*Journal of the Korean Data & Information Science Society*
*2017,* **28**(6), 1229–1244

http://dx.doi.org/10.7465/jkdi.2017.28.6.1229
한국데이터정보과학회지

# 통계적 기계학습에서의 ADMM 알고리즘의 활용

최호식[1] · 최현집[2] · 박상언[3]

[12]경기대학교 응용통계학과 · [3]경기대학교 경영정보학과

접수 2017년 10월 31일, 수정 2017년 11월 14일, 게재확정 2017년 11월 21일

**요 약**

최근 여러 분야에서 데이터에 근거한 분석방법론에 대한 수요가 증대됨에 따라 이를 처리할 수 있는 최적화 방법이 발전되고 있다. 특히 통계학과 기계학습 분야의 문제들에서 요구되는 다양한 제약조건은 볼록 최적화 (convex optimization) 방법으로 해결할 수 있다. 본 논문에서 리뷰하는 alternating direction method of multipliers (ADMM) 알고리즘은 선형 제약 조건을 효과적으로 처리할 수 있으며, 합의 방식을 통해 병렬연산을 수행할 수 있어서 범용적인 표준 최적화 툴로 자리매김되고 있다. ADMM은 원래의 문제보다 최적화가 쉬운 부분문제로 분할하고 이를 취합함으로써 복잡한 원 문제를 해결하는 방식의 근사알고리즘이다. 부드럽지 않거나 복합적인 (composite) 목적 함수를 최적화할 때 유용하며, 쌍대이론과 proximal 작용소 이론을 토대로 체계적으로 알고리즘을 구성할 수 있기 때문에 통계 및 기계학습 분야에서 폭 넓게 활용되고 있다. 본 논문에서는 최근 통계와 관련된 여러 분야에서 ADMM알고리즘의 활용도를 살펴보고자 하며 주요한 두 가지 주제에 중점을 두고자 한다. (1) 목적식의 분할 전략과 증강 라그랑지안 방법 및 쌍대문제의 설명과 (2) proximal 작용소의 역할이다. 알고리즘이 적용된 사례로, 벌점화 함수 추정 등의 조정화 (regularization)를 활용한 방법론들을 소개한다. 모의 자료를 활용하여 lasso 문제의 최적화에 대한 실증결과를 제시한다.

*주요용어*: 벌점함수, 병렬컴퓨팅, 제약조건, 조정화, 최적화.

## 1. 서론

최근 여러 분야에서 데이터에 근거한 분석방법론에 대한 수요가 증대됨에 따라 이를 처리할 수 있는 최적화 방법이 발전되고 있다. 특히 통계학과 기계학습 분야의 문제들에서 요구되는 다양한 제약조건은 볼록 최적화 (convex optimization) 방법으로 해결할 수 있다. 지난 10년간 볼록 최적화 (convex optimization) 분야에서 각광받고 있는 ADMM (alternating direction method of multipliers; Boyd 등, 2010) 알고리즘은 제약조건을 비교적 쉽게 처리할 수 있다. ADMM방법을 리뷰한 Boyd 등 (2010) 논문은 2017년 10월말 현재 약 6000여건의 피인용회수 (Google scholar 검색기준)를 보이고 있다. ADMM방법은 오랜 기간 다양하게 발전되어온 비슷한 아이디어들의 융합에 근거하며, 크게 원시 (primal) 문제와 쌍대 (dual) 문제라 불리우는 두 단계의 교차 최적화와 그에 따른 해의 업데이트 과정으로 구성된다. 교차 갱신과정의 계산복잡도 (computational complexity)가 작을 때, 원 문제 대비 효율성이 크다고 알려져 있다. 많은 경우 ADMM 내부의 부분문제 (sub-problem)에 대한 최적화는 proximal 알고리즘 (Parikh and Boyd, 2013)에 의해 수행된다. Proximal 알고리즘은 최적화할 함

[1] 교신저자: (16938) 경기도 수원시 영통구 광교산로 경기대학교 응용통계학과, 조교수.
E-mail: choi.hosik@gmail.com
[2] (16938) 경기도 수원시 영통구 광교산로 경기대학교 응용통계학과, 교수.
[3] (16938) 경기도 수원시 영통구 광교산로 경기대학교 경영정보학과, 부교수.

수를 보다 부드러운 함수로 만들어 줌으로써 국소 근사해를 찾아준다. 이 알고리즘은 proximal 작용소 (operator)와 해의 방향이나 갱신속도를 제어하는 gradient 작용소로 구성된다.

ADMM 알고리즘을 살펴보기로 한다. $p$차원의 모수벡터 $x \in \mathbb{R}^p$라 놓자. 통계학에서는 모수 (parameter)라 부르나, 본고에서는 참고문헌들에서 사용되는 용어들을 감안하여 변수 (variable)로 혼용하여 쓰기로 한다. 두 개의 함수 $f(x)$와 $g(x)$로 구성된 목적함수에 대한 다음의 최적화 문제를 고려해보자.

$$\min_x f(x) + g(x). \tag{1.1}$$

이 문제와 동일한 문제는 아래와 같이 구성할 수 있다.

$$\min_{x,z} f(x) + g(z) \text{ subject to } x = z. \tag{1.2}$$

원 문제 (1.1)과 달리 문제 (1.2)에서는 새로운 모수벡터 $z$가 추가적으로 도입되었다. 여기서, $z \in \mathbb{R}^p$를 보조변수 (auxiliary variable)로 부르고, 원 변수 $x$의 복제변수로 원 목적식의 최적화 문제를 분할하게 하는 역할을 수행한다. 가장 단순한 형태의 분할은 원변수와 보조변수와의 관계식이 항등관계인 $x = z$이다. 가장 널리 알려진 lasso 예제를 통해 확인하여보자.

