การเปรียบเทียบขั้นตอนวิธีแบบเร็ว สำหรับปัญหาการประมาณที่ใช้นอร์ม-1

A Comparison of Fast Algorithms for ℓ_1-Type Penalized Estimation Problems

นายพุฒิไชย เลิศกุลทานนท์ อาจารย์ที่ปรึกษา อ.ดร.จิตโกมุท ส่งศิริ

กลุ่มวิจัยการควบคุมและการหาค่าเหมาะสุดขั้นสูง

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

บทคัดย่อ

ในโครงงานนี้ เราศึกษาและเปรียบเทียบขั้นตอนวิธีสำหรับแก้ปัญหาการหาค่าเหมาะสุดที่สำคัญสองปัญหา ในการประมาณแบบจำลอง ปัญหาแรกจัดอยู่ในกลุ่มปัญหากำลังสองต่ำสุด (least-squares problem) ที่มีเงื่อนไขบังคับเป็นสมการเชิงเส้น เราจะแสดงให้เห็นว่าผลเฉลยของปัญหาข้อแรกนี้ เป็นผลเฉลยในรูปแบบปิด (closed-form solution) ซึ่งได้จากการแก้ระบบสมการเชิงเส้น ส่วนปัญหาที่สอง มีฟังก์ชันจุดประสงค์ประกอบด้วยสองส่วนคือ ฟังก์ชันสูญเสียแบบกำลังสอง และ ฟังก์ชันลงโทษที่อยู่ในรูปของนอร์ม-1 เนื่องจากฟังก์ชันลงโทษในรูปนอร์ม-1 ไม่สามารถหาอนุพันธ์ได้ เราจึงไม่สามารถใช้เทคนิคพื้นฐาน เช่น วิธีเกรเดียนต์ หรือ วิธีนิวตันในการแก้ปัญหาที่สอง นอกจากนั้น ถึงแม้ว่าปัญหาที่สองนี้ จะเป็นปัญหาคอนเวกซ์ แต่เมื่อปัญหามีขนาดใหญ่ นั่นคือมีตัวแปรในหลักพัน หรือ หมื่นตัว การแก้ปัญหาจะต้องคำนึงถึงข้อจำกัดในด้านทรัพยากรที่ใช้ในการคำนวณด้วย เช่น ข้อจำกัดด้านพื้นที่หน่วยความจำ เป็นต้น ดังนั้นในโครงงานนี้ จึงศึกษาขั้นตอนวิธีในอันดับหนึ่ง (first-order methods) ซึ่งอาศัยเกรเดียนต์ของฟังก์ชันจุดประสงค์ส่วนที่หาอนุพันธ์ได้ในฟังก์ชันจุดประสงค์เป็นหลัก รวมทั้งเปรียบเทียบสมบัติต่างๆ ของขั้นตอนวิธีเหล่านั้น เช่น สมบัติการลู่เข้าในทางปฏิบัติ หรือ เวลาโดยเฉลี่ยที่ใช้ในการคำนวณ เป็นต้น หลังจากนั้นจะนำขั้นตอนวิธีที่ถูกทดสอบแล้ว ว่ามีประสิทธิภาพสูงสุดตามเกณฑ์ที่กำหนดขึ้น ไปประยุกต์ใช้เพื่อแก้ปัญหาการหาแบบจำลองพลวัต และ การเรียนรู้โครงสร้างเชิงสาเหตุในข้อมูลการ ทำงานของสมองต่อไป

คำสำคัญ : การหาค่าเหมาะสุดแบบคอนเวกซ์, ฟังก์ชันลงโทษแบบนอร์ม-1, การถดถอยตัวเองแบบลาซโซกลุ่ม, ขั้นตอนอันดับหนึ่ง

abstract

In this project, we study and compare algorithms for solving two optimization problems in model estimation. The first problem is a least-squares problem with linear equality constraints. We show that this problem has a closed-form solution that can be obtained by solving linear equations. The second problem has an objective function that can be split into two parts; a quadratic loss term and an ℓ_1-norm regularization term. Since the ℓ_1-norm regularization term is nondifferentiable, some well-known classical methods such as gradient and Newton methods are not applicable to this problem. Moreover, the computational cost is an important factor to be considered especially when the problem dimension is high, i.e., having ten thousands variables or more. Thus, in this project, we study first-order methods based on the gradient of the differentiable term in the cost objective and compare their properties. Consequently, we apply the best algorithm according to our criterion to solve the problem of learning causal structure of brain connectivity in fMRI time series.

Keywords : convex optimization, ℓ_1-norm penalty function, group lasso regression, first-order methods

รายงานฉบับสมบูรณ์ วิชาโครงงานวิศวกรรมไฟฟ้า ปีการศึกษา 2555
การเปรียบเทียบขั้นตอนวิธีแบบเร็ว สำหรับปัญหาการประมาณที่ใช้นอร์ม-1
A Comparison of Fast Algorithms for ℓ_1-Type Penalized Estimation Problems

นายพุฒิไชย เลิศกุลทานนท์ 5230367921 กลุ่มวิจัยการควบคุมและการหาค่าเหมาะสุดขั้นสูง
อาจารย์ที่ปรึกษา อ.ดร.จิตโกมุท ส่งศิริ ห้องปฏิบัติการวิจัยระบบควบคุม
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

1 บทนำ

ในโครงงานนี้ เราสนใจปัญหาการประมาณแบบจำลองระบบของข้อมูลทางเวลา ซึ่งเมื่อจัดรูปเป็นปัญหาการหาค่าเหมาะสุดแล้วจะอยู่ในรูป

$$\text{minimize} \qquad f(x) + \lambda \|x\|_1 \tag{1}$$

โดยที่ $x \in \mathbf{R}^n$ เป็นตัวแปรสำหรับปัญหาการหาค่าเหมาะสุด แสดงถึงค่าพารามิเตอร์ของแบบจำลองที่ต้องการประมาณ, $f(x)$ เป็นฟังก์ชันการสูญเสีย เช่น ฟังก์ชันการสูญเสียแบบกำลังสอง (quadratic loss function) หรือ ฟังก์ชันลอการิทึมของความเป็นไปได้ (negative of log-likelihood function), $\|x\|_1$ เรียกว่าเป็นฟังก์ชันลงโทษแบบนอร์ม-1 ทำหน้าที่ควบคุมขนาดของตัวแปร x, และ จำนวนจริง $\lambda > 0$ จะเรียกว่าเป็นพารามิเตอร์ลงโทษ (penalty parameter)

การหาค่าต่ำสุดของ $f(x)$ เพียงอย่างเดียว จะเป็นการหาค่าพารามิเตอร์ของแบบจำลอง ที่สามารถอธิบายข้อมูลที่มีได้ดีที่สุด (อย่างเช่น พารามิเตอร์ที่ทำให้ผลต่างของข้อมูลจริง กับสัญญาณขาออก จากแบบจำลอง มีขนาดน้อยที่สุด เป็นต้น) ส่วนการเพิ่มฟังก์ชันลงโทษรูปแบบใดๆก็ตาม ลงในฟังก์ชันจุดประสงค์ จะเป็นการปรับปรุงคุณสมบัติของผลเฉลยปัญหา (1) ให้มีรูปแบบตามต้องการ ขึ้นกับรูปแบบของฟังก์ชันลงโทษที่เพิ่มลงไป แต่ทั้งนี้ ก็จะต้องแลกกับการยอมให้พารามิเตอร์ที่หาได้ อธิบายข้อมูลจริงได้แย่ลง โดยทั้งหมดนี้ จะมีพารามิเตอร์ λ เป็นตัวกำกับว่า ในการหาผลเฉลยของปัญหาการหาค่าเหมาะสุด เราจะให้ความสำคัญกับ ความสามารถในการอธิบายข้อมูลจริง ของพารามิเตอร์ที่หาได้ หรือ รูปแบบของผลเฉลยที่ตรงตามเงื่อนไขที่ต้องการมากกว่ากัน นั่นคือ หากพารามิเตอร์ λ มีขนาดเล็ก แสดงว่าในปัญหานั้น เราให้ความสำคัญกับความสามารถในการอธิบายข้อมูลจริง ของพารามิเตอร์มากกว่า

เป็นที่ทราบกันดีว่าฟังก์ชันลงโทษแบบนอร์ม-1 ถูกเลือกมาใช้ เมื่อเราต้องการให้พารามิเตอร์ของแบบจำลองบางค่า *มีค่าเป็นศูนย์* [1] เราเรียกลักษณะของผลเฉลย x ที่มีสมาชิกเป็นศูนย์จำนวนมากว่าเป็น *ผลเฉลยเบาบาง* (sparse solution) ผลเฉลย x ในลักษณะนี้ นอกจากจะทำให้เราใช้พารามิเตอร์ในการอธิบายระบบเป็นจำนวนน้อยลงแล้ว ยังอาจเป็นประโยชน์ต่อการนำระบบที่ประมาณได้ ไปวิเคราะห์ตีความต่อไปอีกด้วย สำหรับปัญหา (1) ที่มี f เป็นฟังก์ชันสูญเสียกำลังสอง ปัญหาดังกล่าวเป็นที่รู้จักกันในนาม ปัญหาลาซโซ่ [2]

