Vũ Hữu Tiệp

Convex Optimization - Tối Ưu Lồi

Theo blog: http://machinelearningcoban.com

June 25, 2017

Contents

0

Lời nói đầu

Chào các bạn,

Tài liệu này là một phần trong blog Machine Learning cơ bản của tôi được thực hiện từ đầu năm 2017 cho tới thời điểm này. Khi chuẩn bị tài liệu này, blog đã có 28 bài viết và nhiều ghi chú ngắn về Machine Learning/Artificial Intelligence và Optimization. Hướng tiếp cận của tôi là giới thiệu mỗi thuật toán Machine Learning thông qua việc xây dựng một hàm số đặc biệt được gọi là *hàm mất mát*, hoặc *hàm mục tiêu* và các phương pháp tối ưu hàm mục tiêu. Để có thể hiểu sâu về các thuật toán Machine Learning, tôi luôn cho rằng hiểu rõ cách xây dựng hàm mất mát và cách tối ưu các hàm đó đóng một vai trò quan trọng. Vì vậy, tôi cũng dành thời gian cho một số bài viết liên quan đến Tối Ưu.

Trong lĩnh vực Tối Ưu, Tối Ưu Lồi đóng vai trò quan trọng hơn cả vì những tính chất quan trọng của nó. Tôi chọn ba bài viết về Tối Ưu Lồi để chuẩn bị cho tài liệu này vì tôi biết rằng nhiều bạn đọc trong blog muốn đọc những phần có nhiều toán trên giấy hơn là đọc trên máy. Việc in trực tiếp từ màn hình website ra không sự tốt vì dù sao việc chuyển đổi cũng là tự động.

Không chỉ trong Machine Learning, các lĩnh vực khoa học kỹ thuật và cả tài chính kinh tế cũng rất cần tới tối ưu. Tôi hy vọng rằng tài liệu này sẽ giúp ích cho nhiều người Việt đang học tập và nghiên cứu ở trong và ngoài nước.

Ngôn ngữ trong tài liệu này gần giống với ngôn ngữ trong blog và có liên quan chặt chẽ tới các bài viết khác mà tôi có dẫn links. Tuy nhiên, độc giả chưa đọc blog cũng có thể hiểu được vì đây là kiến thức tổng quan về Tối Ưu Lồi.

Nội dung của tài liệu được dựa trên cuốn **Convex Optimization** của tác giả nổi tiếng Stephen P. Boyd. Tôi cố gắng lược bỏ những phần đi quá sâu vào toán, đồng thời cũng thêm các ví dụ gần với thực tế và chương trình học toán ở Việt Nam. Các hình vẽ trong tài liệu đã được vẽ lại hoàn toàn.

Tài liệu này được tổng hợp trong 1 ngày, nội dung gần như tương tự như trên blog. Tuy nhiên, vì việc chuyển từ ngôn ngữ trên web sang LaTeX khá phức tạp nên chắc chắn tôi

không tránh khỏi sai sót. Nếu thấy phần nào cần phải sửa lại, bạn hãy cho tôi biết qua địa chỉ email vuhuutiep@gmail.com. Tôi sẽ trả lời và chỉnh sửa ngay khi có thể.

Vấn đề bản quyền:

Nội dung trên blog này là hoàn toàn miễn phí. Tôi cũng không sử dụng dịch vụ quảng cáo nào vì không muốn làm phiền các bạn trong khi đọc. Tuy nhiên, nếu bạn thấy nội dung blog và tài liệu này hữu ích và muốn ủng hộ, bạn có thể Mời tôi một ly cà phê bằng cách click vào nút 'Buy me a coffee' ở phía trên cột bên trái của blog, loại cà phê mà bạn vẫn thích uống.

Tôi xin chân thành cảm ơn!

Trân trọng,

Vũ Hữu Tiệp

www.machinelearningcoban.com

Hoa Kỳ, ngày 25 tháng 6 năm 2017.

16

Convex sets và convex functions

16.1 Giới thiệu

Nếu bạn đã đọc các bài trước trong Blog Machine Learning cơ bản, chúng ta đã làm quen với rất nhiều bài toán tối ưu. Học Machine Learning là phải học Toán Tối Ưu, và để hiểu hơn về Toán Tối Ưu, với tôi cách tốt nhất là tìm hiểu các thuật toán Machine Learning. Cho tới lúc này, những bài toán tối ưu các bạn đã nhìn thấy trong blog đều là các bài toán tối ưu không ràng buộc (unconstrained optimization problems), tức tối ưu hàm mất mát mà không có điều kiện ràng buộc (constraints) nào về nghiệm cả.

Không chỉ trong Machine Learning, trên thực tế các bài toán tối ưu thường có rất nhiều ràng buộc khác nhau. Ví dụ:

- Tôi muốn thuê một ngôi nhà cách trung tâm Hà Nội không quá 5km với giá càng thấp càng tốt. Trong bài toán này, giá thuê nhà chính là hàm mất mát (*loss function*, đôi khi người ta cũng dùng *cost function* để chỉ hàm số cần tối ưu), điều kiện khoảng cách không quá 5km chính là ràng buộc (constraint).
- Quay lại bài toán dự đoán giá nhà theo Linear Regression, giá nhà là một hàm tuyến tính của diện tích, số phòng ngủ và khoảng cách tới trung tâm. Rõ ràng, khi làm bài toán này, ta dự đoán rằng giá nhà tăng theo diện tích và số phòng ngủ, giảm theo khoảng cách. Vậy nên một nghiệm được gọi là *có lý một chút* nếu hệ số tương ứng với diện tích và số phòng ngủ là các số dương, hệ số tương ứng với khoảng cách là một số âm. Để tránh các nghiệm ngoại lai không mong muốn, khi giải bài toán tối ưu, ta nên cho thêm các điều kiện ràng buộc này.

Trong Tối Ưu, một bài toán có ràng buộc thường được viết dưới dạng:

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}} & f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, p \end{aligned}$$

Trong đó, vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ được gọi là *biến tối ưu* (*optimization variable*). Hàm số $f_0 : \mathbb{R}^n \to \mathbb{R}$ được gọi là *hàm mục tiêu* (*objective function*, các hàm mục tiêu trong Machine Learning thường được gọi là *hàm mất mát*). Các hàm số $f_i, h_j : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, p$ được gọi là các *hàm ràng buộc* (hoặc đơn giản là *ràng buộc* - constraints). Tập hợp các điểm $\mathbf{x}$ thỏa mãn các *ràng buộc* được gọi là *feasible set*. Mỗi điểm trong *feasible set* được gọi là *feasible point*, các điểm không trong *feasible set* được gọi là *infeasible points*.

Chú ý:

- Nếu bài toán là tìm giá trị lớn nhất thay vì nhỏ nhất, ta chỉ cần đổi dấu của $f_0(\mathbf{x})$.
- Nếu ràng buộc là *lớn hơn hoặc bằng*, tức $f_i(\mathbf{x}) \geq b_i$, ta chỉ cần đổi dấu của ràng buộc là sẽ có điều kiện *nhỏ hơn hoặc bằng* $-f_i(\mathbf{x}) \leq -b_i$.
- Các ràng buộc cũng có thể là *lớn hơn* hoặc *nhỏ hơn*.
- Nếu ràng buộc là *bằng nhau*, tức $h_j(\mathbf{x}) = 0$, ta có thể viết nó dưới dạng hai bất đẳng thức $h_j(\mathbf{x}) \leq 0$ và $-h_j(\mathbf{x}) \leq 0$. Trong một vài tài liệu, người ta bỏ các phương trình ràng buộc $h_j(\mathbf{x}) = 0$ đi.
- Trong bài viết này, $\mathbf{x}, \mathbf{y}$ được dùng chủ yếu để ký hiệu các biến số, không phải là dữ liệu như trong các bài trước. Biến tối ưu chính là biến được ghi dưới dấu $\arg\min$. Khi viết một bài toán Tối Ưu, ta cần chỉ rõ biến nào cần được tối ưu, biến nào là cố định.

Các bài toán tối ưu, nhìn chung không có cách giải tổng quát, thậm chí có những bài chưa có lời giải. Hầu hết các phương pháp tìm nghiệm không chứng minh được nghiệm tìm được có phải là *global optimal* hay không, tức đúng là điểm làm cho hàm số đạt giá trị nhỏ nhất hay lớn nhất hay không. Thay vào đó, nghiệm thường là các *local optimal*, tức các *điểm cực trị*. Trong nhiều trường hợp, các nghiệm *local optimal* cũng mang lại những kết quả tốt.

Để bắt đầu học Tối Ưu, chúng ta cần học một mảng rất quan trọng trong đó, có tên là *Tối Ưu Lồi* (convex optimization), trong đó *hàm mục tiêu* là một *hàm lồi* (convex function), *feasible set* là một *tập lồi* (convex set). Những tính chất đặc biệt về *local optimal* và *global optimal* của một *hàm lồi* khiến Tối Ưu Lồi trở nên cực kỳ quan trọng. Trong bài viết này, tôi sẽ giới thiệu tới các bạn các định nghĩa và tính chất cơ bản của *tập lồi* và *hàm lồi*. *Bài toán tối ưu lồi* (convex optimization problems) sẽ được đề cập trong bài tiếp theo.

Hình 16.1: Các ví dụ về convex sets.

16.2 Convex sets

16.2.1 Định nghĩa

Khái niệm về *convex sets* có lẽ không xa lạ với các bạn học sinh Việt Nam khi chúng ta đã nghe về *đa giác lồi*. *Lồi*, hiểu đơn giản là *phình ra ngoài*, hoặc *nhô ra ngoài*. Trong toán học, *bằng phẳng* cũng được coi là *lồi*.

Định nghĩa 1: Một tập hợp được gọi là *tập lồi* (convex set) nếu đoạn thẳng nối hai điểm *bất kỳ* trong tập hợp hợp đó nằm trọn vẹn trong tập hợp đó.

Một vài ví dụ về convex sets được cho trong Hình 16.1.

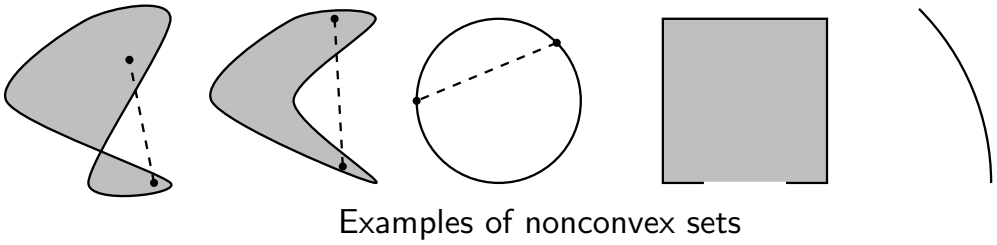
Các hình với đường biên màu đen thể hiện việc bao gồm cả biên, biên màu trắng thể hiện việc biên đó không nằm trong tập hợp đang xét. Đường hoặc đoạn thẳng cũng là một tập lồi theo định nghĩa phía trên.

Một vài ví dụ thực tế:

- Giả sử có một căn phòng có dạng hình *lồi*, nếu ta đặt một bóng đèn đủ sáng ở bất kỳ vị trí nào trong phòng, mọi điểm trong căn phòng đều được chiếu sáng.
- Nếu một đất nước có bản đồ dạng một hình *lồi* thì đường bay nối giữa hai thành phố bất kỳ trong đất nước đó đều nằm trọn vẹn trong không phận của nước đó. (Không như Việt Nam, muốn bay thẳng Hà Nội - Hồ Chí Minh phải bay qua không phận Campuchia).

Hình 16.2 minh hoạ một vài ví dụ về *nonconvex sets*, tức tập hợp mà không phải là lồi:

Ba hình đầu tiên không phải là lồi vì các đường nét đứt chứa nhiều điểm không nằm trong các tập đó. Hình thứ tư, hình vuông không có biên ở đáy, không phải là *tập lồi* vì đoạn



Hình 16.2: Các ví dụ về nonconvex sets.

thẳng nối hai điểm ở đáy có thể chứa phần ở giữa không thuộc tập đang xét (Nếu không có biên thì thình vuông vẫn là một *tập lồi*, nhưng biên nửa vời như ví dụ này thì hãy chú ý). Một đường cong bất kỳ cũng không phải là *tập lồi* vì dễ thấy đường thẳng nối hai điểm bất kỳ không thuộc đường cong đó.

Để mô tả một *tập lồi* dưới dạng toán học, ta sử dụng:

Định nghĩa 2: Một tập hợp $\mathcal{C}$ được gọi là *convex* nếu với hai điểm bất kỳ $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$, điểm $\mathbf{x}_\theta = \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2$ cũng nằm trong $\mathcal{C}$ với bất kỳ $0 \leq \theta \leq 1$.

Có thể thấy rằng, tập hợp các điểm có dạng $(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2)$ chính là *đoạn thẳng* nối hai điểm $\mathbf{x}_1$ và $\mathbf{x}_2$.

Với các định nghĩa này thì *toàn bộ không gian* là một *tập lồi* vì đoạn thẳng nào cũng nằm trong không gian đó. Tập rỗng cũng có thể coi là một trường hợp đặc biệt của *tập lồi*.

Dưới đây là một vài ví dụ hay gặp về *tập lồi*.

16.2.2 Ví dụ

Hyperplanes và halfspaces

Một **hyperplane** (siêu mặt phẳng) trong không gian n chiều là tập hợp các điểm thỏa mãn phương trình:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = \mathbf{a}^T\mathbf{x} = b \tag{16.1}$$

với $b, a_i, i = 1, 2, \ldots, n$ là các số thực.

Hyperplanes là các *tập lồi*. Điều này có thể dễ dàng suy ra từ Định nghĩa 1. Với Định nghĩa 2, chúng ta cũng dễ dàng nhận thấy. Nếu:

$$\mathbf{a}^T\mathbf{x}_1 = \mathbf{a}^T\mathbf{x}_2 = b$$

thì với $0 \leq \theta \leq 1$ bất kỳ:

$$\mathbf{a}^T\mathbf{x}_\theta = \mathbf{a}^T(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2)) = \theta b + (1-\theta)b = b$$

Một **halfspace** (nửa không gian) trong không gian n chiều là tập hợp các điểm thỏa mãn phương trình:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = \mathbf{a}^T\mathbf{x} \leq b$$

với $b, a_i, i = 1, 2, \ldots, n$ là các số thực.

Các halfspace cũng là các tập lồi, bạn đọc có thể dễ dàng nhận thấy theo Định nghĩa 1 hoặc chứng minh theo Định nghĩa 2.

Norm balls

Euclidean balls (hình tròn trong mặt phẳng, hình cầu trong không gian ba chiều) là tập hợp các điểm có dạng:

$$B(\mathbf{x}_c, r) = \left\{\mathbf{x} \;\middle|\; \|\mathbf{x} - \mathbf{x}_c\|_2 \leq r\right\} = \left\{\mathbf{x}_c + r\mathbf{u} \;\middle|\; \|\mathbf{u}\|_2 \leq 1\right\}$$

Theo Định nghĩa 1, chúng ta có thể *thấy* Euclidean balls là các tập lồi, nếu phải chứng minh, ta dùng Định nghĩa 2 và các tính chất của norms. Với $\mathbf{x}_1, \mathbf{x}_2$ bất kỳ thuộc $B(\mathbf{x}_c, r)$ và $0 \leq \theta \leq 1$ bất kỳ:

$$\begin{aligned}\|\mathbf{x}_\theta - \mathbf{x}_c\|_2 &= \|\theta(\mathbf{x}_1 - \mathbf{x}_c) + (1-\theta)(\mathbf{x}_2 - \mathbf{x}_c)\|_2 \\ &\leq \theta\|\mathbf{x}_1 - \mathbf{x}_c\|_2 + (1-\theta)\|\mathbf{x}_2 - \mathbf{x}_c\|_2 \\ &\leq \theta r + (1-\theta)r = r\end{aligned}$$

Vậy nên $\mathbf{x}_\theta \in B(\mathbf{x}_c, r)$.

Euclidean ball sử dụng norm 2 làm khoảng cách. Nếu sử dụng norm bất kỳ là khoảng cách, ta vẫn được một *tập lồi*.

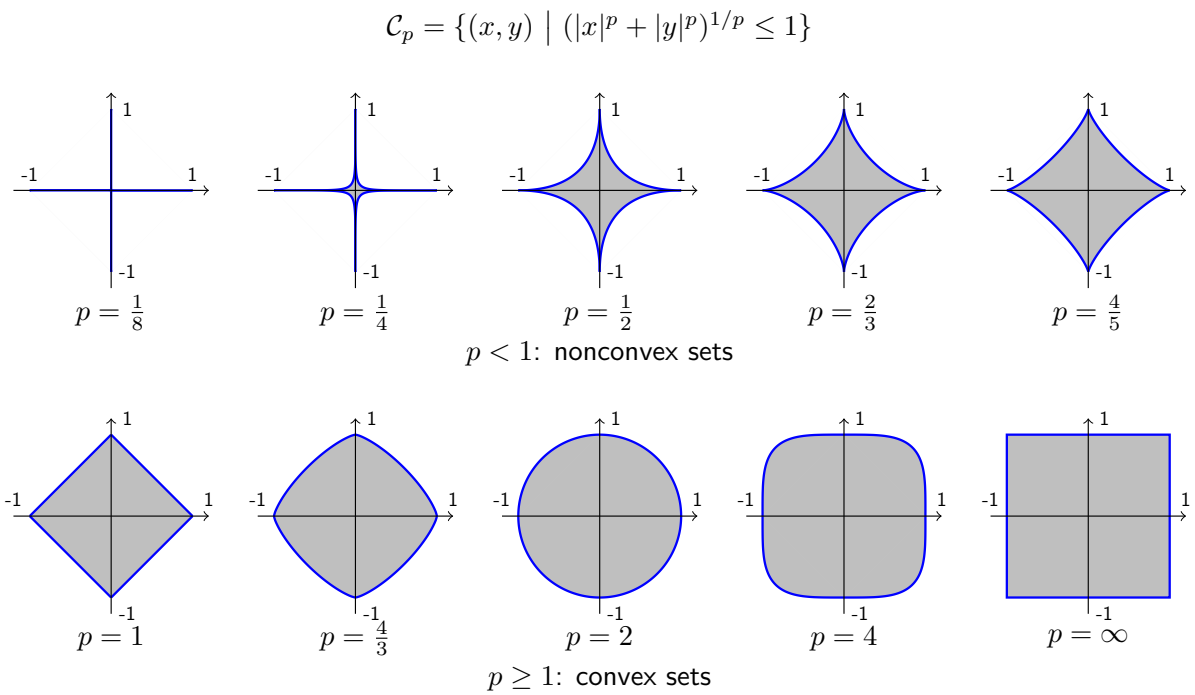
Khi sử dụng norm p:

$$\|\mathbf{x}\|_p = (\|x_1\|^p + \|x_2\|^p + \ldots \|x_n\|^p)^{\frac{1}{p}}$$

với **p là một số thực bất kỳ không nhỏ hơn 1** ta cũng thu được các *tập lồi*.

Hình 16.13 minh họa tập hợp các điểm có tọa độ (x, y) trong không gian hai chiều thỏa mãn:

$$(|x|^p + |y|^p)^{1/p} \leq 1 \tag{16.2}$$



Hình 16.3: Hình dạng của các tập hợp bị chặn bởi pseudo-norms (hàng trên) và norm (hàng dưới).

với hàng trên là các tập với $0 < p < 1$ (không phải norm) và hàng dưới tương ứng với $p \geq 1$.

Chúng ta có thể thấy rằng khi p nhỏ gần bằng 0, tập hợp các điểm thỏa mãn bất đẳng thức (16.2) gần như nằm trên các trục tọa độ và bị chặn trong đoạn $[0, 1]$. Quan sát này sẽ giúp ích cho các bạn khi làm việc với (giả) norm 0 sau này. Khi $p \to \infty$, các tập hợp hội tụ về hình vuông.

Dây cũng là một trong các lý do vì sao cần có điều kiện $p \geq 1$ khi định nghĩa norm.

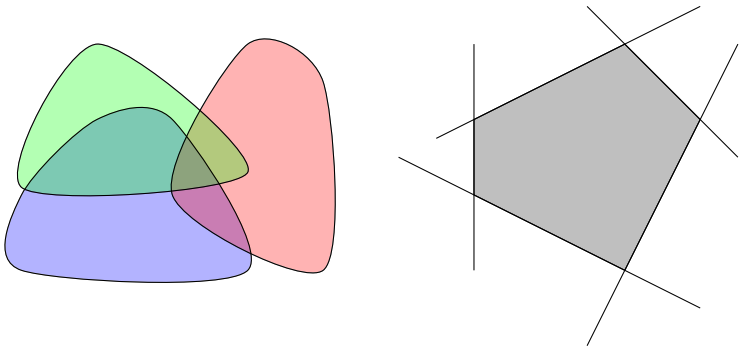
Ellipsoids Các ellipsoids (ellipse trong không gian nhiều chiều) cũng là các *tập lồi*. Thực chất, ellipsoides có mối quan hệ mật thiết tới Khoảng cách Mahalanobis. Khoảng cách này vốn dĩ là một norm nên ta có thể chứng minh theo Định nghĩa 2 được tính chất lồi của các ellipsoids.

Mahalanobis norm của một vector $\mathbf{x} \in \mathbb{R}^n$ được định nghĩa là:

$$\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^T \mathbf{A}^{-1} \mathbf{x}}$$

Với $\mathbf{A}^{-1}$ là một ma trận thỏa mãn:

$$\mathbf{x}^T \mathbf{A}^{-1} \mathbf{x} \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^n \tag{16.3}$$



Hình 16.4: Trái: Giao của các tập lồi là một tập lồi. Phải: giao của các hyperplanes và haflspaces là một tập lồi và được gọi là polyhedron (số nhiều là polyhedra).

Khi một ma trận $\mathbf{A}^{-1}$ thỏa mãn điều kiện (16.3), ta nói ma trận đó *xác định dương* (*positive definite*). Một ma trận là *xác định dương* nếu các trị riêng (eigenvalues) của nó là dương.

