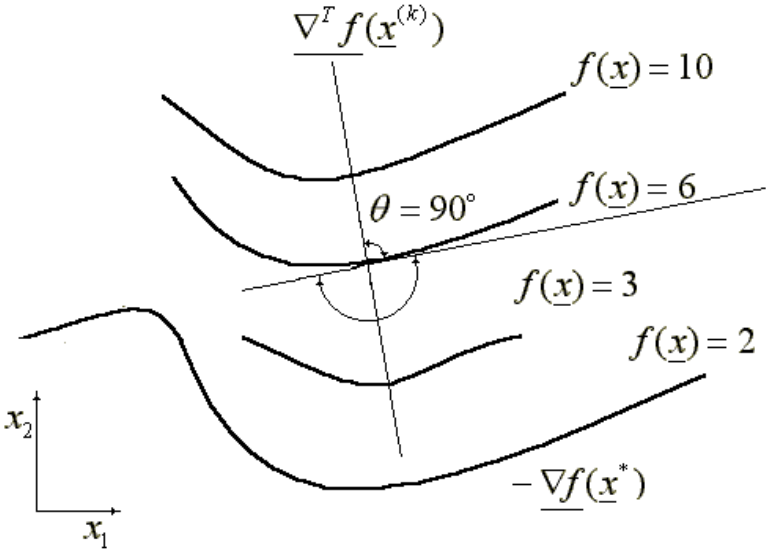
فصل ۸- بهینه‌سازی نامقید، توابع چندمتغیره، روش‌های گرادیان

ورود به مطلب- در این روش‌ها، از گرادیان (مشتق) تابع استفاده می‌شود. فرض می‌شود ابعاد و تعداد متغیرهای سیستم آن‌قدر بزرگ است که حل غیر خطی دستگاه مزبور (روش‌های مبتنی بر قضیه) بسیار سخت می‌باشد. علی‌ای‌حال، روش‌های مبتنی بر گرادیان موسوم به روش‌های غیرمستقیم نیز هستند، چرا که به‌زعمی از خاصیت گرادیان تابع (همان قضیه معروف) استفاده می‌کنند. دوباره در اینجا علت تنوع و تکثر روش‌ها، اختلاف در جهت جستجو و گام جستجو می‌باشد. برای اخذ ایده و قرینه ذهنی به شکل و ادعای زیر توجه کنید.

«برای مسائل مینیمم‌سازی یک جهت جستجوی خوب باید مقدار تابع هدف را در هر تکرار کم کند».

یعنی اگر $\underline{x}^{(0)}$ نقطه شروع باشد و $\underline{x}^{(1)}$ نقطه جدید، باید $f(\underline{x}^{(1)}) \prec f(\underline{x}^{(0)})$ ، اگر این جهت را با $\underline{s}$ نمایش دهیم، موسوم به جهت فروشو[1] خواهد بود و باید در شرط زیر صدق کند: $\nabla^T f(\underline{x})\underline{s} \prec 0$



شکل ۱- یک منحنی تراز نمونه برای تابع دومتغیره.

اثبات: به شکل ۱توجه کنید، ضرب داخلی دو بردار:

$$\nabla^T f(\underline{x})\underline{s}^{(k)} = \left|\nabla^T f(\underline{x}^{(k)})\right| \times \left|\underline{s}^{(k)}\right| \times \cos\theta$$

حال اگر $\theta = 90^\circ$ باشد، یک حالت خنثی داریم، چون $\underline{s}$ بر کنتور مماس است و مقدار تابع نه زیاد می‌شود و نه کم می‌شود و اگر $0 \le \theta \le 90^\circ$، آنگاه هیچ بهبودی نداریم و تابع زیاد می‌شود، بنابراین، باید $\theta \ge 90^\circ$ باشد تا بتوان مقدار تابع را کم کرد.

[1] Descent

در ادامه به تعیین جهت $\underline{s}$ (به جای دامنه و مکان هندسی $\underline{s}$) می‌پردازیم و مشخصاً دو روش کلاسیک «گرادیان» (معروف به تندترین شیب فروشو[1] (فراشو)) و «گرادیان مزدوج» را بحث تفصیلی می‌کنیم.

روش گرادیان یا تندترین شیب فروشو

در این روش، ایده اصلی، مبتنی بر مشتق اول تابع هدف است و می‌گوید جهت جستجوی خوب، عکس جهتی است که تابع بیشترین رشد را دارد، یعنی جهت $-\nabla f$، تندترین یا بیشترین کاهش در f را منجر می‌شود (لااقل به‌طور محلی)؛ یعنی: $\underline{s}^{(k)} = -\underline{\nabla f}(\underline{x}^{(k)})$

واگر دقیق‌تر بگوییم: $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \Delta\underline{x}^{(k)} = \underline{x}^{(k)} + \lambda^{(k)}\underline{s}^{(k)} = \underline{x}^{(k)} - \lambda^{(k)}\underline{\nabla f}(\underline{x}^{(k)})$

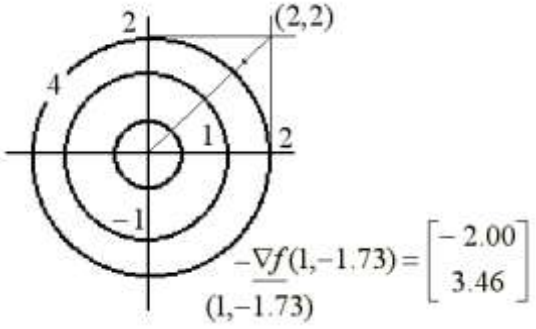
همان‌طور که اشاره شد، روش تندترین شیب فروشو، فقط جهت جستجو را بحث کرده است و راجع به $\lambda^{(k)}$ صحبت نکرده‌است. به‌هر حال، روش‌های مختلف گرادیان یا تندترین شیب فروشو، به‌خاطر نحوه تعیین $\lambda^{(k)}$ با هم تفاوت دارند. دو روش متداول وجود دارد، یکی بهینه‌کردن $\lambda^{(k)}$ یک‌بعدی یا همان بهینه‌سازی در راستای جستجو می‌باشد و دیگری $\lambda^{(k)}$ ثابت (معروف به قانون MIT[2]) می‌باشد. دقت شود که حرکت در جهت $-\nabla f$ و حتی گام بهینه $\lambda^{(k)}$ الزاماً منجر به حصول مینیمم نمی‌شود، مگر اینکه با حالت خاصی از توابع کوادراتیک روبرو باشیم.