***예제 1.1*** [lasso 문제] 설명변수와 종속변수로 구성된 자료쌍 $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$에 대해 다음의 lasso 목적식을 고려하자.

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.3}$$

회귀계수 $\beta \in \mathbb{R}^p$에 대해 보조변수 $\gamma \in \mathbb{R}^p$를 도입하여 $\beta = \gamma$의 관계식으로 설정하면 위와 동일한 문제는

$$\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\gamma_j| \tag{1.4}$$

가 된다. 원 목적식이 $\beta$-최적화 부분 ($\sum_{i=1}^n (y_i - \mathbf{x}_i^T \beta)^2$)과 $\gamma$-최적화 부분 ($\sum_{j=1}^p |\gamma_j|$)으로 분할됨을 알 수 있다. 반면, 최적화할 모수의 수는 $\beta$와 $\gamma$로 2배로 증가한다. 식 (1.3)은 $\beta$에 대한 이차계획법 (quadratic program) 문제이다. 반면 식 (1.4)는 $\beta = \gamma$의 제약식을 제외하면 $\beta$와 $\gamma$의 최적화는 개별로 이루어질 수 있다. 그러므로 선형제약식 $\beta - \gamma = 0$을 만족하게 하면서 교차로 $\beta$와 $\gamma$를 최적화하는 아이디어를 생각할 수 있겠다. 사실 이러한 분할 방법은 1950년대의 Douglas-Rachford splitting으로 편미분 방정식의 수치해를 구하기 위한 방안으로 개발되었다. Bregmann iteration 등으로도 불리는데, 비슷한 유형으로 통계학 및 기계학습분야 뿐 만 아니라 이미지 프로세싱, compressive sensing, matrix completion, 금융 등의 분야에서 발전되어왔다.

ADMM의 특징을 간략히 정리하면 다음과 같다. 첫째, 분할될 수 있는 선형 제약조건을 가진 볼록 문제는 보조변수를 도입하여 모수의 분리함으로써 쉽게 처리될 수 있다. 둘째, ADMM을 통해 큰 문제를 작은 문제로 분할시킬 수 있다. 관측치가 많은 데이터의 경우 데이터를 분할된 블록으로 분해하고 각 블록에 대해 최적화를 수행할 수 있다. 이 경우 각 데이터 블록에 대한 최적화를 통한 해가 전체 자료에 대한 해로 수렴할 수 있도록, 서로 일치하게 하는 제약 조건이 포함되어야 한다. 동일한 방법으로 각 변수를 여러 블록으로 분할할 수 있고, 부분문제를 취합하여 전체 문제에 대한 최적해를 산출하는 방법이라 할 수 있다 (Hastie 등, 2016).

한편 가장 일반적인 형태의 등호 선형 제약식은 $Ax+Bz=c$이다. 여기서 $A$와 $B$는 적절한 행렬이고 $c$는 상수벡터이다. 예를 들어, 모수들이 나무 (tree) 형태의 위계적으로 구조화된 제약이 되어 있다하더라도 ADMM 알고리즘을 적용할 수 있다 (Yan과 Bien, 2015). 최근에는 딥러닝 (deep learning) 분야에 이르기까지 광범위하게 활용되고 있다. Wen 등 (2016)은 CNN (convolutional neural network) 방법에 대해 구조적 성긴 조정학습법 (structured sparsity learning)을 활용하여 필터, 채널, 레이어의 층 등 구조를 조정 (regularize)하기 위한 최적화 방법으로 ADMM을 활용하였다. 이 외에도 Yang 등 (2017)과 Taylor 등 (2016) 등은 범용적인 딥러닝 방법의 분산처리를 위해 ADMM을 활용하였다. 제약식의 구체적 적용사례는 후술할 2.5절의 fused lasso를 비롯한 여러 예제들에서 구체적으로 다루기로 한다.

ADMM의 참고문헌으로는 Boyd 등 (2010), proximal 알고리즘의 리뷰는 Parikh과 Boyd (2013)를 참고하기 바란다. 본 논문은 주로 Boyd 등 (2010)과 Polson 등 (2015)의 리뷰를 기초로 함을 미리 밝히며 구성은 다음과 같다. 2절에서는 ADMM 알고리즘과 이에 필요한 기본적 내용 등을 기술한다. 3절에서는 ADMM의 응용사례을 살펴보고, 간단한 모의 자료를 활용하여 lasso 문제의 최적화에 대한 실증결과를 제시한다. 끝으로 4절에서는 결론 및 GPU 병렬 처리를 위한 향후 계획을 기술한다.

## 2. ADMM 방법론

이 절에서는 제약조건을 가진 최적화 문제를 비제약 (unconstrained) 형식의 벌점화 (penalization)된 문제로 전환하여 이해하고, 이와 관련된 쌍대문제 (dual problem)를 최적화하기 위한 방법을 설명한다.

