ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

KỸ THUẬT LẬP TRÌNH (CO1011)

Đề bài tập lớn 01 (Phiên bản 3.0)

CBGD: TS. LÊ Thành Sách
TS. NGUYỄN Đức Dũng

Tp. Hồ Chí Minh, Tháng 03/2017

Mục lục

1 Lịch sử các phiên bản

Ngày	Phiên bản	Nội dung
29-03-2017	**1.0**	Công bố đề bài BTL Số 01
12-04-2017 (15H)	**2.0**	Cập nhật các điểm sau: **(1)** Cách chia dữ liệu vào tập **TRN** và **TST** - Xem Phần 4.2 **(2)** Bổ sung phần **Hướng dẫn thực hiện** - Xem Phần 5 **(3)** Dời hạn chót nộp bài - Xem Phần 7
12-04-2017 (20H)	**2.1**	Cập nhật các điểm sau: **(1)** Cập nhật ký hiệu số mẫu cho các tập **TRN** và **TST** - Xem Phần 4.2
18-04-2017	**2.2**	Cập nhật các điểm sau: **(1)** Cập nhật công thức xác định V_{min} và V_{max} - Xem Phương trình (9) và (10) **(2)** Giải thích cách tính biểu đồ tần suất - Xem Phần 4.2.1
24-04-2017	**3.0**	Cập nhật các điểm sau: **(1)** Cập nhật yêu cầu định dạng tập tin đầu ra: bỏ 3 dòng văn bản ở đầu, độ rộng và độ định chính xác khác với Phiên bản trước - Xem Phần 4.2 **(2)** Cập nhật cách tính điểm - Xem Phần 6 **(3)** Cập nhật thời gian nộp bài- Xem Phần 7 **(4)** Bổ sung hướng dẫn nộp bài - Xem Phần 8

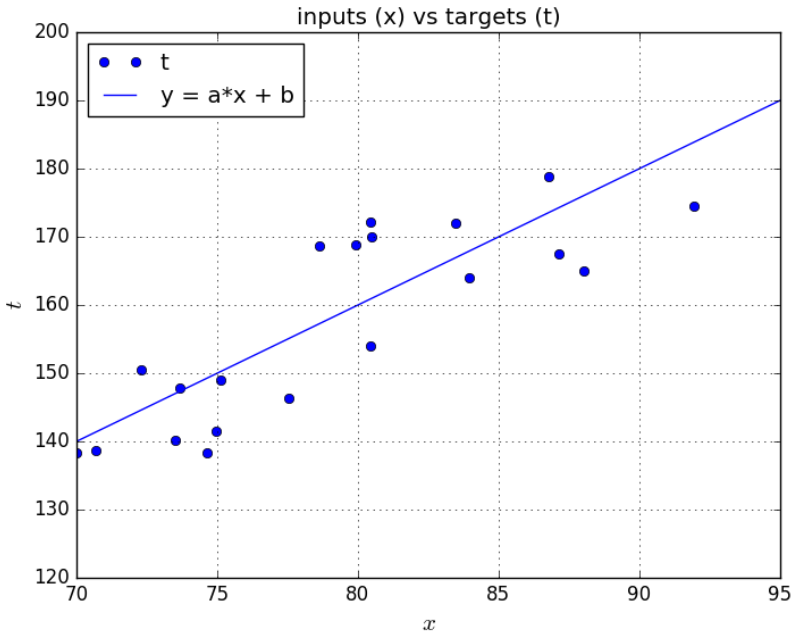
2 Giới thiệu bài toán

Các nhà nghiên cứu trong lĩnh vực sức khoẻ cộng đồng đã quan sát và thấy rằng chiều cao của một người trưởng thành (lúc 18 tuổi) phụ thuộc lớn vào chiều cao của chính người đó lúc 2 tuổi. Việc xác định có hay không quan hệ như vậy có ý nghĩa quan trọng. Mỗi khi có quan hệ đó, người ta tập trung hơn trong việc sớm cải thiện chiều cao của trẻ em ở độ tuổi lên 2, cũng như có thể dự báo chiều cao của một người lúc trưởng thành. Do vậy, nhóm nghiên cứu đã thực hiện thí nghiệm để kiểm chứng quan điểm này.

Để thực hiện thí nghiệm, người ta theo dõi chiều cao của N (đủ lớn) đối tượng (cá thể) từ lúc 2 tuổi đến lúc 18 tuổi, bằng cách đo chiều cao của từng đối tượng lúc 2 tuổi và lúc trưởng thành. Dựa trên số liệu thu thập được, nhóm nghiên cứu dùng một kỹ thuật được gọi là "**tối ưu hoá**" (optimization) để xác định mô hình dự báo chiều cao và kiểm chứng quan điểm trên. Kỹ thuật này sẽ được trình bày trong phần theo sau.