به‌طور مثال، تابع مقیاس‌شده کواردراتیک زیر را در نظر بگیرید و یک نقطه شروع نوعی مثل $[2 \quad 2]^T$ فرض کنید:

$$f(\underline{x}) = x_1^2 + x_2^2 \quad \rightarrow \quad \underline{\nabla f}(\underline{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} \quad \rightarrow \quad \underline{\nabla f}([2 \quad 2]^T) = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$H(\underline{x}) = H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad , \quad \underline{s} = -\underline{\nabla f} = -\begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

مکان کنتوری تابع هدف، دایره‌های متحدالمرکز هستند (چون ترم تداخل نداریم و هر متغیر مقیاس شده‌است) و دقت کنید جهت $-\nabla f$، همیشه به مرکز اشاره می‌کند، یعنی کافیست باندازه مقتضی و مناسب (λ^{opt}) گام برداریم.



شکل ۲. منحنی تراز یک تابع دومتغیره مقیاس‌شده بدون ترم تداخلی.

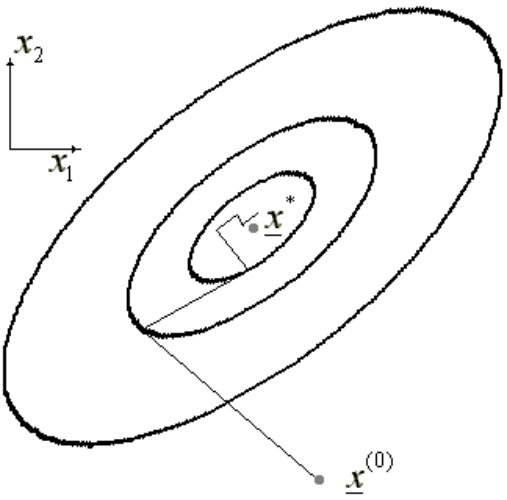
حال فرض کنید تابع کوادراتیک و محدب مزبور، مثل مساله فوق، خوب مقیاس نشده باشد و همچنین کج باشد (به تعبیری عناصر غیرقطری صفر نباشد.) آنگاه جهت $-\nabla f$ متوجه اپتیمم نخواهد بود و به نظر می‌رسد باید با یک تبدیلی

[1] Steepest descent (ascent) Method

[2] MIT Rule

(در جبر خطی، ماتریس می‌تواند علامت تبدیل باشد، فافهم) صاف شود و جهت آن درست شود.

به شکل زیر نگاه کنید:



شکل ۳– منحنی تراز یک تابع دومتغیره کلی.

همانطور که معلوم است در هر تکرار، جهت جستجو عمود بر کنتور است و به خاطر شکل خاص مساله، حالت زیگزاگ بوجود می‌آید. به حال این را (زیگزاگ رفتن) برای هر تابع می‌توان بدست آورد، یعنی λ^* راحساب کنیم:

$$\frac{df}{d\lambda} = \underbrace{\sum \frac{d}{d\lambda} x_i(\lambda) \frac{df}{dx_i}}_{ChainRule} = \sum s_i \frac{\partial f}{\partial x_i} = \underline{s}^T \underline{\nabla f} = 0$$

یعنی می‌خواهیم $\partial f / \partial x_i = 0$ (در انتهای جستجو) بطوریکه $\underline{s}^T \underline{\nabla f}(\underline{x}^{(k+1)}) = 0$ ولی به‌طور استشمامی هم می‌توان فهمید که نیاز نیست اینقدر دقیق باشیم و در هر گام λ^* را دقیقاً حساب کنیم، چون این روش، رفتار نسبتاً نامطلوب زیر را دارد:

در گام‌های (تکرارهای) اول، مقدار تابع به سرعت کاهش می‌یابد ولی وقتی نزدیک مقدار بهینه می‌شویم (نقطه $\underline{x}^*$) سرعت همگرایی کند می‌شود.

خلاصه (الگوریتم) روش تندترین شیب

گام اول: یک نقطه اولیه برای شروع مثل $\underline{x}^{(0)}$ انتخاب کنید. بعد از شروع جستجو، نقطه اولیه، همیشه نقطه جاری یا قبلی خواهد بود و با $\underline{x}^{(k)}$ نمایش می‌دهیم، پس برای گام اول، منظورمان $k = 0$ است.

گام دوم: مقادیر مشتقات پاره‌ای را چه تحلیلی و چه عددی حساب کنید: $\frac{\partial f}{\partial x_j}(\underline{x}^{(k)}) \quad , \quad j = 1, \cdots, n$

گام سوم: جهت جستجو را بدست آورید: $\underline{s} = -\underline{\nabla f}(\underline{x}^{(k)})$

گام چهارم: از رابطه زیر، نقطه بعدی را بدست آورید: $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \lambda^{(k)} \underline{s}^{(k)}$

برای $\lambda^{(k)}$ یا یک مقدار ثابت بگذارید یا از روش جستجوی راستا، مقدار بهینه آن را بدست آورید.

گام پنجم: محک و معیار پایان کار (مثلاً $f(\underline{x}^{(k+1)})$ را با $f(\underline{x}^{(k)})$ مقایسه کنید).