### 2.1. 쌍대문제와 증강 라그랑지안 방법

문제 (1.2)에 대한 라그랑지안 함수 (Lagrangian function)를 생각해 보자.

$$L(x,z,\alpha)=f(x)+g(z)+\alpha^T(x-z),$$

여기서 $\alpha$는 라그랑지안 승수 (Lagrangian multipliers)이다. 먼저 문제 (1.2)를 원시 (primal) 문제라 하고, 모수 $x,z$를 원시모수 (primal parameter)라 한다. $\alpha$가 주어졌을 때, $x,z$에 대한 최소값을 가지는 함수를 쌍대함수 (dual function)라 부르고, 다음으로 정의한다.

$$d(\alpha)=\inf_{x,z} L(x,z,\alpha).$$

제약조건 $x-z=0$하에서 원시문제에서의 최적값을 $p^*=\inf_{x,z}\{f(x)+g(z)\}$으로 놓으면 $d(\alpha)$와 $p^*$와의 관계는 다음과 같음을 증명할 수 있다 (Boyd와 Vandenberghe, 2004).

$$d(\alpha)=\inf_{x,z} L(x,z,\alpha)\le p^*=\inf_{x,z}\{f(x)+g(z)\}.$$

그러므로 $d(\alpha)$는 $p^*$에 대한 하한 (lower bound)이 된다. 따라서 이 하한의 최대값을 구하면 $p^*$와 최대한 근접하게 될 것이라 예상할 수 있다. 이상의 설명을 다음의 문제로 정식화할 수 있게 된다.

$$\max_{\alpha} d(\alpha).$$

이를 원시문제에 대한 쌍대문제라 부른다. 쌍대문제의 최적값을 $d^*=\sup_\alpha d(\alpha)$라 두면 위 관계식에 의해 약쌍대성 (weak duality)

$$d^*\le p^*$$

은 항상 성립한다. 대개 목적함수가 볼록함수 등 적당한 조건이 보장되면 $p^* - d^* = 0$로 쌍대차 (duality gap)가 0이 되어 강쌍대성 (strong duality)이 성립한다. 이는 목표로 하는 원시문제의 최적값을 쌍대문제의 최적값으로 얻을 수 있음을 의미한다. 따라서 쌍대문제는 원시문제보다 풀기가 더 쉬운 경우에 효과적으로 활용될 수 있다. 대표적인 예제로는 support vector machine (Park 등, 2015)을 쌍대문제로 최적화를 푸는 것이 이에 해당된다고 할 수 있겠다.

한편 함수 $h(x)$의 볼록 공액함수 (convex conjugate function) $h^*(\alpha)$는 다음으로 정의된다.

$$h^*(\alpha) = \sup_x \left\{\alpha^T x - h(x)\right\}.$$

볼록 공액 함수의 성질로 $h(x)$가 볼록함수가 아니어도 $h^*(\alpha)$은 볼록함수가 된다는 점을 들 수 있다. 이를 활용하면, 전술한 원시문제가 볼록 (convex) 문제 여부에 관계없이 쌍대문제는 볼록 문제가 된다. 또한, 쌍대함수와 공액함수와의 관계는 다음과 같다 (Boyd와 Vandenberghe, 2004).

$$\begin{aligned} d(\alpha) &= \inf_{x,z} \Big( f(x) + g(z) + \alpha^T (x - z) \Big) \\ &= -[f^*(-\alpha) + g^*(\alpha)]. \end{aligned}$$

조정화 (regularization)에서 함수 $g$는 벌점함수에 해당되는데, 보통 $l_q, q > 0$ 노음 (norm)이 활용된다. 쌍대함수와 공액함수의 관계를 활용하면, 잘 정의되어 있는 노음함수의 공액함수를 통해 최적화 기법을 체계적으로 적용할 수 있게 된다.

상대적으로 풀기가 힘든 원시문제인 경우 이를 직접 최적화하기 보다 쌍대문제를 풀어 원시문제의 해를 구할 수 있다. 대개의 경우 쌍대문제는 볼록 문제이므로 상대적으로 전역 최대해 $d^*$를 구하기 쉽다. 이 경우 쌍대문제에서 최적해 $\alpha^*$를 구한 뒤 원시문제의 최적해 $x^*, z^*$를 구한다. 만약 원시문제가 볼록 문제이고 Slater의 조건 (Slater's condition)을 만족하면 강쌍대성이 성립한다. 보다 자세한 내용은 Boyd와 Vandenberghe (2004)를 참조하기 바란다.

이제 문제 (1.2)의 쌍대문제를 풀어보기로 한다. 가장 기본적인 그래디언트 (gradient) 상승 알고리즘을 통하여 쌍대문제에서의 최대값을 구해보자. 쌍대문제에서의 그래디언트 상승 (dual ascent) 알고리즘의 기본적인 아이디어는 쌍대문제를 $\alpha$에 대한 그래디언트의 반복적인 갱신과정을 통해 최적화하는데 있다. $\alpha$가 주어졌을 때, 원시 해를

$$(x_\alpha, z_\alpha) = \arg\min_{x,z} L(x, z, \alpha)$$

라 두고 $d(\alpha)$를 구한다. $\alpha$에 대한 그래디언트는 $\nabla d(\alpha) = \nabla_\alpha L(x_\alpha, z_\alpha, \alpha) = x_\alpha - z_\alpha$이다. 그러므로, 그래디언트 상승 알고리즘은 먼저, 주어진 $\alpha$에서의 $d(\alpha)$를 구하고, 그래디언트 상승 방법을 적용하여 $\alpha$에 대한 해를 갱신한다. 따라서 쌍대문제에서의 그래디언트 상승 알고리즘은 다음의 두 단계를 수렴할 때까지 반복적으로 실행한다.

$$\begin{aligned} (x^{t+1}, z^{t+1}) &= \arg\min_{x,z} L(x, z, \alpha^t), \\ \alpha^{t+1} &= \alpha^t + \rho(x^{t+1} - z^{t+1}), \end{aligned}$$

여기서 $t$는 반복수이며, $\rho$는 step-size의 역할을 한다. $\rho_t$로 $t$에 의존할 수도 있다. 또한, 그래디언트가 존재하지 않을 경우, 서브 (sub) 그래디언트로 대체될 수 있음을 미리 밝혀 둔다.