Gọi x là chiều cao của một đối tượng lúc 2 tuổi, và t là chiều cao của chính người đó lúc 18 tuổi. Cả x và t được đo cho từng đối tượng. Sau khi thu thập số liệu (x, t) của N đối tượng trong thí nghiệm, các nhà nghiên cứu đã quan sát sự phân phối dữ liệu của N đối tượng. Từ quan sát đó, họ đi đến một giả thiết cụ thể hơn, đó là, chiều cao lúc trưởng thành (y) có thể được dự báo từ chiều cao lúc 2 tuổi (x) theo một mô hình tuyến tính như trong Phương trình (1). Do đó, nhiệm vụ của nhóm nghiên cứu là xác định các hệ số a và b của mô hình tuyến tính trong Phương trình (1). Các điểm (x, t) đo đạc được và đường dự báo y được minh hoạ như Hình 1.

$$y = a \times x + b \tag{1}$$



Hình 1: *Các điểm (x, t) trong thí nghiệm và đường dự báo $y = a \times x + b$*

3 Phương pháp tối ưu hoá

3.1 Sai số dự báo

Với chiều cao x (lúc 2 tuổi) ở đầu vào, chiều cao lúc trưởng thành y là giá trị dự báo được tính từ Phương trình (1). Tuy nhiên, với các đối tượng trong thí nghiệm, chiều cao lúc trưởng thành thực sự đã được đo đạc rồi, đó là t. Chiều cao thực sự này có thể sai lệch (nhỏ hơn hoặc lớn hơn) với chiều cao tính được từ mô hình dự báo y. Độ sai số được tính là $e = t - y$. Giá trị e có thể âm hay dương. Trong kỹ thuật, sai số thường là số không âm, để biểu thị độ lệch lớn hay nhỏ. Một trong các hàm có tính chất đó là "**tổng bình phương sai số**" (sum of squared errors, SSE). Hàm này được định nghĩa trong Phương trình (2).

$$\begin{aligned} L(a,b) &= \frac{1}{2} \times \sum_{i=1}^{i=N} (t_i - y_i)^2 \\ &= \frac{1}{2} \times \sum_{i=1}^{i=N} (t_i - a \times x_i - b)^2 \end{aligned} \tag{2}$$

Vì x và t là các giá trị đã đo được từ N cá thể trong thí nghiệm, trong Phương trình (1), a và b là hai biến cần xác định. Nghĩa là, hàm sai số phụ trên các biến cần xác định là a và b. Hằng số $\frac{1}{2}$ chỉ là một hệ số, nó được đưa vào để làm đơn giản biểu thức đạo hàm sau này.

3.2 Đạo hàm, gradient và sự tăng-giảm của hàm số

Một cách hình thức, các số liệu sau đây là đã biết trước khi dùng giải thuật tối ưu hoá.

1. N bộ số liệu (x, t), là những số đo cho N cá thể trong thí nghiệm.

2. Mô hình dự báo đã biết trước là có dạng như Phương trình (1). Trong mô hình này, có hai thông số cần xác định là a và b.

3. Hàm sai số để biểu diễn tổng sai số dự báo đã biết trước, đó là $L(a,b)$, được định nghĩa trong Phương trình (2).

Giải thuật tối ưu hoá được trình bày ở đây là "**đổ dốc theo đạo hàm**" (gradient descent - GD). GD có mục tiêu là tìm ra bộ tham số a và b mà tại đó hàm $L(a,b)$ đạt giá trị cực tiểu cục bộ. Cực tiểu cục bộ nghĩa là cực tiểu nhưng có thể chưa phải là nhỏ nhất trong toàn bộ không gian tham số (a và b).

Giải thuật này dựa trên một tính chất của đạo hàm như sau. Nếu đạo hàm của một hàm $f(x)$ nào đó là dương tại điểm x_0 bất kỳ, thì hàm $f(x)$ là hàm tăng tại điểm x_0. Nghĩa là $f(x_0 - \alpha \times f^{'}(x_0)) < f(x_0)$, với $\alpha > 0$ và đủ nhỏ. Tương tự, nếu đạo hàm âm thì hàm số giảm, nghĩa là $f(x_0 + \alpha \times f^{'}(x_0)) < f(x_0)$. Tóm lại, **trên trục biến số x, tại điểm x_0, nếu ta di chuyển theo hướng ngược lại với dấu (hướng) của đạo hàm $f^{'}(x_0)$ thì hàm số sẽ giảm dần**.

Áp dụng tính chất này vào hàm $L(a,b)$ ta cũng thu được những quan hệ tương tự. Với hàm nhiều biến sự tăng giảm của hàm số so với đạo hàm được diễn dịch sao cho tất cả các biến có quan hệ nhau. Quan hệ này phụ thuộc trên một số khái niệm được dẫn ra sau đây, do đó, nó sẽ được trình bày sau những khái niệm đó.

Đạo hàm riêng:

Từ Phương trình (2), đạo hàm riêng của $L(a,b)$ theo a và b là:

$$\frac{\partial L}{\partial a} = \sum_{i=1}^{i=N} (a \times x_i + b - t_i) \times x_i \tag{3}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{i=N} (a \times x_i + b - t_i) \tag{4}$$

Gradient vector:

Gradient vector là một vector chứa hai thành phần đạo hàm riêng của $L(a,b)$ theo biến a và b. Gradient vector trong tài liệu này là $\mathbf{g} = [\frac{\partial L}{\partial a}, \frac{\partial L}{\partial b}]^T$.