روش فلچر- ریوز[1] یا گرادیان مزدوج

این روش که توسط این دو نفر ابداع شد[5,6]، جهت جستجو را همیشه مزدوج گرادیان انتخاب کرده و به‌طور دقیق $\underline{\lambda}^*$ را در هر گام به‌دست می‌آورد. همچنین ثابت کرده‌اند اگر تابع کوادراتیک باشد، آنگاه دارای این خاصیت مطلوب است که دارای سرعت همگرایی درجه دوم (کوادراتیک) است. خلاصه الگوریتم:

گام اول: نقطه شروع اولیه‌ای مثل $\underline{x}^{(0)}$ انتخاب کنید و مقدار تابع $f(\underline{x}^{(k)})$ را محاسبه کرده و جهت جستجو را خلاف جهت گرادیان در نظر بگیرید: $\underline{s}^{(0)} = -\underline{\nabla f}(\underline{x}^{(0)})$

گام دوم: جهت قبلی جستجو را حفظ کنید (یعنی $-\underline{\nabla f}(\underline{x}^{(0)})$) و یک گام جلو روید: $\underline{x}^{(1)} = \underline{x}^{(0)} + \lambda^* \underline{s}^{(0)}$ مقدار λ^* را بطور دقیق توسط یکی از روش‌های جستجوی راستا بدست آورید.

گام سوم: مقدار $f(\underline{x}^{(1)})$ و $\underline{\nabla f}(\underline{x}^{(1)})$ را حساب کرده و جهت جدید جستجو را به شکل ترکیبی (ترکیب خطی) زیر محاسبه کنید:

$$\underline{s}^{(1)} = -\underline{\nabla f}(\underline{x}^{(1)}) + \underline{s}^{(0)} \frac{\underline{\nabla f}^T(\underline{x}^{(1)})\underline{\nabla f}(\underline{x}^{(1)})}{\underline{\nabla f}^T(\underline{x}^{(0)})\underline{\nabla f}(\underline{x}^{(0)})}$$

و برای kامین تکرار:

$$\underline{s}^{(k+1)} = -\underline{\nabla f}(\underline{x}^{(k+1)}) + \underline{s}^{(k)} \frac{\underline{\nabla f}^T(\underline{x}^{(k+1)})\underline{\nabla f}(\underline{x}^{(k+1)})}{\underline{\nabla f}^T(\underline{x}^{(k)})\underline{\nabla f}(\underline{x}^{(k)})}$$

برای یک تابع کوادراتیک می‌توان نشان داد که این جهت‌های متوالی با هم مزدوج هستند و بعد از $k = n$ تکرار، باید $\underline{x}^{(n+1)}$ را جایگزین $\underline{x}^{(0)}$ کنید. به‌عبارت دقیق‌تر، جهات جستجو در یک سیکل درونی به‌طور مزدوج حساب می‌شوند. این سیکل دارای n تکرار است و برای شروع مجدد باید $\underline{x}^{(0)}$ را معادل $\underline{x}^{(n+1)}$ قرار داد.

گام چهارم: محکّ اتمام، در صورت صدق‌نکردن، به مرحله سوم برگردید.

گام پنجم: (گام $k = n$) محک اتمام به‌صورت $\left\|\underline{s}^{(k)}\right\| \prec \varepsilon$ در نظرگرفته و اگر صدق نکرد، $\underline{x}^{(0)} = \underline{x}^{(n+1)}$ و به مرحله اول برگردید.

روش پولاک- ریبیره[2]

یک واریانت دیگر از گرادیان مزدوج توسط پولاک و ریبیره پیشنهاد شده‌است[7]. آنها به‌جای مقیاس کردن گرادیان برای انتخاب جهت مزدوج بعدی (در روش فلچر-ریوز) عملا از مقیاس‌کردن تغییر گرادیان استفاده کرده‌اند؛

$$\underline{s}^{(k+1)} = -\underline{\nabla f}(\underline{x}^{(k+1)}) + \underline{s}^{(k)} \frac{\Delta\underline{\nabla f}^T(\underline{x}^{(k-1)})\underline{\nabla f}(\underline{x}^{(k)})}{\underline{\nabla f}^T(\underline{x}^{(k)})\underline{\nabla f}(\underline{x}^{(k)})}$$

[1] Fletcher-Reeves
[2] Polak-Ribiére

نحوه شروع پاول- بییله[1]

تمامی روش‌های گرادیان مزدوج، به‌طور متناوب جهت جستجو را بازمقداردهی[2] به تندترین شیب (منفی گرادیان) می‌کنند، یعنی روش استاندارد اینست که بعد از مقداردهی ، جهت جستجو را معادل منفی گرادیان قرار می‌دهند. این نحوه بازشروع، تنها روش موجود نیست، بلکه واریانت‌هایی مثل پیشنهاد پاول بر مبنای ایده بییله وجود دارد که می‌تواند سرعت الگوریتم تکرار و جستجو را بهبود بخشد [8]. مبنای این روش بر میزان اورتوگونالیتی بین گرادیان جاری و گرادیان قبلی استوارست، بدین معنی که اگر شرط نامساوی زیر (اورتوگونالیتی) برقرار باشد، آنگاه جهت جستجو با مقدار منفی گرادیان مقداردهی می‌شود:

$$\left|\nabla f^{T}(\underline{x}^{(k-1)})\cdot\nabla f(\underline{x}^{(k)})\right| \geq 0.2\left\|\nabla f(\underline{x}^{(k)})\right\|^{2}$$

گرادیان مزدوج مقیاس‌شده یا روش مولر[3]

همان‌طور که قبلا نیز مفصلا بحث شده‌است، بسیاری از روش‌های جستجو در مراحل میانی نیازمند بهینه‌سازی تک‌متغیره (جستجوی راستا) هستند. برای مسایل با مقیاس بزرگ یا توابع هدف پیچیده این مرحله به‌خاطر فراخوانی زیاد تابع هدف، زمان‌بر خواهد بود، لذا مولر پیشنهاد بهبود الگوریتم گرادیان مزدوج را از طریق جستجوی راستای مقیاس‌شده کرده‌است [9]. شرح الگوریتم بسیار طولانی و مفصل است و در اینجا مجال بحث آن نیست؛ فقط به‌طور خلاصه اساس ایده بر این قرارست که مفهوم منطقه قابل اعتماد[4] که بعدا (در فصول و بخش‌های بعدی) پرداخته خواهد شد جایش با مفهوم گرادیان مزدوج عوض می‌شود.