ที่ผ่านมา มีงานวิจัยจำนวนมาก ให้ความสนใจเกี่ยวกับปัญหาการประมาณที่ใช้ฟังก์ชันลงโทษแบบนอร์ม-1 เนื่องจากปัญหาในรูปแบบดังกล่าว มีที่ประยุกต์ใช้งานในหลายด้าน ทั้งในปัญหาการจำแนกหมวดหมู่ (classification problems) เช่น การวิเคราะห์ผู้ป่วยโรคมะเร็งจากข้อมูลโปรตีนในเลือด [3, บทที่ 18], การจำแนกผู้ป่วยเป็นเนื้องอกจากการวิเคราะห์ลำดับยีน เป็นต้น หรือในปัญหาการหาแบบจำลองของระบบ เช่น ปัญหาการหาแบบจำลองข้อมูลคลื่นไฟฟ้าหัวใจ [4], ปัญหาการแก้ไขภาพเบลอ [5] เป็นต้น

ในโครงงานฉบับนี้ จะศึกษาการแก้ปัญหาการประมาณแบบจำลองโครงสร้างเชิงสาเหตุในข้อมูลการทำงานของสมอง [6] โดยข้อมูลการทำงานของสมองในที่นี้คือ ข้อมูลระดับออกซิเจนในเลือดที่วัดได้จากสมองบริเวณต่างๆ ในแต่ละเวลา กระบวนการ functional Magnetic Resonance Imaging (fMRI) เราเลือกใช้แบบจำลองที่ใช้บรรยายโครงสร้างการทำงานของสมอง เป็นแบบจำลองถดถอยตัวเองแบบเวกเตอร์ (Multivariate Autoregressive Model หรือ MAR) ซึ่งสามารถเป็นเขียนสมการได้คือ

$$y(t) = A_1 y(t-1) + A_2 y(t-2) + \cdots + A_p y(t-p) + \eta(t) \tag{2}$$

โดยที่ $y(\cdot) \in \mathbf{R}^n$ คือ เวกเตอร์ข้อมูลระดับออกซิเจนในเลือดที่วัดได้จากสมองบริเวณต่างๆ นั่นคือ y_i จะเป็นข้อมูลที่วัดได้จาก

โนดที่ i ในสมอง ส่วน $A_k \in \mathsf{R}^{n\times n}$, $k = 1,2,\ldots,p$ คือ พารามิเตอร์ของระบบ และ $\eta(t)$ คือ สัญญาณรบกวน

นอกจากนั้น เรายังใช้เงื่อนไขความสัมพันธ์เชิงเหตุและผลแบบเกรนเจอร์ (Granger causality) ในการอธิบายความสัมพันธ์ระหว่างตัวแปรคู่ใดๆในระบบด้วย ซึ่งสามารถกล่าวโดยสรุปคือ หากข้อมูล y_j ไม่ใช่สาเหตุแบบเกรนเจอร์ต่อการเกิด y_i แล้ว

$$[A_k]_{ij} = 0 \qquad (3)$$

ทุกค่า $k = 1,2,\ldots,p$ [7] โดยที่ $[A_k]_{ij}$ หมายถึงสมาชิกตัวที่ (i,j) ของเมทริกซ์ A_k เราเรียก รูปแบบตำแหน่งของสมาชิกที่เป็นศูนย์ ในเมทริกซ์ A_k ใดๆ ว่าเป็น *รูปแบบศูนย์* (sparsity pattern) ของ A_k และ เราจะเรียก A_k สำหรับ $k = 1,2,\ldots,p$ ที่มีสมบัติตาม (3) ว่ามี *รูปแบบศูนย์ร่วม* ดังนั้นหากสมองแต่ละส่วน ไม่ได้มีความสัมพันธ์กับสมองทุกๆส่วน เราสามารถคาดการณ์ได้ว่าเมทริกซ์ A_k แต่ละตัว จะมีรูปแบบศูนย์ร่วมกัน เราจึงใช้สมบัติข้อนี้เป็นแนวทางเพื่อเขียนปัญหาการประมาณแบบจำลองระบบ ให้เป็นปัญหาการหาค่าเหมาะสุดแบบคอนเวกซ์ในรูปแบบ

$$\underset{A}{\text{minimize}} \qquad f(A) + \lambda g(A) \qquad (4)$$

โดย $A = [A_1 \;\; A_2 \;\; \cdots \;\; A_p]$ และ $g(A)$ เป็นฟังก์ชันลงโทษซึ่งจะทำให้ผลเฉลย $A_1,\ A_2,\ \ldots,\ A_p$ ที่ได้มีรูปแบบศูนย์ร่วม

หากปัญหาในรูปแบบ (4) เป็นปัญหาคอนเวกซ์ นั่นคือเป็นปัญหาที่มีฟังก์ชันจุดประสงค์เป็นฟังก์ชันคอนเวกซ์แล้ว วิธีจุดภายใน (interior-point method) เป็นวิธีหนึ่งที่ถูกนำมาใช้กันใช้กันอย่างแพร่หลาย มีงานวิจัยตีพิมพ์มากมายเกี่ยวกับวิธีจุดภายใน [1, บทที่ 11] นอกจากนั้นยังมีโปรแกรมสำเร็จรูปสำหรับแก้ปัญหาคอนเวกซ์โดยเฉพาะอย่างเช่น CVX [8] อีกด้วย แต่ทั้งนี้ วิธีจุดภายในยังคงมีข้อจำกัดคือ มีค่าสิ้นเปลืองในการคำนวณในแต่ละรอบการวนซ้ำสูงมากถ้าหากปัญหามีตัวแปรจำนวนมาก (ระดับหมื่นตัว หรือมากกว่านั้น) เนื่องจากในแต่ละรอบการวนซ้ำ จะต้องคำนวณเมทริกซ์อนุพันธ์อันดับที่สอง (Hessian matrix) ของฟังก์ชันกีดกันแบบลอการิทึม (logarithmic barrier function) และแก้สมการเชิงเส้นซึ่งเกี่ยวข้องกับเมทริกซ์ดังกล่าว ซึ่งทำได้ค่อนข้างยากเมื่อปัญหามีขนาดใหญ่ขึ้น ดังนั้น จากข้อจำกัดทางด้านเวลา และ ตัวประมวลผล สำหรับปัญหาขนาดใหญ่ เราจึงต้องเลือกใช้ขั้นตอนวิธีซึ่งถึงแม้ว่าจะมีสมบัติการลู่เข้าที่ด้อยลงมา แต่ในขณะเดียวกันก็มีค่าสิ้นเปลืองในการคำนวณ ในแต่ละรอบวนซ้ำที่ค่อนข้างต่ำ เมื่อเปรียบเทียบกับวิธีจุดภายในดังกล่าว

ขั้นตอนวิธีที่นำมาใช้ในการแก้ปัญหาการประมาณนี้ คือ ขั้นตอนวิธี ADMM (Alternating Direction Method of Multipliers) [9] และ ขั้นตอนวิธี FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [5] ซึ่งมีสองแบบคือ แบบระยะขั้นคงที่ และ แบบระยะขั้นแปรค่า ขั้นตอนวิธีที่เลือกมาใช้ เป็นขั้นตอนวิธีที่นิยมใช้ในการแก้ปัญหาคอนเวกซ์ นอกจากนั้นยังเป็นขั้นตอนวิธีที่มีค่าสิ้นเปลืองในการคำนวณค่อนข้างต่ำ เนื่องจากอาศัยเพียงเกรเดียนต์ของฟังก์ชันในการคำนวณแต่ละรอบการวนซ้ำเท่านั้น

2 ระเบียบวิธีที่นำเสนอในโครงงาน

2.1 ปัญหาการหาแบบจำลอง MAR

สำหรับข้อมูลอนุกรมเวลา $y(1),\ y(2),\ \ldots,\ y(N)$ เราสามารถใช้แบบจำลองถดถอยตัวเองแบบเวกเตอร์ อันดับ p ซึ่งแสดงได้ด้วย (2) เพื่ออธิบายความสัมพันธ์ของข้อมูลที่เวลาต่างๆ สำหรับโครงงานนี้ เราสนใจว่า ที่อันดับ p ค่าหนึ่ง เราจะหามีวิธีการพารามิเตอร์ $A_1, A_2, \ldots, A_p$ อย่างไร เพื่อให้พารามิเตอร์เหล่านี้สามารถอธิบายข้อมูลได้ดีที่สุด วิธีที่ง่ายวิธีหนึ่งในการใช้เลือกพารามิเตอร์ คือ ใช้ฟังก์ชันสูญเสียแบบกำลังสองเป็นตัวตัดสิน กล่าวคือ เลือกพารามิเตอร์ที่ทำให้ฟังก์ชัน

$$f(A) = (1/2)\sum_{t=p+1}^{N}\left\| y(t) - \sum_{k=1}^{p} A_k y(t-k)\right\|_2^2$$