Điểm trong mặt phẳng (a,b):

Gọi P_0 là một điểm nào đó trong mặt phẳng (a,b). Điểm P_0 có toạ độ (a_0, b_0). Ta có thể biểu diễn P_0 là vector ở dạng $\mathbf{P_0} = [a_0, b_0]^T$

Quan hệ giữa sự tăng-giảm của $L(a,b)$ với gradient vector:

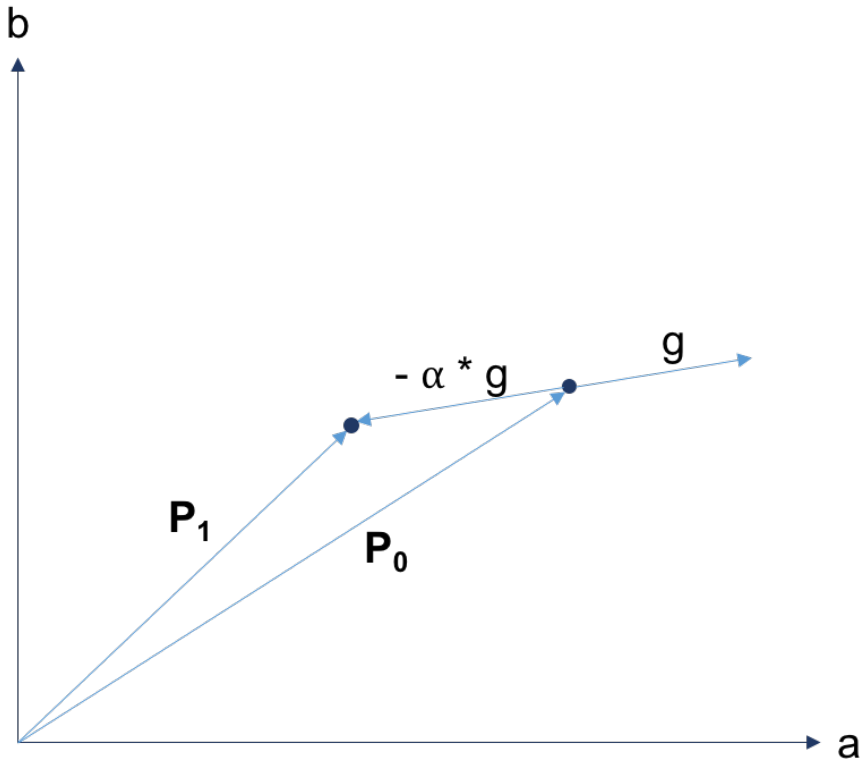
Tương tự như đạo hàm, gradient vector chỉ ra hướng mà di chuyển theo đó thì hàm số tăng. Chúng ta cần tìm cực tiểu, do đó, chúng ta di chuyển theo hướng ngược lại với gradient vector. Việc di chuyển "ngược" này đạt được bằng cách trừ một lượng $\alpha \times \mathbf{g}$, như trong Phương trình (5), được minh hoạ trong Hình 2. Một điểm quan trọng ở đây là, chúng ta cần đến hướng của gradient vector để chúng ta đi ngược một lượng α, và chúng ta hoàn toàn không cần đến độ lớn của gradient vector. Do vậy, gradient vector cần được chuẩn hoá để có độ lớn là đơn vị. Chuẩn hoá bằng cách chia các thành phần của vector cho độ lớn của vector.

Phương trình 5 có thể thực hiện thông qua Phương trình (6) và (7) cho các thành phần trong vector. Ở đó, g_a và g_b là hai thành phần của gradient vector $\mathbf{g}$.

$$\mathbf{P_1} = \mathbf{P_0} - \alpha \times \mathbf{g} \tag{5}$$

$$a_1 = a_0 - \alpha \times g_a \tag{6}$$

$$b_1 = b_0 - \alpha \times g_b \tag{7}$$



Hình 2: *Gradient vector* $\mathbf{g}$ *và việc di chuyển ngược hướng gradient vector một lượng* $-\alpha \times \mathbf{g}$

3.3 Giải thuật tối ưu hoá "Gradient Descent"

Trong phần giải thích ở trên, khi cho trước điểm P_0, nếu P_1 được xác định từ Phương trình (5) hay (6) và 7 thì ta có $L(P_1) < L(P_0)$. Tiếp tục, ta có thể cho P_1 đóng vai trò của P_0 thì chúng ta còn có thể cải thiện giá trị của hàm $L(a, b)$ để nó tiến đến điểm cực trị nữa.

Do vậy, để tìm cực trị chúng ta có thể sử dụng phương pháp lặp như sau:

1. Gọi P là điểm nào đó trong mặt phẳng (a, b). Điểm này có thể gán giá trị nào đó trước, hoặc ngẫu nhiên.
2. Gán giá trị cho α, giá trị thực dương.
3. Lặp cho đến một điều kiện dừng nào đó, thực hiện các bước sau:
   - Tính toán giá trị dự báo y cho tất cả các mẫu số liệu (N mẫu).
   - Tính gradient vector từ Phương trình (3) và (4).
   - Chuẩn hoá gradient vector và thu được $\mathbf{g} = [g_a, g_b]^T$
   - Cập nhật lại toạ độ điểm P dùng Phương trình (6) và (7).