روش‌های جستجوی غیر مستقیم درجه دوم

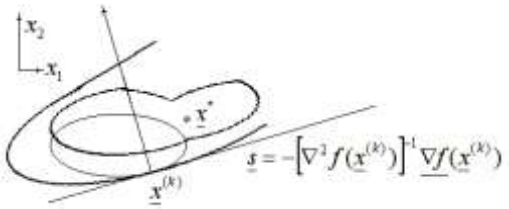
یک راه اصلاح روش گرادیان، این است که در مقام تشابه با حالت خوب مقیاس شده (یعنی کنتورها دایره باشند)، کاری کرد که جهت $-\nabla f$ درمختصات جدیدی بدست آید. یعنی تبدیل مختصاتی انجام دهیم که کنتورها در مختصات جدید، دایره شوند. اگر این کار در فضای جبر خطی صورت بگیرد، عملاً باید ماتریس تبدیل را پیدا کنیم. در ادامه، به جای محاسبه تبدیل مزبور (که نیاز به جبر خطی دارد)، از دیدگاه آنالیز این روش را اصلاح می‌کنیم.

اگر کمی تامل کنیم، می‌بینیم که جهت $-\nabla f$ (یا همان ∇f) عمود بر مماس بر کنتور یا به عبارت دیگر تقریب خطی (تقریب درجه یک) تابع می‌باشد. به شکل نگاه کنید، به نظر می‌رسد که در همان حوالی، اگر از تقریب درجه دوم سیستم استفاده کنیم، بهتر باشد و عملاً مشابه روش میانیابی درجه دوم، از تقریب فانکشنال حاصل، مینیمم را حدس بزنیم یا متشابهاً جهت جستجو را بدست آوریم.

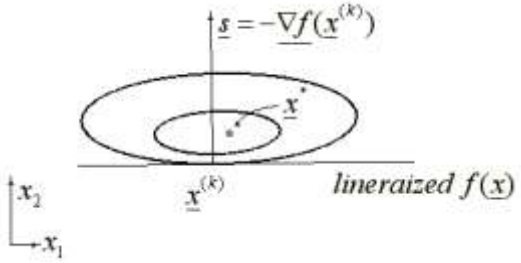
---

[1] Powell-Beale
[2] Reset
[3] Moller
[4] Trust Region

$$f(\underline{x}) \approx \underbrace{\underbrace{f(\underline{x}^{(k)}) + \underline{\nabla f}(\underline{x}^{(k)})^T \Delta \underline{x}^{(k)}}_{Linear\,(first\,order)\,approximation} + \frac{1}{2}(\Delta \underline{x}^{(k)})^T H(\underline{x}^{(k)}) \Delta \underline{x}^{(k)}}_{Quadratic\,(second\,order)\,approximation}$$



شکل ٥- تقریب درجه دوم تابع $f(\underline{x})$ در نقطه $\underline{x}^{(k)}$.



شکل ٤- تقریب درجه اول تابع $f(\underline{x})$ در نقطه $\underline{x}^{(k)}$.

روش نیوتن و آنالوگ‌های آن

این روش از اطلاعات درجه دوم نیز استفاده می‌کند، لذا می‌تواند اثر انحناء را نیز در نظر بگیرد و در نتیجه ایده جستجو را بهبود بخشد. برای محاسبه جهت جستجوی نیوتن،دقت کنید که فرض بر کوادراتیک بوده و هم جهت و هم طول گام در فرمولاسیون نهفته است:

$$\underline{\nabla f}(\underline{x}) = \underline{0} \quad \rightarrow \quad \underline{\nabla f}(\underline{x}) = \underline{\nabla f}(\underline{x}^{(k)}) + H(\underline{x}^{(k)}) \Delta \underline{x}^{(k)} = \underline{0} \quad \rightarrow \quad \text{(شرط مینیمم‌بودن)}$$

$$\Delta \underline{x}^{(k)} \equiv \underline{x}^{(k+1)} - \underline{x}^{(k)} = -\left[H(\underline{x}^{(k)})\right]^{-1} \underline{\nabla f}(\underline{x}^{(k)})$$

رابطه بالا را با حالت یک‌بعدی مقایسه کنید: $\Delta x^{(k)} \equiv x^{(k+1)} - x^{(k)} = -\dfrac{f'(x^{(k)})}{f''(x^{(k)})} = -\left[f''(x^{(k)})\right]^{-1} f'(x^{(k)})$

برای هماهنگی با بقیه روش‌ها و اینکه همه توابع را فقط می‌توان به‌طور تقریبی به‌صورت کوادراتیک فرض کرد، یک طول گام نیز اضافه می‌کنیم:

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \lambda^{(k)} \left[H(\underline{x}^{(k)})\right]^{-1} \underline{\nabla f}(\underline{x}^{(k)})$$

نکته: این رابطه را با حالت یک‌بعدی روش نیوتن بهبود یافته[1] (وجود فاکتور تعدیل[2]) مقایسه کنید.

$$x^{(k+1)} = x^{(k)} - \lambda \frac{g(x^{(k)})}{g'(x^{(k)})} = x^{(k)} - \lambda \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

به‌طوریکه $g(x) \equiv f'(x)$.

