn đáp

- Thời gian và địa điểm vấn đáp: theo lịch của trường
- Các bạn cần đi vấn đáp đầy đủ để ký tên vào danh sách thi
- Khi đi vấn đáp, mỗi nhóm cần nộp cho mình bản in của file báo cáo
  - Ở phần header của file báo cáo, ghi MSSV và họ tên của các thành viên trong nhóm
  - Nội dung file báo cáo (ứng với phần “huấn luyện SVM” và “đánh giá SVM” trong mục 1.2 ở trên): ghi nhận lại các kết quả (nên dùng bảng biểu, đồ thị), các phân tích, nhận xét
  - Báo cáo nên trình bày rõ ràng, ngắn gọn (từ 2-3 trang)

### 2.2 Nộp bài trên moodle

Bạn sẽ nộp bài (gồm báo cáo + code) trên moodle sau buổi vấn đáp.