Những điểm cần chú ý trong giải thuật nêu trên:

- Điểm bắt đầu P: trong trường hợp tổng quát hơn, hàm sai số ($L(a, b)$) có cực trị toàn cục và nhiều cực trị cục bộ, việc chạy thử nghiệm với các điểm bắt đầu khác nhau có thể là giải pháp để chọn điểm bắt đầu phụ hợp, và nó cho ra sai số dự báo tốt hơn. Ở bài tập lớn này, sinh viên sẽ chọn gán điểm bắt đầu từ thông số đọc từ tập tin đầu vào.

- Hệ số α: hệ số này có tên gọi là **hệ số học** (learning rate). Giá trị này có những tính chất sau. Nếu α quá nhỏ thì quá trình điểm P chạy về điểm cực trị rất chậm, vì mỗi lần chỉ có sự thay đổi nhỏ. Ngược lại, quá trình hội tụ có thể nhanh, nhưng cũng có thể vượt qua luôn điểm cực trị và sẽ không gặp điểm cực trị được. Trường hợp này P chỉ chạy qua và chạy lại điểm cực trị. Trong bài tập lớn, hệ số học cũng cho sẵn trong tập tin đầu vào.
- Điều kiện dừng của giải thuật. Có nhiều cách, trong bài tập lớn, điều kiện dừng là chạy đúng số lần lặp được quy định. Số này cho trong tập tin đầu vào.

4 Yêu cầu của Bài tập lớn 01

Ở bài tập lớn này, sinh viên được yêu cầu viết một chương trình để thực hiện các công việc sau đây: đọc dữ liệu đầu vào và đánh giá mô hình dự đoán trong Phương trình (1) dựa vào các số liệu đọc được. Các phần theo sau sẽ mô tả các công việc chi tiết hơn.

4.1 Đọc dữ liệu đầu vào

Tập in đầu vào là: "**assignment1.input.txt**". Tập tin này chứa các tham số phục vụ quá trình tối ưu hoá như sau:

- **trn_params.num_iterations**: số lần lặp trong giải thuật tối ưu hoá "đổ dốc theo đạo hàm" nêu ở Phần 3.3.
- **trn_params.learning_rate**: hệ số học
- **trn_params.start_a và trn_params.start_b**: điểm bắt đầu
- **trn_params.num_folds**: cách chia tập dữ liệu để đánh giá, sẽ trình bày ở phần sau.

Theo sau phần các thông số đã mô tả là phần dữ liệu để đánh giá, đó là các cặp giá trị (x, t) được cho trong phần "Data Samples" của tập tin đầu vào. Hàng dữ liệu đầu tiên bắt đầu dòng thứ 12 (tính từ 1). Mỗi dòng gồm 2 con số x và t, cách nhau khoảng trống. Số mẫu dữ liệu (số hàng) lớn nhất là một hằng số **NUM_SAMPLES = 20000**.

Một tập tin "**assignment1.input.txt**" mẫu được cho như sau:

```
------------------------------------------------------------
Traning and Validation Parameters
------------------------------------------------------------
trn_params.num_iterations:                        50
trn_params.learning_rate:                        0.1
trn_params.start_a:                                0
trn_params.start_b:                                0
trn_params.num_folds:                              3
------------------------------------------------------------
Data samples
------------------------------------------------------------
76.87       153.78
83.76       167.48
```

4.2 Đánh giá mô hình và kết xuất

Gọi M là số mẫu dữ liệu được đọc vào. Mỗi mẫu bao gồm cặp giá trị x và t. Gọi K là giá trị đọc được từ thông số **trn_params.num_folds**.

Việc đánh giá mô hình dự báo ở Phương trình (1) được thực hiện theo hướng dẫn sau đây.

- Chia M mẫu số liệu vào K ngăn (hay tập hợp). Vậy mỗi ngăn có trung bình $D = M/K$ (chia nguyên) mẫu dữ liệu; ngăn cuối cùng (ngăn thứ K) có thể chứa số mẫu dữ liệu **nhiều hơn** các ngăn khác nếu phép chia M cho K không chẵn. Trong bài tập lớn này, sau khi đọc được M mẫu, sinh viên lưu vào một mảng **theo đúng thứ tự** như trong tập tin đầu vào. Như vậy, ngăn số 1 chứa các mẫu số liệu có **chỉ số** từ **0** đến **(D − 1)**, ngăn số 2 chứa các mẫu số liệu có **chỉ số** từ **D** đến **(2D − 1)**, v.v. **Ngăn cuối cùng chứa tất cả các mẫu còn lại sau khi đã đưa các mẫu vào $K - 1$ ngăn trước đó**.

