# CÁC THUẬT TOÁN CHỌN TẬP CON

Mục tiêu của các thuật toán này là vượt qua khó khăn về mặt lưu trữ và tốc độ khi phải đối mặt với cơ sở dữ liệu lớn.

## Phân đoạn (Chunking)

Giải pháp đầu tiên được đề xuất bởi Vapnik (1982) dựa trên sự quan sát rằng chỉ cần có những điểm SV là đủ để ta xây dựng được lời giải cuối cùng. Do đó, nếu ta biết trước các điểm SV và hơn nữa sự xử lý các điểm này phù hợp với bộ nhớ máy tính thì ta chỉ cần phải xử lý vấn đề đã được giảm thiểu này.

Không may thay, ta không thể biết được tập SV trước khi giải quyết vấn đề. Vậy thì ta cứ bắt đầu với một tập ngẫu nhiên (một chunk) mà phù hợp với bộ nhớ, train thuật toán SV với nó, giữ lại những điểm SV và thay chunk bằng các điểm mà sự ước lượng hiện tại gây ra lỗi (chẳng hạn, nằm ngoài ống ). Sau đó, train lại hệ thống và lặp cho đến khi điều kiện KKT được thỏa bởi tất cả các mẫu.

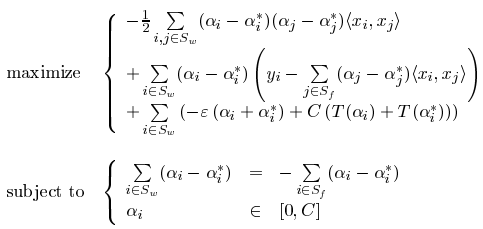
## Các thuật toán tập làm việc (Advanced Working Set Algorithms)

Ta sẽ chỉ dùng một tập con của tập các biến với ý nghĩa là tập làm việc (working set) và tối ưu vấn đề trên tập con này; các biến còn lại ta cố định (freezing).

Không mất tính tổng quát, ta giả sử

Đầu tiên, ta sẽ đưa ra bài toán tối ưu giảm bớt cho tập làm việc trong khi cố định tất cả các biến còn lại.

Gọi là tập làm việc và là tập cố định với . Ta có bài toán tối ưu hóa giảm bớt:



Equation 1

Thuật toán tập làm việc:

1. Khởi tạo αi, αi\* = 0
2. Chọn tập làm việc bất kỳ Sw
3. Lặp

(3.1) Tính các số hạng cặp với tập cố định của Sw

, 

(3.2) Giải quyết baì toán tối ưu hóa giảm bớt

(3.3) Chọn tập Sw mới từ các biến αi, αi\*không thỏa điều kiện KKT

1. Cho đến khi Sw rỗng

Một số nhận xét:

* Mặc dù chưa có chứng minh nào là thuật toán này sẽ hội tụ với một số bước hữu hạn, nhưng nhiều trường hợp trong thực tế, thuật toán này tỏ ra hữu dụng. Tuy nhiên khi áp dụng thật sự, ta cũng phải đề phòng thuật toán sẽ không hội tụ.
* Phần quan trọng của thuật toán là bước (3.3): chọn tập làm việc Sw.

## Các luật lựa chọn

### Đặt vấn đề

Ta đã biết rằng những cặp biến vi phạm điều kiện KKT sẽ được chọn cho tập làm việc Sw mới. Tuy nhiên, trong hầu hết các trường hợp, nhất là ở giai đoạn khởi tạo, tập các cặp biến vi phạm này lớn hơn rất nhiều so với kích thước khả thi cùa Sw. Vậy ta phải chọn những cặp biến vi phạm nào để đưa vào Sw?

### Giải quyết vấn đề

Ý tưởng của ta là chọn những cặp biến vi phạm điều kiện KKT nhiều nhất để đưa vào Sw. Để làm được điều này, với mỗi điểm dữ liệu ta định nghĩa kèm một biến điểm số với thể hiện cho khoảng cách đến lời giải (feasibility gap). Như vậy, cặp biến vi phạm nhiều nhất nghĩa là điểm số tương ứng cao nhất.

