دانشکده علوم ریاضی
دانشگاه صنعتی شریف

تحقیق در عملیات ۱

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

چند مثال در برنامه‌ریزی خطی

جلسه سوم

نگارنده: سجاد ولایی

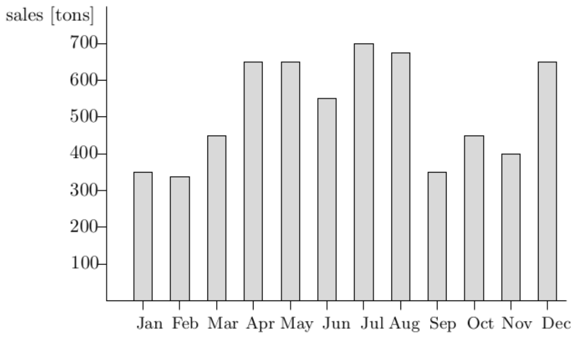
۱ مروری بر مباحث گذشته

پیش‌ازاین گفتیم برنامه ریزی خطی چیست. برنامه‌ریزی خطی وقتی به کار می‌آید که بخواهیم ترکیب خطی متغیرها را کمینه یا بیشینه کنیم در حالی که تعدادی قید خطی روی بعضی متغیرها داریم. در جلسه‌ی پیش سعی کردیم چند مثال از برنامه‌ریزی خطی ببینیم.

۲ مثال‌ها

۱.۲ فروش بستنی

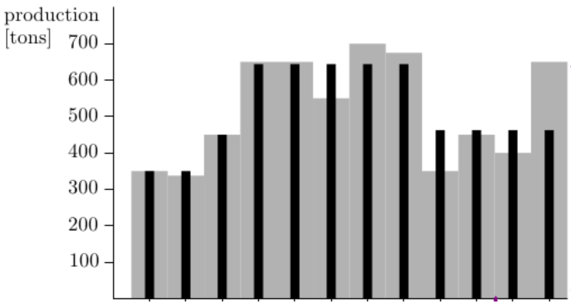
یک شرکت بستنی سازی داریم. فروش بستنی در هرماه را طبق نمودار شکل ۱ بیش‌بینی می‌کنیم. قرار است در هر ماه به میزان پیش‌بینی شده بستنی بفروشیم. این شرکت برای تغییر میزان تولید و نگه‌داری بستنی‌ها دچار هزینه‌هایی می‌شود. می‌خواهیم با تعیین میزان تولید در هر ماه سود شرکت را طی هزاران سال بیشینه کنیم. یا معادلا هزینه‌ها را کمینه کنیم.



شکل ۱: نمودار میزان فروش در ماه

برای نزدیک‌شدن به مسئله می‌توانیم دو جواب بدیهی را بررسی کنیم.

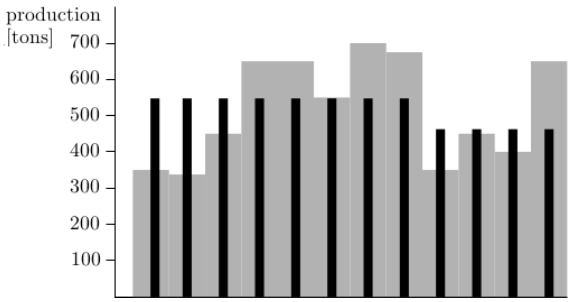
- می‌توانیم میزان تولید هر ماه را برابر میزان فروش آن ماه قرار دهیم.



شکل ۲: نمودار میزان تولید برابر فروش

اما در این حالت هزینه‌ی تغییر میزان تولید که شامل تغییر میزان کارکنان و خطوط تولید می‌شود زیاد خواهد بود.

- می‌توانیم میزان تولید هر ماه را ثابت نگه داریم و اضافی هر ماه را در انبار نگه‌داری کنیم.



شکل ۳: نمودار میزان تولید ثابت

اما در این حالت هزینه‌ی نگه‌داری محصولات در یخچال زیاد خواهد بود.