그래디언트에만 근거한 방법은 일반적으로 불안정한 것으로 알려져 있다. 개선할 수 있는 여러 방법들 중, 증강 (augmented) 라그랑지안 방법을 소개한다 (Bertsekas, 2003; Choi 등, 2015). 이 방법은

$L(x, z, \alpha)$에 제약조건 $x - z = 0$에 대한 $l_2$ 벌점 ($\|x\|_2 = \sqrt{\sum_{j=1}^{p} x_j^2}$)을 다음과 같이 추가적으로 고려한다.

$$L_\rho(x, z, \alpha) = f(x) + g(z) + \alpha^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2,$$

여기서 $\rho$ ($> 0$)는 그래디언트 상승 알고리즘에서와 같이 step-size 역할을 한다. 한편 조정화의 목적식에서 차지하는 벌점항의 가중치를 조율하는 모수 (tuning parameter)로 해석될 수도 있다. 이를 원 문제 관점에서 해석하면, $x - z = 0$ 제약하에서의

$$\min_{x,z} f(x) + g(z) + \frac{\rho}{2}\|x - z\|_2^2, \tag{2.1}$$

문제와 동일함을 알 수 있다. 따라서 라그랑지안 함수가 아닌 원 문제 관점에서 해석하면 $l_2$ 노음 형태의 벌점을 고려하여, 안정적인 최적화를 하는 것이 주된 목적이라 할 수 있다. 전술한 식과 동일하게 증강 라그랑지안 방법의 쌍대함수를

$$d_\rho(\alpha) = \inf_{x,z} L_\rho(x, z, \alpha)$$

으로 정의하자. 이 경우의 $d_\rho(\alpha)$는 $d(\alpha)$와 달리 국소적으로 미분 가능한 함수가 될 수 있다. 이제, 증강 라그랑지안 방법에 그래디언트 상승 알고리즘을 적용하면 다음의 두 단계로 정리할 수 있다.

$$(x^{t+1}, z^{t+1}) = \arg\min_{x,z} L_\rho(x, z, \alpha^t), \tag{2.2}$$

$$\alpha^{t+1} = \alpha^t + \rho(x^{t+1} - z^{t+1}). \tag{2.3}$$

이제 문제 (2.2)와 갱신공식 (2.3)을 표준화된 척도화 형식으로 표현하여 표준적인 ADMM 알고리즘 기술하기로 한다.

### 2.2. 척도화 형식과 ADMM 표준 알고리즘

보통 라그랑지안 함수 $L_\rho(x, z, \alpha)$에서 $\rho$에 대한 의존성을 명시적으로 회피하고자 다음의

$$\alpha^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2 = \frac{\rho}{2}\|x - z + \rho^{-1}\alpha\|_2^2 - \frac{\rho}{2}\|\rho^{-1}\alpha\|_2^2,$$

관계식을 활용하면 $L_\rho(x, z, \gamma^t) = f(x) + g(z) + \frac{\rho}{2}\|x - z + \rho^{-1}\alpha\|_2^2 - \frac{\rho}{2}\|\rho^{-1}\alpha\|_2^2$로 재표현할 수 있다. 식 (2.2)에서 $\delta = \rho^{-1}\alpha$으로 놓으면 식 (2.2)와 (2.3)의 식은 다음과 같이 재기술된다.

$$(x^{t+1}, z^{t+1}) = \arg\min_{x,z} L_\rho(x, z, \delta^t),$$

$$\delta^{t+1} = \delta^t + (x^{t+1} - z^{t+1}).$$

보조변수 $z$를 도입하지 않은 증강 라그랑지안 방법에서는 문제 (1.1)의 $f(x)$와 $g(x)$가 보통 분할되지 않는다. 그러나, 보조변수를 도입한 문제 (1.2)의 목적식에서 $f(x) + g(z)$는 각각 $x$와 $z$에 따라 가법적으로 분할이 된다. ADMM 알고리즘에서 alternating direction 용어는 식 (2.2)에서 $x^{t+1}$와 $z^{t+1}$를 동시에 최적화하여 구하기 보다 $x^{t+1}$와 $z^{t+1}$를 교차 분할하여 구하는 과정에서 유래한다. 그러면 ADMM의 표준적인 세 단계는 다음으로 주어진다.

$$x^{t+1} = \arg\min_{x} \Big( f(x) + \frac{\rho}{2}\|x - z^t + \delta^t\|_2^2 \Big), \tag{2.4}$$

$$z^{t+1} = \arg\min_{z} \Big( g(z) + \frac{\rho}{2}\|x^{t+1} - z + \delta^t\|_2^2 \Big), \tag{2.5}$$

$$\delta^{t+1} = \delta^t + (x^{t+1} - z^{t+1}). \tag{2.6}$$

라그랑지안 승수 갱신공식에서 $\rho$가 사라진 것을 확인할 수 있다. 그러나, 실제적으로 $\rho$에 대한 의존성이 사라진 것이 아니므로 주의가 필요하다. 벡터로 연산이 수행되므로 $x_j^{t+1} - z_j^{t+1} = 0$을 만족하는 $j$축에 대해서는 $\delta_j^{t+1} = \delta_j^t$로 불변이다. 특히, $x^{t+1}$와 $z^{t+1}$를 구하는 최적화 방법은 보통 proximal 작용소를 활용하는 방법으로 주어진다. proximal 작용소에 관심있는 독자는 자세한 내용은 Parikh과 Boyd, 2013을 참조하기 바란다. 본고에서는 proximal 작용소 prox를 다음과 같이 간단히 소개하기로 한다. 함수 $f$에 대한 proximal 작용소는 $\text{prox}_f : \mathbb{R}^p \to \mathbb{R}^p$으로의 사상이며, 아래와 같이 정의된다.

