

تکلیف کامپیوتری: نوشتن برنامه روش BP

فرض کنید $As=b$ یک دستگاه معادلات فرومعین باشد که در آن A یک ماتریس $n \times m$ است (و $m > n$)، b بردار معلوم و s بردار مجهول است. چون تعداد معادلات (n) از تعداد مجهولات (m) کمتر است، دستگاه در حالت کلی بینهایت جواب دارد، ولی ما به دنبال جوابی هستیم که حداکثر تعداد صفر را داشته باشد (تنک‌ترین جواب). به عبارتی به دنبال حل مسأله زیر هستیم:

$$(P_0): \quad \text{Minimize } \|s\|_0 \quad \text{subject to } As = b$$

که در رابطه بالا $\|s\|_0$ (نرم صفر بردار s) به معنی تعداد عناصر غیر صفر بردار s است.

ایده روش Basis Pursuit (BP) آن است که به جای مسأله P_0 بالا، مسأله P_1 زیر را حل کند (یعنی نرم صفر را با نرم یک جایگزین کند):

$$(P_1): \quad \text{Minimize } \|s\|_1 \quad \text{subject to } As = b$$

که در آن $\|s\|_1 = \sum_{i=1}^m |s_i|$ «نرم یک» بردار s است و در درس دیدیم تحت شرایطی، جواب مسأله P_1 با جواب مسأله P_0 یکسان می‌شود.

در این تکلیف شما باید برنامه روش BP را بنویسید. یعنی ابتدا طبق روش گفته‌شده در درس مسأله P_1 را به یک مسأله Linear Programming (LP) تبدیل کنید. لازم نیست برنامه LP را هم خودتان بنویسید، فانکشن MATLAB آن وجود دارد به نام linprog که هم روش Simplex و هم Interior point را شامل می‌شود (بسته به پارامترهایی که به آن می‌فرستید). شما تنها یاد بگیرید که چگونه از تابع linprog در MATLAB استفاده کنید (و چگونه آن را مجبور کنید که از Simplex یا Interior Point استفاده کند).

در نهایت تابعی که شما نوشته‌اید، تابعی به فرم زیر خواهد بود:

$$s_estim = BP(A,b)$$

یعنی این تابع ماتریس A و بردار b را می‌گیرد و با استفاده از روش BP تخمینی از تنک‌ترین جواب دستگاه $As=b$ به دست می‌دهد. اگر خواستید، می‌توانید آرگومان‌های دیگری نیز به تابع اضافه کنید، تا بتوانید چیزهای دیگری را نیز کنترل کنید (مثلاً اینکه به طور داخلی از Simplex یا Interior Point استفاده کند).

تست تابع: برای آزمایش تابع فوق در محیط MATLAB، یک بردار اسپارس تصادفی s تولید کنید (چگونه؟). سپس یک ماتریس تصادفی A نیز تولید کنید و بعد b را از رابطه $b=As$ محاسبه کنید. بعد A و b را به تابع فوق بدهید تا تخمینی از s بدهد (s_estim). سپس مقایسه کنید که چقدر s و s_estim به هم شبیه هستند. برای مقایسه تقریبی می‌توانید هر دو را با دستور stem روی یک گراف رسم کنید، و برای مقایسه دقیق‌تر می‌توانید از نسبت سیگنال به نویز، که با رابطه زیر تعریف می‌شود استفاده کنید:

$$SNR = 10 \log_{10} \frac{\sum_i |s_i|^2}{\sum_i |\hat{s}_i - s_i|^2}$$

همچنین می‌توانید اثر نویز را نیز در شبیه‌سازی‌های خودتان بررسی کنید. دو نوع نویز داریم: یکی نویزی که به S اضافه می‌شود (باعث می‌شود که عناصر «صفر» بردار S دقیقاً صفر نباشند، و در نتیجه نظریه هم‌ارزی نرم صفر و یک به هم بریزد (چطور؟)) و دیگری نویزی که به \mathbf{b} اضافه می‌شود ($\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{v}$ که در آن \mathbf{v} بردار نویز است) که باعث می‌شود که تساوی $\mathbf{b} = \mathbf{A}\mathbf{s}$ به صورت دقیق برقرار نباشد. می‌توانید با شبیه‌سازی ببینید BP چقدر نسبت به این دو نوع نویز مقاومت دارد یا ندارد.