Nhân tiện, một ma trận $\mathbf{B}$ được gọi là **nửa** *xác định dương* (*positive semidefinite*) nếu các *trị riêng* của nó là không âm. Khi đó $\mathbf{x}^T\mathbf{B}\mathbf{x} \geq 0, \forall \mathbf{x}$. Nếu dấu bằng xảy ra khi và chỉ khi $\mathbf{x} = 0$ thì ta nói ma trận đó *xác định dương*. Trong biểu thức (16.3), vì ma trận $\mathbf{A}$ có nghịch đảo nên mọi *trị riêng* của nó phải khác không. Vì vậy, $\mathbf{A}$ là một ma trận *xác định dương*.

Một ma trận $\mathbf{A}$ là *xác định dương* hoặc *nửa xác định dương* sẽ được ký hiệu lần lượt như sau:

$$\mathbf{A} \succ 0, \qquad \mathbf{A} \succeq 0.$$

Cũng lại nhân tiện, khoảng cách Mahalanobis có liên quan đến *khoảng cách từ một điểm tới một phân phối xác suất* (from a point to a distribution).

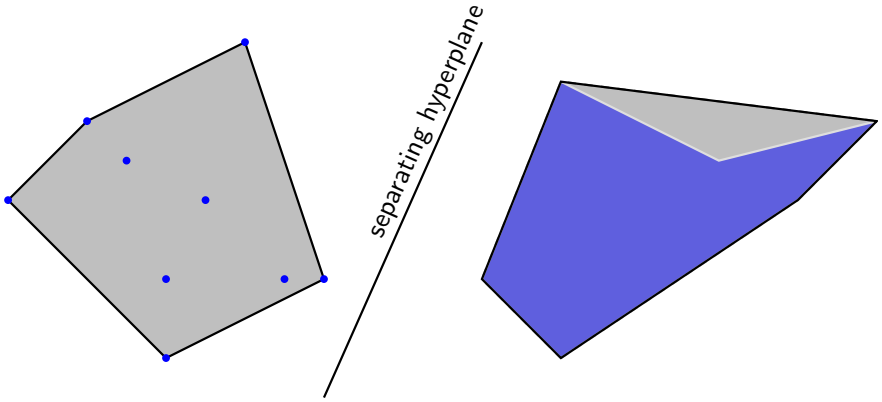
16.2.3 Giao của các tập lồi là một tập lồi.

Việc này có thể nhận dễ nhận thấy với Hình 16.4 (trái). Giao của hai trong ba hoặc cả ba tập lồi đều là các tập lồi.

Việc chứng minh việc này theo Định nghĩa 2 cũng không khó. Nếu $\mathbf{x}_1, \mathbf{x}_2$ thuộc vào giao của các tập lồi, tức thuộc tất cả các tập lồi đã cho, thì $\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2)$ cũng thuộc vào tất cả các tập lồi, tức thuộc vào giao của chúng!

Từ đó suy ra giao của các *haflspaces* và các *hyperplanes* cũng là một tập lồi. Trong không gian hai chiều, tập lồi này chính là *đa giác lồi*, trong không gian ba chiều, nó có tên là *đa diện lồi*.

Trong không gian nhiều chiều, giao của các *haflspaces* và *hyperplanes* được gọi là **polyhedra**.



Hình 16.5: Trái: Giao của các tập lồi là một tập lồi. Phải: giao của các hyperplanes và haflspaces là một tập lồi và được gọi là polyhedron (số nhiều là polyhedra).

Giả sử có m *haflspaces* và p *hyperplanes*. Mỗi một *haflspace*, theo như đã trình bày phía trên, có thể viết dưới dạng $\mathbf{a}_i^T\mathbf{x} \leq b_i,\ \forall i = 1, 2, \ldots, m$. Mỗi một *hyperplane* có thể viết dưới dạng: $\mathbf{c}_i^T\mathbf{x} = d_i,\ \forall i = 1, 2, \ldots, p$.

Vậy nếu đặt $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$, $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T, \mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p]$ và $\mathbf{d} = [d_1, d_2, \ldots, d_p]^T$, ta có thể viết polyhedra dưới dạng tập hợp các điểm $\mathbf{x}$ thỏa mãn:

$$\mathbf{A}^T\mathbf{x} \preceq \mathbf{b}, \quad \mathbf{C}^T\mathbf{x} = \mathbf{d}$$

trong đó $\preceq$ là *element-wise*, tức mỗi phần tử trong vế trái nhỏ hơn hoặc bằng phần tử tương ứng trong vế phải.

16.2.4 Convex combination và Convex hulls

Một điểm được gọi là **convex combination** (*tổ hợp lồi*) của các điểm $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ nếu nó có thể viết dưới dạng:

$$\mathbf{x} = \theta_1\mathbf{x}_1 + \theta_2\mathbf{x}_2 + \cdots + \theta_k\mathbf{x}_k, \quad \text{with} \quad \theta_1 + \theta_2 + \cdots + \theta_k = 1$$

Convex hull của một **tập hợp bất kỳ** là tập hợp tất cả các điểm là *convex combination* của tập hợp đó. *Convex hull* là một *convex set*. *Convexhull* của một *convex set* là chính nó. Một cách dễ nhớ, *convex hull* của một tập hợp là một *convex set* **nhỏ nhất** chứa tập hợp đó. Khái niệm **nhỏ nhất** rất khó định nghĩa, nhưng nó cũng là một cách nhớ trực quan.

Hai tập hợp được gọi là *linearly separable* nếu các *convex hulls* của chúng không có điểm chung.

Trong Hình 16.5, convex hull của các điểm màu xanh là vùng màu xám bao với các đa giác lồi. Ở Hình 16.5 phải, vùng màu xám nằm dưới vùng màu xanh.

Định lý siêu phẳng phân chia (Separating hyperplane theorem): Định lý này nói rằng nếu hai *tập lồi không rỗng* $\mathcal{C}, \mathcal{D}$ là *disjoint* (không giao nhau), thì tồn tại vector $\mathbf{a}$ và số b sao cho:

$$\mathbf{a}^T\mathbf{x} \leq b, \forall \mathbf{x} \in \mathcal{C}, \quad \mathbf{a}^T\mathbf{x} \geq b, \forall \mathbf{x} \in \mathcal{D}$$

Tập hợp tất cả các điểm $\mathbf{x}$ thỏa mãn $\mathbf{a}^T\mathbf{x} = b$ chính là một hyperplane. Hyperplane này được gọi là *separating hyperplane*.

Ngoài ra còn nhiều tính chất thú vị của các tập lồi và các phép toán bảo toàn chính chất *lồi* của một tập hợp, các bạn được khuyến khích đọc thêm Chương 2 của cuốn Convex Optimization trong phần tài liệu tham khảo.

16.3 Convex functions

Hẳn các bạn đã nghe tới khái niệm này khi ôn thi đại học môn toán. Khái niệm hàm lồi có quan hệ tới đạo hàm bậc hai và Bất đẳng thức Jensen (*nếu bạn chưa nghe tới phần này, không sao, bây giờ bạn sẽ biết*).

16.3.1 Định nghĩa

Để trực quan, trước hết ta xem xét các hàm 1 biến, đồ thị của nó là một đường trong một mặt phẳng. Một hàm số được gọi là *lồi* nếu **tập xác định của nó là một tập lồi** và nếu ta nối hai điểm bất kỳ trên đồ thị hàm số đó, ta được một đoạn thẳng nằm về phía trên hoặc nằm trên đồ thị (xem Hình 16.6).

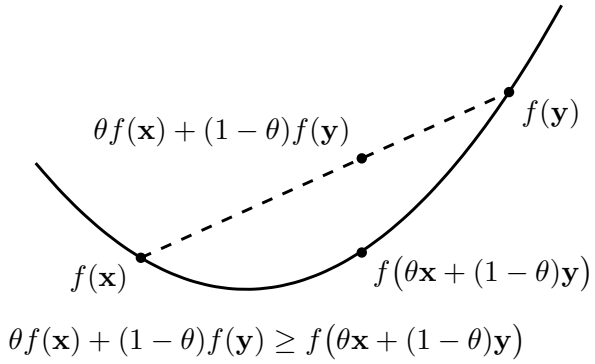
Tập xác định (domain) của một hàm số $f(.)$ thường được ký hiệu là $\text{dom}f$.

Định nghĩa theo toán học:

Định nghĩa convex function: Một hàm số $f : \mathbb{R}^n \to \mathbb{R}$ được gọi là một *hàm lồi* (convex function) nếu $\text{dom}f$ là một *tập lồi*, và:

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y})$$

với mọi $\mathbf{x}, \mathbf{y} \in \text{dom}f, 0 \leq \theta \leq 1$.



Hình 16.6: Định nghĩa hàm lồi. Diễn đạt bằng lời, một hàm số là lồi nếu đoạn thẳng nối 2 điểm bất kỳ trên đồ thị của nó *không nằm dưới* đồ thị đó.

Điều kiện dom$f$ là một *tập lồi* là rất quan trọng, vì nếu không có nó, ta không định nghĩa được $f(\theta\mathbf{x} + (1-\theta)\mathbf{y})$.

Một hàm số f được gọi là **concave** (nếu bạn muốn dịch là *lõm* cũng được, tôi không thích cách dịch này) nếu $-f$ là **convex**. Một hàm số có thể không thuộc hai loại trên. Các hàm tuyến tính vừa *convex*, vừa *concave*.

Định nghĩa strictly convex function: (tiếng Việt có một số tài liệu gọi là *hàm lồi mạnh* hoặc *hàm lồi chặt*) Một hàm số $f : \mathbb{R}^n \to \mathbb{R}$ được gọi là *strictly convex* nếu **dom**f là một *tập lồi*, và:

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) < \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y})$$

với mọi $\mathbf{x}, \mathbf{y} \in \text{dom}f, \mathbf{x} \neq \mathbf{y}, 0 < \theta < 1$.

Tương tự với định nghĩa **strictly concave**.

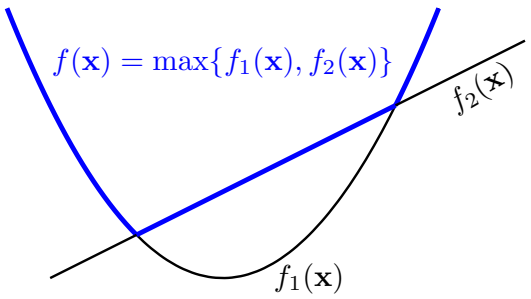
Đây là một điểm quan trọng: **Nếu một hàm số là *strictly convex* và có điểm cực trị, thì điểm cực trị đó là duy nhất và cũng là *global minimum*.**

16.3.2 Các tính chất cơ bản

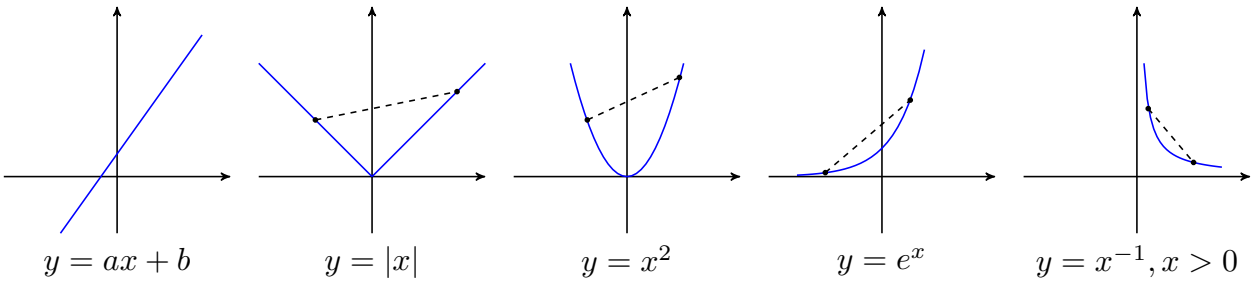
- Nếu $f(\mathbf{x})$ là *convex* thì $af(\mathbf{x})$ là *convex* nếu $a > 0$ và là *concave* nếu $a < 0$. Điều này có thể suy ra trực tiếp từ định nghĩa.
- Tổng của hai *hàm lồi* là một *hàm lồi*, với tập xác định là giao của hai tập xác định kia (nhắc lại rằng giao của hai tập lồi là một tập lồi)
- **Pointwise maximum and supremum:** Nếu các hàm số $f_1, f_2, \ldots, f_m$ là *convex* thì:

  $$f(\mathbf{x}) = \max\{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$$

  cũng là *convex* trên tập xác định là giao của tất cả các tập xác định của các hàm số trên. Hàm max phía trên cũng có thể thay thế bằng hàm sup. Tính chất này có thể chứng minh



Hình 16.7: Ví dụ về Pointwise maximum. Maximum của các hàm lồi là một hàm lồi.



Hình 16.8: Ví dụ về các convex functions một biến.

được theo Định nghĩa. Bạn cũng có thể nhận ra dựa vào hình ví dụ dưới đây. Mọi đoạn thẳng nối hai điểm bất kì trên đường màu xanh đều *không nằm dưới* đường màu xanh.

16.3.3 Ví dụ

Các hàm một biến

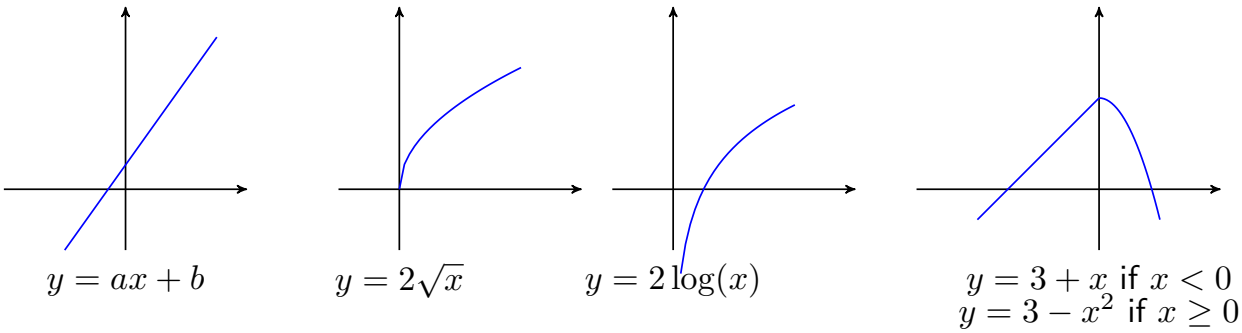
Ví dụ về các *convex functions* một biến:

- Hàm $y = ax + b$ là một *hàm lồi* vì đường nối hai điểm bất kỳ nằm trên chính đồ thị đó.
- Hàm $y = e^{ax}$ với $a \in \mathbb{R}$ bất kỳ.
- Hàm $y = x^a$ trên tập các số thực dương và $a \geq 1$ hoặc $a \leq 0$.
- Hàm *negative entropy* $y = x \log x$ trên tập các số thực dương.

Hình 16.8 minh hoạ đồ thị của một vài *convex functions*:

Ví dụ về các *concave functions* một biến:

- Hàm $y = ax + b$ là một *concave function* vì $-y$ là một *convex function*.



Hình 16.9: Ví dụ về các concave functions một biến.

- Hàm $y = x^a$ trên tập số dương và $0 \leq a \leq 1$.
- Hàm logarithm $y = \log(x)$ trên tập các số dương.

Hình 16.9 minh hoạ đồ thị của một vài *concave functions*:

Affine functions

Các hàm số dạng $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$ vừa là convex, vừa là concave.

Khi biến là một ma trận $\mathbf{X}$, các hàm affine được định nghĩa có dạng:

$$f(\mathbf{X}) = \text{trace}(\mathbf{A}^T\mathbf{X}) + b$$

trong đó trace là hàm số tính tổng các giá trị trên đường chéo của một ma trận vuông, $\mathbf{A}$ là một ma trận có cùng chiều với $\mathbf{X}$ (để đảm bảo phép nhân ma trận thực hiện được và kết quả là một ma trận vuông).

Quadratic forms

Hàm bậc hai một biến có dạng $f(x) = ax^2 + bx + c$ là convex nếu $a > 0$, là concave nếu $a < 0$.

Với biến là một vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, một quadratic form là một hàm số có dạng:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$$

Với $\mathbf{A}$ thường là một ma trận đối xứng, tức $a_{ij} = a_{ji}, \forall i, j$, có số hàng bằng số phần tử của $\mathbf{x}$, $\mathbf{b}$ là một ma trận bất kỳ cùng chiều với $\mathbf{x}$ và c là một hằng số bất kỳ.

Nếu $\mathbf{A}$ là một ma trận (nửa) xác định dương thì $f(\mathbf{x})$ là một *convex function.*

Nếu $\mathbf{A}$ là một ma trận (nửa) xác định âm, tức $\mathbf{x}^T\mathbf{A}\mathbf{x} \leq 0, \forall \mathbf{x}$, thì $f(\mathbf{x})$ là một *concave function*.

Các bạn có thể tìm đọc về ma trận xác định dương và các tính chất của nó trong sách Đại số tuyến tính bất kỳ. Nếu bạn gặp nhiều khó khăn trong phần này, hãy đọc lại kiến thức về Đại số tuyến tính, rất rất quan trọng trong Tối Ưu và Machine Learning.

Hàm mất mát trong Linear Regression có dạng:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \frac{1}{2}\mathbf{y}^T\mathbf{y}\end{aligned}$$

vì $\mathbf{X}^T\mathbf{X}$ là một ma trận xác định dương, hàm mất mát của Linear Regression chính là một convex function.

Norms

Vâng, lại là norms. Một hàm số bất kỳ thỏa mãn ba điều kiện của norm đều là một *convex function*. Bạn đọc có thể chứng minh điều này bằng định nghĩa.

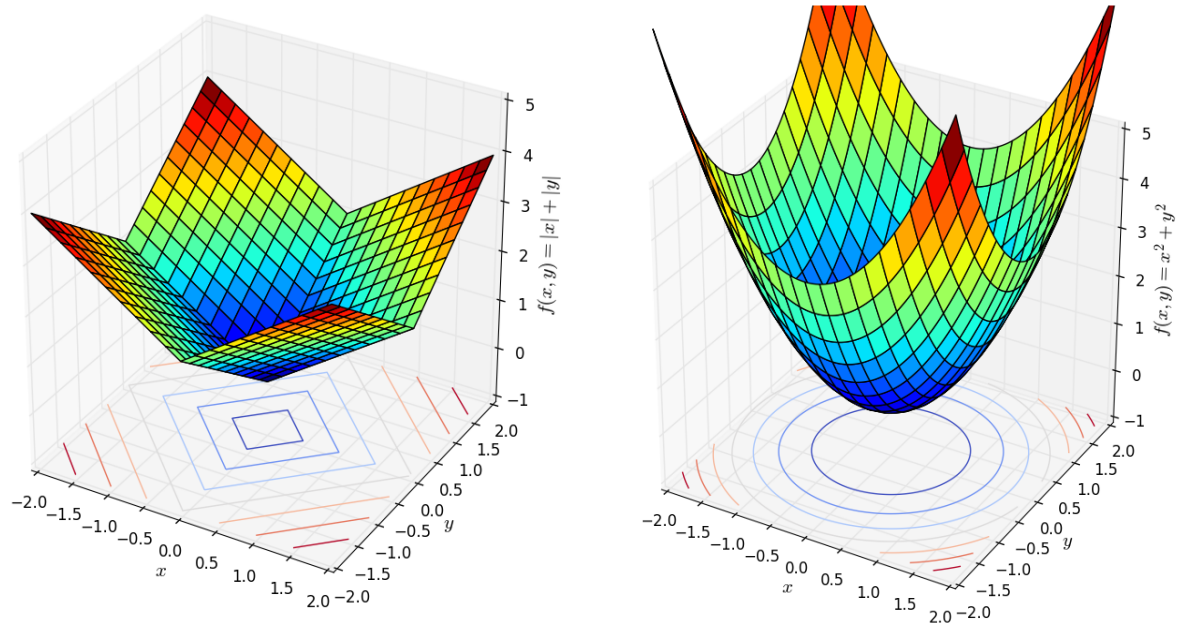
Hình 16.10 minh hoạ hai ví dụ về norm 1 (trái) và norm 2 (phải) với số chiều là 2 (chiều thứ ba trong hình dưới đây là giá trị của hàm số).

Nhận thấy rằng các bề mặt này đều có *một đáy duy nhất* tương ứng với gốc tọa độ (đây chính là điều kiện đầu tiên của norm). Các hàm *strictly convex* khác cũng có dạng tương tự, tức có một *đáy* duy nhất. Điều này cho thấy nếu ta *thả một hòn bi* ở vị trí bất kỳ trên các bề mặt này, cuối cùng nó sẽ *lăn* về đáy. Nếu liên tưởng tới thuật toán Gradient Descent thì việc áp dụng thuật toán này vào các bài toán không ràng buộc với *hàm mục tiêu* là *strictly convex* (và giả sửa là khả vi, tức có đạo hàm) sẽ cho kết quả rất tốt nếu *learning rate* không quá lớn. Đây chính là một trong các lý do vì sao các *convex functions* là quan trọng, cũng là lý do vì sao tôi dành bài viết này chỉ để nói về *convexity*. (Bạn đọc được khuyến khích đọc hai bài về Gradient Descent trong blog này).

Tiện đây, tôi cũng lấy thêm hai ví dụ về các hàm không phải convex (cũng không phải concave). Hàm thứ nhất $f(x, y) = x^2 - y^2$ là một hyperbolic, hàm thứ hai $f(x, y) = \frac{1}{10}(x^2 + 2y^2 - 2\sin(xy))$.

Contours - level sets Với các hàm số phức tạp hơn, khi vẽ các mặt trong không gian ba chiều sẽ khó tưởng tượng hơn, tức khó nhìn được tính *convexity* của nó. Một phương pháp thường được sử dụng là dùng *contours* hay *level sets*. Tôi cũng đã đề cập đến khái niệm này trong Bài Gradient Descent, phần đường đồng mức.