[1] Modified Newton Method
[2] Relaxation factor

نکته: این روش در آموزش شبکه‌های عصبی مصنوعی، معروف به روش نشر برگشتی اطلاعات[1] می‌باشد و اگر λ، مقدار ثابتی نباشد و بلکه در هر تکرار عوض شود، روش برگشتی تطبیقی[2] نام دارد.

نکته: جهت جستجوی $\underline{s}$ در اینجا شده است: $\underline{s} = -\left[H(\underline{x}^{(k)})\right]^{-1}\nabla f(\underline{x}^{(k)})$ و با روش درجه اول مقایسه کنید: $\underline{s} = -\nabla f(\underline{x}^{(k)})$، مقدار $\left[H(\underline{x}^{(k)})\right]^{-1}$ همان تبدیل مختصاتی است که ذکر شد.

نکته: روش نیوتن اصلی[3]، همان رابطه بالاست ولی با $\lambda^{(k)} = 1$.

نکته: می دانیم محاسبه ماتریس معکوس سخت است، لذا اکثراً معادله زیر را حل می کنند (روش نیوتن، $\lambda = 1$):

$$H(\underline{x}^{(k)})\Delta\underline{x}^{(k)} = -\nabla f(\underline{x}^{(k)})$$

مثال ۱- تابع کوادراتیک زیر را با روش نیوتن مینمم کنید.

$$\begin{cases} f(\underline{x}) = 4x_1^2 + x_2^2 - 2x_1x_2 \\ \underline{x}^{(0)} = [1 \quad 1]^T \end{cases}$$

حل: (دقت کنید چون ترم تداخل دارد با روش گرادیان نمی‌توان با یک تکرار به جواب رسید)

$$\nabla f(\underline{x}) = \begin{bmatrix} 8x_1 - 2x_2 \\ 2x_2 - 2x_1 \end{bmatrix} \quad , \quad H(\underline{x}) = \begin{bmatrix} 8 & -2 \\ -2 & 2 \end{bmatrix} \rightarrow H^{-1}(\underline{x}) = \begin{bmatrix} 1/6 & 1/6 \\ 1/6 & 2/3 \end{bmatrix}$$

$$\lambda = 1 \rightarrow \Delta\underline{x}^{(0)} = -H^{-1}\nabla f(\underline{x}^{(0)}) = -\begin{bmatrix} 1/6 & 1/6 \\ 1/6 & 2/3 \end{bmatrix}\begin{bmatrix} 6 \\ 6 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\rightarrow \underline{x}^{(1)} = \underline{x}^{(*)} = \underline{x}^{(0)} + \Delta\underline{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

روش آلترناتیو: برای محاسبه معکوس ، می‌توان دستگاه زیر را حل کرد:

$$\begin{bmatrix} 8 & -2 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \end{bmatrix} = -\begin{bmatrix} 6 \\ 0 \end{bmatrix} \rightarrow \begin{cases} \Delta x_1^{(0)} = -1 \\ \Delta x_2^{(0)} = -1 \end{cases}$$

این روش (نیوتن) و آنالوگ‌هایش (روش‌های شبه‌نیوتن یا v-metrics)، سریعترین و مدرن‌ترین روش‌های جستجوی غیرمستقیم هستند ولی بخاطر عیب‌های زیر، اصلاحاتی بویژه روی ماتریس هسیان صورت پذیرفت:

۱- روش مزبور، الزاماً حل عمومی را پیدا نمی‌کند (گرچه این عیب همه روشهاست).

۲- روش نیازمند محاسبه ماتریس معکوس یا حل دستگاه $n \times n$ است.

۳- روش نیازمند محاسبه تحلیلی مشتق اول و دوم (گرادیان و هسیان) می‌باشد.

۴- اگر در نهایت یا موقع حل، ماتریس هسیان مثبت معین نباشد، جواب یک نقطه زین‌اسبی است و نه مینیمم.

[1] Back propagation
[2] Back-propagation with Adaptive learning rate
[3] Original

به هر حال هر کدام از موارد گفته شده (جز مورد ۱) دارای راه‌حل هستند ولی همان‌طور که ذکر شد بیشترین مانور روی ماتریس هسیان می‌باشد. به‌طور مثال، در روش لونبرگ – مارکوارت[1]، به اجبار ماتریس هسیان را مثبت نگه‌می‌دارند.

$$\tilde{H}(\underline{x}) \equiv [H(\underline{x}) + \beta I] \quad or \quad [\tilde{H}(\underline{x})]^{-1} \equiv [H^{-1}(\underline{x}) + \lambda I]$$

به‌طوریکه β یا λ، ثوابت از پیش‌تعیین‌شده هستند. دقت شود باید از $\tilde{H}(\underline{x})$ به جای $H(\underline{x})$ استفاده کرد. لازم به ذکرست که معمولاً β را کوچکترین مقدار ویژه $H(\underline{x})$ اختیار می‌کنند.

خلاصه (الگوریتم) یک نسخه از روش لونبرگ – مارکوارت

گام اول: $\underline{x}^{(0)}$ و تولرانس (ε) را انتخاب کنید.

گام دوم: قرار دهید $k = 0$ و $\beta^{(0)} = 10^3$

گام سوم: گرادیان $\nabla f(\underline{x}^{(k)})$ را حساب کنید.

گام چهارم: اگر $\left\| \nabla f(\underline{x}^{(k)}) \right\| \prec \varepsilon$ برقرار بود، اختتام و در غیر این‌صورت ادامه دهید.

گام پنجم: جهت جستجو را انتخاب کنید: $\underline{s}^{(k)} = \left[H^{(k)}(\underline{x}) + \beta^{(k)} I\right]^{-1} \nabla f(\underline{x}^{(k)})$

گام ششم: نقطه بعدی را محاسبه کنید: $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \lambda^{(k)} \underline{s}^{(k)}$

گام هفتم: اگر $f(\underline{x}^{(k+1)}) \prec f(\underline{x}^{(k)})$ برقرار بود، به مرحله بعدی و گرنه به مرحله نهم بروید.