- Chương trình thực hiện K lần các công việc xác định mô hình và kiểm định mô hình theo hướng dẫn sau đây. Các lần này thực hiện này được đánh chỉ số từ 1 đến K. Ở lần thứ k ($k = 1, 2, .., K$), chương trình tiến hành các công việc sau.

  1. Chương trình tách dữ liệu đầu vào thành 2 tập hợp, được gọi là tập **TRN** và tập **TST**. Tập **TST** gồm tất cả các mẫu dữ liệu trong ngăn số k. Số mẫu dữ liệu có trong tập **TST** là D. Theo giải thích ở trên, $D = M/K$ cho các ngăn từ 1 đến $(K - 1)$; $D \geq M/K$ cho ngăn cuối cùng. Tập **TRN** bao gồm tất cả các mẫu dữ liệu từ $(K - 1)$ ngăn còn lại. Gọi mẫu có trong tập **TRN** là N. Ta có, $N = M - D$. **Lưu ý**, với mỗi giá trị k khác nhau ($k = 1, 2, .., K$), ta có hai tập **TRN** và **TST** khác nhau.
  2. Chương trình sử dụng các mẫu dữ liệu trong tập **TRN** để xác định các hệ số của mô hình dự báo trong Phương trình (1), theo giải thuật được trình bày trong phần 3.3.
  3. Chương trình dùng mô hình thu được để xác định sai số dự báo trung bình cho các mẫu trong tập **TST**. Sai số này được tính theo Phương trình (8).
  4. Cùng với việc tính ra sai số dự báo chung ở bước trên, chương trình xây dựng biểu đồ tần suất của độ lệch e_i của các mẫu trong tập **TST**. e_i được định nghĩa là $e_i = y_i - t_i$. Ở đó, y_i và t_i là giá trị dự báo và giá trị đo được của cá thể thứ i trong tập **TST**, $i = 1, 2, .., D$. Giá trị e_i có thể âm hoặc dương. Việc tính toán này được trình bày trong Phần 4.2.1.
  5. Chương trình xuất ra tập tin đầu ra một hàng số liệu cho mỗi giá trị k. Hàng số liệu này gồm các số sau đây, từ trái sang phải: hai giá trị của a và b, sai số dự báo của D mẫu số liệu tính ở trên, và 10 giá trị biểu diễn biểu đồ tần suất của các độ lệch. Như vậy, có 13 giá trị trên cùng một hàng. Mỗi giá trị là một số thực định dạng dấu chấm cố định (fixed), độ rộng là **10**, có **năm** số đứng sau dấu chấm thập phân và canh lề phải. Vì chương trình thực hiện K lần kiểm định như vậy, nên trong tập tin đầu ra có K dòng số liệu như đã mô tả (chưa kể các dòng văn bản, xem tập tin mẫu).

$$E_{rmsd} = \sqrt{\frac{1}{D} \times \sum_{i=1}^{i=D} (a \times x_i + b - t_i)^2} \tag{8}$$

$$V_{min} = \bar{e} - 3 \times \sigma \tag{9}$$

$$V_{max} = \bar{e} + 3 \times \sigma \tag{10}$$

$$\sigma = \sqrt{\frac{1}{D} \times \sum_{i=1}^{i=D} (e_i - \bar{e})^2} \tag{11}$$

$$\bar{e} = \frac{1}{D} \times \sum_{i=1}^{i=D} e_i \tag{12}$$

Tập tin đầu ra có tên là "**assigment1.output.txt**". Tập tin này có K dòng số liệu, mỗi dòng có định dạng như mô tả ở trên (Phần 4.2, Bước 5.) và được minh hoạ ở dãy số theo sau.

```
1.99995   0.51018   9.14897 ... (Còn 10 số nữa)
```

4.2.1 Xây dựng biểu đồ tần suất các lỗi

Giả sử chúng ta đã xác định các tham số của mô hình dự báo trong Phương trình 1, khi có D mẫu dữ kiểm tra là (x_i, t_i), với $i = 1, 2, .., D$. Chúng ta thực hiện các bước sau đây để xây dựng biểu đồ tần suất các lỗi.

1. Tính toán các lỗi $e_i = y_i - t_i$, với $i = 1, 2, .., D$. Nhắc lại, y_i và t_i là giá trị dự báo và giá trị đo được của cá thể thứ i trong tập **TST**. Giá trị e_i có thể âm hoặc dương.

2. Xác định giá trị trung bình $\bar{e}$ của các e_i và phương sai σ, theo Phương trình (12) và (11).

3. Xác định cận trái và phải của các giá trị e_i sẽ được xem xét để xây dựng biểu đồ, theo Phương trình (9) và (10). Lưu ý, giá trị cận trái V_{min} lệch về phía trái so với $\bar{e}$ một lượng $3 \times \sigma$. Còn giá trị V_{max} lệch về phía phải của $\bar{e}$ một lượng $3 \times \sigma$.