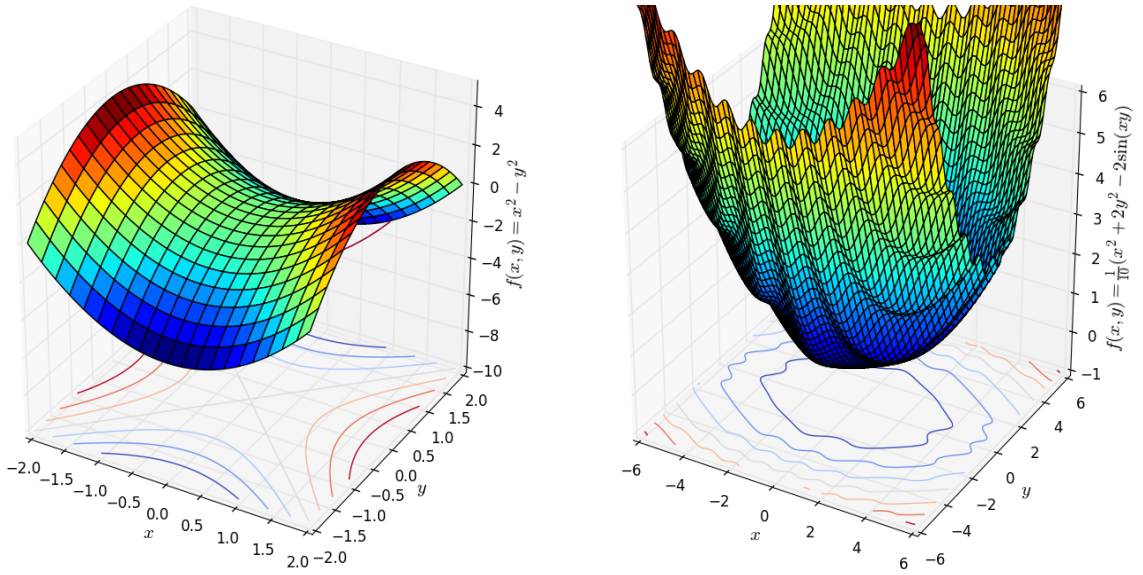
Contours là cách mô tả các mặt trong không gian ba chiều bằng cách chiều nó xuống không gian hai chiều. Trong không gian hai chiều, các điểm thuộc cùng một *đường* tương ứng với



(a) Norm 1

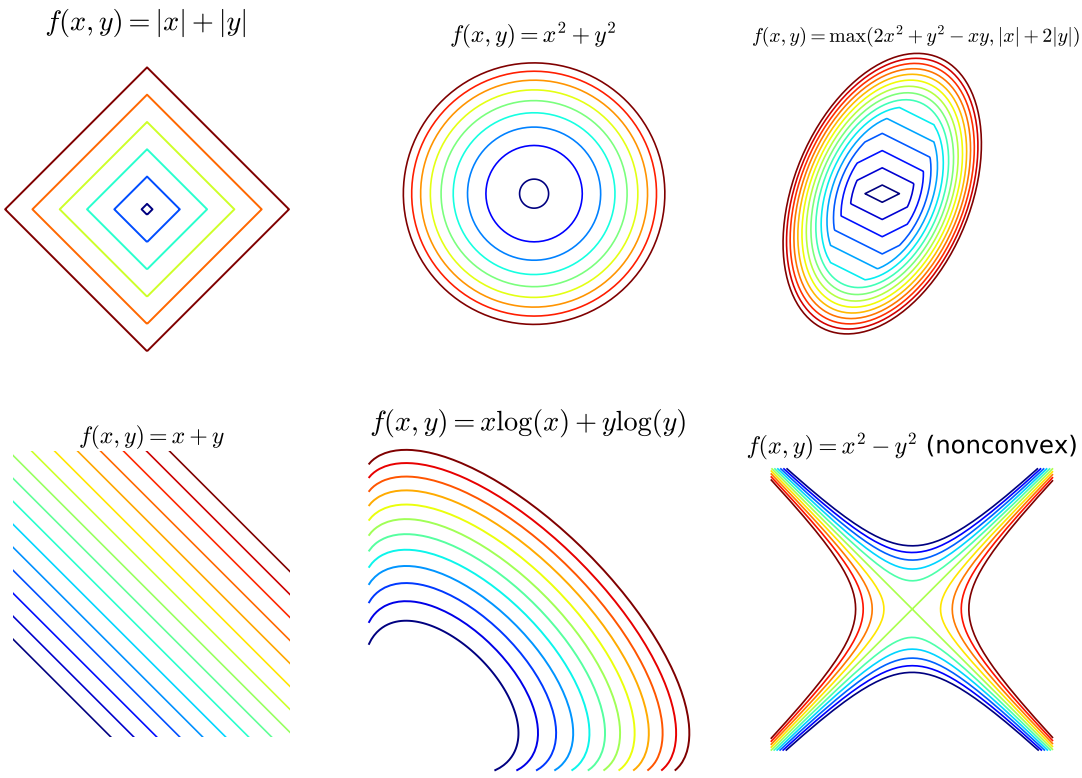
(b) Norm 2

Hình 16.10: Ví dụ về mặt của các norm hai biến.



Hình 16.11: Ví dụ về các hàm hai biến không convex.

các điểm làm cho hàm số có giá trị bằng nhau. Mỗi *đường* đó còn được gọi là một *level set*. Trong Hình 16.10 và Hình 16.11, các đường của các mặt lên mặt phẳng $0xy$ chính là các *level sets*. Một cách hiểu khác, mỗi đường *level set* là một *vết cắt* nếu ta cắt các bề mặt bởi một mặt phẳng song song với mặt phẳng $0xy$.



Hình 16.12: Ví dụ về Countours. Các đường màu càng xanh đậm thì tương ứng với các giá trị càng nhỏ, các đường màu càng đỏ đậm thì tương ứng các giá trị càng lớn.

Khi thể hiện một hàm số hai biến để kiểm tra tính convexity của nó, hoặc để tìm điểm cực trị của nó, người ta thường vẽ *contours* thay vì vẽ các mặt trong không gian ba chiều. Hình 16.12 minh hoạ một vài ví dụ về contours.

Ở hàng trên, các đường *level sets* là các đường khép kín (closed). Khi các đường kín này tập trung nhỏ dần ở một điểm thì các điểm đó là các điểm cực trị. Với các *convex functions* như trong ba ví dụ này, chỉ có 1 điểm cực trị và đó cũng là điểm làm cho hàm số đạt giá trị nhỏ nhất (global optimal). Nếu để ý, bạn sẽ thấy các đường khép kín này tạo thành một *vùng lồi!*.

Ở hàng dưới, các đường không phải khép kín. Hình bên trái tương ứng với một hàm tuyến tính $f(x, y) = x + y$ và đó là một *convex function*. Hình ở giữa cũng là một *convex function* (bạn có thể chứng minh điều này sau khi tính đạo hàm bậc hai, tôi sẽ nói ở phía dưới) nhưng các level sets là các *đường không kín*. Hàm này có log nên tập xác định là góc phần tư thứ nhất tương ứng với các tọa độ dương (chú ý rằng tập hợp các điểm có tọa độ dương cũng là một *tập lồi*). Các *đường không kín* này nếu kết hợp với trục Ox, Oy sẽ tạo thành biên của các *tập lồi*. Hình cuối cùng là contours của một hàm hyperbolic, hàm này không phải là *hàm lồi*.

16.3.4 $\alpha-$ sublevel sets

Định nghĩa: $\alpha-$*sublevel set* của một hàm số $f : \mathbb{R}^n \to \mathbb{R}$ được định nghĩa là:

$$\mathcal{C}_\alpha = \{\mathbf{x} \in \mathbf{dom} f \;\|\; f(\mathbf{x}) \leq \alpha\}$$

Tức tập hợp các điểm trong tập xác định của f mà tại đó hàm số đạt giá trị nhỏ hơn hoặc bằng α.

Quay lại với Hình 16.12, hàng trên, các $\alpha-$ sublevel sets chính là phần bị bao bởi các level sets.

Ở hàng dưới, bên trái, các $\alpha-$ sublevel sets chính là phần nửa mặt phẳng phía dưới xác định bởi các đường thẳng level sets. Ở hình giữa, các $\alpha-$ sublevel sets chính là các vùng bị giới hạn bởi các trục tọa độ và các level sets.

Hàng dưới, bên phải, các $\alpha-$ sublevel sets hơi khó tưởng tượng chút. Với $\alpha > 0$, các level sets là các đường màu vàng hoặc đỏ. Các $\alpha-$ sublevel sets tương ứng là phần *bị bóp vào trong*, giới hạn bởi các đường đỏ cùng màu. Các vùng này, có thể dễ nhận thấy, là *không lồi*.

Định lý: Nếu một hàm số là lồi thì *mọi* $\alpha-$ sublevel sets của nó là lồi. Điều ngược lại chưa chắc đã đúng, tức nếu các $\alpha-$ sublevel sets của một hàm số là *lồi* thì hàm số đó chưa chắc đã *lồi*.

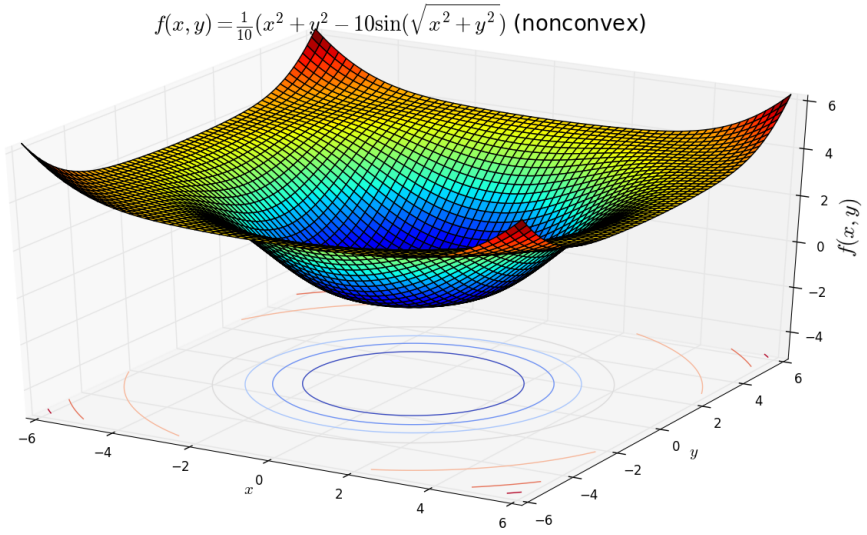
Điều này chỉ ra rằng nếu tồn tại một giá trị α sao cho một $\alpha-$ sublevel set của một hàm số là *không lồi*, thì hàm số đó là *không lồi* (không lồi nhưng không có nghĩa là *concave*, chú ý). Vậy nên Hyperbolic không phải là hàm lồi.

Các ví dụ ở Hình 16.12, trừ hình cuối cùng, đều tương ứng với các hàm lồi.

Một ví dụ về việc một hàm số không *convex* nhưng mọi $\alpha-$ sublevel sets là *convex* là hàm $f(x, y) = -e^{x+y}$. Hàm này có mọi $\alpha-$ sublevel sets là nửa mặt phẳng - là *convex*, nhưng nó không phải là *convex* (trong trường hợp này nó là *concave*).

Dưới đây là một ví dụ khác về việc một hàm số có mọi $\alpha-$ sublevel sets là *lồi* nhưng không phải *hàm lồi*.

Mọi $\alpha-$ sublevel sets của hàm số này đều là các hình tròn - *convex* nhưng hàm số đó không phải là *lồi*. Vì có thể tìm được hai điểm trên mặt này sao cho đoạn thẳng nối hai điểm nằm hoàn toàn phía dưới của mặt (một điểm ở *cánh* và 1 điểm ở *đáy* chẳng hạn).



Hình 16.13: Mọi alpha-sublevel sets là convex sets nhưng hàm số là nonconvex.

Những hàm số có tập xác định là một *tập lồi* và có mọi có $\alpha-$ sublevel sets là *lồi* được gọi chung là *quasiconvex*. Mọi *convex function* đều là *quasiconvex* nhưng ngược lại không đúng. Định nghĩa chính thức của *quasiconvex function* được phát biểu như sau:

Quasiconvex function: Một hàm số $f : \mathcal{C} \to \mathbb{R}$ với $\mathcal{C}$ là một tập con *lồi* của $\mathbb{R}^n$ được gọi là *quasiconvex* nếu với mọi $\mathbf{x}, \mathbf{y}) \in \mathcal{C}$ và mọi $\theta \in [0, 1]$, ta có:

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \max\{f(\mathbf{x}), f(\mathbf{y})\}$$

Định nghĩa này khác với định nghĩa về *convex function* một chút.

16.3.5 Kiểm tra tính chất lồi dựa vào đạo hàm.

Có một cách để nhận biết một hàm số khả vi có là hàm lồi hay không dựa vào các đạo hàm bậc nhất hoặc đạo hàm bậc hai của nó.

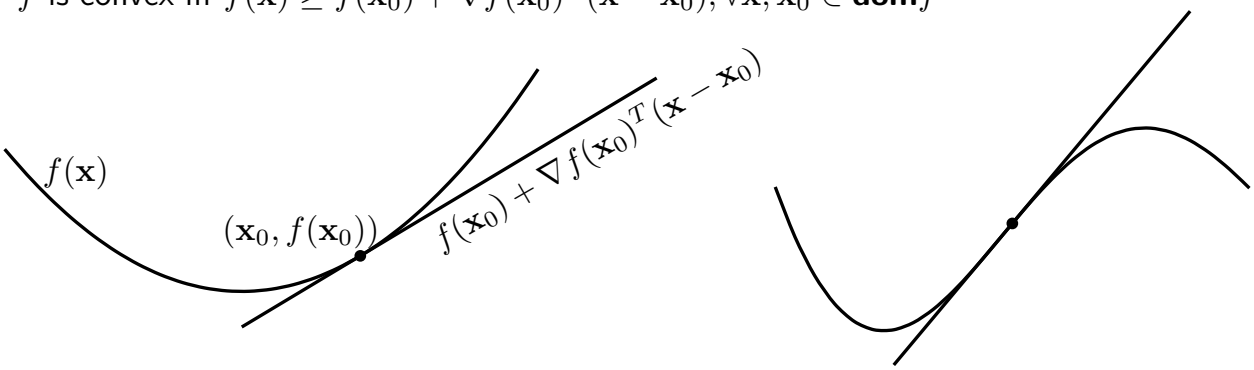
First-order condition

Trước hết chúng ta định nghĩa phương trình đường (mặt) tiếp tuyến của một hàm số f khả vi tại một điểm nằm trên đồ thị (mặt) của hàm số đó $(\mathbf{x}_0, f(\mathbf{x}_0))$. Với hàm một biến, bạn đọc đã quen thuộc:

$$y = f'(x_0)(x - x_0) + f(x_0)$$

f is differentiable with convex domain

f is convex iff $f(\mathbf{x}) \geq f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0), \forall \mathbf{x}, \mathbf{x}_0 \in \mathbf{dom} f$



Hình 16.14: Kiểm tra tính convexity dựa vào đạo hàm bậc nhất. Trái: hàm lồi vì tiếp tuyến tại mọi điểm đều nằm dưới đồ thị hàm số đó, phải: hàm không lồi.

Với hàm nhiều biến, đặt $\nabla f(\mathbf{x}_0)$ là gradient của hàm số f tại điểm $\mathbf{x}_0$, phương trình mặt tiếp tuyến được cho bởi:

$$y = \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) + f(\mathbf{x}_0)$$

First-order condition nói rằng: Giả sử hàm số f có tập xác định là một tập lồi, có đạo hàm tại mọi điểm trên tập xác định đó. Khi đó, hàm số f là *lồi* **nếu và chỉ nếu** với mọi $\mathbf{x}, \mathbf{x}_0$ trên tập xác định của hàm số đó, ta có:

$$f(\mathbf{x}) \geq f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \tag{16.4}$$

Tương tự như thế, một hàm số là *stricly convex* nếu dấu bằng trong (16.4) xảy ra khi và chỉ khi $\mathbf{x} = \mathbf{x}_0$.

Nói một cách trực quan hơn, một hàm số là lồi nếu đường (mặt) tiếp tuyến tại một điểm bất kỳ trên đồ thị (mặt) của hàm số đó **nằm dưới** đồ thị (mặt) đó.

(Đừng quên điều kiện về tập xác định là lồi.)

Dưới đây là ví dụ về *hàm lồi* và *hàm không lồi*.

Hàm bên trái là một hàm lồi. Hàm bên phải không phải là hàm lồi vì đồ thị của nó vừa nằm trên, vừa nằm dưới tiếp tuyến.

(*iff* là viết tắt của *if and only if*)

Ví dụ: Nếu ma trận đối xứng $\mathbf{A}$ là *xác định dương* thì hàm số $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ là *hàm lồi*.

Chứng minh: Đạo hàm bậc nhất của hàm số trên là:

$$\nabla f(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$$

Vậy *first-order condition* có thể viết dưới dạng (chú ý rằng $\mathbf{A}$ là một ma trận đối xứng):

$$\begin{aligned} \mathbf{x}^T\mathbf{A}\mathbf{x} &\geq 2(\mathbf{A}\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0) + \mathbf{x}_0^T\mathbf{A}\mathbf{x}_0 \\ \Leftrightarrow \mathbf{x}^T\mathbf{A}\mathbf{x} &\geq 2\mathbf{x}_0^T\mathbf{A}\mathbf{x} - \mathbf{x}_0^T\mathbf{A}\mathbf{x}_0 \\ \Leftrightarrow (\mathbf{x}-\mathbf{x}_0)^T\mathbf{A}(\mathbf{x}-\mathbf{x}_0) &\geq 0 \end{aligned}$$

Bất đẳng thức cuối cùng là đúng dựa trên định nghĩa của một ma trận *xác định dương*. Vậy hàm số $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ là *hàm lồi*.

First-order condition ít được sử dụng để tìm tính chất lồi của một hàm số, thay vào đó, người ta thường dùng *Second-order condition* với các hàm có đạo hàm tới bậc hai.

Second-order condition

Với hàm nhiều biến, tức biến là một vector, giả sử có chiều là d, đạo hàm bậc nhất của nó là một vector cũng có chiều là d. Đạo hàm bậc hai của nó là một ma trận vuông có chiều là $d \times d$. Đạo hàm bậc hai của hàm số $f(\mathbf{x})$ được ký hiệu là $\nabla^2 f(\mathbf{x})$. Đạo hàm bậc hai còn được gọi là *Hessian*.

Second-order condition: Một hàm số có đạo hàm bậc hai là *convex* nếu $\mathbf{dom}f$ là *convex* và Hessian của nó là một ma trận *nửa xác định dương* với mọi $\mathbf{x}$ trong tập xác định:

$$\nabla^2 f(\mathbf{x}) \succeq 0.$$

Nếu Hessian là một ma trận *xác định dương* thì hàm số đó *strictly convex*. Tương tự, nếu Hessian là một ma trận *xác định âm* thì hàm số đó là *strictly concave*.

Với hàm số một biến $f(x)$, điều kiện này tương đương với $f"(x) \geq 0$ với mọi x thuộc tập xác định (và tập xác định là *lồi*).

Ví dụ:

- Hàm *negative entropy* $f(x) = x\log(x)$ là *stricly convex* vì tập xác định là $x > 0$ là một tập lồi và $f"(x) = 1/x$ là một số dương với mọi x thuộc tập xác định.
- Hàm $f(x) = x^2 + 5\sin(x)$ không là hàm lồi vì đạo hàm bậc hai $f"(x) = 2 - 5\sin(x)$ có thể nhận giá trị âm.
- Hàm *cross entropy* là một hàm *strictly convex*. Xét ví dụ đơn giản với chỉ hai xác suất x và $1-x$ với a là một hằng số thuộc đoạn $[0, 1]$ và $0 < x < 1$: $f(x) = -(a\log(x) + (1-a)\log(1-x))$ có đạo hàm bậc hai là $\frac{a}{x^2} + \frac{1-a}{(1-x)^2}$ là một số dương.

- Nếu $\mathbf{A}$ là một ma trận xác định dương thì $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$ là lồi vì Hessian của nó chính là $\mathbf{A}$ là một ma trận xác định dương.

- Xét hàm số *negative entropy* với hai biến: $f(x, y) = x\log(x) + y\log(y)$ trên tập các giá trị dương của x và y. Hàm số này có đạo hàm bậc nhất là $[\log(x) + 1, \log(y) + 1]^T$ và Hessian là $\begin{bmatrix} 1/x & 0 \\ 0 & 1/y \end{bmatrix}$, là một ma trận đường chéo với các thành phần trên đường chéo là dương nên là một ma trận xác định dương. Vậy *negative entropy* là một hàm *strictly convex*.(*Chú ý rằng một ma trận là xác định dương nếu các trị riêng của nó đều dương. Với một ma trận là ma trận đường chéo thì các trị riêng của nó chính là các thành phần trên đường chéo.*)

Ngoài ra còn nhiều tính chất thú vị của các *hàm lồi*, các bạn được khuyến khích đọc thêm Chương 3 của cuốn Convex Optimization trong phần tài liệu tham khảo.

16.4 Tóm tắt

- Machine Learning và Optimization có quan hệ mật thiết với nhau. Trong Optimization, Convex Optimization là quan trọng nhất. Một bài toán là convex optimization nếu *hàm mục tiêu* là convex và tập hợp các điểm thỏa mãn các điều kiện ràng buộc là một *convex set*.

- Trong *convex set*, mọi đoạn thẳng nối hai điểm bất kỳ trong tập đó sẽ nằm hoàn toàn trong tập đó. Tập hợp các giao điểm của các *convex sets* là một *convex set*.

- Một hàm số là *convex* nếu đoạn thẳng nối hai điểm bất kỳ trên đồ thì hàm số đó không nằm dưới đồ thị đó.

- Một hàm số khả vi là *convex* nếu tập xác định của nó là *convex* và đường (mặt) tiếp tuyến *không nằm phía trên* đồ thị (bề mặt) của hàm số đó.

- Các norms là các hàm lồi, được sử dụng nhiều trong tối ưu.

16.5 Tài liệu tham khảo

[1] Convex Optimization – Boyd and Vandenberghe, Cambridge University Press, 2004.

17

Convex Optimization Problems

Nội dung trong bài viết này chủ yếu được dịch từ Chương 4 của cuốn *Convex Optimization* trong phần Tài liệu tham khảo.

