Introduction to Machine Learning - HW3 - Q2

Professors: Abolghasemi & Arabi

Student: Mohamad Mahdi Samadi

Student ID: 810101465

الف) روش نیتون برای بهینه سازی را با ذکر روابط ریاضی بیان کنید.
ب)مشکلات روش نیوتن را بیان کنید. تحت چه شرایطی این روش خوب کار نمی کند.
پ) روش های نیوتن تصحیح شده رو بیان کنید. این روش ها برای حل چه مشکلی از روش نیتون آمده اند.
ت) مساله بهینه سازی روش های شبه نیوتن(DFB , BFGS) رو بیان کنید و بیان کنید در این مساله بهینه سازی هر بخش از این مساله چه مفهومی دارد.

الف: روش نیوتون یک روش برای پیدا کردن نقاط stationary با استفاده از تقریب تیلور درجه 2 است. ابتدا رابطه تیلور را تا جمله دوم حول نقطه θ_1 می‌نویسیم:

$$f(\theta) = f(\theta_1) + (\theta - \theta_1)^T \nabla\{f(\theta)\}_{\theta=\theta_1} + \frac{1}{2}(\theta - \theta_1)^T H\{f(\theta)\}_{\theta=\theta_1}(\theta - \theta_1)$$

حال فرض کنید مقدار بهینه مسئله با تقریب درجه دو در نقطه θ رخ می‌دهد. پس گرادیان در این نقطه را صفر می‌گذاریم. به علت فرضی که کردیم، این روش برای توابع درجه دو در یک گام به مقدار بهینه می‌رسد.

$$\nabla f(\theta) = 0 \rightarrow 0 + \nabla\{f(\theta)\}_{\theta=\theta_1} + H\{f(\theta)\}_{\theta=\theta_1}(\theta - \theta_1) = 0$$

حال به ساده‌سازی نتیجه بالا می‌پردازیم:

$$\rightarrow H\{f(\theta)\}_{\theta=\theta_1}(\theta - \theta_1) = -\nabla\{f(\theta)\}_{\theta=\theta_1} \rightarrow \theta - \theta_1 = -H^{-1}\{f(\theta)\}_{\theta=\theta_1}\nabla\{f(\theta)\}_{\theta=\theta_1}$$

در نهایت داریم:

$$\rightarrow \theta = \theta_1 - H^{-1}\{f(\theta)\}_{\theta=\theta_1}\nabla\{f(\theta)\}_{\theta=\theta_1}$$

رابطه iterative بالا تا جایی ادامه پیدا می‌کند که یا به هدف برسیم یا پیشرفت محسوسی رخ ندهد. این توضیحات از تدریس دکتر اعرابی برداشته شده‌اند.

ب:

- همانطور که بیان کردیم، با فرض اینکه تقریب دوجمله‌ای، تقریب مناسب و دقیقی است به بهینه‌سازی پرداختیم. پس اگر این چنین نباشد این روش چندان مناسب نمی‌باشد. البته در نقاط نزدیک به نقطه بهینه می‌توان تصور کرد که تقریب دوجمله‌ای مناسب است و این روش به خوبی در آن‌جا عمل می‌کند.

- نیاز به محاسبه ماتریس هسین و معکوس کردن آن و در نتیجه پیچیدگی زمانی $O(n^3)$ دارد. در مسائل با فضای بزرگ بسیار هزینه‌بر است.
- اصلا ممکن است ماتریس هسین وارون‌پذیر نباشد.
- نقطه اولیه خوبی انتخاب نکرده باشیم.
- اگر ماتریس هسین positive definite نباشد، ممکن است الگوریتم همگرا نشود یا به saddle point همگرا شود.

پ:

1. ماتریس هسین را تخمین می‌زنیم. در کلاس روش BFGS صحبت شد که به طور کامل در بخش ت سوال توضیح می‌دهیم.
2. در این روش سعی در positive definite کردن ماتریس هسین می‌کنیم. در واقع ماتریس جدیدی تعریف می‌کنیم که به اجزای قطر اصلی ماتریس هسین مقدار ثابتی اضافه کرده تا آن PD شود. پس کوچک‌ترین λ را انتخاب می‌کنیم که ماتریس H_2 جدید PD شود. بدین صورت مشکل همگرا نشدن الگوریتم رفع می‌شود.

$$H_2\{f(\theta)\} = H\{f(\theta)\} + \lambda I$$

3. اضافه کردن طول گام: به فرمول روش نیوتون یک طول گام اضافه می‌کنیم. تا الگوریتم سریع‌تر همگرا شود.

$$\theta = \theta_1 - \alpha \times H^{-1}\{f(\theta)\}_{\theta=\theta_1} \nabla\{f(\theta)\}_{\theta=\theta_1}$$

ت:

BFGS:

از رابطه زیر استفاده می‌کند:

$$\theta_{i+1} = \theta_i - \alpha_i S_i g_i$$

که g_i همان گرادیان، α_i همان طول گام و S_i تقریب ما از وارون ماتریس هسین است.