4. Với đoạn $[V_{min}, V_{max}]$ vừa xác định, ta chia đoạn này thành 10 khoảng có độ dài bằng nhau. Như vậy, chúng ta cần 9 giá trị chia, gọi lại mức. Các mức đó được ký hiệu là $L_1, L_2, .., L_9$. Bằng cách đó, khoảng số 1 tính từ trái sang phải là: $[V_{min}, L_1)$; lưu ý, khoảng này **bao gồm** đầu mút V_{min}, nhưng **không bao gồm** đầu mút L_1. Các khoảng còn lại là: $[L_1, L_2)$, $[L_2, L_3)$, .., $[L_9, V_{max}]$. Theo đó, chỉ khoảng cuối cùng $[L_9, V_{max}]$ là bao gồm cả hai đầu mút.

5. Dùng một mảng gồm 10 con số để biểu diễn biểu đồ này. Con số đầu tiên biểu diễn **số lượng** các lỗi e_i ở trên có giá trị rơi vào khoảng $[V_{min}, L_1)$. Tương tự, con số kế tiếp trên mảng biểu diễn số lượng các lỗi e_i rơi vào khoảng $[L_1, L_2)$, và vân vân.

6. Cuối cùng là đưa mảng về biểu đồ tần suất. Để làm việc này, hãy chia các giá trị trong mảng cho **tổng các giá trị trong mảng**. Lưu ý, tổng các giá trị trong mảng trước khi thực hiện phép chia có thể không phải là D (nhỏ hơn D), vì có một số lỗi e_i nhỏ hơn V_{min} hay lớn hơn V_{max}. Cũng như chú ý, tổng các giá trị trong mảng sau khi thực hiện phép chia sẽ là 1. Tuy vậy, sau khi in ra thì có thể tổng này không còn bằng 1 nữa vì tác dụng của hàm **setprecision**.

5 Hướng dẫn thực hiện

Ở bài tập lớn này, các bạn được rèn luyện việc sử dụng các cấu trúc điều khiển, kiểu dữ liệu có cấu trúc và mảng trong phát triển chương trình. Bài tập lớn này chưa đặt nặng vấn đề về tổ chức chương trình như thiết kế các hàm, tổ chức chương trình vào các tập tin, v.v. Do đó, các bạn có thể thực hiện bài tập lớn này theo các hướng dẫn sau đây.

1. Tạo một dự án phần mềm trong công cụ phát triển phần mềm, ví dụ, Visual Studio.

2. Tạo một tập tin "**program.cpp**". Mã nguồn của chương trình các bạn tạo ra **PHẢI** được chứa trong tập tin có tên vừa ghi ra. Lý do của việc này là, sau này chương trình chấm tự động sẽ **CHỈ** biên dịch tập tin "**program.cpp**" ra chương trình thực thi để chạy. Với các bạn đã thiết kế chương trình thành các hàm và các tập tin như hướng dẫn trước của Thầy Nguyễn Đức Dũng, các bạn có thể copy vào tập tin "**program.cpp**". Việc thiết kế chương trình sẽ được quan tâm hơn trong Bài tập lớn số 2. Để chấm tiêu chí đó, các giảng viên sẽ dùng các Trợ giảng - sẽ thông báo sau.

3. Về mặt giải thuật, chương trình của các bạn khá đơn giản chỉ gồm hai bước như được trình bày ở phần 4, đó là (1) Đọc dữ liệu đầu vào và (2) Đánh giá mô hình dự báo theo dữ liệu đọc được.

4. Ở bước đọc dữ liệu:

   - **Việc đọc dữ liệu từ tập tin như thế nào? TRẢ LỜI:** Tôi đã có một ví dụ tương tự, đã cho trước trên trang môn học.
   - **Đọc xong lưu vào đâu? TRẢ LỜI:** Các bạn phải tạo ra những cấu trúc (**struct**) để mô tả dữ liệu. Vì chúng ta có cả hàng ngàn mẫu số liệu. Đến thời điểm này, các bạn đã biết mảng (array) có thể hỗ trợ. Các bạn cũng có thể sử dụng kiểu **vector** (có sẵn trong thư viện <**vector**>), như được minh hoạ trong tài liệu tải lên trước. Tuy nhiên, các bạn được yêu cầu rèn luyện sử dụng mảng trong bài tập lớn này. Nếu các bạn không rèn luyện ở đây, tôi không thể đảm bảo các bạn có thể làm được các câu hỏi trong bài thi cuối khoá.
   - **Cho đến thời điểm này, khai báo mảng cần biết trước số phần tử. Vậy có bao nhiêu mẫu số liệu tối đa trong tập tin đầu vào? TRẢ LỜI:** Trong tập tin đầu vào trong quá trình chấm sẽ không có nhiều hơn 20000 mẫu số liệu. Các bạn nên sử dụng **macro** để thay thế con số vừa nêu.

5. Ở bước đánh giá mô hình dự báo:

   - Khi chương trình vào đến bước này nghĩa là đã đọc được và đúng dữ liệu đầu vào.
   - Dữ liệu đầu vào có thông số **trn_params.num_folds**. Gọi giá trị này là K, chúng ta có K lần đánh giá (dùng cấu trúc lặp). Mỗi lần như vậy phải có các bước sau: (1) Tạo hai tập dữ liệu **TRN** và **TST**, (2) Huấn luyện mô hình dự báo với tập **TRN**, nghĩa là xác định hai tham số a và b của Phương trình 1, (3) Đánh giá mô hình vừa xác định dùng tập **TST**, và (4) Phân tích phân phối của các sai số dự báo cho các mẫu, nghĩa là Tốixây dựng biểu đồ tần suất.