17.1 Giới thiệu

Tôi xin bắt đầu bài viết này bằng ba bài toán khá gần với thực tế:

17.1.1 Bài toán nhà xuất bản

Bài toán

Một nhà xuấn bản (NXB) nhận được đơn hàng 600 bản của cuốn "Machine Learning cơ bản" tới Thái Bình và 400 bản tới Hải Phòng. NXB đó có 800 cuốn ở kho Nam Định và 700 cuốn ở kho Hải Dương. Giá chuyển phát một cuốn sách từ Nam Định tới Thái Bình là 50,000 VND (50k), tới Hải Phòng là 100k. Giá chuyển phát một cuốn từ Hải Dương tới Thái Bình là 150k, trong khi tới Hải Phòng chỉ là 40k. Hỏi để tốn ít chi phí chuyển phát nhất, công ty đó nên phân phối mỗi kho chuyển bao nhiêu cuốn tới mỗi địa điểm?

Phân tích

Để cho đơn giản, ta xây dựng bảng số lượng chuyển sách từ nguồn tới đích như sau:

Tổng chi phí (objective function) sẽ là $f(x, y, z, t) = 5x + 10y + 15z + 4t$. Các điều kiện ràng buộc (constraints) viết dưới dạng biểu thức toán học là:

- Chuyển 600 cuốn tới Thái Bình: $x + z = 600$.

Nguồn	**Đích**	**Đơn giá (×10k)**	**Số lượng**
Nam Định	Thái Bình	5	x
Nam Định	Hải Phòng	10	y
Hải Dương	Thái Bình	15	z
Hải Dương	Hải Phòng	4	t

- Chuyển 400 cuốn tới Hải Phòng: $y + t = 400$.
- Lấy từ kho Nam Định không quá 800: $x + y \leq 800$.
- Lấy từ kho Hải Dương không quá 700: $z + t \leq 700$.
- x, y, z, t là các số tự nhiên. Ràng buộc là số tự nhiên sẽ khiến cho bài toán rất khó giải nếu số lượng biến là rất lớn. Với bài toán này, ta giả sử rằng x, y, z, t là các số thực dương. Khi tìm được nghiệm, nếu chúng không phải là số tự nhiên, ta sẽ lấy các giá trị tự nhiên gần nhất.

Vậy ta cần giải bài toán tối ưu sau đây:

Bài toán NXB:

$$(x, y, z, t) = \arg\min_{x,y,z,t} 5x + 10y + 15z + 4t \tag{17.1}$$

$$\text{subject to:} \quad x + z = 600 \tag{17.2}$$

$$y + t = 400 \tag{17.3}$$

$$x + y \leq 800 \tag{17.4}$$

$$z + t \leq 700 \tag{17.5}$$

$$x, y, z, t \geq 0 \tag{17.6}$$

Nhận thấy rằng hàm mục tiêu (objective function) là một hàm tuyến tính của các biến x, y, z, t. Các điều kiện ràng buộc đều có dạng *hyperplanes* hoặc *haflspaces*, đều là các ràng buộc tuyến tính (linear constraints). Bài toán tối ưu với cả *objective function* và *constraints* đều là *linear* được gọi là **Linear Programming (LP)**. Dạng tổng quát và cách thức lập trình để giải một bài toán thuộc loại này sẽ được cho trong phần sau của bài viết này.

Nghiệm cho bài toán này có thể nhận thấy ngay là $x = 600, y = 0, z = 0, t = 400$. Nếu ràng buộc nhiều hơn và số biến nhiều hơn, chúng ta cần một lời giải có thể tính được bằng cách lập trình.

17.1.2 Bài toán canh tác

Bài toán

Một anh nông dân có tổng cộng 10ha (10 hecta) đất canh tác. Anh dự tính trồng cà phê và hồ tiêu trên số đất này với tổng chi phí cho việc trồng này là không quá 16T (triệu đồng). Chi phí để trồng cà phê là 2T cho 1ha, để trồng hồ tiêu là 1T/ha/. Thời gian trồng cà phê là 1 ngày/ha và hồ tiêu là 4 ngày/ha; trong khi anh chỉ có thời gian tổng cộng là 32 ngày. Sau khi trừ tất cả các chi phí (bao gồm chi phí trồng cây), mỗi ha cà phê mang lại lợi nhuận 5T, mỗi ha hồ tiêu mang lại lợi nhuận 3T. Hỏi anh phải trồng như thế nào để tối đa lợi nhuận? (*Các số liệu có thể vô lý vì chúng đã được chọn để bài toán ra nghiệm đẹp*)

Phân tích

Gọi x và y lần lượt là số ha cà phê và hồ tiêu mà anh nông dân nên trồng. Lợi nhuận anh ấy thu được là $f(x, y) = 5x + 3y$ (triệu đồng).

Các ràng buộc trong bài toán này là:

- Tổng diện tích trồng không vượt quá 10: $x + y \leq 10$.
- Tổng chi phí trồng không vượt quá 16T: $2x + y \leq 16$.
- Tổng thời gian trồng không vượt quá 32 ngày: $x + 4y \leq 32$.
- Diện tích cà phê và hồ tiêu là các số không âm: $x, y \geq 0$.

Vậy ta có bài toán tối ưu sau đây:

Bài toán canh tác:

$$(x, y) = \arg\max_{x,y} 5x + 3y \tag{17.7}$$

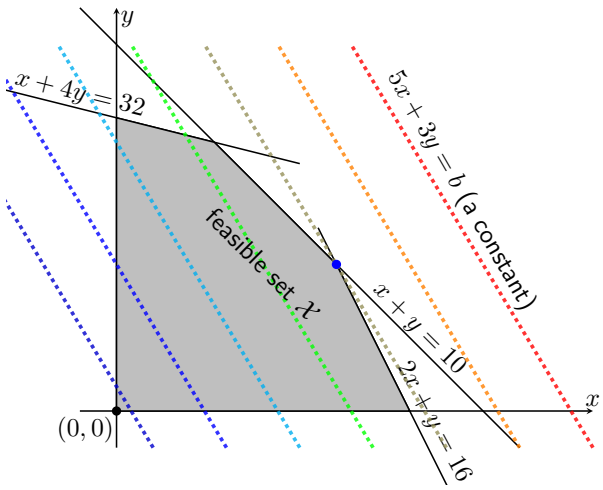
$$\text{subject to:} \quad x + y \leq 10 \tag{17.8}$$

$$2x + y \leq 16 \tag{17.9}$$

$$x + 4y \leq 32 \tag{17.10}$$

$$x, y \geq 0 \tag{17.11}$$

Bài toán này hơi khác một chút là ta cần *tối đa hàm mục tiêu* thay vì tối thiểu nó. Việc chuyển bài toán này về bài toán *tối thiểu* có thể được thực hiện đơn giản bằng cách đổi dấu



Hình 17.1: Minh hoạ nghiệm cho bài toán canh tác. Phần ngũ giác màu xám thể hiện tập hợp các điểm thoả mãn các ràng buộc. Các đường nét đứt thể hiện các đường đồng mức của hàm mục tiêu với màu càng đỏ tương ứng với giá trị càng cao. Nghiệm tìm được chính là điểm màu xanh, là giao điểm của hình ngũ giác xám và đường đồng mức ứng với giá trị cao nhất. Minh hoạ nghiệm cho bài toán canh tác.

hàm mục tiêu. Khi đó hàm mục tiêu vẫn là *linear*, các ràng buộc vẫn là các *linear constraints*, ta lại có một bài toán **Linear Programming (LP)** nữa.

Bạn cũng có thể dựa vào Hình 17.1 để suy ra nghiệm của bài toán.

Vùng màu xám có dạng *polyhedron* (trong trường hợp này là đa giác) chính là tập hợp các điểm thoả mãn các ràng buộc từ (17.8) đến (17.11). Các đường nét đứt có màu chính là các đường đồng mức của hàm mục tiêu $5x + 3y$, mỗi đường ứng với một giá trị khác nhau với đường càng đỏ ứng với giá trị càng cao. Một cách trực quan, nghiệm của bài toán có thể tìm được bằng cách di chuyển đường nét đứt màu xanh về phía bên phải (phía làm cho giá trị của hàm mục tiêu lớn hơn) đến khi nó không còn điểm chung với phần đa giác màu xám nữa.

Có thể nhận thấy nghiệm của bài toán chính là điểm màu xanh là giao điểm của hai đường thẳng $x + y = 10$ và $2x + y = 16$. Giải hệ phương trình này ta có $x^\} = 6$ và $y^\} = 4$. Tức anh nông dân nên trồng 6ha cà phê và 4ha hồ tiêu. Lúc đó lợi nhuận thu được là $5x^\} + 3y^\} = 42$ triệu đồng, trong khi anh chỉ mất thời gian là 22 ngày. (*Chịu tính toán cái là khác ngay, làm ít, hưởng nhiều*).

Với nhiều biến hơn và nhiều ràng buộc hơn, chúng ta liệu có thể vẽ được hình như thế này để nhìn ra nghiệm hay không? Câu trả lời của tôi là nên tìm một công cụ để với nhiều biến hơn và với các ràng buộc khác nhau, chúng ta có thể tìm ra nghiệm gần như ngay lập tức.

17.1.3 Bài toán đóng thùng

Bài toán

Một công ty phải chuyển 400 m^3 cát tới địa điểm xây dựng ở bên kia sông bằng cách thuê một chiếc xà lan. Ngoài chi phí vận chuyển một lượt đi về là 100k của chiếc xà lan, công ty đó phải thiết kế một thùng hình hộp chữ nhật đặt trên xà lan để đựng cát. Chiếc thùng này

không cần nắp, chi phí cho các mặt xung quanh là $1\text{T}/m^2$, cho mặt đáy là $2\text{T}/m^2$. Hỏi kích thước của chiếc thùng đó như thế nào để tổng chi phí vận chuyển là nhỏ nhất. Để cho đơn giản, giả sử cát chỉ được đổ ngang hoặc thấp hơn với phần trên của thành thùng, không có ngọn. Giả sử thêm rằng xà lan *rộng vô hạn* và chứa được sức nặng vô hạn, giả sử này khiến bài toán dễ giải hơn.

Phân tích

Giả sử chiếc thùng cần làm có chiều dài là x (m), chiều rộng là y và chiều cao là z. Thể tích của thùng là xyz (đơn vị là m^3). Có hai loại chi phí là:

- *Chi phí thuê xà lan:* số chuyến xà lan phải thuê là $\frac{400}{xyz}$ (ta hãy tạm giả sử rằng đây là một số tự nhiên, việc làm tròn này sẽ không thay đổi kết quả đáng kể vì chi phí vận chuyển một chuyến là nhỏ so với chi phí làm thùng). Số tiền phải trả cho xà lan sẽ là $0.1\frac{400}{xyz} = \frac{40}{xyz}$.
- *Chi phí làm thùng:* Diện tích xung quanh của thùng là $2(x+y)z$. Diện tích đáy là xy. Vậy tổng chi phí làm thùng là $2(x+y)z + 2xy = 2(xy+yz+zx)$.

Tổng toàn bộ chi phí là $f(x,y,z) = 40x^{-1}y^{-1}z^{-1} + 2(xy+yz+zx)$. Điều kiện ràng buộc duy nhất là kích thước thùng phải là các số dương. Vậy ta có bài toán tối ưu sau:

Bài toán vận chuyển:

$$\begin{aligned}(x,y) = \arg\min_{x,y,z} \; & 40x^{-1}y^{-1}z^{-1} + 2(xy+yz+zx) \\ \text{subject to:} \quad & x,y,z > 0\end{aligned} \tag{17.12}$$

Bài toán này thuộc loại **Geometric Programming (GP)**. Định nghĩa của GP và cách dùng công cụ tối ưu sẽ được trình bày trong phần sau của bài viết.

Nhận thấy rằng bài này hoàn toàn có thể dùng bất đẳng thức Cauchy để giải được, nhưng tôi vẫn muốn một lời giải cho bài toán tổng quát sao cho có thể lập trình được.

(Lời giải:

$$f(x,y,z) = \frac{20}{xyz} + \frac{20}{xyz} + 2xy + 2yz + 2zx \geq 5\sqrt[5]{3200}$$

dấu bằng xảy ra khi và chỉ khi $x = y = z = \sqrt[5]{10}$. Bài này có lẽ hợp với các kỳ thi vì dữ kiện quá đẹp. Cá nhân tôi thích các đề bài ra kiểu này hơn là yêu cầu đi tìm giá trị nhỏ nhất của một biểu thức nhàm chán, nhiều học sinh cho rằng không biết học bất đẳng thức để làm gì!)

Nếu có các ràng buộc về kích thước của thùng và trọng lượng mà xà lan tải được thì có thể tìm được lời giải đơn giản như thế này không?

Những bài toán trên đây đều là các bài toán tối ưu. Chính xác hơn nữa, chúng đều là các bài toán tối ưu lồi (*convex optimization problems*) như các bạn sẽ thấy ở phần sau. Và việc tìm lời giải có thể không mấy khó khăn, thậm chí giải bằng tay cũng có thể ra kết quả. Tuy nhiên, mục đích của bài viết này không phải là hướng dẫn các bạn giải các bài toán trên *bằng tay*, mà là cách nhận diện các bài toán và đưa chúng về các dạng mà các toolboxes sẵn có có thể giúp chúng ta. Trên thực tế, lượng dữ kiện và số biến cần tối ưu lớn hơn nhiều, chúng ta không thể giải các bài toán trên *bằng tay* được.

Trước hết, chúng ta cần hiểu các khái niệm về *convex optimization problems* và tại sao *convex* lại quan trọng. (Bạn đọc có thể đọc tới phần 4 nếu không muốn biết các khái niệm và định lý toán trong phần 2 và 3.)

17.2 Nhắc lại bài toán tối ưu

17.2.1 Các khái niệm cơ bản

Tôi xin nhắc lại bài toán tối ưu ở dạng tổng quát:

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}} & f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, \\ & h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, p \end{aligned} \tag{17.13}$$

Phát biểu bằng lời: Tìm giá trị của biến $\mathbf{x}$ để tối thiểu hàm $f_0(\mathbf{x})$ trong số các giá trị của $\mathbf{x}$ thoả mãn các điệu hiện ràng buộc. Ta có bảng các tên gọi tiếng Anh và tiếng Việt như trong Bảng 17.1.

Ngoài ra:

- Khi $m = p = 0$, bài toán (17.13) được gọi là *unconstrained optimization problem* (bài toán tối ưu không ràng buộc).
- $\mathcal{D}$ chỉ là tập xác định, tức giao của tất cả các tập xác định của mọi hàm số xuất hiện trong bài toán. Tập hợp các điểm thoả mãn mọi điều kiện ràng buộc, thông thường, là một tập con của $\mathcal{D}$ được gọi là *feasible set* hoặc *constraint set*. Khi *feasible set* là một tập rỗng thì ta nói bài toán tối ưu (17.13) là *infeasible*. Nếu một điểm nằm trong *feasible set*, ta gọi điểm đó là *feasible*.
- *Optimal value* (*giá trị tối ưu*) của bài toán tối ưu (17.13) được định nghĩa là:

Ký hiệu	Tiếng Anh	Tiếng Việt
$\mathbf{x} \in \mathbb{R}^n$	optimization variable	biến tối ưu
$f_0 : \mathbb{R}^n \to \mathbb{R}$	objective/loss/cost/function	hàm mục tiêu
$f_i(\mathbf{x}) \leq 0$	inequality constraint	bất đẳng thức ràng buộc
$f_i : \mathbb{R}^n \to \mathbb{R}$	inequality constraint function	hàm bất đẳng thức ràng buộc
$h_j(\mathbf{x}) = 0$	equality constraint	đẳng thức ràng buộc
$h_j : \mathbb{R}^n \to \mathbb{R}$	equality constraint function	hàm đẳng thức ràng buộc
$\mathcal{D} = \bigcap_{i=0}^m \mathbf{dom} f_i \cap \bigcap_{pj=1}^p \mathbf{dom} h_i$	domain	tập xác định

Bảng 17.1: Bảng các thuật ngữ trong các bài toán tối ưu.

$$p^* = \inf \{f_0(\mathbf{x}) | f_i(\mathbf{x}) \leq 0, i = 1, \ldots, m; h_j(\mathbf{x}) = 0, j = 1, \ldots, p\}$$

trong đó inf là viết tắt của hàm infimum. p^* có thể nhận các giá trị $\pm\infty$. Nếu bài toán là *infeasible*, ta coi $p^* = +\infty$, Nếu hàm mục tiêu không bị chặn dưới (*unbounded below*) trong tập xác định, ta coi $p^* = -\infty$.

17.2.2 Optimal and locally optimal points

Một điểm $\mathbf{x}^*$ được gọi là một điểm *optimal point* (*điểm tối ưu*), hoặc là *nghiệm* của bài toán (17.13) nếu $\mathbf{x}^*$ là *feasible* và $f_0(\mathbf{x}^*) = p^*$. Tất hợp tất cả các *optimal points* được gọi là *optimal set*.

Nếu *optimal set* là một tập *không* rỗng, ta nói bài toán (17.13) là *solvable* (*giải được*). Ngược lại, nếu *optimal set* là một tập rỗng, ta nói *optimal value* là *không thể đạt được* (*not attained/ not achieved*).

Ví dụ: xét hàm mục tiêu $f(x) = 1/x$ với ràng buộc $x > 0$. *Optimal value* của bài toán này là $p^\} = 0$ nhưng *optimal set* là một tập rỗng vì không có giá trị nào của x để hàm mục tiêu đạt giá trị 0. Lúc này ta nói *giá trị tối ưu* là *không đạt được*.

Với hàm một biến, một điểm là *cực tiểu* của một hàm số nếu tại đó, hàm số đạt giá trị nhỏ nhất trong một lân cận (và lân cận này thuộc tập xác định của hàm số). Trong không gian 1 chiều, *lân cận* được hiểu là trị tuyệt tối của hiệu 2 điểm nhỏ hơn một giá trị nào đó.

Trong toán tối ưu (thường là không gian nhiều chiều), ta gọi một điểm $\mathbf{x}$ là **locally optimal** (cực tiểu) nếu tồn tại một giá trị (thường được gọi là bán kính) R sao cho:

$$\begin{aligned} f_0(\mathbf{x}) = \quad & \inf \big\{ f_0(\mathbf{z}) | f_i(\mathbf{z}) \leq 0, i = 1, \ldots, m, \\ & h_j(\mathbf{z}) = 0, j = 1, \ldots, p, \|\mathbf{z} - \mathbf{x}\|_2 \leq R \big\} \end{aligned} \tag{17.14}$$

Nếu một điểm *feasible* $\mathbf{x}$ thoả mãn $f_i(\mathbf{x}) = 0$, ta nói rằng bất đẳng thức ràng buộc thứ $i : f_i(\mathbf{x}) = 0$ là *active*. Nếu $f_i(\mathbf{x}) < 0$, ta nói rằng ràng buộc này là *inactive* tại $\mathbf{x}$.

17.2.3 Một vài lưu ý

Mặc dù trong định nghĩa bài toán tối ưu (17.13) là cho bài toán *tối thiểu hàm mục tiêu* với các ràng buộc thoả mãn các điều kiện nhỏ hơn hoặc bằng 0, các bài toán tối ưu với *tối đa hàm mục tiêu* và điều kiện ràng buộc ở dạng khác đều có thể đưa về được dạng này:

- $\max f_0(\mathbf{x}) \Leftrightarrow \min -f_0(\mathbf{x})$.
- $f_i(\mathbf{x}) \leq g(\mathbf{x}) \Leftrightarrow \ f_i(\mathbf{x}) - g(\mathbf{x}) \leq 0$.
- $f_i(\mathbf{x}) \geq 0 \Leftrightarrow \ -f_i(\mathbf{x}) \leq 0$.
- $a \leq f_i(\mathbf{x}) \leq b \Leftrightarrow \ f_i(\mathbf{x}) - b \leq 0$ và $a - f_i(\mathbf{x}) \leq 0$.
- $f_i(\mathbf{x}) \leq 0 \Leftrightarrow f_i(\mathbf{x}) + s_i = 0$ và $s_i \geq 0$. s_i được gọi là *slack variable*. Phép biến đổi đơn giản này trong nhiều trường hợp lại tỏ ra hiệu quả vì bất đẳng thức $s_i \geq 0$ thường dễ giải quyết hơn là $f_i(\mathbf{x}) \leq 0$.

17.3 Bài toán tối ưu lồi

Trong toán tối ưu, chúng ta đặc biệt quan tâm tới những bài toán mà hàm mục tiêu là một hàm lồi, và *feasible set* cũng là một tập lồi.

17.3.1 Định nghĩa

Một *bài toán tối ưu lồi* (*convex optimization problem*) là một bài toán tối ưu có dạng:

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}} & f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m \\ & \mathbf{a}_j^T \mathbf{x} - b_j = 0, j = 1, \ldots, \end{aligned} \tag{17.15}$$

trong đó $f_0, f_1, \ldots, f_m$ là các hàm lồi.

So với bài toán tối ưu (17.13), bài toán tối ưu lồi (17.15) có thêm ba điều kiện nữa:

- *Hàm mục tiêu* là một *hàm lồi*.

- Các hàm bất đẳng thức ràng buộc f_i là các hàm lồi.
- Hàm đẳng thức ràng buộc h_j là *affine* (hàm *linear* cộng với một hằng số nữa được gọi là *affine*).

Một vài nhận xét:

- Tập hợp các điểm thoả mãn $h_j(\mathbf{x}) = 0$ là một tập lồi vì nó có dạng một *hyperplane*.
- Khi f_i là một *hàm lồi* thì tập hợp các điểm thoả mãn $f_i(\mathbf{x}) \leq 0$ chính là 0-sublevel set của f_i và là một tập lồi.
- Như vậy tập hợp các điểm thoả mãn mọi điều kiện ràng buộc chính là giao điểm của các *tập lồi*, vì vậy nó là một *tập lồi*.

Vậy, trong một bài toán tối ưu lồi, ta *tối thiểu một hàm mục tiêu lồi* trên một *tập lồi*.

17.3.2 Cực tiểu của bài toán tối ưu lồi chính là điểm tối ưu.

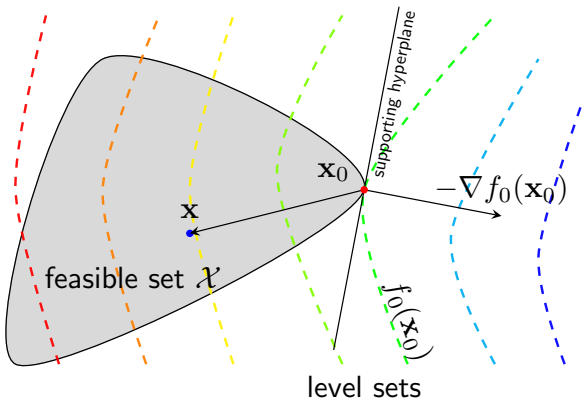
TÍnh chất quan trọng nhất của bài toán tối ưu lồi chính là bất kỳ *locally optimal point* chính là một điểm *(globally) optimal point*.