در نتیجه انتظار داریم جواب بهینه چیزی بین دو جواب مذکور باشد. برای دقیق‌تر شدن نگاهمان لازم است بدانیم در مسئله‌ی مذکور به ازای هر واحد تغییر میزان تولید در دو ماه متوالی ۵۰ واحد هزینه و به ازای نگه‌داری هر واحد در ماه ۲۰ واحد هزینه خواهیم داد.

نکته‌ی قابل توجه این است که مقدار نگه‌داری شده در ماه اول ممکن است صفر نباشد. و صفر نبودن آن طی سالیان دراز به نفع شرکت تمام شود. تعداد سال‌ها آن‌قدر زیاد است که عملا می‌توانیم از جزییات ماه اول اولین سال شرکت صرف نظر کنیم. (میتوان ثابت کرد میزان نگه‌داری در یکی از ماه‌ها صفر خواهد بود)

برای ماه i از ۱ تا ۱۲ متغیر x_i را میزان تولید و s_i را میزان ذخیره شده و d_i را میزان فروش در ماه i در نظر می‌گیریم. مشخص است که d_i را از قبل داریم. برای آنکه بتوانیم میزان فروش را تامین کنیم شرط زیر لازم است.

$$x_i + s_{i-1} \geq d_i \Rightarrow x_i - s_i + s_{i-1} = d_i$$

در نتیجه برای نوشتن برنامه‌ریزی خطی در ابتدا داریم:

$$\begin{aligned}
\text{کمینه کن}\quad & 20 \times \sum_{i=1}^{i\leq 12} + 50 \times (\sum_{i=2}^{i\leq 12} |x_i - x_{i-1}| + |x_1 - x_{12}|) \\
\text{که}\quad & x_1 + s_{12} - d1 = s_1 \\
& x_i + s_{i-1} - d_i = s_i, \quad i \in \{2, \ldots, 12\} \\
& x_i \geq 0, \quad i \in \{1, \ldots, 12\} \\
& s_i \geq 0, \quad i \in \{1, \ldots, 12\}
\end{aligned}$$

اما همانطور که مشاهده می‌شود تابع هدف به دلیل وجود تابع قدر مطلق یک تابع خطی نیست و نمی‌توان آن را به این صورت با برنامه‌ریزی خطی حل کرد. برای حل این مشکل از دو متغیر اضافه کمک می‌گیریم. متغیر y_i را برابر میزان افزایش تولید و متغیر z_i را برابر میزان کاهش تولید می‌گیریم. طبق تعریف داریم:

$$\begin{aligned}
x_1 - x_{12} = y_1 - z_1, \quad x_i - x_{i-1} = y_i - z_i, \quad i \in \{2, \ldots, 12\} \\
|x_1 - x_{12}| = y_1 + z_1, \quad |x_i - x_{i-1}| = y_i + z_i, \quad i \in \{2, \ldots, 12\}
\end{aligned}$$

پس می‌توانیم برنامه‌ریزی خطی را به این صورت بنویسیم:

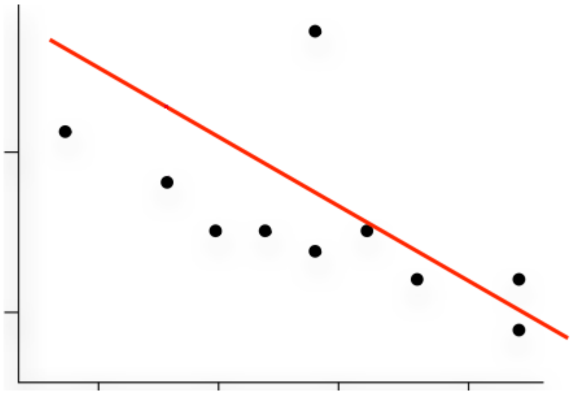
$$\begin{aligned}
\text{کمینه کن}\quad & 20 \times \sum_{i=1}^{i\leq 12} + 50 \times \sum_{i=2}^{i\leq 12} y_i + z_i \\
\text{که}\quad & x_1 + s_{12} - d1 = s_1, \quad x_i + s_{i-1} - d_i = s_i, \quad i \in \{2, \ldots, 12\} \\
& x_1 - x_{12} = y_1 - z_1, \quad x_i - x_{i-1} = y_i - z_i, \quad i \in \{2, \ldots, 12\} \\
& x_i \geq 0, \quad i \in \{1, \ldots, 12\} \\
& s_i \geq 0, \quad i \in \{1, \ldots, 12\} \\
& z_i \geq 0, \quad i \in \{1, \ldots, 12\} \\
& y_i \geq 0, \quad i \in \{1, \ldots, 12\}
\end{aligned}$$