มีค่าต่ำสุด ทั้งนี้เราอาจเขียนฟังก์ชัน f ให้อยู่ในรูปแบบที่กระชับขึ้นเป็น

$$f(A) = (1/2)\|Y - AH\|_F^2, \qquad (5)$$

โดยที่

$$\begin{aligned} A &= [A_1 \;\; A_2 \;\; \cdots \;\; A_p] \in \mathsf{R}^{n\times np},\\ Y &= [y(p+1) \;\; y(p+2) \;\; \cdots \;\; y(N)] \in \mathsf{R}^{n\times(N-p)}, \end{aligned}$$

$$H = \begin{bmatrix} y(p) & y(p+1) & \cdots & y(N-1) \\ y(p-1) & y(p) & \cdots & y(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ y(1) & y(2) & \cdots & y(p) \end{bmatrix} \in \mathbf{R}^{np\times(N-p)}.$$

ปัญหากำลังสองต่ำสุดนั้น เป็นที่ทราบกันดีว่าเป็นปัญหาที่ง่าย เนื่องจากมีผลเฉลยในรูปแบบปิด หากเราพิจารณาปัญหาการหาแบบจำลอง ที่คำนึงถึงเงื่อนไขความสัมพันธ์เชิงเหตุและผลแบบเกรนเจอร์ด้วย เราสามารถแบ่งปัญหาดัง กล่าวออกเป็นสองกรณี ดังต่อไปนี้

กรณีทราบรูปแบบศูนย์ร่วมของ $A_1,\ A_2,\ \ldots,\ A_p$

สำหรับกรณีนี้ เราสามารถเขียนปัญหาการหาแบบจำลองให้อยู่ในรูปปัญหาการหาค่าเหมาะสุดที่มีเงื่อนไขบังคับเป็นสมการได้ คือ

$$\begin{array}{ll} \underset{A}{\text{minimize}} & \|Y - AH\|_F^2 \\ \text{subject to} & (A_k)_{ij} = 0,\ \ (i,j) \notin \mathcal{V},\ \ k = 1,2,\ldots,p \end{array} \tag{6}$$

โดยที่ $\mathcal{V}$ คือเซตของตำแหน่งสมาชิกที่ไม่เป็นศูนย์ของพารามิเตอร์ $A_k,\ k = 1,2,\ldots,p$

ถ้าตัดตัวแปรที่เป็นศูนย์ออกไป เราจะสามารถเขียนปัญหานี้ให้อยู่ในรูปปัญหากำลังสองต่ำสุดได้ n ปัญหา โดยจะแสดงการพิสูจน์ไว้ในภาคผนวก ดังนั้นปัญหานี้จึงมีผลเฉลยในรูปแบบปิดเหมือนกับปัญหากำลังสองต่ำสุดทั่วๆไป ซึ่งง่ายต่อการคำนวณ และ มีค่าสิ้นเปลืองในการคำนวณน้อยกว่าการหาผลเฉลยของ (6) โดยใช้วิธีวนซ้ำ เป็นอย่างมาก หาก M^2 คือความหนาแน่นของสมาชิกที่ไม่เป็นศูนย์ในพารามิเตอร์ ในการแก้ปัญหากรณีนี้ จะมีค่าสิ้นเปลืองในการคำนวณอยู่ในอันดับประมาณ $\mathcal{O}(n(Mp)^3)$ ซึ่งแปรผันตามจำนวนตัวแปรในแบบจำลอง MAR (n) ด้วยอัตราเชิงเส้นเท่านั้น สำหรับการศึกษาและ เปรียบเทียบขั้นตอนวิธีที่จะนำมาใช้แก้ปัญหาการหาแบบจำลองในโครงงานนี้ จึงกล่าวถึงเพียงปัญหาแบบที่สอง ซึ่งจะได้กล่าวถึงต่อไป

กรณีไม่ทราบรูปแบบศูนย์ร่วมของ $A_1,\ A_2,\ \ldots,\ A_p$

สำหรับกรณีนี้ เราต้องการเสนอรูปแบบปัญหา (4) ที่จะทำให้ผลเฉลย $A_1,\ A_2,\ \ldots,\ A_p$ มีสมาชิกเป็นศูนย์จำนวนมาก รวมทั้งมี*รูปแบบศูนย์ร่วม*ด้วย เราจึงเสนอให้เพิ่มพจน์ฟังก์ชันลงโทษ ซึ่งเป็นผลรวมของนอร์ม-2 ลงในฟังก์ชันจุดประสงค์ปัญหานี้จึงเขียนได้เป็น

$$\begin{array}{ll} \underset{A}{\text{minimize}} & (1/2)\|Y - AH\|_F^2 + \\ & \lambda \sum_{i\neq j} \left\| \left[(A_1)_{ij}\ \ (A_2)_{ij}\ \ \cdots\ \ (A_p)_{ij} \right] \right\|_2 \end{array} \tag{7}$$

โดย λ เป็นพารามิเตอร์ลงโทษ ปัญหาในรูปแบบนี้ถูกจัดเป็นปัญหาแบบลาซโซกลุ่ม (group lasso problem) ซึ่งถูกเสนอโดย [10] ผลบวกของนอร์ม-2 ทำหน้าที่เป็นเหมือนกับฟังก์ชันลงโทษแบบนอร์ม-1 ที่สามารถทำให้พจน์ของนอร์ม-2 สำหรับบางค่า (i,j) มีค่าเป็นศูนย์ และ หากพจน์นอร์ม-2 ที่ (i,j) ใดๆ มีค่าเป็นศูนย์ จะได้ว่า $(A_1)_{ij} = (A_2)_{ij} = \ldots = (A_p)_{ij} = 0$ การเพิ่มฟังก์ชันลงโทษแบบดังกล่าวลงในฟังก์ชันจุดประสงค์ จึงเป็นการทำให้พารามิเตอร์ของแบบจำลองมีตำแหน่งศูนย์ร่วมกันตามต้องการ

จากปัญหา (7) จะเห็นว่าพจน์ฟังก์ชันลงโทษ จะไม่รวมนอร์ม-2 กรณี (i,j) ซึ่ง $i \neq j$ เนื่องจาก $(A_k)_{ii}$ สำหรับ $k = 1,2,\ldots,p$ เป็นพารามิเตอร์ที่แสดงถึงความสัมพันธ์ของสมองโนดที่ i ที่มีต่อตัวเอง เราจึงมีสมมติฐานว่าพารามิเตอร์ดังกล่าวอาจจะไม่เป็นศูนย์

ในหัวข้อต่อไป จะกล่าวถึงรายละเอียดของขั้นตอนวิธีนำมาศึกษาในโครงงานนี้ รวมถึงการประยุกต์ใช้ขั้นตอนวิธีเหล่านั้นกับปัญหา (7) ด้วย

2.2 ขั้นตอนวิธี ADMM

เราจะพิจารณาขั้นตอนวิธี ADMM เพื่อแก้ปัญหา (7) ขั้นตอนวิธี ADMM ถูกเสนอขึ้นเพื่อแก้ปัญหาการหาค่าเหมาะสุดแบบคอนเวกซ์ที่อยู่ในรูป

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{8}$$

โดย $x \in \mathbf{R}^n, z \in \mathbf{R}^m$ เป็นตัวแปรในการหาค่าเหมาะสุด $A \in \mathbf{R}^{p\times n}, B \in \mathbf{R}^{p\times m}$ และ $c \in \mathbf{R}^p$

สำหรับการใช้ขั้นตอนวิธี ADMM เราจะต้องสร้างลากรานเจียนต่อเติม L_ρ ขึ้นมาก่อน ซึ่ง L_ρ เป็นฟังก์ชันของตัวแปร x, z และ ตัวแปรคู่กัน (dual variable) $y \in \mathbf{R}^p$ นิยามโดย

$$\begin{aligned} L_\rho(x,z,y) = f(x) + g(z) + y^T(Ax + bz - c) + \\ (\rho/2)\|Ax + bz - c\|_2^2 \end{aligned} \tag{9}$$

โดยที่ $\rho > 0$ เป็นพารามิเตอร์ของวิธี ADMM ซึ่งมองได้ว่าเป็นพารามิเตอร์ลงโทษ สำหรับเศษเหลือจากสมการเงื่อนไขบังคับในปัญหา (8)