$$\text{prox}_f(x) = \arg\min_{\nu} \Big[ f(\nu) + \frac{1}{2}\|\nu - x\|_2^2 \Big].$$

많은 경우 상수 $\eta$를 감안한 다음의 척도화 형식으로 활용된다.

$$\text{prox}_{\eta f}(x) = \arg\min_{\nu} \Big[ f(\nu) + \frac{1}{2\eta}\|\nu - x\|_2^2 \Big].$$

해석으로는 현재 해 $x$에 가까운 $f(x)$의 최소값을 찾는 과정이라고 할 수 있다. $\eta$가 충분히 작고, $f$가 미분가능할 때, 보다 직접적으로 해석할 수 있다. 이를 확인하기 위해 Moreau envelope (Moreau-Yosida regularization)으로 불리는

$$R^\eta(x) = \inf_{\nu} \Big[ f(\nu) + \frac{1}{2\eta}\|\nu - x\|_2^2 \Big]$$

을 고려하자. 편의상 Moreau envelope이 미분가능한 함수 $f$에 대해 잘 정의되었다고 가정하자. 그러면 $R^\eta(x)$는 함수 $f$의 최소값을 현재 해 $x$와 유사한 범위에서 찾게한다. 간단한 연산을 통하면

$$\nabla R^\eta(x) = \frac{1}{\eta}\Big( x - \text{prox}_{\eta f}(x) \Big)$$

으로 주어짐을 확인할 수 있다. 이는 $\text{prox}_{\eta f}(x) = x - \eta \nabla R^\eta(x)$으로 재표현할 수 있다. 따라서 그래디언트 강하 (gradient descent)방법의 해의 갱신과정에서 $\text{prox}_{\eta f}(x)$는 현재 해 $x$에서 갱신된 해로 해석할 수 있다. 한편 proximal gradient 방법은 문제 (1.1)에서 보조변수 $z$를 도입하지 않고 proximal 작용소와 gradient 작용소를 활용하는 방법이다. 이는 Moreau envelope이 상계 (upper bound)의 역할을 하는 majorization-minimization (MM, Lange, 2016) 방법의 특별한 경우로 해석될 수 있다.

예제 (1.1)의 lasso의 경우에 대해 ADMM의 세단계 (2.4)-(2.6)를 proximal 작용소를 활용하여 기술하면 다음과 같다.

$$\begin{aligned}
\beta^{t+1} &= \arg\min_{\beta} \Big( \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\rho}{2}\|\beta - (\gamma^t - \delta^t)\|_2^2 \Big) \\
&= \text{prox}_{(1/\rho)f}(\gamma^t - \delta^t) \text{ where } f(\beta) = 1/2\|y - X\beta\|_2^2 \\
&= (X^T X + \rho I)^{-1}\Big( X^T y + \rho(\gamma^t - \delta^t) \Big), \\
\gamma^{t+1} &= \arg\min_{\gamma} \Big( \lambda\|\gamma\|_1 + \frac{\rho}{2}\|\beta^{t+1} - \gamma + \delta^t\|_2^2 \Big) \\
&= \text{prox}_{(1/\rho)g}(\beta^{t+1} + \delta^t) \text{ where } g(\gamma) = \lambda\|\gamma\|_1 \\
&= \text{sign}(\beta_j^{t+1} + \delta_j^t)\max(|\beta_j^{t+1} + \delta_j^t| - \lambda/\rho, 0), \forall j \\
\delta^{t+1} &= \delta^t + (\beta^{t+1} - \gamma^{t+1}).
\end{aligned}$$

첫번째 업데이트는 수정된 반응변수에 대한 능형회귀추정량 형태로 주어지며, 두 번째 업데이트에서는 보조변수의 각 축마다 약분계 작용소 (soft thresholding operator)를 적용한 식으로 주어진다. 여기서 $\|x\|_1 = \sum_{j=1}^p |x_j|$이며, 약분계 작용소는 $S(z, \lambda) = \text{sign}(z)\max(|z| - \lambda, 0)$이다.

ADMM은 문제 (1.1)에서 $f+g$의 proximal 작용소의 복잡도 대비 $f$, $g$ 각각의 proximal 작용소의 복잡도가 작을 때 효과적이라 할 수 있다. 또한, Nesterov의 가속화 (accerleration) 단계 (Beck과 Teboulle, 2009)를 추가하면 보다 빠른 계산속도를 얻을 수 있다. 여러 알고리즘의 세부적인 차이에 따른 수렴 속도의 비교는 Polson 등 (2015)의 표에서 제시되어 있으니 참고하기 바란다.