گام هشتم: قرار دهید: $\beta^{(k+1)} = \frac{1}{4}\beta^{(k)}$, $k = k+1$ ، برگردید به گام سوم.

گام نهم: قرار دهید $\beta^{(k+1)} = 2\beta^{(k)}$ ، بروید مرحله پنجم.

نکته: روش‌های دیگری نیز برای بهبود هسیان هست، نظیر فاکتوریزاسیون خالتسکی[2]: $H(\underline{x}^{(k)}) + D = LL^T$ به‌طوریکه D یک ماتریس قطری با عناصر غیر صفر می‌باشد.

روش دیویدن[3]، فلچر و پاول – DFP

در سال ۱۹۵۹، دیویدن، معضل منفی شدن ماتریس هسیان را گوشزد کرد و چهار سال بعد فلچر و پاول متدولوژی به‌روزرسانی ماتریس هسیان را تعمیم داده و قضایائی برای سرعت همگرایی آن ارائه دادند. بنا به قول فلچر [۱]، این روش در زمره قویترین و مقبول‌ترین روش‌های شبه‌نیوتن (معروف به روش‌های مقیاس متغیر[4]) می‌باشد و ایده اصلی آن به اینصورت است که در ابتدای تکرار (سیکل درونی)، ماتریس تبدیل مختصات به‌صورت ماتریس واحد در نظر گرفته می‌شود ولی در تکرارهای بعدی، ماتریس مربوطه به‌روز شده تا در نهایت به معکوس هسیان تبدیل می‌شود. اگر موتور تکرار را به‌صورت روبرو نمایش دهیم: $\underline{x}^{(k+1)} = \underline{x}^{(k)} - \lambda^{(k+1)} H^{(k)} \nabla f(\underline{x}^{(k)})$

ماتریس $H^{(k)}$ به این صورت به‌روز می‌شود: $H^{(k)} = H^{(k-1)} + A^{(k)} + B^{(k)}$

---

[1] Levenberg – Marquardt
[2] Cholsky
[3] Davidon
[4] Variable Metrics

ماتریس‌های $A^{(k)}, B^{(k)}$ توسط روابط زیر ارائه شده‌اند:

$$A^{(k)} = \frac{\left(\underline{x}^{(k)} - \underline{x}^{(k-1)}\right)\left(\underline{x}^{(k)} - \underline{x}^{(k-1)}\right)^T}{\left(\underline{x}^{(k)} - \underline{x}^{(k-1)}\right)^T \left(\underline{\nabla f}(\underline{x}^{(k)}) - \underline{\nabla f}(\underline{x}^{(k-1)})\right)}$$

$$B^{(k)} = \frac{H^{(k-1)}\left(\underline{\nabla f}(\underline{x}^{(k)}) - \underline{\nabla f}(\underline{x}^{(k-1)})\right)\left(\underline{\nabla f}(\underline{x}^{(k)}) - \underline{\nabla f}(\underline{x}^{(k-1)})\right)^T H^{(k-1),T}}{\left(\underline{\nabla f}(\underline{x}^{(k)}) - \underline{\nabla f}(\underline{x}^{(k-1)})\right)^T H^{(k-1)} \left(\underline{\nabla f}(\underline{x}^{(k)}) - \underline{\nabla f}(\underline{x}^{(k-1)})\right)}$$

الگوریتم با جستجو در راستای گرادیان و از نقطه $\underline{x}^{(0)}$ شروع می‌شود ($k=0$):

$$\underline{x}^{(1)} = \underline{x}^{(0)} - \lambda^{(1)} H^{(0)} \underline{\nabla f}(\underline{x}^{(0)})$$

به‌طوری‌که $H^{(0)} = I$ می‌باشد (I ماتریس واحد است).

نکته: اگر از روش‌های دقیق جستجوی راستا استفاده کنیم و تابع هدف به فرم کوادراتیک باشد، آنگاه الگوریتم مزبور بعد از n تکرار به نقطه بهینه همگرا می‌شود.

نکته: دو ماتریس $A^{(k)}, B^{(k)}$، طوری نرمالیزه و مشابه مشتق اول و دوم پیشنهاد شده‌اند که نهایتاً به معکوس هسیان و جمله خنثی ماتریس واحد برسند، به‌عبارت دقیق‌تر، ماتریس‌های $A^{(k)}, B^{(k)}$ دارای خواص زیر هستند:

$$\sum_{k=1}^{n} A_k = H^{-1} \quad , \quad \sum_{k=1}^{n} B_k = -H^{(0)} = -I$$

مثال۲- تابع سه متغیره کوادراتیک زیر را در نظر گرفته و تحقیق کنید که با استفاده از روش DFP، تنها با سه تکرار به جواب می‌رسیم، نقطه اولیه را به دلخواه، مبدا ($\underline{x}^{(0)} = [0 \quad 0 \quad 0]^T$) فرض کنید:

$$Minimize \quad f(\underline{x}) = 5x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3 - 2x_1x_3 - 6x_3$$

حل:

$$\underline{\nabla f} = \begin{bmatrix} 10x_1 + 2x_2 - 2x_3 \\ 2x_1 + 4x_2 + 2x_3 \\ -2x_1 + 2x_2 + 4x_3 - 6 \end{bmatrix} \quad , \quad H = \begin{bmatrix} 10 & 2 & -2 \\ 2 & 4 & 2 \\ -2 & 2 & 4 \end{bmatrix}$$

استفاده از فرمول تکرار:

$$\underline{x}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \lambda^{(1)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ -6 \end{bmatrix} = \quad (\text{جستجو در راستا} \rightarrow \lambda^{*(1)} = 1/4) = \begin{bmatrix} 0 \\ 0 \\ 3/2 \end{bmatrix} \rightarrow \underline{\nabla f}(\underline{x}^{(1)}) = \begin{bmatrix} -3 \\ -3 \\ 0 \end{bmatrix}$$