ในการใช้ขั้นตอนวิธี ADMM แก้ปัญหา (7) เราเขียนปัญหาให้อยู่ในรูปแบบของ ADMM ได้คือ

$$\begin{array}{ll}\underset{A}{\text{minimize}} & (1/2)\|Y - AH\|_F^2 + \\ & \lambda\sum_{i\neq j}\left\|\left[(Z_1)_{ij}\ (Z_2)_{ij}\ \cdots\ (Z_p)_{ij}\right]\right\|_2 \quad (10)\\ \text{subject to} & A - Z = 0\end{array}$$

โดยมีตัวแปรคือ $A = [A_1\ A_2\ \cdots\ A_p] \in \mathbf{R}^{n\times np}$, $Z = \begin{bmatrix}Z_1\ Z_2\ \cdots\ Z_p\end{bmatrix} \in \mathbf{R}^{n\times np}$ และ $Y \in \mathbf{R}^{n\times(N-p)}$, $H \in \mathbf{R}^{np\times(N-p)}$

เมื่อประยุกต์ใช้ขั้นตอนวิธี ADMM กับรูปแบบปัญหานี้ จะมีขั้นตอนการปรับตัวแปรต่างๆดังนี้

<u>กำหนดให้</u> $A^{(0)}, Z^{(0)}, U^{(0)}$ เป็นค่าเริ่มต้นของการวนซ้ำ, $\rho > 0$, $k = 0$

<u>ทำซ้ำ</u> ปรับตัวแปรเรียงลำดับดังนี้

1. $$A^{(k+1)} := \underset{A}{\text{argmin}}\ \frac{1}{2}\|Y - AH\|_F^2 + \frac{\rho}{2}\|A - Z^{(k)} + U^{(k)}\|_F^2 \quad (11)$$
2. $$Z^{(k+1)} := \underset{Z}{\text{argmin}}\ \frac{\rho}{2}\left\|Z - \left(A^{(k+1)} + U^{(k)}\right)\right\|_F^2 + \lambda\sum_{i\neq j}\left\|\left[(Z_1)_{ij}\ (Z_2)_{ij}\ \cdots\ (Z_p)_{ij}\right]\right\|_2 \quad (12)$$
3. $$U^{(k+1)} := U^{(k)} + A^{(k+1)} - Z^{(k+1)} \quad (13)$$

<u>จนกระทั่ง</u> จุดประมาณของจุดต่ำสุด หรือ ค่าประมาณของค่าต่ำสุด ของฟังก์ชันจุดประสงค์ สอดคล้องกับเกณฑ์การหยุด (stopping criterion) ที่กำหนดไว้

ต่อไปจะกล่าวถึงรายละเอียดในการคำนวณสูตรการปรับตัวแปรตาม (11) และ (12)

จาก (11) $A^{(k+1)}$ ได้จากผลเฉลยของปัญหาการหาค่าเหมาะสุดแบบไม่มีเงื่อนไขบังคับ เมื่อใช้เงื่อนไขความเหมาะที่สุด (optimality condition) นั่นคือ เงื่อนไขเกรเดียนต์เป็นศูนย์ (zero-gradient condition) ซึ่งเขียนได้เป็น

$$\nabla_A\left(\frac{1}{2}\|Y - AH\|_F^2 + \frac{\rho}{2}\|A - Z^{(k)} + U^{(k)}\|_F^2\right) = 0$$

เราจะได้ว่าขั้นตอนการปรับตัวแปร A มีรูปแบบปิดคือ

$$A^{(k+1)} = \left(\rho\big(Z^{(k)} - U^{(k)}\big) + YH^T\right)\left(HH^T + \rho I\right)^{-1} \quad (14)$$

ซึ่งเป็นการแก้สมการเชิงเส้น จาก (14) จะเห็นว่า $\left(HH^T + \rho I\right) \succ 0$ เสมอ การคำนวณจึงเริ่มจากการแยกตัวประกอบโชเลสกี้ (Cholesky factorization) ของพจน์ $\left(HH^T + \rho I\right)$ เก็บไว้ก่อนในหน่วยความจำ เนื่องจากเป็นพจน์คงที่ในทุกรอบการวนซ้ำ การแยกตัวประกอบนี้มีค่าสิ้นเปลืองในการคำนวณอยู่ที่ $\mathcal{O}((1/3)(np)^3)$ จากนั้นจึงใช้ตัวประกอบที่แยกมาได้ เพื่อคำนวณ $A^{(k+1)}$ เราจะได้ว่า สำหรับขั้นตอนการปรับค่าตัวแปร A ตาม (11) มีค่าสิ้นเปลืองในการคำนวณอยู่ในอันดับ $\mathcal{O}(((1/3)p^3 + 2p^2)n^3)$

ส่วนขั้นตอนการปรับตัวแปร Z ตาม (12) จะเห็นว่าเราสามารถคำนวณที่แต่ละดัชนี (i, j) แยกกันได้ ดังนั้นขั้นตอนนี้จึงเป็นการแก้ปัญหาการหาค่าเหมาะสุดย่อยๆ จำนวน n ปัญหา โดยเมื่อกำหนดให้

$$z_{ij} = \begin{bmatrix}(Z_1)_{ij}\\(Z_2)_{ij}\\ \vdots \\(Z_p)_{ij}\end{bmatrix} \in \mathbf{R}^p,\quad w_{ij} = \begin{bmatrix}(A_1 + U_1)_{ij}\\(A_2 + U_2)_{ij}\\ \vdots \\(A_p + U_p)_{ij}\end{bmatrix} \in \mathbf{R}^p$$

แต่ละปัญหาย่อยจะมีค่าเหมาะสุดในตัวแปร z_{ij} อยู่ในรูป

$$z_{ij} = \underset{z}{\text{argmin}}\ (1/2)\|z - w_{ij}\|_2^2 + (\lambda/\rho)\|z\|_2$$

ซึ่งมีพจน์ $\|z\|_2$ ที่ไม่สามารถหาอนุพันธ์ได้ เมื่อใช้เงื่อนไขความเหมาะที่สุดสำหรับฟังก์ชันที่หาอนุพันธ์ไม่ได้ ซึ่งใช้หลักการของแคลคูลัสเกรเดียนต์ย่อย (subgradient calculus) แล้ว เราจะได้ว่า ผลเฉลย z_{ij} มีรูปแบบปิดคือ

$$z_{ij} = \begin{cases}\dfrac{w_{ij}}{\|w_{ij}\|_2}\Big(\|w_{ij}\|_2 - (\lambda/\rho)\Big) & ,\ \|w_{ij}\|_2 \geq (\lambda/\rho)\\ 0 & ,\ \|w_{ij}\|_2 < (\lambda/\rho).\end{cases} \quad (15)$$

ซึ่ง เราเรียกการกระทำบนตัวแปร z_{ij} ตาม (15) ว่า*การขีดกั้นแบบอ่อน (soft thresholding)* ซึ่งตัวกระทำดังกล่าว ใช้ค่าสิ้นเปลืองในการคำนวณที่น้อยมาก

สำหรับขั้นตอนการปรับตัวแปร U เป็นเพียงการบวกลบเมทริกซ์ธรรมดา โดยมีค่าสิ้นเปลืองในการคำนวณอยู่ที่

$\mathcal{O}(2pn^2)$ ดังนั้น เราจะเห็นว่าในการคำนวณแต่ละรอบการวนซ้ำ ขั้นตอนการปรับตัวแปร A จะมีค่าสิ้นเปลืองมากที่สุด และทำให้การคำนวณของขั้นตอนวิธี ADMM โดยรวมมีค่าสิ้นเปลืองอยู่ในอันดับ $\mathcal{O}(((1/3)p^3 + 2p^2)n^3)$

2.3 ขั้นตอนวิธี FISTA

ขั้นตอนวิธี FISTA ใช้กับปัญหาคอนเวกซ์ที่อยู่ในรูปแบบ

$$\text{minimize} \quad f(x) + g(x) \tag{16}$$

โดยที่ f เป็นฟังก์ชันที่หาอนุพันธ์ได้ ส่วน g เป็นฟังก์ชันที่หาอนุพันธ์ไม่ได้ ซึ่งขั้นตอนวิธี FISTA จะเหมาะกับ g ที่สามารถคำนวณตัวกระทำพร็อกซิมิตี้ได้ง่าย โดยตัวกระทำพร็อกซิมิตี้สำหรับฟังก์ชันคอนเวกซ์ g นิยามโดย

$$\mathbf{prox}_g(x) = \underset{u}{\text{argmin}} \left(g(u) + \frac{1}{2}\|u - x\|_2^2 \right) \tag{17}$$