### 2.3. 정지 기준과 수렴

2.1절에서 살펴본 그래디언트 상승 알고리즘의 수렴성은 다음의 가정에 기초하고 있다. $\alpha^*$를 쌍대문제의 최적해, $x^*, z^*$를 원시문제에서의 최적해라고 놓으면, 모든 $x, z, \alpha$에 대해

$$L_0(x^*, z^*, \alpha) \leq L_0(x^*, z^*, \alpha^*) \leq L_0(x, z, \alpha^*)$$

이 만족된다. 즉, 증강되지 않은 라그랑지안 함수 $L_0$가 안장점 (saddle point)을 가진다 (Bertsekas, 2003). 일반적으로 이러한 가정하에서 ADMM 알고리즘은 수렴하는 것으로 알려져 있다. 구체적인 증명은 Boyd 등 (2010)의 부록을 참조하자. 식의 우측 부등식은 $\alpha^*$가 주어졌을 때, $x, z$에 관한 최소화 관점이고 좌측 부등식은 $x^*, z^*$가 주어졌을 때의 쌍대문제의 최대화에 관한 것이다. 결국 쌍대 그래디언트 상승 알고리즘의 두 단계는 좌우의 최소화와 최대화 과정을 의미한다. $\rho$가 상수로 고정된 경우, proximal 작용소를 활용하기 때문에 수렴속도는 반복수 (iteration) $t$에 대해 $O(1/t)$로 알려져 있다 (He와 Yuan, 2012; Fang 등, 2015). 여기서 $O(1/t)$는 최적값과의 차이에 대한 허용한계보다 작게 되는 속도가 $1/t$로 $t$에 반비례한다는 것을 의미한다. 만약 $f$와 $g$중 최소한 하나의 함수가 강볼록 (strongly convex) 함수이면 $O(1/t^2)$로 개선된 속도를 얻을 수 있는 것으로 알려져 있다.

ADMM은 원시문제와 쌍대문제를 동시에 푸는 것으로 해석될 수 있기 때문에 두 종류의 가변영역 (feasible region)과 관련된 잔차를 구할 수 있다. 첫번째는 원시잔차 (primal residual)로 불리우며, primal 가변성 (feasibility)과 관련된다. 두번째는 쌍대잔차 (dual residual)로 불리우며, dual 가변성과 관련된다. primal/dual 가변성은 ADMM 알고리즘의 수렴성에 대한 필요충분한 조건이다. 논의의 편의상 $f$와 $g$ 모두 미분가능하다고 하자. $x^*, z^*$는 문제 (1.2)에 대한 최적해이므로 제약조건을 만족하고, primal 가변성

$$x^* - z^* = 0$$

이 성립한다. 한편 dual 가변성은 $L_0$의 편미분을 0으로 놓은 식으로부터 두 개의 방정식

$$\nabla f(x^*) + \alpha^* = 0, \nabla g(z^*) - \alpha^* = 0$$

을 얻는다. $z^{t+1}$은 정의에 의해 $L_\rho(x^{t+1}, z, \alpha^t)$를 최소화한다. 그러면 다음의 관계식을 만족한다.

$$\begin{aligned} 0 &= \nabla g(z^{t+1}) - \alpha^t - \rho(x^{t+1} - z^{t+1}) \\ &= \nabla g(z^{t+1}) - \alpha^t - \rho r^{t+1} \\ &= \nabla g(z^{t+1}) - \alpha^{t+1}, \end{aligned}$$

여기서, $r^t = x^t - z^t$는 $t$번째에서의 원시잔차이다. 그러므로, 두번째 방정식은 알고리즘의 업데이트 과정에서 자동적으로 만족한다. 한편 $x^{t+1}$는 $L_\rho(x, z^t, \alpha^t)$의 최소해이므로,

$$\begin{aligned} 0 &= \nabla f(\alpha^{t+1}) + \alpha^t + \rho(x^{t+1} - z^t) \\ &= \nabla f(\alpha^{t+1}) + \left(\alpha^t + \rho r^{t+1} - \rho(z^t - z^{t+1})\right) \\ &= \nabla f(\alpha^{t+1}) + \alpha^{t+1} - \rho(z^t - z^{t+1}) \end{aligned}$$

을 만족한다. $\nabla f(\alpha^{t+1}) + \alpha^{t+1} = -\rho(z^{t+1} - z^t)$로부터 $s^{t+1} = -\rho(z^{t+1} - z^t)$로 놓자. $s^t$가 0으로의 수렴여부를 모니터링하면 dual 가변성 조건의 첫번째 조건 충족 여부를 살펴볼 수 있다. 이를 $r^t$에 대한 쌍대잔차라 부른다. 가장 표준적인 알고리즘 정지규칙은 반복과정에서 원시/쌍대 잔차의 각각의 $l_2$노음 $\|r^t\|_2$와 $\|s^t\|_2$을 주어진 허용한계 $\epsilon$보다 작을 때까지 반복하는 것이다. 보다 다양한 정지기준은 Boyd *et al.*, 2010의 3.3.1절을 참조하기 바란다.

수렴성에 관련된 가정은 보통 데이터의 크기 $n$의 증가속도와 관련되기 보다는 함수 $f$나 $g$의 성질에 많이 의존한 non-asymptotic 특징이 있다 (Davis와 Yin, 2016). 예를 들면, 함수 $f$와 $g$, 혹은 $\nabla f$와 $\nabla g$가 Lipschitz 조건, 강볼록 함수를 각각 만족하거나 동시 만족하느냐 등의 가정을 들 수 있다. 선형제약 조건의 경우는 $Ax + Bz - c = 0$에서 행렬 $A$와 $B$의 차수 (rank)나 노음에 대한 가정을 들 수 있다. 이론적으로는 ADMM은 유한의 어떠한 $\rho$값에 대해서도 수렴하는 것으로 알려져 있지만, 많은 경우에 ADMM의 효율성은 $\rho$의 선택에 매우 민감하다 (Yin 등, 2008). 벌점함수를 활용하는 조정화 관점에서, $\rho$는 조율의 대상이 되는 모수라 할 수 있으므로 $\rho$를 자료에 적응하여 선택하는 것은 자연스럽다고 할 수 있다 (Xu 등, 2017).