برای تکرار بعدی: $\underline{x}^{(2)} = \underline{x}^{(1)} - \lambda^{(2)} H^{(1)} \underline{\nabla f}(\underline{x}^{(1)})$، به‌طوریکه

$$H^{(1)} = H^{(0)} + A^{(1)} + B^{(1)} \quad , \quad A^{(1)} = \frac{\begin{bmatrix} 0 \\ 0 \\ 3/2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 3/2 \end{bmatrix}}{\begin{bmatrix} 0 & 0 & 3/2 \end{bmatrix} \begin{bmatrix} -3 \\ 3 \\ 6 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

$$B^{(1)} = -\frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 3 \\ 6 \end{bmatrix} \begin{bmatrix} -3 & 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}{\begin{bmatrix} -3 & 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 3 \\ 6 \end{bmatrix}} = \begin{bmatrix} -1/6 & 1/6 & 1/3 \\ 1/6 & -1/6 & -1/3 \\ 1/3 & -1/3 & -2/3 \end{bmatrix}$$

$$H^{(1)}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}+\begin{bmatrix}0&0&0\\0&0&0\\0&0&1/4\end{bmatrix}+\begin{bmatrix}-1/6&1/6&1/3\\1/6&-1/6&-1/3\\1/3&-1/3&2/3\end{bmatrix}=\begin{bmatrix}5/6&1/6&1/3\\1/6&5/6&-1/3\\1/3&-1/3&7/12\end{bmatrix}$$

با جستجوی در راستا، مقدار دقیق $\lambda^{*(2)}$، معادل $1/2$ بدست می‌آید.

و برای تکرار سوم:

$$\begin{cases}\underline{x}^{(3)}=\underline{x}^{(2)}-\lambda^{(3)}H^{(2)}\underline{\nabla f}(\underline{x}^{(2)})\\ H^{(2)}=H^{(1)}+A^{(2)}+B^{(2)}=\begin{bmatrix}1/16&-1/6&1/6\\-1/6&29/30&-17/30\\1/6&-17/30&37/60\end{bmatrix}\end{cases}$$

$$\lambda^{*(3)}=5/12 \quad (\text{با جستجوی دقیق در راستا})=$$

و با جاگذاری، $\underline{x}^{(3)}$ به‌صورت زیر بدست می‌آید که همان نقطه اپتیمم است:

$$\underline{x}^{(3)}=\begin{bmatrix}1\\-1\\5/2\end{bmatrix}-(5/12)\begin{bmatrix}1/6&-1/6&1/6\\-1/6&29/30&-17/30\\1/6&-17/30&37/30\end{bmatrix}\begin{bmatrix}3\\3\\0\end{bmatrix}=\begin{bmatrix}1\\-2\\3\end{bmatrix}=\underline{x}^{(*)}$$

روش برویدن[1]، فلچر، گلفارب[2] و شانو[3] – BFGS

این روش تعمیم روش DFP برای به روز کردن ماتریس بهره یا ماتریس تبدیل می‌باشد. یک خاصیت جالب فرمول به‌روزرسانی این است که برای محاسبه λ^{opt}، الزامی به استفاده از حل دقیق نیست، بلکه حتی با روش‌های حذف بازه یک متغیره مثل فیبوناچی نیز کار می‌کند و نهایتا به جواب بهینه می‌رسد. این خاصیت به‌خاطر محدودیت محاسباتی کامپیوترها و امکان تجمع خطای عددی، مورد استقبال کدهای کامپیوتری برای کاربردهای صنعتی و مقیاس بزرگ شده است. روند الگوریتم مثل روش DFP است با این تفاوت که فرمول تغییر ماتریس تبدیل از رابطه زیر به‌دست می‌آید:

$$H^{(k+1)}=H^{(k)}-\left[\frac{H^{(k)}\underline{\gamma}^{(k)}\underline{\delta}^{(k),T}+\underline{\delta}^{(k)}\underline{\gamma}^{(k),T}H^{(k)}}{\underline{\delta}^{(k),T}\underline{\gamma}^{(k)}}\right]+\left[\frac{\underline{\gamma}^{(k),T}H^{(k)}\underline{\gamma}^{(k)}}{\underline{\gamma}^{(k),T}\underline{\gamma}^{(k)}}\right]\left[\frac{\underline{\delta}^{(k),T}H^{(k)}\underline{\delta}^{(k)}}{\underline{\delta}^{(k),T}\underline{\delta}^{(k)}}\right]$$

به‌طوریکه

$$\underline{\gamma}^{(k)}=\underline{\nabla f}(\underline{x}^{(k+1)})-\underline{\nabla f}(\underline{x}^{(k)})=\Delta\underline{\nabla f}(\underline{x}^{(k)})$$
$$\underline{\delta}^{(k)}=\underline{x}^{(k+1)}-\underline{x}^{(k)}=\Delta\underline{x}^{(k)}$$

[1] Broyden
[2] Golfarb
[3] Shanno

روش وتری تک‌گام[1]

این روش عملا ذات‌البین روش‌های درجه دومی و نیوتنی نظیر BFGS و روش‌های غیر گرادیانی مثل فلچر – ریوز میباشد. انگیزه این مصالحه و پل‌زدن بین دو فامیلی مزبور برخاسته از صرفه‌جویی در حافظه و عملیات محاسباتی زیاد است که منجر به این روش شبه‌نیوتنی (مقیاس متغیر یا وتری) شده‌است[10]. در این روش فرض می‌شود که ماتریس هسیان در مرحله قبلی یک ماتریس یکه می‌باشد، لذا دیگر نیازی به محاسبه معکوس یا حل دستگاه نمی‌باشد! برای جزئیات باید به مرجع مربوطه مراجعه کرد.