Tính chất quan trọng này có thể chứng minh bằng phản chứng như sau. Gọi $\mathbf{x}_0$ là một điểm *locally optimal*, tức:

$$f_0(\mathbf{x}_0) = \inf\{f_0(\mathbf{x}) | \mathbf{x} \text{ is feasible}, \|\mathbf{x} - \mathbf{x}_0\|_2 \leq R\}$$

với $R > 0$ nào đó. Giả sử $\mathbf{x}_0$ không phải là *globally optimal point*, tức tồn tại một *feasible point* $\mathbf{y}$ sao cho $f(\mathbf{y}) < f(\mathbf{x}_0)$ (hiển nhiên rằng $\mathbf{y}$ không nằm trong lân cận đang xét). Ta có thể tìm được $\theta \in [0, 1]$ đủ nhỏ sao cho $\mathbf{z} = (1 - \theta)\mathbf{x}_0 + \theta\mathbf{y}$ nằm trong lân cận của $\mathbf{x}_0$, tức $\|\mathbf{z} - \mathbf{x}_0\|_2 < R$. Chú ý rằng $\mathbf{z}$ cũng là một *feasible point* vì *feasible set* là một *tập lồi*. Hơn nữa, vì *hàm mục tiêu* f_0 là một hàm lồi, ta có:

$$\begin{aligned}
f_0(\mathbf{z}) &= f_0((1 - \theta)\mathbf{x}_0 + \theta\mathbf{y}) && (17.16)\\
&\leq (1 - \theta)f_0(\mathbf{x}_0) + \theta f_0(\mathbf{y}) && (17.17)\\
&< (1 - \theta)f_0(\mathbf{x}_0) + \theta f_0(\mathbf{x}_0) && (17.18)\\
&= f_0(\mathbf{x}_0) && (17.19)
\end{aligned}$$



Hình 17.2: Biểu diễn hình học của điều kiện tối ưu cho hàm mục tiêu khả vi. Các đường nét đứt có màu tương ứng với các level sets (đường đồng mức).

điều này mâu thuẫn với giả thiết $\mathbf{x}_0$ là một điểm cực tiểu. Vậy giả sử sai, tức $\mathbf{x}_0$ chính là *globally optimal point* và ta có điều phải chứng minh.

Chứng minh bằng lời: giả sử một điểm cực tiểu không phải là điểm làm cho hàm số đạt giá trị nhỏ nhất. Với điều kiện *feasible set* và *hàm mục tiêu* là lồi, ta luôn tìm được một điểm khác trong lân cận của điểm cực tiểu đó sao cho giá trị của hàm mục tiêu tại điểm mới này nhỏ hơn giá trị của hàm mục tiêu tại điểm cực tiểu. Sự mâu thuẫn này chỉ ra rằng với một bài toán tối ưu lồi, điểm cực tiểu phải là điểm làm cho hàm số đạt giá trị nhỏ nhất.

17.3.3 Điều kiện tối ưu cho hàm mục tiêu khả vi

Nếu hàm mục tiêu f_0 là khả vi (differentiable), theo first-order condition, với mọi $\mathbf{x}, \mathbf{y} \in \text{dom} f_0$, ta có:

$$f_0(\mathbf{x}) \geq f_0(\mathbf{x}_0) + \nabla f_0(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \tag{17.20}$$

Đặt $\mathcal{X}$ là *feasible set*. **Điều kiện cần và đủ** để một điểm $\mathbf{x}_0 \in \mathcal{X}$ là *optimal point* là:

$$\nabla f_0(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \geq 0, \ \forall \mathbf{x} \in \mathcal{X} \tag{17.21}$$

Tôi xin được bỏ qua việc chứng minh điều kiện cần và đủ này, bạn đọc có thể tìm trong trang 139-140 của cuốn Convex Optimization trong Tài liệu tham khảo.

Một cách hình học, điều kiện này nói rằng: Nếu $\mathbf{x}_0$ là điểm *optimal* thì với mọi $\mathbf{x} \in \mathcal{X}$, vector đi từ $\mathbf{x}_0$ tới $\mathbf{x}$ hợp với vector $-\nabla f_0(\mathbf{x}_0)$ một góc tù (Xem Hình 17.2). Nói cách khác, nếu ta vẽ *mặt tiếp tuyến* của hàm mục tiêu tại $\mathbf{x}_0$ thì mọi điểm *feasible* nằm về một phía so với *mặt tiếp tuyến này*. Hơn nữa, *feasible set* nằm về phía làm cho hàm mục tiêu đạt giá trị cao hơn $f_0(\mathbf{x}_0)$. Mặt tiếp tuyến này chính là *supporting hyperplane* của *feasible set* tại điểm $\mathbf{x}_0$. Nhắc lại rằng khi vẽ các *level set*, tôi thường dùng màu lam để chỉ giá trị nhỏ, màu đỏ để chỉ giá trị lớn của hàm số.

(Một mặt phẳng đi qua một điểm trên biên của một tập hợp sao cho mọi điểm trong tập hợp đó nằm về một phía (hoặc nằm trên) so với mặt phẳng đó được gọi là *supporting hyperplane*

(*siêu phẳng hỗ trợ*). Nếu một tập hợp là *lồi*, tồn tại *supporting hyperplane* tại mọi điểm trên biên của nó.)

Nếu tồn tại một điểm $\mathbf{x}_0$ trong *feasible set* sao cho $\nabla f_0(\mathbf{x}_0) = 0$, đây chính là *optimal point*. Điều này dễ hiểu vì đó chính là điểm làm cho gradient bằng 0, tức điểm cực tiểu của hàm mục tiêu. Nếu $\nabla f_0(\mathbf{x}_0) \neq 0$, vector $-\nabla f_0(\mathbf{x}_0)$ chính là *vector pháp tuyến* của *supporting hyperplane* tại $\mathbf{x}_0$.

17.3.4 Giới thiệu thư viện CVXOPT

CVXOPT là một thư viện miễn phí trên Python giúp giải rất nhiều các bài toán trong cuốn sách Convex Optimization ở phần Tài liệu tham khảo. Tác giả thứ hai của cuốn sách này, Lieven Vandenberghe, chính là đồng tác giả của thư viện này. Hướng dẫn cài đặt, tài liệu hướng dẫn, và các ví dụ mẫu của thư viện này cũng có đầy đủ trên trang web CVXOPT.

Trong phần còn lại của bài viết, tôi sẽ giới thiệu 3 bài toán rất cơ bản trong Convex Optimization: Linear Programming, Quadratic Programming, và Geometric Programming. Tôi cũng sẽ cùng các bạn lập trình để giải các ví dụ đã nêu ở phần đầu bài viết dựa trên thư viện CVXOPT này.

17.4 Linear Programming

Chúng ta cùng bắt đầu với lớp các bài toán đơn giản nhất trong Convex Optimization - Linear Programming (LP, một số tài liệu cũng gọi là *Linear Program*), trong đó hàm mục tiêu f_0 và hàm bất đẳng thức ràng buộc $f_i, i = 1, \ldots, m$ đều là các hàm tuyến tính cộng với một hằng số (tức hàm *affine*).

17.4.1 Dạng tổng quát của LP

A general LP:

$$\begin{aligned} \mathbf{x} = \arg\min_{\mathbf{x}} \; & \mathbf{c}^T\mathbf{x} + d \\ \text{subject to:} \quad & \mathbf{G}\mathbf{x} \preceq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{17.22}$$

Trong đó $\mathbf{G} \in \mathbb{R}^{m \times n}, \mathbf{h} \in \mathbb{R}^m$ và, $\mathbf{A} \in \mathbb{R}^{p \times n}, \mathbf{b} \in \mathbb{R}^p$. $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ và d là một số vô hướng (số vô hướng này có thể bỏ qua vì nó không ảnh hưởng tới nghiệm của bài toán tối ưu, nó chỉ

làm thay đổi giá trị của hàm mục tiêu). Nhắc lại rằng ký hiệu $\preceq$ nghĩa là mỗi phần tử trong vector (ma trận) ở vế trái nhỏ hơn hoặc bằng phần tử tương ứng trong vector (ma trân) ở về phải.

Chú ý rằng nhiều bất đẳng thức dạng $\mathbf{g}_i\mathbf{x} \leq h_i$, với $\mathbf{g}_i$ là các vector hàng, có thể viết gộp dưới dạng $\mathbf{G}\mathbf{x} \preceq \mathbf{h}$ trong đó mỗi hàng của $\mathbf{G}$ ứng với một $\mathbf{g}_i$, mỗi phần tử của $\mathbf{h}$ tương ứng với một h_i.

17.4.2 Dạng tiêu chuẩn của LP

Trong dạng tiêu chuẩn (*standard form*) LP, các bất đẳng thức ràng buộc chỉ là điều kiện các nghiệm có thành phần không âm:

A standard form LP:

$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \succeq \mathbf{0} \end{aligned} \tag{17.23}$$

Bài toán (17.22) có thể đưa về bài toán (17.23) bằng cách đặt thêm biến *slack* $\mathbf{s}$.

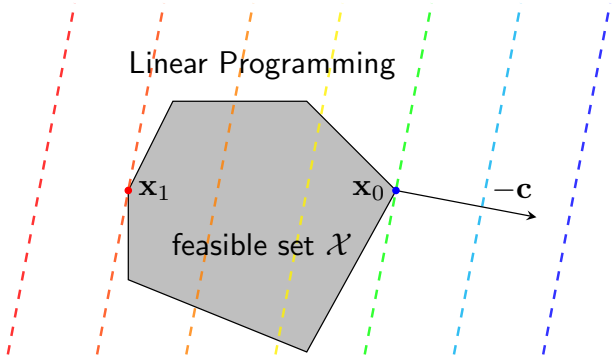
$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x},\mathbf{s}} \mathbf{c}^T\mathbf{x} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{G}\mathbf{x} + \mathbf{s} = \mathbf{h} \\ & \mathbf{s} \succeq \mathbf{0} \end{aligned} \tag{17.24}$$

Tiếp theo, nếu ta biểu diễn $\mathbf{x}$ dưới dạng hiệu của hai vector mà thành phần của nó đều không âm, tức: $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$, với $\mathbf{x}^+, \mathbf{x}^- \succeq 0$. Ta có thể tiếp tục viết lại (17.24) dưới dạng:

$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}^+,\mathbf{x}^-,\mathbf{s}} \mathbf{c}^T\mathbf{x}^+ - \mathbf{c}^T\mathbf{x}^- \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x}^+ - \mathbf{A}\mathbf{x}^- = \mathbf{b} \\ & \mathbf{G}\mathbf{x}^+ - \mathbf{G}\mathbf{x}^- + \mathbf{s} = \mathbf{h} \\ & \mathbf{x}^+ \succeq 0, \mathbf{x}^- \succeq 0, \mathbf{s} \succeq \mathbf{0} \end{aligned} \tag{17.25}$$

Tới đây, bạn đọc có thể thấy rằng (17.25) có thể viết gọn lại như (17.23).

Bài toán nhà xuất bản và Bài toán canh tác trong phần đầu của bài viết này chính là các LP.



Hình 17.3: Biểu diễn hình học của Linear Programming.

17.4.3 Minh hoạ bằng hình học của bài toán LP

Các bài toán LP có thể được minh hoạ như Hình 17.3.

Điểm $\mathbf{x}_0$ chính là điểm làm cho hàm mục tiêu đạt giá trị nhỏ nhất, điểm $\mathbf{x}_1$ chính là điểm làm cho hàm mục tiêu đạt giá trị lớn nhất. Với các bài toán LP, nghiệm, nếu có, thường là một điểm ở *đỉnh* của polyheron hoặc là một *mặt* của polyhedron đó (trong trường hợp các đường level sets của hàm mục tiêu song song với mặt đó, và trên mặt đó, hàm mục tiêu đạt giá trị tối ưu).

Về LP, các bạn có thể tìm thấy rất nhiều tài liệu cả tiếng Việt (Quy hoạch tuyến tính) và tiếng Anh. Có rất nhiều các bài toán trong thực tế có thể đưa về dạng LP. Phương pháp thường được dùng để giải bài toán này có tên là *simplex* (*đơn hình*). Tôi sẽ không đề cập đến các phương pháp này, thay vào đó, tôi sẽ hướng dẫn các bạn dùng thư viện CVXOPT để giải quyết các bài toán thuộc dạng này.

17.4.4 Giải LP bằng CVXOPT

Tôi sẽ dùng thư viện CVPOPT để giải Bài toán canh tác ở phía trên. Nhắc lại bài toán này:

Bài toán canh tác:

$$\begin{aligned} (x, y) = \arg\max_{x,y} \; & 5x + 3y \\ \text{subject to:} \quad & x + y \leq 10 \\ & 2x + y \leq 16 \\ & x + 4y \leq 32 \\ & x, y \geq 0 \end{aligned} \tag{17.26}$$

Các điều kiện ràng buộc có thể viết lại dưới dạng $\mathbf{G}\mathbf{x} \preceq \mathbf{h}$, trong đó:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{h} = \begin{bmatrix} 10 \\ 16 \\ 32 \\ 0 \\ 0 \end{bmatrix}$$

Lời giải cho bài toán này khi dùng CVXOPT là:

```
from cvxopt import matrix, solvers
c = matrix([-5., -3.])
G = matrix([[1., 2., 1., -1., 0.], [1., 1., 4., 0., -1.]])
h = matrix([10., 16., 32., 0., 0.])

solvers.options['show_progress'] = False
sol = solvers.lp(c, G, h)

print('Solution"')
print(sol['x'])
```

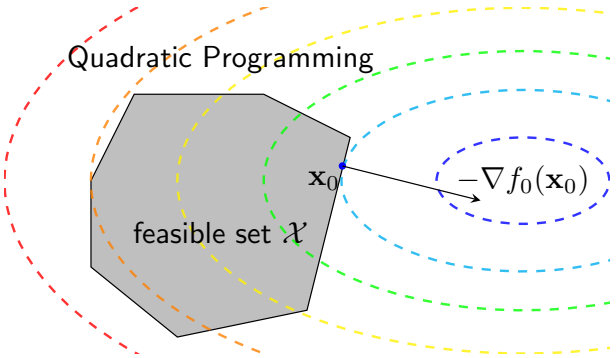
```
Solution:
[ 6.00e+00]
[ 4.00e+00]
```

Nghiệm này chính là nghiệm mà tôi đã tìm được trong phần đầu của bài viết.

Một vài lưu ý:

- Hàm `solvers.lp` của `cvxopt` giải bài toán (17.24).
- Trong bài toán của chúng ta, vì ta cần tìm giá trị lớn nhất nên ta phải đổi hàm mục tiêu về dạng $-5x - 3y$. Chính vì vậy mà `c = matrix([−5., −3.])`.
- Hàm `matrix` nhận đầu vào là một **`list`** (trong Python), **`list`** này thể hiện một vector cột. Nếu muốn biểu diễn một ma trận, đầu vào của `matrix` là một **`list`** của **`list`**, trong đó mỗi **`list`** bên trong thể hiện một vector cột của ma trận đó.
- Các hằng số trong bài toán cần ở dạng số thực. Nếu chúng là các số nguyên, ta cần thêm dấu `.` vào sau các số đó thể thể hiện đó là số thực. (Tôi thấy điểm này hơi thừa, nhưng nếu không có dấu `.` thì chương trình sẽ báo lỗi.)
- Với đẳng thức ràng buộc $\mathbf{Ax} = \mathbf{b}$, `solvers.lp` lấy giá trị mặc định của `A` và `b` là `None`, tức nếu không khái báo thì nghĩa là không có đẳng thức ràng buộc nào.
- Với các tuỳ chọn khác, bạn đọc có thể tìm trong Tài liệu của CVXOPT.

Việc giải Bài toán nhà xuất bản bằng CVXOPT xin nhường lại cho bạn đọc như một bài tập đơn giản.



Hình 17.4: Biểu diễn hình học của Quadratic Programming.

17.5 Quadratic Programming

17.5.1 Định nghĩa bài toán Quadratic Programming

Một dạng Convex Optimization mà các bạn sẽ gặp rất nhiều trong các bài sau của blog là *Quadratic Programming* (QP, hoặc *Quadratic Program*). Khác biệt duy nhất của QP so với LP là hàm mục tiêu có dạng *Quadratic form*:

Quadratic Programming:

$$\begin{aligned} \mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + \mathbf{r} \\ \text{subject to:} \quad \mathbf{G}\mathbf{x} \preceq \mathbf{h} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{17.27}$$

Trong đó $\mathbf{P} \in \mathbb{S}_+^n$ (tập các ma trận vuông nửa xác định dương có số cột là n), $\mathbf{G} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{p \times n}$. Điều kiện $\mathbf{P}$ là *nửa xác định dương* để đảm bảo hàm mục tiêu là *convex*.

Chúng ta có thể thấy rằng LP chính là một trường hợp đặc biệt của QP với $\mathbf{P} = \mathbf{0}$.

Diễn đạt bằng lời: trong QP, chúng ta tối thiểu một hàm quadratic lồi trên một *polyhedron* (Xem Hình 17.4).

17.5.2 Ví dụ về QP

Bài toán vui: Có một hòn đảo mà hình dạng của nó có dạng một đa giác lồi. Một con thuyền ở ngoài biển thì cần đi theo hướng nào để tới đảo nhanh nhất, giả sử rằng tốc độ của sóng và gió bằng 0.

Bài toán khoảng cách từ một điểm tới một polyhedron được phát biểu như sau:

Cho một polyhedron được biểu diễn bởi $\mathbf{Ax} \preceq \mathbf{b}$ và một điểm $\mathbf{u}$, tìm điểm $\mathbf{x}$ thuộc polyhedron đó sao cho khoảng cách Euclidean giữa $\mathbf{x}$ và $\mathbf{u}$ là nhỏ nhất.

Bài toán này có thể phát biểu như sau:

$$\begin{aligned} \mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 \\ \text{subject to:} \quad \mathbf{Gx} \preceq \mathbf{h} \end{aligned}$$

Hàm mục tiêu đạt giá trị nhỏ nhất bằng 0 nếu $\mathbf{u}$ nằm trong polyheron đó và *optimal point* chính là $\mathbf{x} = \mathbf{u}$. Khi $\mathbf{u}$ không nằm trong polyhedron, ta viết:

$$\frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 = \frac{1}{2}(\mathbf{x}-\mathbf{u})^T(\mathbf{x}-\mathbf{u}) = \frac{1}{2}\mathbf{x}^T\mathbf{x} - \mathbf{u}^T\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{u}$$

Biểu thức này có dạng hàm mục tiêu như trong (17.27) với $\mathbf{P} = \mathbf{I}, \mathbf{q} = -\mathbf{u}, \mathbf{r} = \frac{1}{2}\mathbf{u}^T\mathbf{u}$, trong đó $\mathbf{I}$ là ma trận đơn vị.

17.5.3 Ví dụ về giải QP bằng CVXOPT

Xét bài toán sau đây (xem Hình 17.5):

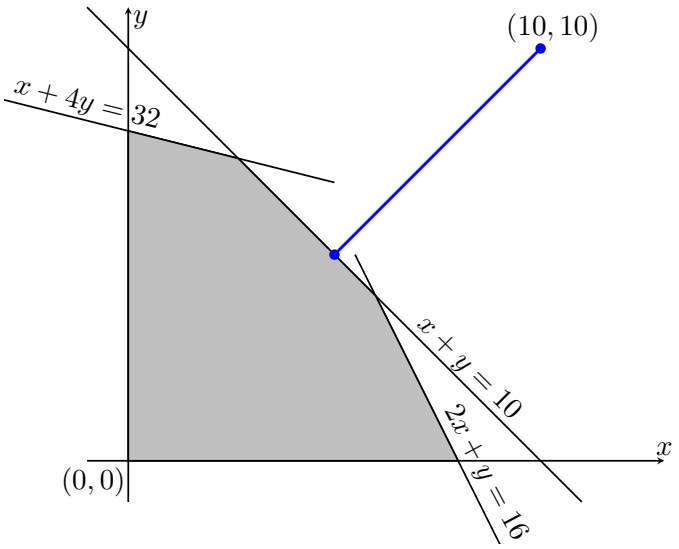
$$\begin{aligned} (x, y) = \arg\min_{x,y} (x-10)^2 + (y-10)^2 \\ \text{subject to:} \quad \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 4 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \preceq \begin{bmatrix} 10 \\ 16 \\ 32 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Feasible set trong bài toán này tôi lấy trực tiếp từ Bài toán canh tác và $\mathbf{u} = [10, 10]^T$. Bài toán này có thể được giải bằng CVXOPT như sau:

```
from cvxopt import matrix, solvers
P = matrix([[1., 0.], [0., 1.]])
q = matrix([-10., -10.])
G = matrix([[1., 2., 1., -1., 0.], [1., 1., 4., 0., -1.]])
h = matrix([10., 16., 32., 0., 0])

solvers.options['show_progress'] = False
sol = solvers.qp(P, q, G, h)

print('Solution:')
print(sol['x'])
```



Hình 17.5: Ví dụ về khoảng cách giữa một điểm và một polyhedron.

```
Solution:
[ 5.00e+00]
[ 5.00e+00]
```

Trong các thuật toán Machine Learning, các bạn sẽ gặp các bài toán về tìm *hình chiếu* (projection) của một điểm lên một *tập lồi* nói chung rất nhiều. Tới từng phần, tôi sẽ đề cập hướng giải quyết của các bài toán đó.

17.6 Geometric Programming

Trong mục này, chúng ta sẽ thấy một lớp các bài toán *không lồi* khi nhìn vào hàm mục tiêu và các hàm ràng buộc, nhưng có thể được biến đổi về dạng *lồi* bằng một vài kỹ thuật.