โดยทั่วไป สำหรับปัญหา (16) หากใช้วิธีเกรเดียนต์ย่อย และต้องการให้ค่าต่ำสุดที่คำนวณได้ กับค่าต่ำสุดจริงของปัญหาคลาดเคลื่อนกันไม่เกิน ϵ จะต้องใช้จำนวนรอบการวนซ้ำในอันดับ $\mathcal{O}(1/\epsilon^2)$ แต่เมื่อมีการเสนอขั้นตอนวิธี FISTA ขึ้นมาใน [5] สมบัติการลู่เข้าสำหรับปัญหา (16) ถูกปรับปรุงขึ้นจากการใช้เกรเดียนต์ย่อยเป็นอย่างมาก โดยได้มีการพิสูจน์ใน [5] ว่า หาก ∇f เป็นฟังก์ชันต่อเนื่องแบบลิพชิทซ์ (Lipschitz continuous function) ด้วยแล้ว จำนวนรอบการวนซ้ำเมื่อใช้ขั้นตอนวิธี FISTA จะอยู่ในอันดับเพียง $\mathcal{O}(1/\sqrt{\epsilon})$

ในการใช้ขั้นตอนวิธี FISTA แก้ปัญหาการหาแบบจำลองกรณีไม่ทราบรูปแบบศูนย์ร่วม เราจะกำหนดให้

$$f(A) = (1/2)\|Y - AH\|_F^2$$
$$g(A) = \lambda \sum_{i \neq j} \left\| \left[(Z_1)_{ij} \;\; (Z_2)_{ij} \;\; \cdots \;\; (Z_p)_{ij} \right] \right\|_2$$

ดังนั้น เมื่อประยุกต์ใช้ขั้นตอนวิธี FISTA กับรูปแบบปัญหานี้ จะได้ขั้นตอนการปรับตัวแปรต่างๆ ดังนี้

<u>กำหนดให้</u> $A^{(0)}$ เป็นค่าเริ่มต้นของการวนซ้ำ, $A^{(-1)} = A^{(0)}$, $k = 0$

<u>ทำซ้ำ</u> ปรับตัวแปรดังนี้

$$1. \quad V \quad := A^{(k)} + \frac{k-2}{k+1}\left(A^{(k)} - A^{(k-1)}\right) \tag{18}$$

$$2. \quad A^{(k+1)} := \mathbf{prox}_{t_k g}(V - t_k \nabla f(V)) \tag{19}$$

<u>จนกระทั่ง</u> จุดประมาณของจุดต่ำสุด หรือ ค่าประมาณของค่าต่ำสุด ของฟังก์ชันจุดประสงค์ สอดคล้องกับเกณฑ์การหยุด (stopping criterion) ที่กำหนดไว้

เมื่อเรากำหนดให้ $C = V - t_k \nabla f(V)$ และอาศัยนิยามจาก (17) เราจะได้สูตรการปรับตัวแปร A คือ

$$A^{(k+1)} = \underset{U}{\text{argmin}} \;\; (1/2)\|U - C\|_F^2 + \lambda t_k \sum_{i \neq j} \left\| \left[(U_1)_{ij} \;\; (U_2)_{ij} \;\; \cdots \;\; (U_p)_{ij} \right] \right\|_2$$

เราจะสังเกตเห็นว่า ขั้นตอนการปรับตัวแปร A ด้านบนนี้ มีรูปแบบเดียวกับการปรับตัวแปร Z ในขั้นตอนวิธี ADMM จาก (12) นั่นคือตัวแปร A จะถูกคำนวณโดยตัวกระทำขีดกั้นแบบอ่อน (soft thresholding operator) เช่นเดียวกับ (15)

สำหรับระยะขั้น t_k เราสามารถใช้ได้สองแบบ คือ

แบบระยะขั้นคงที่

กรณีนี้ เรากำหนดให้ระยะขั้นคงที่ตลอดการวนซ้ำ และเลือกค่าระยะขั้นเป็น $t_k = 1/L$ โดยที่ L เป็นค่าคงที่ลิพชิทซ์ของฟังก์ชัน ∇f ซึ่งสำหรับกรณีนี้ เราจะได้ว่า

$$L = \sqrt{\lambda_{\max}(HH^T HH^T)}$$

แบบระยะขั้นแปรค่า

กรณีนี้ เราต้องการใช้ค่า t_k ที่ทำให้อสมการ

$$f(A^{(k+1)}) \leq f(U) + \nabla f(U)^T (A^{(k+1)} - U) + (1/2t_k)\|A^{(k+1)} - U\|_F^2 \tag{20}$$

เป็นจริงในทุกๆรอบการวนซ้ำ ซึ่งอสมการ (20) สามารถมองได้ว่าเป็นเงื่อนไขอันดับที่หนึ่ง (first-order condition) ของฟังก์ชันคอนเวกซ์อย่างมาก (strongly convex function) หากค่า t_k ไม่สอดคล้องกับเงื่อนไขข้างต้น เราจะลดค่า t_k ลงตามสูตร $t_k := \gamma t_k$ เมื่อ $\gamma \in (0, 1)$ จนกว่าค่า t_k ที่ได้ จะทำให้อสมการ (20) เป็นจริง

3 ผลการทดลอง

การทดลองในโครงงานนี้ เป็นการเปรียบเทียบสมบัติต่างๆ ของขั้นตอนวิธี ADMM, ขั้นตอนวิธี FISTA แบบระยะขั้นคงที่ และ ขั้นตอนวิธี FISTA แบบระยะขั้นแปรค่า เมื่อนำขั้นตอนวิธีเหล่านั้นมาใช้แก้ปัญหาการหาแบบจำลอง AR จากข้อมูลที่สังเคราะห์ขึ้นมาเอง โดยคอมพิวเตอร์ที่ใช้เป็นระบบปฏิบัติการ ubuntu 12.04 LTS 32−บิต พื้นที่หน่วยความจำ 3.9 กิกะไบต์ มีตัวประมวลผลคือ Intel® Xeon(R) CPU X3450 2.67GHz ×8

3.1 การสังเคราะห์ข้อมูล

ในการสังเคราะห์แบบจำลอง AR ตาม (2) เราจะเริ่มต้นด้วยการสังเคราะห์พารามิเตอร์ $A_1,\ A_2,\ \ldots,\ A_p$ ที่ทำให้แบบจำลอง AR มีเสถียรภาพ ขั้นแรกจะสร้างรูปแบบศูนย์ร่วมของ $A_1,\ A_2,\ \ldots,\ A_p$ ขึ้นมาก่อน จากนั้นจึงสร้างเมทริกซ์ $A_k,\ k = 1,2,\ldots,p$ ให้ตำแหน่งของสมาชิกที่ไม่เป็นศูนย์ ตรงกับรูปแบบศูนย์ร่วมที่สร้างขึ้นมา สุดท้ายจึงตรวจสอบเสถียรภาพของระบบ โดยพิจารณาจากสมการสถานะของระบบเชิงเส้นเวลาวิยุต

$$x(t+1) = \mathcal{A}x(t) + \mathcal{B}\nu(t), \tag{21}$$

$x \in \mathbf{R}^{np},\ \mathcal{A} \in \mathbf{R}^{np\times np},\ \mathcal{B} \in \mathbf{R}^{n\times np}$ โดยที่

$$x(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \\ \vdots \\ y(t-p) \end{bmatrix},\ \mathcal{A} = \begin{bmatrix} A_1 & A_2 & \cdots & A_p \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

เนื่องจากระบบเชิงเส้นเวลาวิยุตตาม (21) มีเสถียรภาพก็ต่อเมื่อ ค่าเจาะจงทุกตัวของ $\mathcal{A}$ อยู่ในวงกลมหนึ่งหน่วย ดังนั้นหากเมทริกซ์ $A_1,\ A_2,\ \ldots,\ A_p$ ที่สังเคราะห์ขึ้นมา ไม่สอดคล้องกับเงื่อนไขเสถียรภาพดังกล่าว เราก็จะเริ่มสังเคราะห์เมทริกซ์ $A_1,\ A_2,\ \ldots,\ A_p$ ใหม่

หลังจากสังเคราะห์ได้พารามิเตอร์ที่ทำให้แบบจำลองมีเสถียรภาพแล้ว ก็จะเริ่มสังเคราะห์สัญญาณออก $y(t)$ สำหรับ $t = 1,2,\ldots,N$ ตาม (2) โดยกำหนดให้สัญญาณรบกวนมีการแจกแจงแบบปรกติ