6. Chạy và kiểm tra kết quả.

7. Khi có thông báo nộp bài thì tải tập tin "**program.cpp**" lên hệ thống chấm trực tuyến. Nếu điểm thấp, tinh chỉnh chương trình và nộp lại lên hệ thống; có thể nộp lại nhiều lần, miễn sao trước hạn chót

6 Cách tính điểm

Các giảng viên cũng sẽ thực hiện bài tập lớn này, nghĩa là tạo ra chương trình mẫu, gọi là **CTM**. Kết quả của chương trình do sinh viên tạo ra sẽ được so sánh với kết quả tạo ra của **CTM**, so sánh trên tập tin **assignment1.output.txt**. Nếu kết quả **giống nhau** thì xem như sinh viên hoàn thành đúng, ngược lại xem như chưa hoàn thành. Khi so sánh hai con số, hai số được gọi là **giống nhau**, nếu chúng lệch nhau không quá 10^{-4}. Kết xuất có **năm** con số sau dấu chấm là vậy.

Cụ thể, điểm Bài tập lớn 01 được tính như sau:

[**Điểm**] = 0.2* [**Điểm 1**] + 0.8 * [**Điểm 2**].

Ở đó, [**Điểm 1**] là điểm chấm trên tập dữ liệu đầu vào đã công bố với sinh viên. Đó là tập tin **assignment1.input.txt**. Chương trình chấm điểm cũng chạy chương trình của sinh viên với các tập dữ liệu ẩn (**không công bố cho sinh viên**). Ở các tập dữ liệu ẩn này, giảng viên sẽ cho thay đổi cả thông số, số lượng và giá trị dữ liệu (x, t) trong tập tin đầu vào. Kết quả chấm với các tập dữ liệu mới này là [**Điểm 2**].

Cả [**Điểm 1**] và [**Điểm 2**] đều gồm ba cột thành phần, tương ứng chấm cho kết quả so sánh các hệ số của mô hình dự báo và sai số dự báo (nghĩa là sử dụng 3 ba trị đầu trong các dòng đầu ra) và kết quả so sánh 10 giá trị cuối của mỗi hàng dữ liệu đầu ra, xem công thức theo sau. Có nhiều hàng dữ liệu; do đó, chương trình chấm sẽ lấy trung bình.

[**Điểm 1**] hay [**Điểm 2**] = 0.5*[**Điểm mô hình**] + 0.2 *[**Điểm sai số dự báo**] + 0.3 * [**Điểm biểu đồ**]

Các trường hợp sinh viên sẽ nhận điểm 0:

1. Gian lận trong làm bài (xem trang môn học). Các giảng viên sẽ sử dụng một dịch vụ phát hiện gian lận mã nguồn để làm việc này. Trong trường hợp các sinh viên có mã nguồn giống nhau (cho dù là đổi tên biến, hàm, v.v.), tất cả các sinh viên đó đều phải nhận điểm 0 cho bài tập lớn này. Danh sách các sinh viên có gian lận cũng sẽ được nộp về Khoa và GVCN để theo dõi và xem xét trừ điểm rèn luyện.

2. Chương trình biên dịch không thành công.

3. Chương trình chạy được nhưng ra kết quả khác với kết quả chuẩn.

Các trường hợp điểm chấm cho một tập dữ liệu là 0:

1. Tập tin đầu ra không thể đọc được thành ma trận số liệu có K hàng (số folds, xem **trn_params.num_folds**) và 13 cột.

2. Tập tin đọc được, nhưng có kết quả so sánh khác với kết quả chuẩn.

7 Thời gian nộp bài

1. Hệ thống chấm bài trực tuyến sẽ được mở cho nộp từ lúc: **19h00 Ngày 24 Tháng 04 Năm 2017**.
2. Hệ thống sẽ đóng lại vào lúc: **13h00 Ngày 28 Tháng 04 Năm 2017**.
3. Sinh viên có thể nộp bài nhiều lần, từ lúc hệ thống mở đến lúc hệ thống đóng. **Điểm được lấy là điểm trong lần nộp sau cùng**.

.

8 Hướng dẫn nộp bài

8.1 Kiểm tra chương trình trước khi nộp

Hệ thống chấm tự động sẽ dùng bộ biên dịch "**g++**" chạy trên hệ điều hành Linux để dịch chương trình các bạn ra tập tin thực thi. Việc biên dịch cũng ép buộc chương trình các bạn tuân thủ **Phiên bản năm 2011 của C++, gọi tắt là C++ 11**. Sinh viên có thể đọc thêm phần mô tả **C++ 11** để biết thêm. Rất may mắn, ở bài tập lớn này các bạn sẽ chưa gặp vấn đề gì đặc biệt về **C++ 11**. Do đó, hầu như chương trình các bạn sẽ được biên dịch thành công với **g++** và thông số **C++ 11** trên hệ thống chấm trực tuyến.