Trước hết, chúng ta cần có một vài định nghĩa:

17.6.1 Monomials và posynomials

Một hàm số $f : \mathbb{R}^n \to \mathbb{R}$ với tập xác đinh $\text{dom} f = \mathbf{R}^n_{++}$ (tất cả các phần tử đều là số dương) có dạng:

$$f(\mathbf{x}) = cx_1^{a_1}x_2^{a_2}\dots x_n^{a_n} \tag{17.28}$$

trong đó $c > 0$ và $a_i \in \mathbb{R}$, được gọi là một *monomial function* (khái niệm này khá giống với *đơn thức* khi tôi học lớp 8, nhưng khi đó SGK định nghĩa với c bất kỳ và a_i là các số tự nhiên).

Tổng của các monomials:

$$f(\mathbf{x}) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \dots x_n^{a_{nk}}$$

trong đó các $c_k > 0$ được gọi là *posynomial function* (*đa thức*), hoặc đơn giản là *posynomial*.

17.6.2 Geometric Programming

Một bài toán tối ưu có dạng:

$$\begin{aligned} \mathbf{x} = \arg\min_{\mathbf{x}} \; & f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(x) \leq 1, \quad i = 1, 2, \dots, m \\ & h_j(x) = 1, \quad j = 1, 2, \dots, p \end{aligned} \tag{17.29}$$

trong đó $f_0, f_1, \dots, f_m$ là các *posynomials* và $h_1, \dots, h_p$ là các *monomials*, được gọi là *Geometric Programming* (GP). Điều kiện $\mathbf{x} \succ 0$ được ẩn đi.

Chú ý rằng nếu f là một *posynomial*, h là một *monomial* thì f/h là một *posynomial*.

Ví dụ:

$$\begin{aligned} (x, y, z) = \arg\min_{x,y,z} \; & x/y \\ \text{subject to:} \quad & 1 \leq x \leq 2 \\ & x^3 + 2y/z \leq \sqrt{y} \\ & x/y = z \end{aligned} \tag{17.30}$$

Có thể được viết lại dưới dạng GP:

$$\begin{aligned} (x, y, z) = \arg\min_{x,y,z} \; & xy^{-1} \\ \text{subject to:} \quad & x^{-1} \leq 1 \\ & (1/2)x \leq 1 \\ & x^3 y^{-1/2} + 2y^{1/2} z^{-1} \leq 1 \\ & xy^{-1} z^{-1} = 1 \end{aligned} \tag{17.31}$$

Bài toán này rõ ràng là *nonconvex* vì cả hàm mục tiêu và điều kiển ràng buộc đều không lồi.

17.6.3 Biến đổi GP về dạng convex

GP có thể được biến đổi về dạng lồi như sau:

Đặt $y_i = \log(x_i)$, tức $x_i = \exp(y_i)$. Nếu f là một *monomial function* của $\mathbf{x}$ thì:

$$f(\mathbf{x}) = c(\exp(y_1))^{a_1} \dots (\exp(y_n))^{a_n} = \exp(\mathbf{a}^T\mathbf{y} + b)$$

với $b = \log(c)$. Lúc này, hàm số $g(y) = \exp(\mathbf{a}^T\mathbf{y} + b)$ là một hàm lồi theo $\mathbf{y}$. (Bạn đọc có thể chứng minh theo định nghĩa rằng hợp của hai hàm lồi là một hàm lồi. Trong trường hợp này, hàm exp và hàm *affine* trên đều là các hàm lồi.)

Tương tự như thế, *posynomial* trong đẳng thức (24) có thể viết dưới dạng:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \exp(\mathbf{a}_k^T\mathbf{y} + b_k)$$

trong đó $\mathbf{a}_k = [a_{1k}, \dots, a_{nk}]^T$ và $b_k = \log(c_k)$. Lúc này, *posynomial* đã được viết dưới dạng tổng của các hàm exp của các hàm *affine* (và vì vậy là một hàm lồi, nhớ lại rằng tổng của các hàm lồi là một hàm lồi).

Bài toán GP (17.29) được viết lại dưới dạng:

$$\begin{aligned} \mathbf{y} = \arg\min_{\mathbf{y}} & \sum_{k=1}^{K_0} \exp(\mathbf{a}_{0k}^T\mathbf{y} + b_{0k}) \\ \text{subject to:} \quad & \sum_{k=1}^{K_i} \exp(\mathbf{a}_{ik}^T\mathbf{y} + b_{ik}) \leq 1, \quad , i = 1, \dots, m \\ & \exp(\mathbf{g}_j^T\mathbf{y} + h_j) = 1, \ j = 1, \dots, p \end{aligned} \tag{17.32}$$

với $\mathbf{a}_{ik} \in \mathbb{R}^n, i = 1, \dots, p$ và $\mathbf{g}_i \in \mathbb{R}^n$.

Với chú ý rằng hàm số $\log \sum_{i=1}^{m} \exp(g_i(\mathbf{x}))$ là môt hàm *lồi* nếu g_i là các hàm *lồi* (tôi xin bỏ qua phần chứng minh), ta có thể viết lại bài toán (17.32) dưới dạng *lồi* bằng cách lấy log của các hàm như sau:

GP in convex form:

$$\begin{aligned} \text{minimize}_{\mathbf{y}} \tilde{f}_0(\mathbf{y}) &= \log\left(\sum_{k=1}^{K_0} \exp(\mathbf{a}_{0k}^T\mathbf{y} + b_{i0})\right) \\ \text{subject to: } \tilde{f}_i(\mathbf{y}) &= \log\left(\sum_{k=1}^{K_i} \exp(\mathbf{a}_{ik}^T\mathbf{y} + b_{ik})\right) \leq 0, \quad i = 1, \dots, m \\ \tilde{h}_j(\mathbf{y}) &= \mathbf{g}_j^T\mathbf{y} + h_j = 0, \quad j = 1, \dots, p \end{aligned} \tag{17.33}$$

Lúc này, ta có thể nói rằng GP tương đương với một bài toán tối ưu lồi vì hàm mục tiêu và các hàm bất đẳng thức ràng buộc trong (17.33) đều là hàm lồi, đồng thời điều hiện đẳng thức cuối cùng chính là dạng *affine*. Dạng này thường được gọi là *geometric program in convex form* (để phân biệt nó với định nghĩa của GP).

17.6.4 Giải GP bằng CVXOPT

Chúng ta quay lại ví dụ về Bài toán đóng thùng *không có ràng buộc* và hàm mục tiêu là $f(x, y, z) = 40x^{-1}y^{-1}z^{-1} + 2xy + 2yz + 2zx$ là một posynomial. Vậy đây là một GP.

Code cho việc tìm *optimal point* của bài toán này bằng CVXOPT như sau:

```
from cvxopt import matrix, solvers
from math import log, exp# gp
from numpy import array
import numpy as np

K = [4]
F = matrix([[-1., 1., 1., 0.],
            [-1., 1., 0., 1.],
            [-1., 0., 1., 1.]])
g = matrix([log(40.), log(2.), log(2.), log(2.)])
solvers.options['show_progress'] = False
sol = solvers.gp(K, F, g)

print('Solution:')
print(np.exp(np.array(sol['x'])))

print('\nchecking sol^5')
print(np.exp(np.array(sol['x']))**5)
```

```
Solution:
[[ 1.58489319]
 [ 1.58489319]
 [ 1.58489319]]

checking sol^5
[[ 9.9999998]
 [ 9.9999998]
 [ 9.9999998]]
```

Nghiệm thu được chính là $x = y = z = \sqrt[5]{10}$. Bạn đọc được khuyến khích đọc thêm chỉ dẫn của hàm `solvers.gp` để hiểu cách thiết lập bài toán.

17.7 Tóm tắt

- Các bài toán tối ưu xuất hiện rất nhiều trong thực tế, trong đó Tối Ưu Lồi đóng một vai trò quan trọng. Trong bài toán Tối Ưu Lồi, nếu tìm được cực trị thì cực trị đó chính là một điểm *optimal* của bài toán (nghiệm của bài toán).
- Có nhiều bài toán tối ưu không được viết dưới dạng *convex* nhưng có thể biến đổi về dạng *convex*, ví dụ như bài toán Geometric Programming.
- Linear Programming và Quadratic Programming đóng một vài trò quan trọng trong toán tối ưu, được sử dụng nhiều trong các thuật toán Machine Learning.
- Thư viện CVXOPT được dùng để tối ưu nhiều bài toán tối ưu lồi, rất dễ sử dụng và thời gian chạy tương đối nhanh.

17.8 Tài liệu tham khảo

[1] Convex Optimization – Boyd and Vandenberghe, Cambridge University Press, 2004.

[2] CVXOPT.

18

Duality

Trong bài viết này, chúng ta giả sử rằng các đạo hàm đều tồn tại.

Bài viết này chủ yếu được dịch lại từ Chương 5 của cuốn *Convex Optimization* trong tài liệu tham khảo.

18.1 Giới thiệu

Trong Bài 16, chúng ta đã làm quen với các khái niệm về tập hợp lồi và hàm số lồi. Tiếp theo đó, trong Bài 17, tôi cũng đã trình bày về các bài toán tối ưu lồi, cách nhận dạng và cách sử dụng thư viện để giải các bài toán lồi cơ bản. Trong bài này, chúng ta sẽ tiếp tục tiếp cận một cách sâu hơn: các điều kiện về nghiệm của các bài toán tối ưu, cả lồi và không lồi; bài toán đối ngẫu (dual problem) và điều kiện KKT.

Trước tiên, chúng ta lại bắt đầu bằng những kỹ thuật đơn giản cho các bài toán cơ bản. Kỹ thuật này có lẽ các bạn đã từng nghe đến: Phương pháp nhân tử Lagrange (method of Lagrange multipliers). Đây là một phương pháp giúp tìm các điểm cực trị của hàm mục tiêu trên feasible set của bài toán.

Nhắc lại rằng giá trị lớn nhất và nhỏ nhất (nếu có) của một hàm số $f_0(\mathbf{x})$ khả vi (và tập xác định là một *tập mở*) đạt được tại một trong các điểm cực trị của nó. Và điều kiện cần để một điểm là điểm cực trị là đạo hàm của hàm số tại điểm này $f_0'(x) = 0$. Chú ý rằng một điểm thoả mãn $f_0'(\mathbf{x}) = 0$ thì được gọi là *điểm dừng* hay *stationary point*. Điểm cực trị là một điểm dừng nhưng không phải điểm dừng nào cũng là điểm cực trị. Ví dụ hàm $f(x) = x^3$ có 0 là một điểm dừng nhưng không phải là điểm cực trị.

Với hàm nhiều biến, ta cũng có thể áp dụng quan sát này. Tức chúng ta cần đi tìm nghiệm của phương trình đạo hàm *theo mỗi biến* bằng 0. Tuy nhiên, đó là với các bài toán không ràng buộc (unconstrained optimization problems), với các bài toán có ràng buộc như chúng ta đã gặp trong Bài 17 thì sao?

Trước tiên chúng ta xét bài toán mà ràng buộc chỉ là một phương trình:

$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}} f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_1(\mathbf{x}) = 0 \end{aligned} \tag{18.1}$$

Bài toán này là bài toán tổng quát, không nhất thiết phải lồi. Tức hàm mục tiêu và hàm ràng buộc không nhất thiết phải lồi.

18.2 Phương pháp nhân tử Lagrange

Nếu chúng ta đưa được bài toán này về một bài toán không ràng buộc thì chúng ta có thể tìm được nghiệm bằng cách giải hệ phương trình đạo hàm theo từng thành phần bằng 0 (giả sử rằng việc giải hệ phương trình này là khả thi).

Điều này là động lực để nhà toán học Lagrange sử dụng hàm số: $\mathcal{L}(\mathbf{x}, \lambda) = f_0(\mathbf{x}) + \lambda f_1(\mathbf{x})$. Chú ý rằng, trong hàm số này, chúng ta có thêm một biến nữa là λ, biến này được gọi là nhân tử Lagrange (Lagrange multiplier). Hàm số $\mathcal{L}(\mathbf{x}, \lambda)$ được gọi là *hàm hỗ trợ* (*auxiliary function*), hay *the Lagrangian*. Người ta đã chứng minh được rằng, điểm *optimal value* của bài toán (18.1) thoả mãn điều kiện $\nabla_{\mathbf{x},\lambda}\mathcal{L}(\mathbf{x}, \lambda) = 0$ (tôi xin được bỏ qua chứng minh của phần này). Điều này tương dương với:

$$\nabla_{\mathbf{x}} f_0(\mathbf{x}) + \lambda \nabla_{\mathbf{x}} f_1(\mathbf{x}) = 0 \tag{18.2}$$

$$f_1(\mathbf{x}) = 0 \tag{18.3}$$

Để ý rằng điều kiện thứ hai chính là $\nabla_\lambda \mathcal{L}(\mathbf{x}, \lambda) = 0$, và cũng chính là ràng buộc trong bài toán (18.1).

Việc giải hệ phương trình (18.2) - (18.3), trong nhiều trường hợp, đơn giản hơn việc trực tiếp đi tìm *optimal value* của bài toán (18.1).

Xét các ví dụ đơn giản sau đây.

18.2.1 Ví dụ

Ví dụ 1: Tìm giá trị lớn nhất và nhỏ nhất của hàm số $f_0(x, y) = x + y$ thoả mãn điều kiện $f_1(x, y) = x^2 + y^2 = 2$. Ta nhận thấy rằng đây không phải là một bài toán tối ưu lồi vì *feasible set* $x^2 + y^2 = 2$ không phải là một tập lồi (nó chỉ là một đường tròn).

Lời giải:

Lagrangian của bài toán này là: $\mathcal{L}(x, y, \lambda) = x + y + \lambda(x^2 + y^2 - 2)$. Các điểm cực trị của hàm số Lagrange phải thoả mãn điều kiện:

$$\nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda)=0 \Leftrightarrow \begin{cases} 1+2\lambda x &= 0 \\ 1+2\lambda y &= 0 \\ x^2+y^2 &= 2 \end{cases} \tag{18.4}$$

Từ hai phương trình đầu của (18.4) ta suy ra $x=y=\frac{-1}{2\lambda}$. Thay vào phương trình ta sẽ có $\lambda^2=\frac{1}{4} \Rightarrow \lambda=\pm\frac{1}{2}$. Vậy ta được 2 cặp nghiệm $(x,y)\in\{(1,1),(-1,-1)\}$. Bằng cách thay các giá trị này vào hàm mục tiêu, ta tìm được giá trị nhỏ nhất và lớn nhất của hàm số cần tìm.

Ví dụ 2: Cross-entropy. Trong Chương 10 và Chương 13, chúng ta đã được biết đến hàm mất mát ở dạng *cross entropy*. Chúng ta cũng đã biết rằng hàm cross entropy được dùng để đo sự giống nhau của hai phân phối xác suất với giá trị của hàm số này càng nhỏ thì hai xác suất càng gần nhau. Chúng ta cũng đã phát biểu rằng giá trị nhỏ nhất của hàm cross entopy đạt được khi từng gặp xác suất là giống nhau. Bây giờ, tôi xin phát biểu lại và chứng minh nhận định trên.

Cho một phân bố xác xuất $\mathbf{p}=[p_1,p_2,\ldots,p_n]^T$ với $p_i\in[0,1]$ và $\sum_{i=1}^n p_i=1$. Với một phân bố xác suất bất kỳ $\mathbf{q}=[q_1,q_2,\ldots,q_n]$ và giả sử rằng $q_i\neq 0,\forall i$, hàm số cross entropy được định nghĩa là:

$$f_0(\mathbf{q})=-\sum_{i=1}^n p_i\log(q_i)$$

Hãy tìm $\mathbf{q}$ để hàm cross entropy đạt giá trị nhỏ nhất.

Trong bài toán này, ta có ràng buộc là $\sum_{i=1}^n q_i=1$. *Lagrangian* của bài toán là:

$$\mathcal{L}(q_1,q_2,\ldots,q_n,\lambda)=-\sum_{i=1}^n p_i\log(q_i)+\lambda(\sum_{i=1}^n q_i-1)$$

Ta cần giải hệ phương trình:

$$\nabla_{q_1,\ldots,q_n,\lambda}\mathcal{L}(q_1,\ldots,q_n,\lambda)=0 \Leftrightarrow \begin{cases} -\frac{p_i}{q_i}+\lambda &= 0, \quad i=1,\ldots,n \\ q_1+q_2+\cdots+q_n &= 1 \end{cases}$$

Từ phương trình thứ nhất ta có $p_i=\lambda q_i$. Vậy nên: $1=\sum_{i=1}^n p_i=\lambda\sum_{i=1}^n q_i=\lambda \Rightarrow \lambda=1 \Rightarrow q_i=p_i,\forall i$.

Qua đây, chúng ta đã hiểu rằng vì sao hàm số cross entropy được dùng để *ép* hai xác suất *gần nhau*.

18.3 Hàm đối ngẫu Lagrange (The Lagrange dual function)

18.3.1 Lagrangian

Với bài toán tối ưu tổng quát:

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, p \end{aligned} \tag{18.5}$$

với miền xác đinh $\mathcal{D} = (\cap_{i=0}^{m} \text{dom} f_i) \cap (\cap_{j=1}^{p} \text{dom} h_j)$. Chú ý rằng, chúng ta đang không giả sử về tính chất lồi của hàm tối ưu hay các hàm ràng buộc ở đây. Giả sử duy nhất ở đây là $\mathcal{D} \neq \emptyset$ (tập rỗng).

Lagrangian cũng được xây dựng tương tự với mỗi nhân tử Lagrange cho một (bất) phương trình ràng buộc:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{p} \nu_j h_j(\mathbf{x})$$

với $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]; \boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_p]$ là các vectors và được gọi là *dual variables* (*biến đối ngẫu*) hoặc *Lagrange multiplier vectors* (vector nhân tử Lagrange). Lúc này nếu biến chính $\mathbf{x} \in \mathbb{R}^n$ thì tổng số biến của hàm số này sẽ là $n + m + p$.

18.3.2 Hàm đối ngẫu Lagrange

Hàm đối ngẫu Lagrange của bài toán tối ưu (hoặc gọn là *hàm số đối ngẫu*) (18.5) là một hàm của các biến đối ngẫu, được định nghĩa là giá trị nhỏ nhất theo $\mathbf{x}$ của *Lagrangian*:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{18.6}$$

$$= \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{p} \nu_j h_j(\mathbf{x}) \right) \tag{18.7}$$

Nếu *Lagrangian không bị chặn dưới*, hàm đối ngẫu tại $\boldsymbol{\lambda}, \boldsymbol{\nu}$ sẽ lấy giá trị $-\infty$.

Đặc biệt quan trọng:

- inf được lấy trên miền $x \in \mathcal{D}$, tức miền xác định của bài toán (là giao của miền xác định của mọi hàm trong bài toán). Miền xác định này khác với *feasible set*. Thông thường, *feasible set* là tập con của miền xác định $\mathcal{D}$.

- Với mỗi $\mathbf{x}$, *Lagrangian* là một hàm *affine* của $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, tức là một hàm *concave*. Vậy, *hàm đối ngẫu* chính là *pointwise infimum* của (có thể vô hạn) các hàm concave, tức là một hàm concave. Vậy **hàm đối ngẫu của một bài toán tối ưu bất kỳ là một hàm concave, bất kể bài toán ban đầu có phải là convex hay không**. Nhắc lại rằng *pointwise supremum* của các hàm *convex* là một hàm *convex*, và một hàm là *concave* nếu khi đổi dấu hàm đó, ta được một hàm *convex*.

18.3.3 Chặn dưới của giá trị tối ưu

Nếu p^* là *optimal value* (giá trị tối ưu) của bài toán (18.5), thì với các biến đối ngẫu $\lambda_i \geq 0, \forall i$ và ν *bất kỳ*, chúng ta sẽ có:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^* \tag{18.8}$$

Tính chất này có thể được chứng minh dễ dàng. Giả sử $\mathbf{x}_0$ là một điểm *feasible* bất kỳ của bài toán (18.5), tức thoả mãn các điều kiện ràng buộc $f_i(\mathbf{x}_0) \leq 0, \forall i = 1, \dots, m; h_j(\mathbf{x}_0) = 0, \forall j = 1, \dots, p$, ta sẽ có:

$$\sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}_0) + \sum_{j=1}^{p} \nu_j h_j(\mathbf{x}_0) \leq 0 \Rightarrow \mathcal{L}(\mathbf{x}_0, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\mathbf{x}_0)$$

Vì điều này đúng với mọi $\mathbf{x}_0$ *feasible*, ta sẽ có tính chất quan trọng sau đây:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq \mathcal{L}(\mathbf{x}_0, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\mathbf{x}_0).$$

Khi $\mathbf{x}_0 = \mathbf{x}^*$, ta có bất đẳng thức (18.8).

18.3.4 Ví dụ

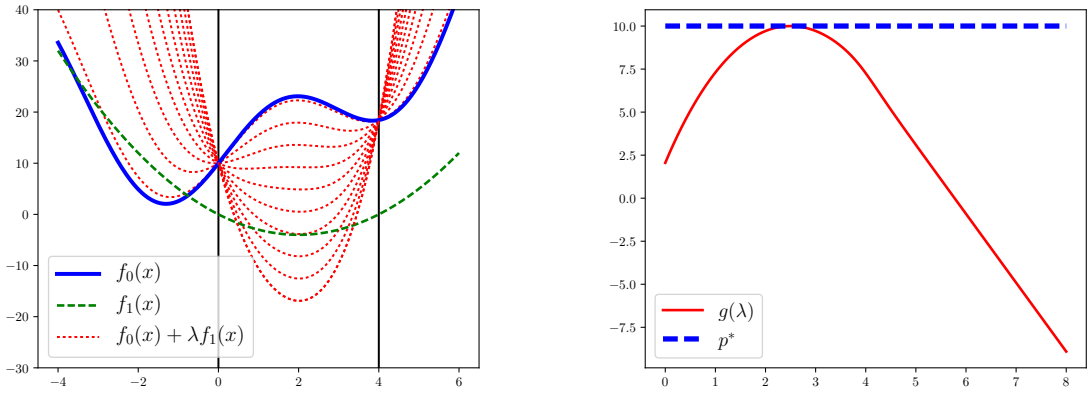
Ví dụ 1

Xét bài toán tối ưu sau:

$$\begin{aligned} x = \arg\min_x x^2 + 10\sin(x) + 10 \\ \text{subject to:} \quad (x-2)^2 \leq 4 \end{aligned} \tag{18.9}$$

Chú ý: Với bài toán này, miền xác định $\mathcal{D} = \mathbb{R}$ nhưng *feasible set* là $0 \leq x \leq 4$.