### 2.4. 병렬화

ADMM 방법은 가법 분할가능한 (separable) 목적식과 제약식 각각에 대해 proximal 작용소를 활용하는데, 이는 구조적으로 병렬로 연산할 수 있음을 의미한다. 통신비용 (communication cost)을 잘 제어하면 최적화의 복잡도를 개선할 수 있다. 분할방식은 자료의 분할과 변수의 분할로 두 가지 방식을 활용할 수 있다. 이절에서는 보편적인 분할 방식인 합의 (consensus) 문제를 살펴보도록 한다. 주어진 자료를 $K$개의 분할된 블럭으로 나눈 경우를 고려해 보자. $k$번째 블럭에 속하는 자료의 첨자집합을 $I_k \subset \{1, \cdots, n\}$라 하자. 목적함수를 $K$개의 블럭으로 분할하고 이를 합한 함수는 $f(x) = \sum_{k=1}^K f_k(x)$으로 표현할 수 있다. 협력하는 (collaborating) 변수로 공통된 전역변수 (common global variable) $z$를 두고, 각 데이터 블럭마다의 변수를 $x_k, k = 1, \cdots, K$라 두자. $x_k$들은 전역변수의 복제변수라 할 수 있다. $x_k - z = 0, \forall k$ 제약하에서

$$\min_{x_1, \cdots, x_K} \sum_{k=1}^K f_k(x_k)$$

문제를 고려할 수 있다. 이를 원문제에 대한 전역 합의 문제 (global consensus problem)라 일컫는다. 블럭마다의 국소 변수들이 전역변수 $z$와 모두 동일해야 함을 알 수 있다. 원 변수 $x$보다 $K$개의 블럭마다의 국소 변수들 $x_k$로 모수의 복잡도가 늘어났으나 원자료를 분할하므로 알고리즘이 자료의 개수에 의존할수록 합의의 효과는 크다고 할 수 있다. 일반적으로는 자료와 분할과의 관계를 그래프로 표현하여 활용한다. 예를 들어, SVM에 대한 ADMM방법을 적용한 예제는 Forero 등 (2010)과 Choi 등 (2013) 등에서 확인할 수 있다.

선형제약식 $x_k - z = 0, \forall k$ 하에서 식 (1.2)에 대한 합의문제는

$$\min_{x_1, \cdots, x_K, z} \sum_{k=1}^K f(x_k) + g(z) \tag{2.7}$$

으로 표현된다. 이제 식 (2.7)에 ADMM은 $x_k - z = 0, k = 1, \cdots, K$에 해당하는 라그랑지안 승수 $\alpha_k, k = 1, \cdots, K$들이 필요하다. 편의상 총 $K$개의 병렬화가 가능하다고 하자. 각 $k = 1, 2, \cdots, K$에

대해 다음의 ADMM 갱신공식을 수행하면 된다.

$$x_k^{t+1} = \arg\min_{x_k} \Big[ f_k(x_k) + (\alpha_k^t)^T (x_k - z^t) + \frac{\rho}{2} \|x_k - z^t\|_2^2 \Big], \forall k \tag{2.8}$$

$$z^{t+1} = \arg\min_{z} \Big[ g(z) + \sum_{k=1}^{K} \Big( - (\alpha_k^t)^T z + \frac{\rho}{2} \|x_k^{t+1} - z\|_2^2 \Big) \Big], \tag{2.9}$$

$$\alpha_k^{t+1} = \alpha_k^t + \rho(x_k^{t+1} - z^{t+1}), \forall k, \tag{2.10}$$

여기서, $f_k(x_k)$는 $k$번째 집합 $I_k$에 속하는 데이터 블록에서 정의된다.

## 3. 응용사례 및 모의실험

이 절에서는 ADMM을 활용한 방법론을 소개하고, 간단한 모의 자료를 활용하여 lasso 문제의 최적화에 대한 실증결과를 제시한다.

### 3.1. 응용사례

***예제 3.1*** [fused lasso 문제] 단백질 질량 분광 (protein mass spectroscopy) 데이터는 전하량 대비 질량 비 (mass over charge ratio, m/z)에 따라 구성 단백질을 정렬하였다. m/z의 크기에 따라 정렬되었기 때문에 인접한 m/z들은 강상관되어 있다. 예제 (1.1)의 문제 (1.3)과 유사하지만, 인접한 m/z의 회귀계수가 비슷하게 하는 fused 벌점 (Tibshirani 등, 2005)

$$\sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|$$

을 활용하면, 다음의 문제를 고려할 수 있다.