3.2 การเปรียบเทียบสมบัติของขั้นตอนวิธีต่างๆ

ในหัวข้อนี้ เราแบ่งการทดลองออกเป็นสองกรณีคือ กรณีพารามิเตอร์ A หนาแน่น และ กรณีพารามิเตอร์ A เบาบาง ทั้งสองกรณีใช้แบบจำลอง MAR อันดับ 3 มีจำนวนข้อมูลทั้งหมด 500 ตัว แต่ละตัวเป็นเวกเตอร์ใน $\mathbf{R}^{100}$ ในแต่ละกรณี เราทำการทดลองที่ค่า λ แตกต่างกัน 11 ค่าซึ่งเว้นระยะห่างเท่าๆ กันตามมาตราส่วนเชิงลอการิทึมในช่วง $[0.01\lambda_{\max}, \lambda_{\max}]$ โดยที่ $\lambda_{\max}$ คือค่าพารามิเตอร์ลงโทษที่น้อยที่สุดที่ทำให้ผลเฉลยของปัญหา (7) เป็นเมทริกซ์ทแยงมุม

ข้อมูลที่บันทึกในการทดลองที่แต่ละค่า λ ได้แก่ ค่าของฟังก์ชันจุดประสงค์ในแต่ละรอบการวนซ้ำ, ตำแหน่งของสมาชิกที่ไม่เป็นศูนย์ของผลเฉลย, และ เวลาที่ใช้ในการคำนวณจนได้ผลเฉลยออกมา โดยสำหรับการบันทึกเวลา ในแต่ละค่า λ จะคำนวณซ้ำ 50 ครั้ง แล้วจึงนำค่าเฉลี่ยของเวลาที่บันทึกได้มาใช้เปรียบเทียบผลการทดลอง

นอกจากนั้น สำหรับขั้นตอนวิธี ADMM จะมีพารามิเตอร์ ρ (จาก (9)) ด้วย ในการทดลองจึงจัดให้มีการเปลี่ยนแปลงค่า ρ ด้วย โดยค่าที่ใช้ได้แก่ $\rho = 0.1\lambda,\ 0.5\lambda,\ \lambda,\ 5\lambda,\ 10\lambda,\ 50\lambda,$ และ 100λ

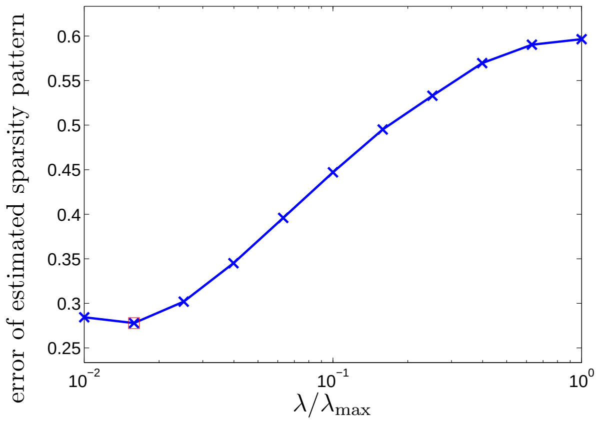
หลังจากทำการทดลองเสร็จทั้งหมด จึงนำข้อมูลที่บันทึกไว้มาประมวลผล ขั้นแรกจะพิจารณาว่าที่ λ ค่าใดให้ความคลาดเคลื่อนของแบบจำลองที่สร้างขึ้น กับ แบบจำลองที่เป็นต้นแบบน้อยที่สุด ความคลาดเคลื่อนดังกล่าวคำนวณโดย

$$\text{ความคลาดเคลื่อน} = \frac{\text{จำนวนตำแหน่งสมาชิกของพารามิเตอร์ที่ทำนายพลาด}}{\text{จำนวนสมาชิกของพารามิเตอร์ทั้งหมด}} \tag{22}$$

สำหรับขั้นตอนวิธี ADMM นั้น จะพิจารณาด้วยว่า ที่ λ แต่ละค่า พารามิเตอร์ ρ มีผลกระทบต่อเวลาที่ใช้ในการคำนวณอย่างไรบ้าง หลังจากนั้นจะเปรียบเทียบผลจากทั้งสามขั้นตอนวิธีทั้งด้าน เวลาที่ใช้ในการคำนวณ และ ความคลาดเคลื่อนสัมพัทธ์ระหว่างค่าฟังก์ชันจุดประสงค์ของ (7) ในแต่ละรอบการวนซ้ำ กับค่าต่ำสุดของ (7) ที่ประมาณจากการใช้โปรแกรม CVX ซึ่งถือว่ามีความแม่นยำสูงกว่า โดยใช้ค่า λ ที่ให้ความคลาดเคลื่อนตาม (22) น้อยที่สุด

กรณีแบบจำลองจริงมีพารามิเตอร์ A หนาแน่น

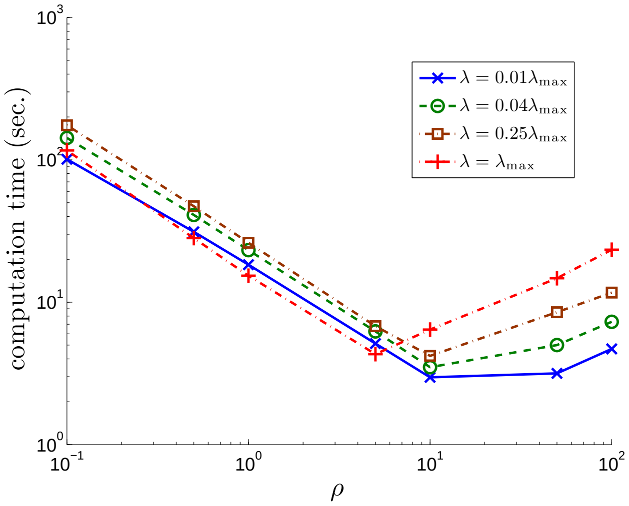
แบบจำลอง MAR จริงที่สร้างได้ในกรณีนี้ มีความหนาแน่นของสมาชิกที่ไม่เป็นศูนย์ของพารามิเตอร์ $A_1, A_2, \ldots, A_p$ เท่ากับ 0.60 โดยได้ค่า $\lambda_{\max} = 47.82$ จากผลการทดลองพบว่า $\lambda = 0.016\lambda_{\max}$ ให้ค่าความคลาดเคลื่อนต่ำที่สุดดังกราฟ



รูปที่1: ผลของค่า λ ที่มีต่อความคลาดเคลื่อนของแบบจำลอง โดยเครื่องหมายสี่เหลี่ยมแสดงค่า $\lambda/\lambda_{\max}$ ที่ให้ความคลาดเคลื่อนต่ำสุด

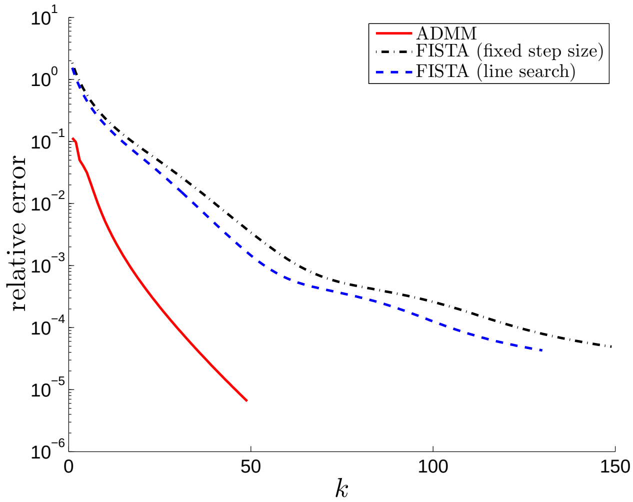
จากผลการทดลอง จะเห็นว่าการใช้ค่า λ เป็นอัตราส่วนที่ต่ำเมื่อเทียบกับ $\lambda_{\max}$ จะทำให้ผลเฉลยของ (7) หนาแน่น

สำหรับขั้นตอนวิธี ADMM ในการทดลองนี้พบว่าค่า ρ ที่ส่งผลให้ใช้เวลาในการคำนวณน้อยที่สุด คือ $\rho = 10\lambda$ สำหรับทุกค่า λ ยกเว้นกรณีเดียวคือกรณี $\lambda = \lambda_{\max}$ ซึ่ง $\rho = 5\lambda$ ให้ผลดีที่สุด จะเห็นได้จากกราฟด้านล่าง