ابهامی که ممکن است پیش بیاید این است که متغیرهای بیهوده‌ی زیادی در مسئله وجود دارد. یا به عبارتی ما به همراه اضافه کردن ۲۴ متغیر فقط ۱۲ معادله افزوده‌ایم. چرا مشکلی پیش نمی‌آید؟

نکته‌ی قابل توجه در مورد این ابهام آن است که در حالت بهینه همیشه یکی از دو متغیر y_i و z_i صفر است. و می‌توان به سادگی می‌توان اثبات کرد اگر هر دو ناصفر باشند تابع هدف کمینه نشده‌است(چراکه در آن صورت می‌توانیم از هر دو متغیر ۱ واحد کم کنیم که در نتیجه‌ی آن تابع هدف کم‌تر می‌شود و شروط پابرجا باقی می‌ماند). و اصلا به همین دلیل است که توانستیم این جمع را جایگزین تابع قدر مطلق کنیم.

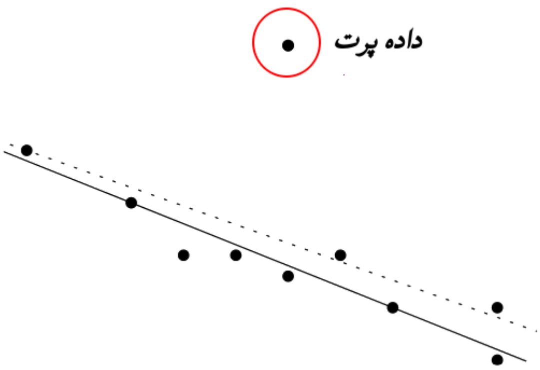
۲.۲ برازش خط

تعدادی نقطه روی صفحه‌ی اعداد حقیقی داریم. می‌خواهیم خطی در این صفحه رسم کنیم که به طور متوسط رفتار نقاط را برای ما شبیه‌سازی کند. همان‌طور که می‌دانیم این مسئله در نمودارهای آماری بسیار کاربردی است. مخصوصا در حالاتی که دو المان نمودار با یکدیگر رابطه‌ی خطی دارند.



شکل ۴: نمودار داده‌ها

$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ داده‌های ما هستند و ما می‌خواهیم خطی با معادله خطی $ax + b - y = 0$ از میان آن‌ها رد کنیم به طوری که این خط توصیفی از داده‌های ما باشد. می‌خواهیم خطی را بیابیم که میزان خطای آن کمینه باشد. خطا را برخی به صورت مجموع مربعات در نظر می‌گیرند که با عبارت $\sum_{i=1}^{n}(ax_i + b - y_i)^2$ بدست می‌آید. مشکل این روش آن است که اگر داده‌ی پرتی داشته باشیم که مقدار آن از متوسط رفتار داده‌های دیگر بسیار دور است. این داده‌ی پرت تاثیر زیادی روی خط نهایی ما خواهد گذاشت.



شکل ۵: نقطه‌چین خطی‌ایست که با روش کمینه‌ی مربعات برای تمام نقاط و خط دیگر خطی‌است که با همان روش برای تمام نقاط جز داده‌ی پرت تعبیه شده است.