برای α_i الگوریتم line search استفاده می‌کنیم.

تقریب وارون ماتریس هسین از رابطه زیر آپدیت می‌شود که معمولا با $S_1 = I$ شروع می‌کنیم:

$$S_{i+1} = \left(I - \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i}\right) S_i \left(I - \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i}\right)^T + \frac{\Delta\theta_i \Delta\theta_i^T}{\Delta\theta_i^T \Delta g_i}$$

$$S_{i+1} = S_i + \frac{\Delta\theta_i \Delta\theta_i^T}{\Delta\theta_i^T \Delta g_i} - S_i \frac{\Delta g_i \Delta\theta_i^T}{\Delta\theta_i^T \Delta g_i} - \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i} S_i + \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i} S_i \frac{\Delta g_i \Delta\theta_i^T}{\Delta\theta_i^T \Delta g_i}$$

که اپراتور Δ را بدین صورت تعریف می‌کنیم:

$$\Delta\theta_i = \theta_{i+1} - \theta_i \ \ and \ \ \Delta g_i = g_{i+1} - g_i$$

که S_i تخمین کنونی، S_{i+1} تخمین جدید، $\Delta\theta_i$ اندازه گام و Δg_i تغییرات گرادیان را نشان می‌دهند.
فرمول بالا تضمین می‌کند که اگر از یک تخمین PD و متقارن شروع کنیم، در هر گام تخمین PD و متقارن می‌ماند. ما می‌خواهیم وارون ماتریس هسین نیز چنین خاصیتی داشته باشد.
جمله $(I - \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i}) S_i (I - \frac{\Delta\theta_i \Delta g_i^T}{\Delta\theta_i^T \Delta g_i})^T$ از فرمول Sherman-Morrison آمده که برای آپدیت وارون ماتریس به کار می‌رود. جمله $\frac{\Delta\theta_i \Delta\theta_i^T}{\Delta\theta_i^T \Delta g_i}$ تخمین را بهبود می‌بخشد.

DFP:

از رابطه زیر استفاده می‌کند:

$$\theta_{i+1} = \theta_i - \alpha_i S_i g_i$$

که g_i همان گرادیان، α_i همان طول گام و S_i تقریب ما از وارون ماتریس هسین است.
برای α_i الگوریتم line search استفاده می‌کنیم.
تقریب وارون ماتریس هسین از رابطه زیر آپدیت می‌شود که معمولا با $S_1 = I$ شروع می‌کنیم:

$$S_{i+1} = (I - \frac{\Delta g_i \Delta\theta_i^T}{\Delta g_i^T \Delta\theta_i}) S_i (I - \frac{\Delta g_i \Delta\theta_i^T}{\Delta g_i^T \Delta\theta_i})^T + \frac{\Delta g_i \Delta g_i^T}{\Delta g_i^T \Delta\theta_i}$$

$$S_{i+1} = S_i + \frac{\Delta g_i \Delta g_i^T}{\Delta g_i^T \Delta\theta_i} - S_i \frac{\Delta\theta_i \Delta g_i^T}{\Delta g_i^T \Delta\theta_i} - \frac{\Delta g_i \Delta\theta_i^T}{\Delta g_i^T \Delta\theta_i} S_i + \frac{\Delta g_i \Delta\theta_i^T}{\Delta g_i^T \Delta\theta_i} S_i \frac{\Delta\theta_i \Delta g_i^T}{\Delta g_i^T \Delta\theta_i}$$

که اپراتور Δ را بدین صورت تعریف می‌کنیم:

$$\Delta\theta_i = \theta_{i+1} - \theta_i \ and \ \ \Delta g_i = g_{i+1} - g_i$$

که S_i تخمین کنونی، S_{i+1} تخمین جدید، $\Delta\theta_i$ اندازه گام و Δg_i تغییرات گرادیان را نشان می‌دهند.
فرمول بالا تضمین می‌کند که اگر از یک تخمین PD و متقارن شروع کنیم، در هر گام تخمین PD و متقارن می‌ماند. ما می‌خواهیم وارون ماتریس هسین نیز چنین خاصیتی داشته باشد.

هر دو روش از اطلاعات گرادیان و خود نقطه قبلی و جدید استفاده می‌کنند، اما هر کدام به شیوه‌ای از آن بهره می‌برند. به طور کلی BFGS روش robust تر و در محاسبات پایدارتر است.
توضیحات BFGS و DFP از سایت ویکی‌پدیا گرفته شده‌اند.