بسط لاپلاس دترمینان

بسط لاپلاس (یا بسط همسازه‌ای) برای محاسبه‌ی دترمینان ماتریس مرتبه‌ی n*n ، به فرم زیر است:

$$|A| = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} |A_{ij}| \qquad , \qquad 1 \leq i \leq n$$

انتخاب سطر یا ستون مناسب برای محاسبه‌ی دترمینان با استفاده از این روش وابسته به مقادیر درایه‌های آن است .به عنوان مثال اگر تعداد درایه‌های صفر یک سطر یا یک ستون زیاد باشد، بهتر است از آن سطر یا ستون برای بسط استفاده کنیم .مثلا در ماتریس زیر، بهتر است از ستون اول برای بسط استفاده کنیم.

پیچیدگی زمانی بسط لاپلاس

همانطور که از تعریف مشخص است، در روش بسط لاپلاس، محاسبه‌ی دترمینان یک ماتریس مرتبه‌ی n، به محاسبه‌ی دترمینان n ماتریس کهاد از مرتبه‌ی $n-1$ شکسته می‌شود. اگر عمل اصلی محاسبات را اعمال ضرب و جمع در نظر گرفته و T1(n) تعداد این اعمال را برای محاسبه‌ی دترمینان ماتریس مرتبه‌ی n به روش بسط لاپلاس نشان دهد، می‌توان نوشت:

$$T_1(n) = nT_1(n-1) + n + n + n - 1 = nT_1(n-1) + 3n - 1 \qquad , \qquad T_1(1) = 0$$

$nT_1(n-1)$: تعداد اعمال لازم برای محاسبه‌ی زیرمسائل

n: تعداد ضرب‌های بین a_{ij} و توان‌های زوج یا فرد منفی یک

n: تعداد ضرب‌های بین a_{ij} و $det(A_{ij})$

$n-1$: تعداد جمع‌های لازم برای محاسبه‌ی نهایی

این رابطه‌ی بازگشتی نشان می‌دهد $T_1(n)$ از مرتبه‌ی $O(n!)$ است که برای nهای بزرگ کارایی ندارد.

روش گاوس برای محاسبه‌ی دترمینان ماتریس

برای محاسبه‌ی دترمینان یک ماتریس مربعی، خواصی وجود دارد که به اعمال مقدماتی سطری و ستونی مشهور بوده و عموما از روش بسط لاپلاس ثابت می‌شوند. تعدادی از این خواص به شرح زیر هستند:

1- جابجا کردن دو سطر (یا دو ستون) ماتریس، مقدار دترمینان را قرینه می‌کند.

2- اگر تمام درایه‌های یک سطر (یا یک ستون) ماتریس در عددی مانند k ضرب شود، حاصل دترمینان نیز k برابر می‌شود.:

3- اگر ضریب ثابتی از درایه‌های یک سطر (یا یک ستون) ماتریس به سطر (یا ستون) دیگری اضافه شود، مقدار دترمینان تغییر نمی‌کند.

4- دترمینان یک ماتریس مثلثی (ماتریسی که تمامی درایه‌های بالای قطر اصلی یا پایین قطر اصلی و یا هر دو صفر باشند) برابر حاصلضرب درایه‌های قطر اصلی آن است.

5- ماتریسی که تمامی درایه‌های یک سطر (یا یک ستون) آن صفر باشد، دترمینان آن نیز صفر خواهد بود:

در روش گاوس مراحل زیر انجام می‌شود:

مرحله‌ی اول :اگر درایه‌ی سطر اول و ستون اول صفر است، سطری را که مقدار درایه‌ی ستون اول آن صفر نباشد به سطر اول منتقل می‌کنیم .این عمل مقدار دترمینان را تغییر علامت می‌دهد .اگر چنین سطری یافت نشد، یعنی تمامی درایه‌های ستون اول صفر هستند .پس بنا به خاصیت شماره‌ی پنج، مقدار دترمینان صفر شده و انجام مراحل بعدی نیاز نیست.

مرحله‌ی دوم :ضریب مناسبی از مقدار درایه‌ی سطر اول و ستون اول را که درایه‌ی ستون اول هر سطر را صفر کند، به هر سطر به صورت مجزا اضافه می‌کنیم .اگر مقدار درایه‌ی ستون اول آن ستون، از قبل صفر باشد، نیاز به انجام عمل خاصی نیست .این عمل مقدار دترمینان را تغییر نمی‌دهد.
در مثال زیر، ستون اول سطر دوم مقدار صفر دارد. پس نیاز به انجام عملیات خاصی نیست. اما مقدار درایه‌ی ستون اول سطر سوم غیرصفر است. پس با اضافه کردن ضریب مناسبی از درایه‌های سطر اول به این سطر، مقدار آن را نیز صفر می‌کنیم. مقدار این ضریب با توجه به مقدار درایه‌ی سطر اول و ستون اول مشخص می‌شود

مرحله‌ی سوم :در ماتریس به دست آمده، ستون اول آن، به غیر از سطر اول همه صفر هستند .بسط لاپلاس دترمینان ماتریس را بر اساس ستون اول انجام می‌دهیم:

مرحله‌ی چهارم :محاسبه‌ی دترمینان ماتریس از مرتبه‌ی n به محاسبه‌ی دترمینان ماتریس مرتبه‌ی $n - 1$ تقلیل یافته است .با ادامه‌ی این مراحل برای این ماتریس، تا رسیدن به ماتریسی از مرتبه‌ی یک، مقدار دترمینان اصلی محاسبه می‌شود:

پیچیدگی زمانی روش گاوس

اعمال ضرب و جمع را اعمال اصلی این روش در نظر گرفته و T2(n) را تعداد این اعمال برای محاسبه‌ی دترمینان به روش گاوس تعریف می‌کنیم. برای صفر کردن ستون اول هر سطر، ضریب مشخصی را محاسبه می کنیم که به یک عمل تقسیم (هم‌ارز ضرب) نیاز دارد. سپس حاصلضرب این ضریب در درایه‌های سطر اول را به درایه‌های متناظر آن سطر اضافه می‌کنیم. در نتیجه این مرحله برای هر سطر n عمل ضرب و n عمل جمع دارد که برای n−1 سطر باید اعمال شود. در پایان نیز با بسط لاپلاس روی ستون اول و یک عمل ضرب، به محاسبه‌ی دترمینان مرتبه n−1 می‌رسیم. پس می‌توان نوشت:

$$T_2(n) = 1 + (n-1)(n+n) + T_2(n-1) + 1 = T_2(n-1) + 2n^2 - 2n + 2 \quad , \quad T_2(1) = 0$$

چنین رابطه‌ی بازگشتی‌ای از مرتبه‌ی O(n3) است که بهبود چشمگیری در مقایسه با روش بسط لاپلاس با مرتبه‌ی O(n!) دارد.

پیچیدگی زمانی متد جدید(رضایی فر–رضایی):

متد جدید از تمام متد های قبلی بهینه تر بوده به طوری که اوردر زمانی آن خطی است و در ماتریس هایی با ابعاد بزرگتر بسیار بهتر عمل میکند . o(n)

روزبه غزوی 98522274