اگر بخواهیم خطا را طوری تعریف کنیم که مشکل مذکور حل شده باشد می‌توانیم خطا را مجموع فاصله‌ی عرضی هر نقطه از خط برابر عبارت $\sum_{i=1}^{n}|ax_i + b - y_i|$ در نظر بگیریم. بنابراین می‌توان نوشت:

$$\text{کمینه کن} \quad \sum_{i=1}^{n}|ax_i + b - y_i|$$

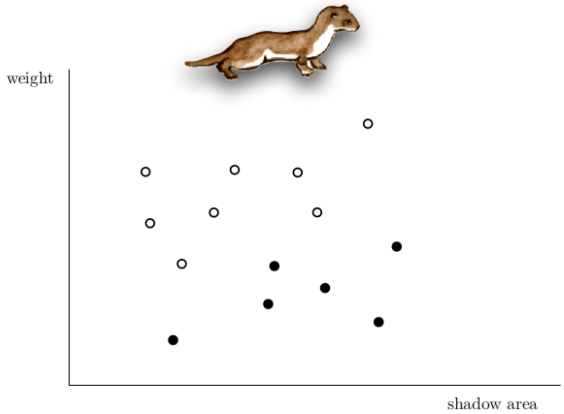
برای حل کردن مشکل وجود قدر مطلق می‌توان از روش قبل استفاده کرد. اما در این قسمت می‌خواهیم از روش دیگری بهتره بگیریم. کافی است متغیر e_i را طوری تعریف کنیم که در بهینه‌ترین حالت برابر مقدار قدر مطلق مذکور باشد. برنامه‌ریزی خطی را به این صورت می‌نویسیم:

$$\begin{aligned} \text{کمینه کن} \quad & \sum_{i=1}^{n} e_i \\ \text{که} \quad & e_i \geq (ax_i + b - y_i) \quad i \in 1, \cdots, n \\ & e_i \geq -(ax_i + b - y_i) \quad i \in 1, \cdots, n \end{aligned}$$

که مشخصا برای هر i کمترین مقدار e_i برابر مقدار $|ax_i + b - y_i|$ خواهد بود.

۳.۲ جدا کردن نقاط(SVM)

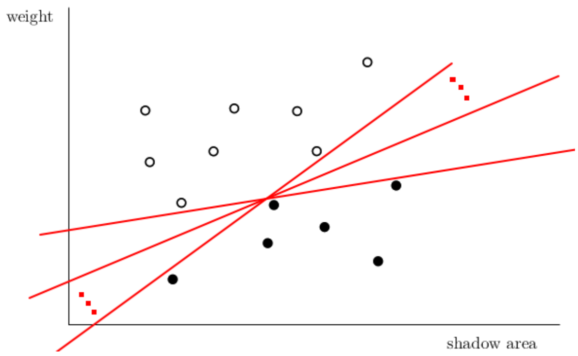
دستگاهی داریم که مساحت سایه و وزن موجود روی خود را اندازه می‌گیرد. برای تعدادی خرگوش و تعدادی سمور این داده‌ها را محاسبه کرده‌ایم و نموداری مانند شکل ۶ کشیده‌ایم. می‌خواهیم با خطی طوری این دو دسته نقاط را از هم جدا کنیم که ازین پس بتوانیم با خطای کمی خرگوش‌ها و سمورها را با این دستگاه تمایز دهیم. با توجه به شکل مشخص است با خطی افقی یا عمودی نمی‌توانیم این کار را انجام دهیم چرا که حداقل انتظار ما این است که خط مورد نظر بتواند سفیدها و سیاه‌های تصویر را از یکدیگر جدا کند.