Tuy nhiên, để chắc chắn rằng chương trình của các bạn sẽ được biên dịch thành công trên hệ thống chấm tự động, các bạn có thể kiểm tra theo các bước sau đây.

1. Truy cập website `https://www.tutorialspoint.com/compile_cpp11_online.php`. Đây là môi trường phát triển trực tuyến. Website này cho phép các bạn **upload** mã nguồn chương trình và dữ liệu đầu vào, sau đó biên dịch và chạy.
2. Sau khi truy cập, các bạn xoá tập tin "main.cpp" được tạo mặc nhiên, bằng cách nhấn chuột phải lên nó và chọn "Delete file".
3. Dùng menu "**File**", sau đó là "**Upload file**" để upload tập tin "**program.cpp**" và "**assignment1.input.txt**" lên trang này. Nhớ chọn "root" mỗi lần upload (hơi bất tiện!).
4. Chọn menu "**Project**", sau đó là "**Compile Options**" để xác nhận lệnh biên dịch và lệnh chạy chương trình. Một hộp thoại sẽ được hiển thị ra, nó có 2 ô trống. Nhập các lệnh theo hướng dẫn.
   - Nhập lệnh biên dịch vào ô có tên "Compilation Command". Lệnh là (nhớ copy và dán): **g++ -std=c++11 -o program -c program.cpp**
   - Nhập lệnh chạy chương trình vào ô có tên "Execution Command". Lệnh là (nhớ copy và dán): **program < "assignment1.input.txt" > "output.txt"**
5. Chọn menu "**Compile**" (trên giao diện) để biên dịch chương trình. Nếu biên dịch không thành công thì lỗi sẽ xuất hiện trong cửa sổ bên dưới của màn hình. Trong trường hợp này, nhiệm vụ của các bạn là **PHẢI** hiệu chỉnh mã nguồn sao cho biên dịch thành công.
6. Nếu như biên dịch thành công, cửa sổ bên dưới màn hình cũng sẽ thông báo. Trường hợp này, các bạn nhấn nút "**Execute**" để chạy chương trình vừa biên dịch xong, theo lệnh đã nhập phía trên. Các bạn có thể mở tập tin "**output.txt**" để xem kết quả. Nhớ "refresh" lại cửa sổ bên trái để nhìn thấy tập tin "**output.txt**".

8.2 Nộp bài

Sinh viên làm theo các bước sau đây để nộp bài lên hệ thống chấm bài tự động:

1. Truy cập vào website: http://cse.hcmut.edu.vn/onlinejudge/
2. Chọn nút Reset password, ở góc trên, bên trái màn hình. Chương trình sẽ hiển thị một hộp thoại để sinh viên nhập mã số sinh viên.
3. Nhập mã số sinh viên vào ô trống trên hộp thoại. Ví dụ: "**1610129**". Chương trình sẽ gửi một email chứa mật khẩu (password) vào hộp thư điện tử của sinh viên, theo địa chỉ email Nhà trường đã cấp cho các bạn. Ở ví dụ này là "**1610129@hcmut.edu.vn**". Lúc này, sinh viên đã có tên tài khoản (là mã số sinh viên) và mật khẩu để đăng nhập vào hệ thống chấm bài.
4. Đăng nhập vào hệ thống chấm bài với tên tài khoản và mật khẩu đã nói trên.
5. Chọn tab "**Nộp bài**", rồi thực hiện các bước sau:
   (a) Ở mục "**Task**", chọn "**KTLT-assignment-1**".
   (b) Xem hạn chót nộp bài ở mục "**Deadline**".
   (c) Nhấn vào nút "**Choose File**" để chọn tập tin chương trình của các bạn.
   (d) Chọn tập tin mã nguồn C/C++ của chương trình các bạn. Chương trình của các bạn **PHẢI** có tên "**program.cpp**". **Lưu ý, tất cả các ký tự đều là chữ thường. Nếu sai tên thì chương trình chấm bài sẽ không tìm ra, không biên dịch; do đó, các bạn sẽ bị điểm 0**.
   (e) Nhấn "**Send**" để hoàn tất việc nộp bài. Nếu việc nộp thành công, chương trình sẽ cho hiển thị thời điểm bài được nộp và dòng chữ "**Upload done**" để báo các bạn biết.
6. Sau khi nộp bài xong, chọn tab "**Kết quả**" để xem kết quả bài làm của mình. Tab này có nhiều dòng thông tin về các lần nộp bài của các bạn. Trong tab này sẽ có cột "**Score**"; đó là điểm của lần nộp tương ứng. Các bạn có thể nhấn vào mã số sinh viên của các bạn hay nhấn trên điểm để xem chi tiết về kết quả chấm bài.

Lưu ý: Thường thì hệ thống sẽ cho biết điểm trong vài giây. Tuy nhiên, cũng có thể kéo dài hơn, tuỳ vào số lượng người dùng đăng nhập và nộp bài lúc đó.