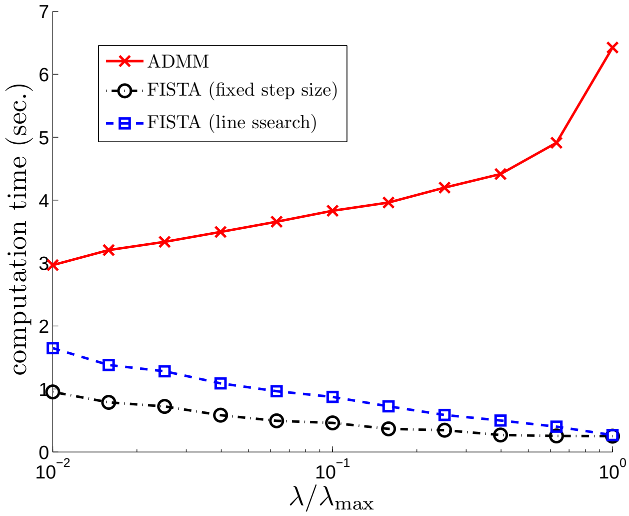


รูปที่2: ผลของค่า ρ ต่อเวลาที่ใช้ในการคำนวณของขั้นตอนวิธี ADMM เมื่อใช้ค่า λ ต่างๆ กัน

ดังนั้น ต่อไปจึงเปรียบเทียบผลการทดลองจากทั้งขั้นตอนวิธี ADMM และ ขั้นตอนวิธี FISTA ทั้งสองแบบ โดยใช้ค่า $\lambda = 0.016\lambda_{\max}$ และสำหรับขั้นตอนวิธี ADMM จะใช้ $\rho = 10\lambda$



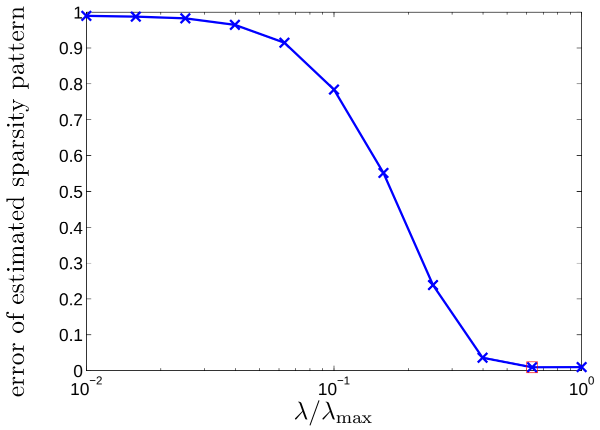
รูปที่3: ความคลาดเคลื่อนสัมพัทธ์ของค่าฟังก์ชันจุดประสงค์ของปัญหา (7) ในแต่ละรอบการวนซ้ำ กับ ค่าต่ำสุดของปัญหา (7)



รูปที่4: เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี ADMM และ FISTA ที่ค่า λ ต่างๆกัน

กรณีแบบจำลองจริงมีพารามิเตอร์ A เบาบาง

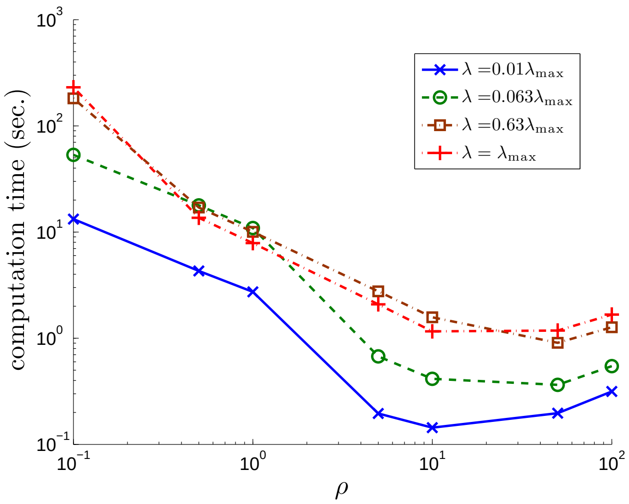
แบบจำลอง MAR จริงที่สร้างได้ในกรณีนี้ มีความหนาแน่นของสมาชิกที่ไม่เป็นศูนย์ของพารามิเตอร์ $A_1, A_2, \ldots, A_p$ เท่ากับ 0.020 โดยได้ค่า $\lambda_{\max} = 1.78$ จากผลการทดลองพบว่า $\lambda = 0.63\lambda_{\max}$ ให้ค่าความคลาดเคลื่อนต่ำที่สุดดังกราฟ



รูปที่5: ผลของค่า λ ที่มีต่อความคลาดเคลื่อนของแบบจำลอง โดยเครื่องหมายสี่เหลี่ยมแสดงค่า $\lambda/\lambda_{\max}$ ที่ให้ความคลาดเคลื่อนน้อยที่สุด

จากผลการทดลอง จะเห็นว่าการใช้ค่า λ เป็นอัตราส่วนที่สูงเมื่อเทียบกับ $\lambda_{\max}$ จะทำให้ผลเฉลยของ (7) เบาบาง

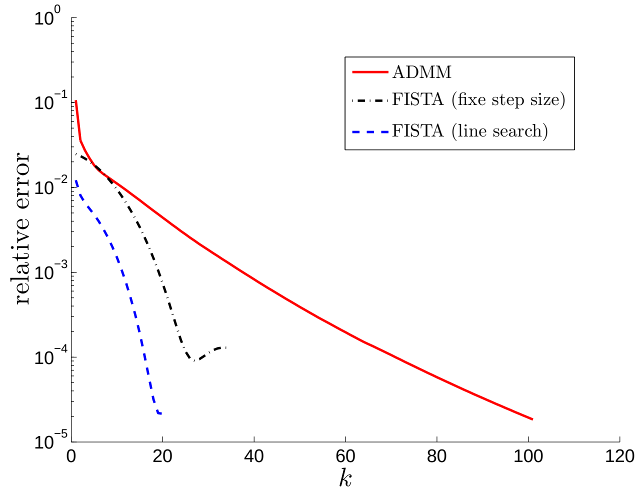
สำหรับค่า ρ ในขั้นตอนวิธี ADMM ที่ส่งผลให้ใช้เวลาในการคำนวณน้อยที่สุด มีสองค่าคือ $\rho = 10\lambda$ และ $\rho = 50\lambda$ ขึ้นอยู่กับค่า λ ที่ใช้ สำหรับค่า λ ที่ให้ความคลาดเคลื่อนน้อยสุดในการทดลองนี้ พบว่า $\rho = 50\lambda$ ให้ผลดีที่สุด



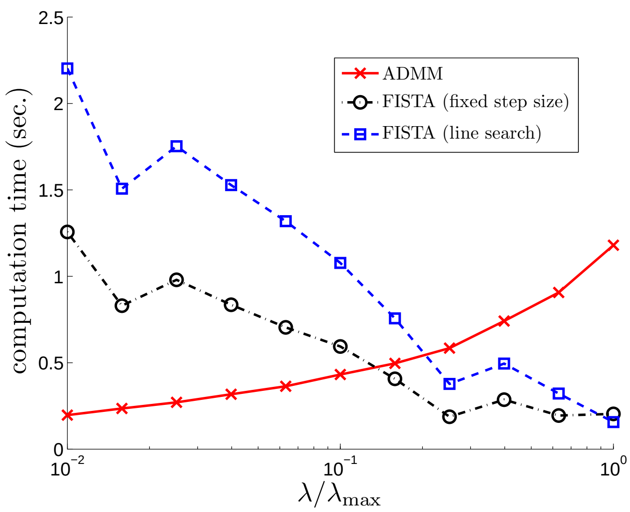
รูปที่6: ผลของค่า ρ ต่อเวลาที่ใช้ในการคำนวณของขั้นตอนวิธี ADMM เมื่อใช้ค่า λ ต่างๆ กัน

ในเปรียบเทียบผลการทดลองจากทั้งขั้นตอนวิธี ADMM และ ขั้นตอนวิธี FISTA ทั้งสองแบบ จะใช้ค่า $\lambda = 0.63\lambda_{\max}$ และสำหรับขั้นตอนวิธี ADMM จะใช้ $\rho = 50\lambda$

ผลการทดลองที่ได้ เป็นดังกราฟต่อไปนี้



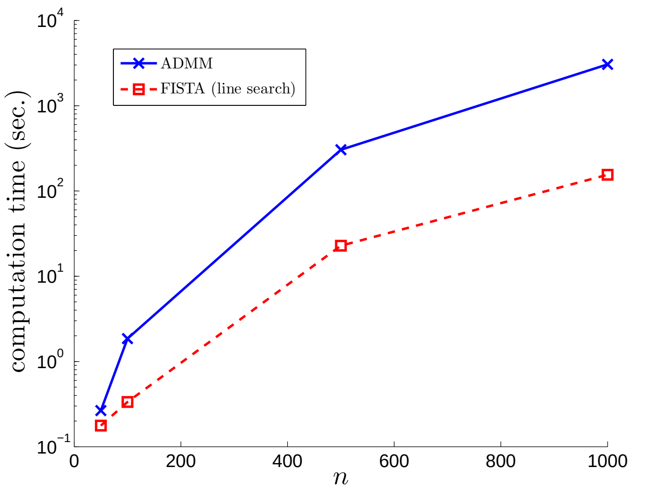
รูปที่7: ความคลาดเคลื่อนสัมพัทธ์ของค่าฟังก์ชันจุดประสงค์ของปัญหา (7) ในแต่ละรอบการวนซ้ำ กับ ค่าต่ำสุดของปัญหา (7)