شکل ۶: نقاط سفید در نمودار بیانگر خرگوش‌ها و نقاط سیاه بیانگر سمورها هستند

به ترتیب p_i و q_i را خرگوش و سمور iام معرفی می‌کنیم. می‌خواهیم خط مناسبی به صورت معادله خط $ax + b - y = 0$ در ابتدا قیودی را برای متغیرها پیدا می‌کنیم که هر جوابی برای آن خطی باشد که نقاط سیاه و سفید نمودارمان را از یکدیگر جدا کند. برای این کافی است خطی را بیابیم که نقاط سیاه در پایین آن و نقاط سفید در بالای آن ظاهر شوند پس:

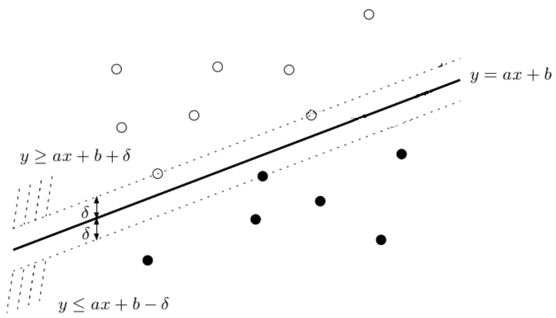
$$y > ax_{q_i} + b \quad i \in \{1, \cdots, m\}$$
$$y < ax_{p_i} + b \quad i \in \{1, \cdots, n\}$$



شکل ۷: همه‌ی این خطوط نقاط را جداسازی می‌کنند

به طور شهودی می‌توان گفت خطی همه‌ی خرگوش‌ها را درست شناسایی می‌کند که از داده‌های سفید فاصله داشته باشد و به داده‌های سمورها نزدیک‌تر باشد. و همینطور خطی همه‌ی سمورها را درست شناسایی می‌کند که از داده‌های سیاه فاصله‌ی بیشتری داشته باشد و به داده‌های سمورها نزدیک‌تر باشد. بنابراین شهودا خطی مناسب‌ترین خط است که فاصله‌ی آن از هر دو نوع نقطه زیاد باشد. برای اینکه بتوانیم بهترین خط را انتخاب کنیم که با داده‌های موجود طوری خطوط را جدا کند که بتواند نتیجه‌ی آزمایش‌های آینده را پیشبینی کند ازین برنامه‌ریزی خطی استفاده می‌کنیم:

$$\begin{aligned} \text{بیشینه کن} \quad & \delta \\ \text{که} \quad & y_{p_i} \geq ax_{p_i} + b + \delta \quad i \in \{1, \cdots, n\} \\ & y_{q_i} \leq ax_{q_i} + b - \delta \quad i \in \{1, \cdots, m\} \end{aligned}$$



شکل ۸: خطی پیدا می‌کنیم که فاصله‌اش از دو نوع نقطه بیشینه باشد

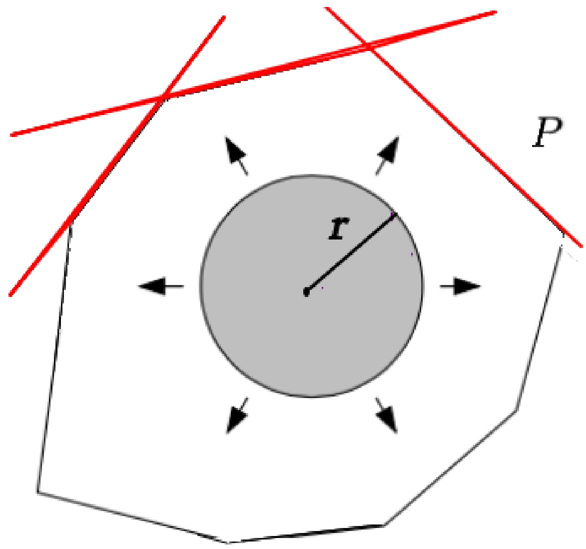
ممکن است این تابع جداکننده نقاط یک تابع خطی نباشد. بلکه تابعی سهی باشد. اگر این تابع سهمی درجه دو باشد می‌توانیم برنامه‌ریزی خطی زیر را برای آن بنویسیم:

$$\begin{array}{lll} \text{بیشینه کن} & \delta & \\ \text{که} & y_{p_i} \geq ax_{p_i}^۲ + bx_{p_i} + c + \delta & i \in \{۱, \cdots, n\} \\ & y_{q_i} \leq ax_{p_i}^۲ + bx_{p_i} + c - \delta & i \in \{۱, \cdots, m\} \end{array}$$