Với hàm mục tiêu là đường đậm màu xanh lam trong Hình 18.1. Ràng buộc thực ra $0 \leq x \leq 4$, nhưng tôi viết ở dạng này để bài toán thêm phần thú vị. Hàm số ràng buộc $f_1(x) = (x-2)^2 - 4$ được cho bởi đường nét đứt màu xanh lục. Optimal value của bài toán này có thể được nhận



Hình 18.1: Ví dụ về dual function. Trái: Đường màu lam đậm thể hiện hàm mục tiêu. Đường nét đứt mà lục thể hiện hàm số ràng buộc. Các đường nét đứt màu đỏ thể hiện dual function ứng với các λ khác nhau. Phải: Đường nét đứt thể hiện giá trị tối ưu của bài toán . Đường màu đỏ thể hiện dual function. Với mọi λ, giá trị của hàm dual function nhỏ hơn hoặc bằng giá trị tối ưu của bài toán gốc.

ra là điểm trên đồ thị có hoành độ bằng 0. Chú ý rằng hàm mục tiêu ở đây không phải là hàm lồi nên bài toán tối ưu này cũng không phải là lồi, mặc dù hàm bất phương trình ràng buộc $f_1(x)$ là lồi.

Lagrangian của bài toàn này có dạng:

$$\mathcal{L}(x, \lambda) = x^2 + 10\sin(x) + 10 + \lambda((x-2)^2 - 4)$$

Các đường dấu chấm màu đỏ trong Hình 1 là các đường ứng với các λ khác nhau. Vùng bị chặn giữa hai đường thẳng đứng màu đen thể hiện miền *feasible* của bài toán tối ưu.

Với mỗi λ, *dual function* được định nghĩa là:

$$g(\lambda) = \inf_x \left(x^2 + 10\sin(x) + 10 + \lambda((x-2)^2 - 4)\right), \quad \lambda \geq 0.$$

Từ hình 1 bên trái, ta có thể thấy ngay rằng với các λ khác nhau, $g(\lambda)$ hoặc tại điểm có hoành độ bằng 0, hoặc tại một điểm thấp hơn điểm tối ưu của bài toán. Đồ thị của hàm $g(\lambda)$ được cho bởi đường liền màu đỏ ở Hình 1 bên phải. Đường nét đứt màu lam thể hiện *optimal value* của bài toán tối ưu ban đầu. Ta có thể thấy ngay hai điều:

- Đường liền màu đỏ luôn nằm dưới (hoặc có đoạn trùng) với đường nét đứt màu lam.
- Hàm $g(\lambda)$ có dạng một hàm *concave*, tức nếu ta *lật* đồ thị này theo hướng trên-dưới thì ta sẽ có đồ thị của một hàm *convex*. (Mặc dù bài toán tối ưu gốc là không phải là một bài toán lồi.)

(*Để vẽ được hình bên phải, tôi đã dùng Gradient Descent để tìm giá trị nhỏ nhất ứng với mỗi* λ.)

Ví dụ 2

Xét một bài toán Linear Programming:

$$\begin{aligned} x = \arg\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \\ \text{s.t.:} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \succeq 0 \end{aligned} \tag{18.10}$$

Hàm ràng buộc cuối cùng có thể được viết lại là: $f_i(\mathbf{x}) = -x_i, i = 1, \dots, n$. Lagrangigan của bài toán này là:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \mathbf{c}^T\mathbf{x} - \sum_{i=1}^{n} \lambda_i x_i + \boldsymbol{\nu}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = -\mathbf{b}^T\boldsymbol{\nu} + (\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda})^T\mathbf{x}$$

(đừng quên điều kiện $\boldsymbol{\lambda} \succeq 0$.)

Dual function là:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{18.11}$$

$$= -\mathbf{b}^T\nu + \inf_{\mathbf{x}}(\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda})^T\mathbf{x} \tag{18.12}$$

Nhận thấy rằng một hàm tuyến tính $\mathbf{d}^T\mathbf{x}$ của $\mathbf{x}$ bị chặn dưới khi vào chỉ khi $\mathbf{d} = 0$. Vì chỉ nếu một phần tử d_i của $\mathbf{d}$ khác 0, ta chỉ cần chọn x_i rất lớn và ngược dấu với d_i, ta sẽ có một giá trị nhỏ tuỳ ý.

Nói cách khác, $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -\infty$ trừ khi $\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda} = 0$. Tóm lại:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \begin{cases} -\mathbf{b}^T\boldsymbol{\nu} & \text{if } \mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda} = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{18.13}$$

Trường hợp thứ hai khi $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -\infty$ các bạn sẽ gặp rất nhiều sau này. Trường hợp này không nhiều thú vị vì hiển nhiên $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^*$. Vì mục đích chính là đi tìm chặn dưới của p^* nên ta sẽ chỉ quan tâm tới các giá trị của $\boldsymbol{\lambda}$ và $\boldsymbol{\nu}$ sao cho $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ càng lớn càng tốt. Trong bài toán này, ta sẽ quan tâm tới các $\boldsymbol{\lambda}$ và $\boldsymbol{\nu}$ sao cho $\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda} = 0$.

18.4 Bài toán đối ngẫu Lagrange (The Lagrange dual problem)

Với mỗi cặp $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, hàm đối ngẫu Lagrange cho chúng ta một chặn dưới cho *optimal value* p^* của bài toán gốc (18.5). Câu hỏi đặt ra là: với cặp giá trị nào của $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, chúng ta sẽ có một chặn dưới tốt nhất của p^*? Nói cách khác, ta đi cần giải bài toán:

$$\boldsymbol{\lambda}^*, \boldsymbol{\nu}^* = \arg\max_{\boldsymbol{\lambda}, \boldsymbol{\nu}} g(\boldsymbol{\lambda}, \boldsymbol{\nu})$$
$$\text{subject to:} \quad \boldsymbol{\lambda} \succeq 0 \tag{18.14}$$

Một điểm quan trọng: vì $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ là *concave* và hàm ràng buộc $f_i(\boldsymbol{\lambda}) = -\lambda_i$ là các hàm *convex*. Vậy bài toán (18.14) chính là một bài toán lồi. Vì vậy trong nhiều trường hợp, lời giải có thể dễ tìm hơn là bài toán gốc. Chú ý rằng, bài toán đối ngẫu (18.14) là lồi bất kể bài toán gốc (18.5) có là lồi hay không.

Bài toán này dược gọi là *Lagrange dual problem* (bài toán đối ngẫu Largange) ứng với bài toán (18.5). Bài toán (18.5) còn có tên gọi khác là *primal problem* (bài toán gốc). Ngoài ra, có một khái niệm nữa, gọi là *dual feasible* tức là *feasible set* của bài toán đối ngẫu, bao gồm điều kiện $\boldsymbol{\lambda} \succeq 0$ và điều kiện ẩn $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) > -\infty$ (vì ta đang đi tìm giá trị lớn nhất của hàm số nên $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -\infty$ rõ ràng là không thú vị).

Nghiệm của bài toán (18.14), ký hiệu là $\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*$ được gọi là *dual optimal* hoặc *optimal Lagrange multipliers*.

Chú ý rằng điều kiện ẩn $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) > -\infty$, trong nhiều trường hợp, cũng có thể được viết cụ thể. Quay lại với ví dụ phía trên, điệu kiện ẩn có thể được viết thành $\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda} = 0$. Đây là một hàm affine. Vì vậy, khi có thêm ràng buộc này, ta vẫn được một bài toán lồi.

18.4.1 Weak duality

Ký hiệu giá trị tối ưu của bài toán đối ngẫu (18.14) là d^*. Theo (18.14), ta đã biết rằng:

$$d^* \leq p^*$$

ngay cả khi bài toán gốc không phải là lồi.

Tính chất đơn giản này được gọi là *weak duality*. Tuy đơn giản nhưng nó cực kỳ quan trọng.

Từ đây ta quan sát thấy hai điều:

- Nếu bài toán gốc không bị chặn dưới, tức $p^* = -\infty$, ta phải có $d^* = -\infty$, tức là bài toán đối ngẫu Lagrange là *infeasible* (tức không có giá trị nào thoả mãn ràng buộc).
- Nếu bài toàn đối ngẫu là không bị chặn trên, tức $d^* = +\infty$, chúng ta phải có $p^* = +\infty$, tức bài toán gốc là *infeasible*.

Giá trị $p^* - d^*$ được gọi là *optimal duality gap* (dịch thô là *khoảng cách đối ngẫu tối ưu*). Khoảng cách này luôn luôn là một số không âm.

Đôi khi có những bài toán (lồi hoặc không) rất khó giải, nhưng ít nhất nếu ta có thể tìm được d^*, ta có thể biết được chặn dưới của bài toán gốc. Việc tìm d^* thường có thể thực hiện được vì bài toán đối ngẫu luôn luôn là lồi.

18.4.2 Strong duality và Slater's constraint qualification

Nếu đẳng thức $p^* = d^*$ thoả mãn, *the optimal duality gap* bằng không, ta nói rằng *strong duality* xảy ra. Lúc này, việc giải bài toán đối ngẫu đã giúp ta tìm được *chính xác* giá trị tối ưu của bài toán gốc.

Thật không may, *strong duality* không thường xuyên xảy ra trong các bài toán tối ưu. Tuy nhiên, nếu bài toán gốc là lồi, tức có dạng:

$$\begin{aligned} x = \arg\min_{\mathbf{x}} & f_0(\mathbf{x}) \\ \text{subject to:} \quad & f_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, m \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{18.15}$$

trong đó $f_0, f_1, \ldots, f_m$ là các hàm lồi, chúng ta *thường* (không luôn luôn) có *strong duality*. Có rất nhiều nghiên cứu thiết lập các điều kiện, ngoài tính chất lồi, để *strong duality* xảy ra. Những điều kiện đó thường có tên là *constraint qualifications*.

Một trong các *constraint qualification* đơn giản nhất là *Slater's condition*.

Định nghĩa: Một điểm *feasible* của bài toán (18.15) được gọi là *strictly feasible* nếu:

$$f_i(\mathbf{x}) < 0, \; i = 1, 2, \ldots, m, \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

Định lý Slater: Nếu tồn tại một điểm *strictly feasible* (và bài toán gốc là lồi), thì *strong duality* xảy ra.

Điều kiện khá đơn giản sẽ giúp ích cho nhiều bài toán tối ưu sau này.

Chú ý:

- *Strong duality* không thường xuyên xảy ra. Với các bài toán lồi, việc này xảy ra thường xuyên hơn. Tồn tại những bài toán lồi mà *strong duality* không xảy ra.
- Có những bài toán không lồi nhưng *strong duality* vẫn xảy ra. Ví dụ như bài toán trong Hình 1 phía trên.

18.5 Optimality conditions

18.5.1 Complementary slackness

Giả sử rằng *strong duality* xảy ra. Gọi $\mathbf{x}^*$ là một điểm *optimal* của bài toán gốc và (λ^*, ν^*) là cặp điểm *optimal* của bài toán đối ngẫu. Ta có:

$$f_0(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) \tag{18.16}$$

$$= \inf_{\mathbf{x}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}) + \sum_{j=1}^{p} \nu_j^* h_i(\mathbf{x}) \right) \tag{18.17}$$

$$\leq f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{j=1}^{p} \nu_j^* h_j(\mathbf{x}^*) \tag{18.18}$$

$$\leq f_0(\mathbf{x}^*) \tag{18.19}$$

- Dòng đầu là do chính là *strong duality*.
- Dòng hai là do định nghĩa của hàm đối ngẫu.
- Dòng ba là hiển nhiên vì infimum của một hàm nhỏ hơn giá trị của hàm đó tại bất kỳ một điểm nào khác.
- Dòng bốn là vì các ràng buộc $f_i(\mathbf{x}^*) \leq 0, \lambda_i \geq 0, i = 1, 2, \ldots, m$ và $h_j(\mathbf{x}^*) = 0$.

Từ đây có thể thế rằng dấu đẳng thức ở dòng ba và dòng bốn phải đồng thời xảy ra. Và ta lại có thêm hai quan sát thú vị nữa:

- $\mathbf{x}^*$ chính là một điểm *optimal* của $g(\lambda^*, \nu^*)$.
- Thú vị hơn:

$$\sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) = 0$$

Vì mỗi phần tử trong tổng trên là không dương do $\lambda_i^* \geq 0, f_i \leq 0$, ta kết luận rằng:

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, i = 1, 2, \ldots, m$$

Điều kiện cuối cùng này được gọi là *complementary slackness*. Từ đây có thể suy ra:

$$\lambda_i^* > 0 \Rightarrow f_i(\mathbf{x}^*) = 0 \tag{18.20}$$

$$f_i(\mathbf{x}^*) < 0 \Rightarrow \lambda_i^* = 0 \tag{18.21}$$

Tức ta luôn có một trong hai giá trị này bằng 0.

18.5.2 KKT optimality conditions

Chúng ta vẫn giả sử rằng các hàm đang xét có đạo hàm và bài toán tối ưu không nhất thiết là lồi.

KKT condition cho bài toán *không* lồi

Giả sử rằng *strong duality* xảy ra. Gọi $\mathbf{x}^*$ và $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ là *bất kỳ primal và dual optimal points*. Vì $\mathbf{x}^*$ tối ưu hàm khả vi $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$, ta có đạo hàm của Lagrangian tại $\mathbf{x}^*$ phải bằng 0.

Điều kiện Karush-Kuhn-Tucker (KKT)) nói rằng $\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*$ phải thoả mãn điều kiện:

$$f_i(\mathbf{x}^*) \leq 0, i = 1, 2, \ldots, m \tag{18.22}$$

$$h_j(\mathbf{x}^*) = 0, j = 1, 2, \ldots, p \tag{18.23}$$

$$\lambda_i^* \geq 0, i = 1, 2, \ldots, m \tag{18.24}$$

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, i = 1, 2, \ldots, m \tag{18.25}$$

$$\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{p} \nu_j^* \nabla h_j(\mathbf{x}^*) = 0 \tag{18.26}$$

Đây là *điều kiện cần* để $\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*$ là nghiệm của hai bài toán.

KKT conditions cho bài toán lồi

Với các bài toán lồi và *strong duality* xảy ra, các điệu kiện KKT phía trên cũng là *điều kiện đủ*. Vậy với các bài toán lồi với hàm mục tiêu và hàm ràng buộc là khả vi, bất kỳ điểm nào thoả mãn các điều kiện KKT đều là *primal và dual optimal* của bài toán gốc và bài toán đối ngẫu.

Từ đây ta có thể thấy rằng: Với một bài toán lồi và điều kiện Slater thoả mãn (suy ra *strong duality*) thì các điều kiện KKT là điều cần và đủ của nghiệm.

Các điều kiện KKT rất quan trọng trong tối ưu. Trong một vài trường hợp đặc biệt (chúng ta sẽ thấy trong bài Support Vector Machine sắp tới), việc giải hệ (bất) phương trình các điều kiện KKT là khả thi. Rất nhiều các thuật toán tối ưu được xây dựng giả trên việc giải hệ điều kiện KKT.

Ví dụ: *Equality constrained convex quadratic minimization*. Xét bài toán:

$$\begin{aligned} \mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r \\ \text{subject to:} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{18.27}$$

trong đó $\mathbf{P} \in \mathbb{S}_+^n$ (tập các ma trận đối xứng nửa xác định dương).

Lagrangian:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r + \boldsymbol{\nu}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$$

Diều kiện KKT cho bài toán này là:

$$\mathbf{A}\mathbf{x}^* = \mathbf{b} \tag{18.28}$$

$$\mathbf{P}\mathbf{x}^* + \mathbf{q} + \mathbf{A}^T\boldsymbol{\nu}^* = 0 \tag{18.29}$$

Phương trình thứ hai chính là phương trình đạo hàm của Lagrangian tại $\mathbf{x}^*$ bằng 0.

Hệ phương trình này có thể được viết lại đơn giản là:

$$\begin{bmatrix} \mathbf{P} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \boldsymbol{\nu}^* \end{bmatrix} = \begin{bmatrix} -\mathbf{q} \\ \mathbf{b} \end{bmatrix}$$

đây là một phương trình tuyến tính đơn giản!

18.6 Tóm tắt

Giả sử rằng các hàm số đều khả vi:

- Các bài toán tối ưu với chỉ ràng buộc là đẳng thức có thể được giải quyết bằng phương pháp nhân tử Lagrange. Ta cũng có định nghĩa về Lagrangian. Điều kiện cần để một điểm là nghiệm của bài toán tối ưu là nó phải làm cho đạo hàm của Lagrangian bằng 0.

- Với các bài toán tối ưu có thêm ràng buộc là bất đẳng thức (không nhất thiết là lồi), chúng ta có Lagrangian tổng quát và các biến Lagrange λ, ν. Với các giá trị (λ, ν) cố định, ta có định nghĩa về **hàm đối ngẫu Lagrange** (Lagrange dual function) $g(\lambda, \nu)$ được xác định là infimum của Lagrangian khi $\mathbf{x}$ thay đổi trên miền xác định của bài toán.

- Miền xác định và tập các điểm *feasible* thường khác nhau. *Feasible set* là tập con của tập xác định.

- Với mọi $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^*$.

- Hàm số $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ **là lồi** bất kể bài toán tối ưu có là lồi hay không. Hàm số này được gọi là *dual Lagrange fucntion* hay *hàm đối ngẫu Lagrange*.

- Bài toán đi tìm giá trị lớn nhất của hàm đối ngẫu Lagrange với điều kiện $\boldsymbol{\lambda} \succeq 0$ được gọi là bài toán *đối ngẫu* (*dual problem*). Bài toán này **là lồi** bất kể bài toán gốc có lồi hay không.

- Gọi giá trị tối ưu của bài toán đối ngẫu là d^* thì ta có: $d^* \leq p^*$. Đây được gọi là *weak duality*.
- *Strong duality* xảy ra khi $d^* = p^*$. Thường thì *strong duality* không xảy ra, nhưng với các bài toán lồi thì *strong duality* thường (không luôn luôn) xảy ra.
- Nếu bài toán là lồi và điều kiện Slater thoả mãn, thì *strong duality* xảy ra.
- Nếu bài toán lồi và có *strong duality* thì nghiệm của bài toán thoả mãn các điều kiện KKT (điều kiện cần và đủ).
- Rất nhiều các bài toán tối ưu được giải quyết thông qua KKT conditions.

18.7 Kết luận

Trong ba bài 16, 17, 18, tôi đã giới thiệu *sơ lược* về tập lồi, hàm lồi, bài toán lồi, và các điệu kiện tối ưu được xây dựng thông qua *duality*. Ý định ban đầu của tôi là tránh phần này vì khá nhiều toán, tuy nhiên trong quá trình chuẩn bị cho bài Support Vector Machine, tôi nhận thấy rằng cần phải giải thích về Lagrangian - kỹ thuật được sử dụng rất nhiều trong Tối ưu. Thêm nữa, để giải thích về Lagrangian, tôi cần nói về các bài toán lồi. Chính vì vậy tôi thấy có trách nhiệm *phải* viết về ba bài này.

Trong loạt bài tiếp theo, chúng ta sẽ lại quay lại với các thuật toán Machine Learning với rất nhiều ví dụ, hình vẽ và code mẫu. Nếu bạn nào có cảm thấy hơi đuối sau ba bài tối ưu này thì cũng đừng lo, mọi chuyện rồi sẽ ổn cả thôi.

18.8 Tài liệu tham khảo

[1] Convex Optimization – Boyd and Vandenberghe, Cambridge University Press, 2004.

[2] Lagrange Multipliers - Wikipedia.

51

Ôn tập Đại số tuyến tính

51.1 Lưu ý về ký hiệu

Trong các bài viết của tôi, các số vô hướng được biểu diễn bởi các chữ cái viết ở dạng không in đậm, có thể viết hoa, ví dụ x_1, N, y, k. Các vector được biểu diễn bằng các chữ cái thường in đậm, ví dụ $\mathbf{y}, \mathbf{x}_1$. Nếu không giải thích gì thêm, các vector được mặc định hiểu là các vector cột. Các ma trận được biểu diễn bởi các chữ viết hoa in đậm, ví dụ $\mathbf{X}, \mathbf{Y}, \mathbf{W}$.

Đối với vector, $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ được hiểu là một vector hàng. Trong khi $\mathbf{x} = [x_1; x_2; \ldots; x_n]$ được hiểu là vector cột. Chú ý sự khác nhau giữa dầu phẩy (,) và dấu chấm phẩy (;). Đây chính là ký hiệu được Matlab sử dụng.

Tương tự, trong ma trận, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ được hiểu là các vector cột $\mathbf{x}_j$ được đặt cạnh nhau theo thứ tự từ trái qua phải để tạo ra ma trận $\mathbf{X}$. Trong khi $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_m]$ được hiểu là các vector $\mathbf{x}_i$ được đặt chồng lên nhau theo thứ tự từ trên xuống dưới để tạo ra ma trận $\mathbf{X}$. Các vector được ngầm hiểu là có kích thước phù hợp để có thể xếp cạnh hoặc xếp chồng lên nhau.

Cho một ma trận $\mathbf{W}$, nếu không giải thích gì thêm, chúng ta hiểu rằng $\mathbf{w}_i$ là **vector cột** thứ i của ma trận đó. Chú ý sự tương ứng giữa ký tự viết hoa và viết thường.

51.2 Norms (chuẩn)

Trong không gian một chiều, việc đo khoảng cách giữa hai điểm đã rất quen thuộc: lấy trị tuyệt đối của hiệu giữa hai giá trị đó. Trong không gian hai chiều, tức mặt phẳng, chúng ta thường dùng khoảng cách Eclid để đo khoảng cách giữa hai điểm. Khoảng cách này chính là cái chúng ta thường nói bằng ngôn ngữ thông thường là *đường chim bay*. Đôi khi, để đi từ một điểm này tới một điểm kia, con người chúng ta không thể đi bằng đường chim bay được mà còn phụ thuộc vào việc đường đi nối giữa hai điểm có dạng như thế nào nữa.

Việc đo khoảng cách giữa hai điểm dữ liệu nhiều chiều, tức hai vector, là rất cần thiết trong Machine Learning. Chúng ta cần đánh giá xem điểm nào là điểm gần nhất của một điểm khác; chúng ta cũng cần đánh giá xem độ chính xác của việc ước lượng; và trong rất nhiều ví dụ khác nữa.