รูปที่8: เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี ADMM และ FISTA ที่ค่า λ ต่างๆกัน

3.3 การทดลองแปรขนาดปัญหา

ในการทดลองนี้ เราทดสอบขั้นตอนวิธี ADMM และ FISTA แบบระยะขั้นแปรค่า กับปัญหาการจำลอง MAR อันดับ $p = 3$ ขนาดต่างๆคือ $n = 50, 100, 500,$ และ 1000 ปัญหาดังกล่าวจะมีจำนวนตัวแปร เท่ากับ n^2p ตัวอย่างเช่น ปัญหาการหาแบบจำลอง MAR อันดับ 3 ที่มี $n = 1000$ จะมีจำนวนตัวแปรมากถึง $3,000,000$ ตัว



รูปที่9: เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี ADMM และ FISTA ที่ปัญหาขนาดต่างๆกัน

4 บทสรุป

สำหรับขั้นตอนวิธี ADMM จะเห็นว่าค่าพารามิเตอร์ ρ ส่งผลต่อเวลาที่ใช้ในการคำนวณอย่างเห็นได้ชัด และ พบว่า เมื่อสมาชิกที่ไม่เป็นศูนย์ของพารามิเตอร์แบบจำลองที่เป็นตัวต้นแบบ มีความหนาแน่นน้อยลง จะต้องใช้ค่า ρ ที่ส่งผลให้ใช้เวลาในการคำนวณน้อยที่สุด จะเพิ่มมากขึ้น และจากผลการทดลอง จะเห็นว่าควรใช้ค่า ρ ในช่วง 10 ถึง 50 เท่าของ λ

ในการทดลองกรณีพารามิเตอร์ A หนาแน่น พบว่าถึงแม้ขั้นตอนวิธี ADMM จะใช้เวลาในการคำนวณมากกว่า แต่จะใช้จำนวนรอบการวนซ้ำที่น้อยกว่า ในทางกลับกัน กรณีพารามิเตอร์ A เบาบาง พบว่าขั้นตอนวิธี ADMM จะใช้ผลการลู่เข้าที่ไม่ดีนัก เมื่อเปรียบเทียบกับขั้นตอนวิธี FISTA ทั้งสองแบบ แต่ทั้งนี้สำหรับทั้งสองกรณีพบว่า ค่าต่ำสุดของฟังก์ชันจุดประสงค์ที่ได้จากขั้นตอนวิธี ADMM เข้าใกล้ค่าต่ำสุดจริงที่ได้จากการคำนวณโดยใช้โปรแกรม CVX มากกว่าขั้นตอนวิธี FISTA ทั้งสองแบบ

สำหรับทั้งขั้นตอนวิธี ADMM และ FISTA ถือว่าใช้เวลาในการคำนวณในระดับที่ยอมรับได้ โดยเมื่อปัญหามีตัวแปรมากถึง $3,000,000$ ตัว ยังใช้เวลาในการคำนวณเพียงไม่เกิน 1 ชั่วโมงเท่านั้น ในขณะที่ปัญหาขนาดใหญ่เช่นนี้ ไม่สามารถใช้ตัวแก้ปัญหาทั่วไป (general solver) ที่เขียนขึ้นโดยใช้วิธีจุดภายใน (interior-point method) ได้

5 กิตติกรรมประกาศ

ขอขอบพระคุณ อ.ดร.จิตโกมุท ส่งศิริ อาจารย์ที่ปรึกษาโครงงาน ที่ให้คำแนะนำในการทำโครงงานเสมอมา รวมถึงสละเวลาเพื่อตรวจทาน และ แก้ไขข้อบกพร่องในการเขียนรายงานของข้าพเจ้าด้วย

6 เอกสารอ้างอิง

[1] S.P. Boyd and L.Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] R.Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[3] J.Friedman, T.Hastie, and R.Tibshirani. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[4] S.Ghosh and Y.Rudy. Application of ℓ_1-norm regularization to epicardial potential solution of the inverse electrocardiography problem. *Annals of Biomedical Engineering*, 37(5):902–912, 2009.

[5] A.Beck and M.Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] A.Pongrattanakul and J.Songsiri. Learning causal structures of brain connectivity in fnri time series. Technical report, Electrical Engineering Department, Chulalongkorn University, 2013.

[7] H.Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Cambridge University Press, 2005.

[8] M.Grant and S.Boyd. CVX: Matlab software for disciplined convex programming (web page and software). `http://stanford.edu/~boyd/cvx`, August 2008.

[9] S.Boyd, N.Parikh, E.Chu, B.Peleato, and J.Eckstein. *Distributed Optimization and Statistical Learning via The Alternating Direction Method of Multipliers*. Now Publishers, 2011.

[10] M.Yuan and Y.Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2005.

ภาคผนวก

ในการแก้ปัญหาการหาแบบจำลอง AR กรณีที่ไม่ทราบรูปแบบศูนย์ร่วม ซึ่งมีรูปแบบปัญหาคือ

$$\begin{array}{ll} \underset{A}{\text{minimize}} & \|Y - AH\|_F^2 \\ \text{subject to} & (A_k)_{ij} = 0, \ \ (i,j) \notin \mathcal{V}, \ \ k = 1,2,\ldots,p \end{array}$$

โดยที่ $\mathcal{V}$ คือเซตของตำแหน่งสมาชิกที่ไม่เป็นศูนย์ของพารามิเตอร์ A_k เราจะจัดกลุ่มข้อมูล และ ตัวแปรของปัญหาในรูปแบบใหม่ โดยนิยามตัวแปร

$$a_{ij} = [(A_1)_{ij} \ \ (A_2)_{ij} \ \ \cdots \ \ (A_p)_{ij}]^T \in \mathsf{R}^p \tag{23}$$

เพื่อความสอดคล้องกันกับตัวแปรที่กำหนดขึ้นใหม่ ให้

$$Y = \begin{bmatrix} \mathsf{y}_1^T \\ \mathsf{y}_2^T \\ \vdots \\ \mathsf{y}_n^T \end{bmatrix}, \quad X_j = \begin{bmatrix} H_{(j,1)} & H_{(j+n,1)} & \cdots & H_{(j+(p-1)n,1)} \\ H_{(j,2)} & H_{(j+n,2)} & \cdots & H_{(j+(p-1)n,2)} \\ \vdots & \vdots & \ddots & \vdots \\ H_{(j,N-p)} & H_{(j+n,N-p)} & \cdots & H_{(j+(p-1)n,N-p)} \end{bmatrix}$$

สุดท้ายเราจะได้ว่า ปัญหาเดิมสามารถเขียนใหม่เป็นปัญหากำลังสองต่ำสุดย่อยๆได้ n ปัญหา คือ

$$\begin{array}{ll} \underset{a}{\text{minimize}} & \dfrac{1}{2}\left\| \mathsf{y}_i - \displaystyle\sum_{j=1}^{n} X_j a_{ij} \right\|_2^2 \\ \text{subject to} & a_{ij} = 0, \ \ (i,j) \notin \mathcal{V} \end{array}$$

สำหรับ $i = 1,2,\ldots,n$ นอกจากนั้นสังเกตว่าสำหรับ $(i_0, j_0) \notin \mathcal{V}$ เมทริกซ์ X_{j_0} จะไม่เกี่ยวข้องกับการคำนวณเลย ดังนั้น ที่ $i = i_0$ใดๆ เราสามารถหา $a_{i_0 j)}$ สำหรับ $(i_0, j) \in \mathcal{V}$ ได้จากผลเฉลยของปัญหา

$$\underset{\mathsf{a}}{\text{minimize}} \quad \frac{1}{2}\left\|\mathsf{y}_{i_0} - \mathbf{X}\mathsf{a}\right\|_2^2$$

โดยที่ $\mathsf{a} \in \mathsf{R}^{\mathsf{Mp}}$ ได้จากการนำเวกเตอร์ $a_{(i_0 j)}, (i_0, j) \in \mathcal{V}$ มาเรียงต่อกันตามแนวตั้ง, $\mathbf{X} \in \mathsf{R}^{(N-p)\times Mp}$ ได้จากการนำเมทริกซ์ $X_j, (i_0, j) \in \mathcal{V}$ มาเรียงต่อกันตามแนวนอน และ M คือจำนวนดัชนี j ซึ่ง $(i_0, j) \in \mathcal{V}$

เมื่อ $\mathbf{X}$ เป็นเมทริกซ์แบบเต็มขั้น เราจะได้ว่า

$$\mathsf{a} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathsf{y}_{i_0} \tag{24}$$