با همین روش می‌توان برنامه‌ریزی خطی هایی نوشت برای دو دسته نقاط روی صفحه یا روی فضا با بعدهای بیشتر که با تابعی به صورت $F(x) = a_۱ f_۱(x) + a_۲ f_۲(x) + \cdots + a_k f_k(x)$ این دو دسته نقطه در صورت امکان از یکدیگر جدا شود. به این صورت که توابع f_i توابع مشخصی هستند که می‌توانند خطی نباشند و a_i ضرایب حقیقی هستند. شهودا باید شروط $F(x_{p_i}) > ۰$ و $F(x_{q_i}) < ۰$ برقرار شوند.

۴.۲ بزرگ‌ترین دیسک در چندضلعی محدب

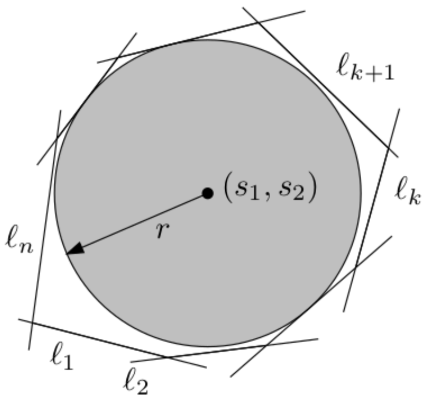
می‌خواهیم بیشترین r را بیابیم که دایره‌ای به شعاع r در چندضلعی مشخصی قرار بگیرد. برای هر ضلع i از چندضلعی (a_i, b_i) داریم که معادله خط آن ضلع به صورت $l_i : y = a_i x + b_i$ خواهد بود.



شکل ۹: دیسک و چندضلعی

اگر مرکز دایره را $(s_۱, s_۲)$ در نظر بگیریم فاصله‌ی ضلع i از نقطه که به دلیل محدب بودن چندضلعی با فاصله‌ی خط l_i از نقطه برابر است به صورت عبارت $\dfrac{|s_۲ - a_i s_۱ - b_i|}{\sqrt{a_i^۲ + ۱}}$ به دست می‌آید. نکته‌ی قابل توجه آن است که اگر خط بالای نقطه‌ی مورد نظر باشد مقدار داخل قدر مطلق منفی

و اگر خط پایین نقطه باشد این مقدار مثبت است. هر دایره‌ای را درون چند ضلعی در نظر بگیریم خط l_i یا در پایین مرکز آن قرار دارد یا بالای آن. خط‌ها را طوری مطابق شکل شماره‌گذاری می‌کنیم که خط‌های l_1 تا l_k زیر نقطه‌ی مرکز و باقی بالای آن باشند.



شکل ۱۰: نحوه‌ی شماره‌گذاری خطوط

$$
\begin{aligned}
r &\leq \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}, \quad i \in \{1, 2, \cdots, k\} \\
-r &\geq \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}, \quad i \in \{k+1, k+2, \cdots, n\}
\end{aligned}
$$

پس در برنامه‌ریزی خطی می‌توان داشت:

$$
\begin{aligned}
\text{بیشینه کن} \quad & r \\
\text{که} \quad & r \leq \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}, \quad i \in \{1, 2, \cdots, k\} \\
& -r \geq \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}, \quad i \in \{k+1, k+2, \cdots, n\}
\end{aligned}
$$

مراجع

[GM07] Bernd Gärtner and Jiří Matoušek. Understanding and using linear programming. Springer, 2007.