Và đó chính là lý do mà khái niệm norm (chuẩn) ra đời. Có nhiều loại norm khác nhau mà các bạn sẽ thấy ở dưới đây:

Để xác định khoảng cách giữa hai vector $\mathbf{y}$ và $\mathbf{z}$, người ta thường áp dụng một hàm số lên vector hiệu $\mathbf{x} = \mathbf{y} - \mathbf{z}$. Một hàm số được dùng để đo các vector cần có một vài tính chất đặc biệt.

51.2.1 Định nghĩa

Một hàm số $f()$ ánh xạ một điểm $\mathbf{x}$ từ không gian n chiều sang tập số thực một chiều được gọi là norm nếu nó thỏa mãn ba điều kiện sau đây:

1. $f(\mathbf{x}) \geq 0$. Dấu bằng xảy ra $\Leftrightarrow \mathbf{x} = \mathbf{0}$.
2. $f(\alpha\mathbf{x}) = \|\alpha\|f(\mathbf{x}), \quad \forall\alpha \in \mathbb{R}$
3. $f(\mathbf{x}_1) + f(\mathbf{x}_2) \geq f(\mathbf{x}_1 + \mathbf{x}_2), \quad \forall\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{R}^n$

Điều kiện thứ nhất là dễ hiểu vì khoảng cách không thể là một số âm. Hơn nữa, khoảng cách giữa hai điểm $\mathbf{y}$ và $\mathbf{z}$ bằng 0 nếu và chỉ nếu hai điểm nó trùng nhau, tức $\mathbf{x} = \mathbf{y} - \mathbf{z} = \mathbf{0}$.

Điều kiện thứ hai cũng có thể được lý giải như sau. Nếu ba điểm $\mathbf{y}, \mathbf{v}$ và $\mathbf{z}$ thẳng hàng, hơn nữa $\mathbf{v} - \mathbf{y} = \alpha(\mathbf{v} - \mathbf{z})$ thì khoảng cách giữa $\mathbf{v}$ và $\mathbf{y}$ sẽ gấp $\|\alpha\|$ lần khoảng cách giữa $\mathbf{v}$ và $\mathbf{z}$.

Điều kiện thứ ba chính là bất đẳng thức tam giác nếu ta coi $\mathbf{x}_1 = \mathbf{w} - \mathbf{y}, \mathbf{x}_2 = \mathbf{z} - \mathbf{w}$ với $\mathbf{w}$ là một điểm bất kỳ trong cùng không gian.

51.2.2 Một số chuẩn thường dùng

Giả sử các vectors $\mathbf{x} = [x_1; x_2; \ldots; x_n]$, $\mathbf{y} = [y_1; y_2; \ldots; y_n]$.

Nhận thấy rằng khoảng cách Euclid chính là một norm, norm này thường được gọi là **norm 2**:

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots x_n^2} \tag{51.1}$$

Với p **là một số không nhỏ hơn 1** bất kỳ, hàm số sau đây:

$$\|\mathbf{x}\|_p = (\|x_1\|^p + \|x_2\|^p + \ldots \|x_n\|^p)^{\frac{1}{p}} \tag{51.2}$$

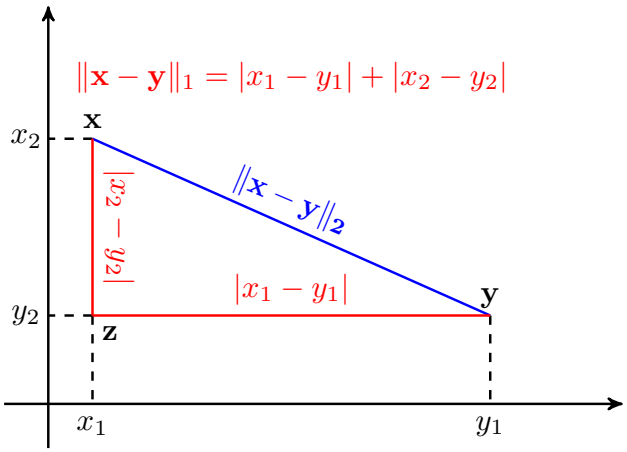
được chứng minh thỏa mãn ba điều kiện bên trên, và được gọi là **norm p**.

Nhận thấy rằng khi $p \to 0$ thì biểu thức bên trên trở thành *số các phần tử khác 0 của* $\mathbf{x}$. Hàm số (51.2) khi $p = 0$ được gọi là giả chuẩn (pseudo-norm) 0. Nó không phải là norm vì nó không thỏa mãn điều kiện 2 và 3 của norm. Giả-chuẩn này, thường được ký hiệu là $\|\mathbf{x}\|_0$, khá quan trọng trong Machine Learning vì trong nhiều bài toán, chúng ta cần có ràng buộc "sparse", tức số lượng thành phần "active" của $\mathbf{x}$ là nhỏ.

Có một vài giá trị của p thường được dùng:

1. Khi $p = 2$ chúng ta có norm 2 như ở trên.

2. Khi $p = 1$ chúng ta có:
$$\|\mathbf{x}\|_1 = \|x_1\| + \|x_2\| + \ldots \|x_n\| \tag{51.3}$$
là tổng các trị tuyệt đối của từng phần tử của $\mathbf{x}$. Norm 1 thường được dùng như xấp xỉ của norm 0 trong các bài toán có ràng buộc "sparse". Dưới đây là một ví dụ so sánh norm 1 và norm 2 trong không gian hai chiều:



Hình 51.1: Minh họa norm 1 và norm 2

Norm 2 (màu xanh) chính là đường thẳng "chim bay" nối giữa hai vector $\mathbf{x}$ và $\mathbf{y}$. Khoảng cách norm 1 giữa hai điểm này (màu đỏ) có thể diễn giải như là đường đi từ $\mathbf{x}$ tới $\mathbf{y}$ trong một thành phố mà đường phố tạo thành hình bàn cờ. Chúng ta chỉ có cách đi dọc theo cạnh của bàn cờ mà không được đi thẳng.

3. Khi $p \to \infty$, ta có norm p chính là trị tuyệt đối của phần tử lớn nhất của vector đó:
$$\|\mathbf{x}\|_\infty = \max_{i=1,2,\ldots,n} \|x_i\| \tag{51.4}$$

51.2.3 Chuẩn của ma trận

Với một ma trận $\mathbf{A} \in \mathbb{R}^{m \times n}$, chuẩn thường được dùng nhất là chuẩn Frobenius, ký hiệu là $\|\mathbf{A}\|_F$ là căn bậc hai của tổng bình phương tất cả các phần tử của ma trận đó.

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$$

51.3 Đạo hàm của hàm nhiều biến

Trong mục này, chúng ta sẽ giả sử rằng các đạo hàm tồn tại. Chúng ta sẽ xét hai trường hợp: i) Hàm số nhận giá trị là ma trận (vector) và cho giá trị là một số thực vô hướng; và ii) Hàm số nhận giá trị là một số vô hướng hoặc vector và cho giá trị là một vector.

51.3.1 Hàm cho giá trị là một số vô hướng

Đạo hàm (gradient) của một hàm số $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ **theo vector x** được định nghĩa như sau:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) \triangleq \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n \tag{51.5}$$

trong đó $\frac{\partial f(\mathbf{x})}{\partial x_i}$ là đạo hàm của hàm số theo thành phần thứ i của vector $\mathbf{x}$. Đạo hàm này được lấy khi giả sử tất cả các biến còn lại là hằng số.

Nếu không có thêm biến nào trong hàm số, $\nabla_{\mathbf{x}} f(\mathbf{x})$ thường được viết gọn là $\nabla f(\mathbf{x})$.

Điều quan trọng cần nhớ: **đạo hàm của hàm số này là một vector có cùng chiều với vector đang lấy đạo hàm**. Tức nếu vector viết ở dạng cột thì đạo hàm cũng phải viết ở dạng cột.

Đạo hàm bậc hai (second-order gradient) của hàm số trên còn được gọi là *Hessian* và được định nghĩa như sau:

$$\nabla^2 f(\mathbf{x}) \triangleq \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix} \in \mathbb{S}^n \tag{51.6}$$

với $\mathbb{S}^n \in \mathbb{R}^{n \times n}$ là tập các ma trận vuông đối xứng có số cột là n.

Đạo hàm của một hàm số $f(\mathbf{X}) : \mathbb{R}^{n \times m} \to \mathbb{R}$ **theo ma trận X** được định nghĩa là:

$$\nabla f(\mathbf{X}) = \begin{bmatrix} \frac{\partial f(\mathbf{X})}{\partial x_{11}} & \frac{\partial f(\mathbf{X})}{\partial x_{12}} & \cdots & \frac{\partial f(\mathbf{X})}{\partial x_{1m}} \\ \frac{\partial f(\mathbf{X})}{\partial x_{21}} & \frac{\partial f(\mathbf{X})}{\partial x_{22}} & \cdots & \frac{\partial f(\mathbf{X})}{\partial x_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{X})}{\partial x_{n1}} & \frac{\partial f(\mathbf{X})}{\partial x_{n2}} & \cdots & \frac{\partial f(\mathbf{X})}{\partial x_{nm}} \end{bmatrix} \in \mathbb{R}^{n \times m} \tag{51.7}$$

Một lần nữa, đạo hàm của một hàm số theo ma trận là một ma trận có chiều giống với ma trận đó.

Hiểu một cách đơn giản, đạo hàm của một hàm số (có đầu ra là 1 số vô hướng) theo một ma trận được tính như sau. Trước tiên, tính đạo hàm của hàm số đó theo từng thành phần của ma trận *khi toàn bộ các thành phần khác được giả sử là hằng số*. Tiếp theo, ta *ghép* các đạo hàm thành phần tính được thành một ma trận đúng theo thứ tự như trong ma trận đó. Chú ý rằng vector là một trường hợp của ma trận.

Ví dụ: Xét hàm số: $f : \mathbb{R}^2 \to \mathbb{R}$, $f(\mathbf{x}) = x_1^2 + 2x_1x_2 + \sin(x_1) + 2$.

Đạo hàm bậc nhất theo $\mathbf{x}$ của hàm số đó là:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 + 2x_2 + \cos(x_1) \\ 2x_1 \end{bmatrix}$$

Đạo hàm bậc hai theo $\mathbf{x}$, hay *Hessian* là:

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial f^2(\mathbf{x})}{\partial x_1 x_2} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 x_1} & \frac{\partial f^2(\mathbf{x})}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 - \sin(x_1) & 2 \\ 2 & 0 \end{bmatrix}$$

Chú ý rằng *Hessian* luôn là một ma trận đối xứng.

51.3.2 Hàm cho giá trị là một vector

Những hàm số cho giá trị là một vector được gọi là *vector-valued function* trong tiếng Anh.

Giả sử một hàm số với **đầu vào là một số thực** $v(x) : \mathbb{R} \to \mathbb{R}^n$:

$$v(x) = \begin{bmatrix} v_1(x) \\ v_2(x) \\ \vdots \\ v_n(x) \end{bmatrix} \tag{51.8}$$

Đạo hàm của nó là một **vector hàng** như sau:

$$\nabla v(x) \triangleq \begin{bmatrix} \frac{\partial v_1(x)}{\partial x} & \frac{\partial v_2(x)}{\partial x} & \dots & \frac{\partial v_n(x)}{\partial x} \end{bmatrix} \tag{51.9}$$

Đạo hàm bậc hai của hàm số này có dạng:

$$\nabla^2 v(x) \triangleq \begin{bmatrix} \frac{\partial^2 v_1(x)}{\partial x^2} & \frac{\partial^2 v_2(x)}{\partial x^2} & \dots & \frac{\partial^2 v_n(x)}{\partial x^2} \end{bmatrix} \tag{51.10}$$

Ví dụ: Cho vector $\mathbf{a} \in \mathbb{R}^n$ và *vector-valued function* $v(x) = x\mathbf{a}$, thế thì:

$$\nabla v(x) = \mathbf{a}^T, \quad \nabla^2 v(x) = \mathbf{0} \in \mathbb{R}^{n \times n} \tag{51.11}$$

với $\mathbf{0}$ là ma trận với các thành phần đều là 0.

Xét một *vector-valued function* với **đầu vào là một vector** $h(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}^n$, đạo hàm bậc nhất của nó là:

$$\nabla h(\mathbf{x}) \triangleq \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial x_1} & \frac{\partial h_2(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial h_n(\mathbf{x})}{\partial x_1} \\ \frac{\partial h_1(\mathbf{x})}{\partial x_2} & \frac{\partial h_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_n(\mathbf{x})}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_1(\mathbf{x})}{\partial x_k} & \frac{\partial h_2(\mathbf{x})}{\partial x_k} & \dots & \frac{\partial h_n(\mathbf{x})}{\partial x_k} \end{bmatrix} \tag{51.12}$$

$$= \begin{bmatrix} \nabla h_1(\mathbf{x}) & \nabla h_2(\mathbf{x}) & \dots & \nabla h_n(\mathbf{x}) \end{bmatrix} \in \mathbf{R}^{k \times n} \tag{51.13}$$

Một quy tắc dễ nhớ ở đây là nếu một hàm số $g : \mathbb{R}^m \to \mathbb{R}^n$ thì đạo hàm của nó là một ma trận thuộc $\mathbb{R}^{m \times n}$.

Đạo hàm bậc hai của hàm số trên là một *ma trận ba chiều*, tôi xin không đề cập ở đây.

<hr> Với các hàm số *matrix-valued* nhận giá trị đầu vào là ma trận, tôi cũng xin không đề cập ở đây. Tuy nhiên, ở phần dưới, khi tính toán đạo hàm cho các hàm cho giá trị là số thực, chúng ta vẫn có thể sẽ sử dụng khái niệm này.

Trước khi đến phần tính đạo hàm của các hàm số thường gặp, chúng ta cần biết hai tính chất quan trọng khá giống với đạo hàm của hàm một biến được học trong chương trình cấp ba.

51.3.3 Hai tính chất quan trọng

Product rules

Để cho tổng quát, ta giả sử biến đầu vào là một ma trận (vector và số thực là các trường hợp đặt biệt của ma trận). Giả sử rằng các hàm số có chiều phù hợp để các phép nhân thực hiện được. Ta có:

$$\nabla \left(f(\mathbf{X})^T g(\mathbf{X})\right) = \left(\nabla f(\mathbf{X})\right) g(\mathbf{X}) + \left(\nabla g(\mathbf{X})\right) f(\mathbf{X}) \tag{51.14}$$

Biểu thức này giống như biểu thức chúng ta đã quá quen thuộc:

$$(f(x)g(x))' = f'(x)g(x) + g'(x)f(x)$$

Chú ý rằng với vector và ma trận, chúng ta không được sử dụng tính chất giao hoán.

Chain rules

Khi có các hàm hợp thì:

$$\nabla_{\mathbf{X}} g(f(\mathbf{X})) = \nabla_{\mathbf{X}} f^T \nabla_f g \tag{51.15}$$

Quy tắc này cũng giống với quy tắc trong hàm một biến:

$$(g(f(x))' = f'(x)g'(f)$$

Nhắc lại rằng khi tính toán với ma trận, chúng ta cần chú ý tới chiều của các ma trận, và nhân ma trận không có tính chất giao hoán.

51.3.4 Đạo hàm của các hàm số thường gặp

$\boldsymbol{f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}}$

Giả sử $\mathbf{a}, \mathbf{x} \in \mathbb{R}^n$, ta viết lại:

$$f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

Có thể nhận thấy rằng:

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = a_i, \ \forall i = 1, 2 \ldots, n$$

Vậy nên:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \mathbf{a}$$

Thêm nữa, vì $\mathbf{a}^T\mathbf{x} = \mathbf{x}^T\mathbf{a}$ nên:

$$\nabla(\mathbf{x}^T\mathbf{a}) = \mathbf{a}$$

$\boldsymbol{f(x) = Ax}$

Đây là một *vector-valued function* $f : \mathbb{R}^n \to \mathbb{R}^m$ với $\mathbf{x} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{m\times n}$. Giả sử rằng $\mathbf{a}_i$ là **hàng** thứ i của ma trận $\mathbf{A}$. Ta có:

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{a}_1\mathbf{x} \\ \mathbf{a}_2\mathbf{x} \\ \vdots \\ \mathbf{a}_m\mathbf{x} \end{bmatrix}$$

Theo định nghĩa (13.2), và công thức (17), ta có thể suy ra:

$$\nabla_{\mathbf{x}}(\mathbf{Ax}) = \begin{bmatrix} \mathbf{a}_1^T & \mathbf{a}_2^T & \dots & \mathbf{a}_m^T \end{bmatrix} = \mathbf{A}^T \tag{51.16}$$

Từ đây ta có thể suy ra đạo hàm của hàm số $f(\mathbf{x}) = \mathbf{x} = \mathbf{Ix}$, với $\mathbf{I}$ là ma trận đơn vị với chiều phù hợp, là:

$$\nabla\mathbf{x} = \mathbf{I}$$

$\boldsymbol{f(x) = x^TAx}$

với $\mathbf{x} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{n\times n}$. Áp dụng Product rules (14) ta có:

$$\begin{aligned} \nabla f(\mathbf{x}) &= \nabla\left(\left(\mathbf{x}^T\right)(\mathbf{Ax})\right) \\ &= (\nabla(\mathbf{x}))\,\mathbf{Ax} + (\nabla(\mathbf{Ax}))\,\mathbf{x} \\ &= \mathbf{IAx} + \mathbf{A}^T\mathbf{x} \\ &= (\mathbf{A} + \mathbf{A}^T)\mathbf{x} \end{aligned} \tag{51.17}$$

Từ (51.17) và (51.16), ta có thể suy ra: $\nabla^2\mathbf{x}^T\mathbf{Ax} = \mathbf{A}^T + \mathbf{A}$

Nếu $\mathbf{A}$ là một ma trận đối xứng, ta sẽ có:

$$\nabla\mathbf{x}^T\mathbf{Ax} = 2\mathbf{Ax}, \quad \nabla^2\mathbf{x}^T\mathbf{Ax} = 2\mathbf{A} \tag{51.18}$$

Nếu $\mathbf{A}$ là ma trận đơn vị, tức $f(\mathbf{x}) = \mathbf{x}^T\mathbf{Ix} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2$, ta có:

$$\nabla\|\mathbf{x}\|_2^2 = 2\mathbf{x}, \quad \nabla^2\|\mathbf{x}\|_2^2 = 2\mathbf{I} \tag{51.19}$$

$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$

Có hai cách tính đạo hàm của hàm số này:

Cách 1: Trước hết, biến đổi:

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = (\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = (\mathbf{x}^T\mathbf{A}^T - \mathbf{b}^T)(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ &= \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{b} \end{aligned}$$

Lấy đạo hàm cho từng số hạng rồi cộng lại ta có:

$$\nabla\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} = 2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$$

Cách 2: Dùng Chain rule. Sử dụng $\nabla(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{A}^T$ và $\nabla\|\mathbf{x}\|_2^2 = 2\mathbf{x}$ và công thức chain rules (51.15), ta sẽ thu được kết quả tương tự.

$f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}\mathbf{x}^T\mathbf{b}$

Bằng cách viết lại $f(\mathbf{x}) = (\mathbf{a}^T\mathbf{x})(\mathbf{x}^T\mathbf{b})$, ta có thể dùng Product rules (14) và ra kết quả:

$$\nabla(\mathbf{a}^T\mathbf{x}\mathbf{x}^T\mathbf{b}) = \mathbf{a}\mathbf{x}^T\mathbf{b} + \mathbf{b}\mathbf{a}^T\mathbf{x} = \mathbf{a}\mathbf{b}^T\mathbf{x} + \mathbf{b}\mathbf{a}^T\mathbf{x} = (\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)\mathbf{x}$$

trong đây tôi đã sử dụng tính chất $\mathbf{y}^T\mathbf{z} = \mathbf{z}^T\mathbf{y}$ và tích của một số thực với một vector cũng bằng tích của vector và số thực đó.

51.3.5 Bảng các đạo hàm thường gặp

Cho vector

$f(\mathbf{x})$	$\nabla f(\mathbf{x})$
$\mathbf{a}^T\mathbf{x}$	$\mathbf{a}$
$\mathbf{x}^T\mathbf{A}\mathbf{x}$	$\mathbf{A} + \mathbf{A}^T)\mathbf{x}$
$\mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2$	$2\mathbf{x}$
$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$	$2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$
$\mathbf{a}^T\mathbf{x}^T\mathbf{x}\mathbf{b}$	$2\mathbf{a}^T\mathbf{b}\mathbf{x}$
$\mathbf{a}^T\mathbf{x}\mathbf{x}^T\mathbf{b}$	$(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)\mathbf{x}$

Bảng 51.1: Đạo hàm theo vector

$f(\mathbf{X})$	$\nabla f(\mathbf{X})$
$\|\mathbf{X}\|_F^2$	$2\mathbf{X}$
$\mathbf{AX}$	$\mathbf{A}^T$
$\|\mathbf{AX}-\mathbf{B}\|_F^2$	$2\mathbf{A}^T(\mathbf{AX}-\mathbf{B})$
$\|\mathbf{XA}-\mathbf{B}\|_F^2$	$2(\mathbf{XA}-\mathbf{B})\mathbf{A}^T$
$\mathbf{a}^T\mathbf{X}^T\mathbf{Xb}$	$\mathbf{X}(\mathbf{ab}^T+\mathbf{ba}^T)$
$\mathbf{a}^T\mathbf{XX}^T\mathbf{b}$	$(\mathbf{ab}^T+\mathbf{ba}^T)\mathbf{X},$
$\mathbf{a}^T\mathbf{YX}^T\mathbf{b}$	$\mathbf{ba}^T\mathbf{Y}$
$\mathbf{a}^T\mathbf{Y}^T\mathbf{Xb}$	$\mathbf{Yab}^T$
$\mathbf{a}^T\mathbf{XY}^T\mathbf{b}$	$\mathbf{ab}^T\mathbf{Y}$
$\mathbf{a}^T\mathbf{X}^T\mathbf{Yb}$	$\mathbf{Yba}^T$

Bảng 51.2: Đạo hàm theo ma trận

Cho ma trận

51.3.6 Tài liệu tham khảo

[1] Matrix calculus

Index