Cuối cùng, ta chú ý đến một số heuristic có thể cải thiện hiệu năng đáng kể như là gán các cờ sticky-flag cho các biến ở biên, lưu (caching) ma trận tích vô hướng đã được tính, …

## Tối ưu hóa từng phần nhỏ liên tiếp (SMO – Sequential Minimal Optimization)

### Tổng quan

Ý tưởng của SMO là phân đoạn với kích thước đoạn (tập làm việc) là 2. Điểm “key” ở đây là với tập làm việc chỉ có 2 phần tử, bài toán tối ưu hóa con trên tập này có thể được giải mà không cần dùng đến bộ tối ưu hóa bậc 2.

Chú ý: Những lập luận dưới đây chỉ áp dụng trên hàm tổn thất .

### Mẫu phụ thuộc hằng chuẩn hóa (Pattern Dependent Regularization)

Xét bài toán tối ưu hóa có ràng buộc ở Equation 1với 2 chỉ số i, j. Mẫu phụ thuộc hằng chuẩn hóa nghĩa là Ci có thể khác nhau đối với mỗi mẫu. Đối với bài toán hồi qui, ta phải phân ra làm 4 trường hợp:

Từ ràng buộc tổng có ngay:



Equation 2

và với , ta có 

trong đó L, H được định nghĩa như sau:

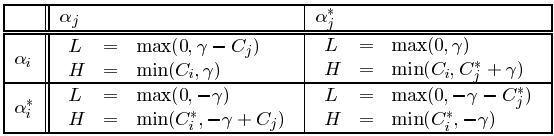
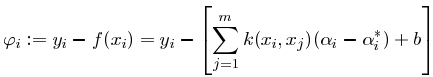


Table 1

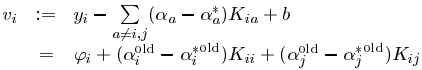
### Giải quyết bài toán tối ưu hóa con

Kế đến ta phải giải quyết vấn đề tối ưu hóa cho hai biến.

Đặt:



Equation 3



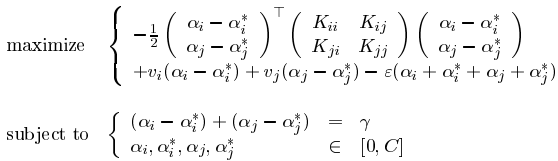
Equation 4

Từ đây, ta có:



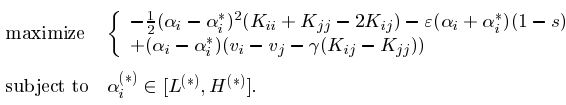
Equation 5

Giới hạn Equation 1 chỉ với (i, j), ta có:



Equation 6

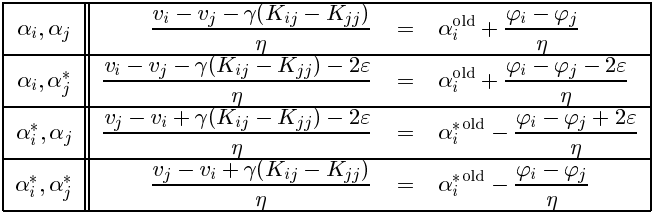
Kế đến, ta sẽ khử bằng ràng buộc tổng:



Equation 7

Để ý rằng (“Lực đẩy” Lagrange chỉ hoặc là kéo hoặc là đẩy.)

Ta có bảng giá trị max của Equation 7:



Table

với .

Nếu một trong hai biến của cặp đang chọn bằng 0, ta phải kiểm cặp còn lại dựa vào Table 2 (chẳng hạn cặp đang chọn là mà thì ta phải kiểm thêm cặp .)

### Luật lựa chọn

Ta phải chọn cặp (i, j) để hàm mục tiêu đạt cực đại.

Chọn i: Ta duyệt qua tất cả các mẫu vi phạm mà trước hết là nhân tử Lagrange của chúng không nằm trong khoảng ràng buộc, khi tất cả đều thỏa ta mới xét đến sự vi phạm điều kiện KKT.

Chọn j: Vì rằng sẽ rất tốn chi phí nếu ta tính cho tất cả các cặp (i, j) có thể, ta sẽ dùng heuristic để chọn j sao cho phần tử số ở Table 2 đạt cực đại. Nếu heuristic này thất bại, tất cả các j còn lại được xem xét cho đến khi tìm được mẫu phù hợp.