مثال۳- همان مثال قبلی (روش DFP) را با روش BFGS، حل کنید:

$$\underline{x}^{(0)}=[0\quad 0\quad 0]^T\ ,\ \underline{\nabla f}(\underline{x}^{(0)})=[0\quad 0\quad -6]^T$$
$$\underline{x}^{(1)}=[0\quad 0\quad 3/2]^T\ ,\ \underline{\nabla f}(\underline{x}^{(1)})=[-3\quad 3\quad 0]^T$$
$$\underline{x}^{(2)}=\underline{x}^{(1)}-\lambda^{(2)}H^{(1)}\underline{\nabla f}(\underline{x}^{(1)})$$

به‌طوریکه:

$$H^{(1)}=H^{(0)}-\left[\frac{H^{(0)}\underline{\gamma}^{(0)}\underline{\delta}^{(0),T}+\underline{\delta}^{(0)}\underline{\gamma}^{(0),T}H^{(0}}{\underline{\delta}^{(0),T}\underline{\gamma}^{(0)}}\right]+\left[\frac{\underline{\gamma}^{(0),T}H^{(0)}\underline{\gamma}^{(0)}}{\underline{\gamma}^{(0),T}\underline{\gamma}^{(0)}}\right]\left[\frac{\underline{\delta}^{(0),T}H^{(0)}\underline{\delta}^{(0)}}{\underline{\delta}^{(0),T}\underline{\delta}^{(0)}}\right]$$

$$H^{(0)}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\ ,\ \underline{\delta}^{(0)}=\underline{x}^{(1)}-\underline{x}^{(0)}=\begin{bmatrix}0\\0\\3/2\end{bmatrix}\ ,\ \underline{\gamma}^{(0)}=\underline{\nabla f}(\underline{x}^{(1)})-\underline{\nabla f}(\underline{x}^{(0)})=\begin{bmatrix}-3\\3\\6\end{bmatrix}$$

$$\underline{\delta}^{(0).T}\underline{\gamma}^{(0)}=9\ ,\ \underline{\gamma}^{(0),T}H^{(0)}\underline{\gamma}^{(0)}=54\ \rightarrow$$

$$H^{(1)}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}-\begin{bmatrix}0&0&-1/2\\0&0&1/2\\-1/2&1/2&2\end{bmatrix}+\begin{bmatrix}0&0&0\\0&0&0\\0&0&7/4\end{bmatrix}=\begin{bmatrix}1&0&1/2\\0&1&-1/2\\1/2&-1/2&3/4\end{bmatrix}$$

مقدار $\lambda^{(2)}$ با مینیمم سازی یک متغیره، به‌طور دقیق معادل $1/3$ محاسبه می‌شود، در نتیجه:

$$\underline{x}^{(2)}=[1\quad -1\quad 5/2]^T\ ,\ \underline{\nabla f}(\underline{x}^{(2)})=[3\quad 3\quad 0]^T$$

تکرار بعدی:

$$\underline{x}^{(3)}=\underline{x}^{(2)}-\lambda^{(3)}H^{(2)}\underline{\nabla f}(\underline{x}^{(2)})$$
$$\underline{\delta}^{(1)}=[1\quad -1\quad 1]^T\ ,\ \underline{\gamma}^{(1)}=[6\quad 0\quad 0]^T\ ,\ \underline{\delta}^{(1),T}\underline{\gamma}^{(1)}=6\ ,\ \underline{\gamma}^{(1),T}H\underline{\gamma}^{(1)}=36$$
$$H^{(2)}=\begin{bmatrix}1&0&1/2\\0&1&-1/2\\1/2&-1/2&3/4\end{bmatrix}-\begin{bmatrix}2&-1&3/2\\-1&0&-1/2\\3/2&-1/2&1\end{bmatrix}+\begin{bmatrix}7/6&-7/6&7/6\\-7/6&7/6&-7/6\\7/6&-7/6&7/6\end{bmatrix}$$
$$=\begin{bmatrix}1/6&-1/6&1/6\\-1/6&13/6&-7/6\\1/6&-7/6&11/12\end{bmatrix}$$

مقدار $\lambda^{*(3)}$ با استفاده از روش‌های مینیمم‌سازی یک متغیره معادل $1/6$ به‌دست می‌آید و با جایگذاری به همان مقدار بهینه $\underline{x}^{(3)}=\underline{x}^{(*)}=[1\quad -2\quad 3]^T$ می‌رسیم.

[1] One Step Secant Algorithm - OSS

مراجع

[1]. Fletcher, R.; *Practical Methods of Optimization , Vol. I, Unconstrained Optimization*, John wiley & Sons, Inc., New York, 1981.

[2]. Gill, P.E., Murray, E., Wright, M.H., *Practical optimization*, Acaemic Press, New York, 1981.

[3]. McCormick, G.P., *Nonlinear Programming: Theory, Algorithms and Applications*, John Wiley & Sons, Inc., New York, 1983.

[4]. Himmelblau, D.M., *Applied Nonlinear Programming*, McGraw-Hill Book Co., New york, 1972.

[5]. Fletcher, R. and C. M. Reeves, "Function Minimization by Conjugate Gradients", *Computer J*., Vol. 7, pp. 149-154, 1964.

[6]. Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996.

[7]. Powell, M. J. D., "Restart Procedures for Conjugate Gradient Method", *Mathematical Programming*, Vol. 12, pp. 241-254, 1977.

[8]. Beale, E. M. L., "*A Derivation of Conjugate Gradients*", in F.A. Lootsma, ed., Numerical Methods for nonlinear Optimization, London: Academic Press, 1972.

[9]. Moller, M. F., "*A Scaled Conjugate Gradient Algorithm for fast Supervised Learning*", Neural Networks, Vol. 6, pp. 525-533, 1993.

[10]. Battiti, R., "*First and Second Order Methods for Learning: Between Steepeest Descent and Newton's Method*", Neural Computation, Vol. 4, No. 2, pp. 141-166, 1992.