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|. \tag{3.1}$$

모수들간의 선형제약 대비행렬 $D \in \mathbb{R}^{(p-1)\times p}$

$$D = \begin{pmatrix} -1 & 1 & 0 & \cdots & & 0 \\ 0 & -1 & 1 & \cdots & & 0 \\ \vdots & & & & & 0 \\ 0 & 0 & 0 & & -1 & 1 \end{pmatrix}$$

를 도입하고, 보조변수 $\gamma \in \mathbb{R}^{p-1}$를 $\gamma = D\beta$의 관계식으로 설정한다. 그러면 인접 회귀계수에 대한 벌점은 $\sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}| = \sum_{j=1}^{p-1} |\gamma_j|$이 된다. $D$의 행의 크기는 $p-1$이 되는데, 이는 선형제약식의 수가 된다. 이 문제의 쌍대문제에서는 선형제약식 개수가 추정해야할 모수의 개수가 된다. $D\beta = \gamma$와 같은 구조에서의 ADMM 방식을 특히 Bregmann 형식이라 부른다. 일반화한 $D$에 대한 완전자취 해 (entire solution path) 형식의 알고리즘은 문제 (3.1)의 제곱 손실함수 (loss function)에서만 유효하다 (Tibshirani와 Talyor, 2011). Zhu (2017)는 ADMM을 활용하여 일반화 lasso 문제의 복잡도를 개선하였다.

***예제 3.2*** [Trend filtering] fused lasso를 비선형 함수추정에 적용하여 조각별 선형 (piecewise linear)인 추세 (trend)를 평활 스플라인 (smoothing spline)으로 추정하는 방법을 고려하자. Tibshirani (2014)는 평활 스플라인 방법과 거의 동일한 효과를 가지는 벌점화 방법론에 근거한 적응형 추세방법을 제안하였다.

$$\min_{\beta\in\mathbb{R}^n}\frac{1}{2}\sum_{i=1}^{n}(y_i-\beta_i)^2+\lambda_k\|D^k\beta\|_1, \tag{3.2}$$

여기서, $\lambda_k$는 고차의 모수들간의 선형제약 대비행렬 $D^k=\underbrace{D\times D\cdots\times D}_{k}$에 상응하는 조율모수이다. 보조변수 $\gamma$를 $\gamma=D^k\beta$로 두고 ADMM을 적용할 수 있다. Ramdas와 Tibshirani (2016)는 추세를 조각별 선형으로 평활하는 방법과 고차의 선형제약 대비행렬에 관한 보조변수를 설정하는 새로운 방안을 제시함으로써 효율적인 ADMM 알고리즘을 개발하였다. 한편 이와 유사하게 회귀계수의 순서제약식을 다루는 등위회귀분석 (isotonic regression)에서 경우에도 적용할 수 있다 (Tibshirani 등, 2011).

***예제 3.3*** [Network 벌점화] 모든 노드들이 연결되어 있는 가장 복잡한 그래프 구조 (complete network)를 가진다고 하자. 그러면 회귀분석에서 다음의 벌점을 생각할 수 있다 (Hallac 등, 2015; Jeon과 Choi, 2016; Jeon 등, 2017).

$$\min_{\beta\in\mathbb{R}^p}\sum_{i=1}^{n}(y_i-\mathbf{x}_i^T\beta)^2+\lambda\sum_{i<j}|\beta_i-\beta_j|. \tag{3.3}$$

노드 (vortex)와 이들의 연결 (edge)을 가진 그래프 $G=\{V,E\}$를 고려하자. 그러면 변수 첨자 $i$와 $j$를 노드라고 하고, 두 노드 $(i,j)$사이의 연결강도에 fused 벌점을 적용할 수 있다. 따라서 벌점을 그래프상에서 정의된 $\sum_{(i,j)\in G}|\beta_i-\beta_j|$와 같이 정의할 수 있다. 이를 네트웍 벌점이라 부른다. 따라서 노드사이에 연결이 있는 $D$의 원소에만 -1과 1의 값을 적절히 대비시키면 된다. 만약 complete 네트웍인 경우, $D$는 $p(p-1)/2\times p$행렬로 행별 합은 0이면서, 첨자들의 쌍별 조합 ($\{(i,j)|1\le i<j\le p\}$)의 두 노드에만 -1과 1의 값을 가진다.

***예제 3.4*** [행렬 차수/구조 벌점화] 주성분 분석, 요인분석, nonnegative matrix factorization 등 자료행렬을 분해하거나 저차수 행렬 (low rank)을 추정하는 방법론 분야에서, 조정화를 통해 행렬의 차수를 줄이는 방법을 살펴보자. 회귀계수가 행렬을 이루는 경우 행렬의 차수를 줄이기 위해 특이치 (singular value)를 축소추정 (shrinkage estimation)할 수 있다. 양(반)정치 (positive (semi) definite) 행렬 $X$의 특이치를 $\sigma_1\ge\sigma_2\ge\cdots\ge\sigma_p\ge 0$라 하자. 그러면 특이치의 합 (nuclear norm)인 $\sum_k\sigma_k$을 벌점함수로 한, 다음의 문제를 고려할 수 있다.

$$\min_{\Theta\in\mathbb{R}^{n\times p}}\frac{1}{2}\|X-\Theta\|_F^2+\lambda\|\Theta\|_* \tag{3.4}$$

여기서, $\|\cdot\|_F^2$는 Frobenius 노음의 제곱으로 행렬원소의 제곱합이며, $\|\Theta\|_*=\sum_k\sigma_k$는 모수 행렬 $\Theta$의 특이치의 합이다. 이 경우 보조변수 행렬 $\Gamma$를 도입하여 $\Theta=\Gamma$로 놓으면 원시문제에서 $\Theta$와 $\Gamma$의 최적화가 가법분해가 된다. ADMM 알고리즘의 두번째 업데이트 (2.5)에서, 행렬에 proximal 작용소를 적용하면 $\Gamma$의 특이치에 대한 약분계 작용소로 간단히 구해진다 (Xu 등, 2012; Parekh과 Selesnick, 2017).

***예제 3.5*** [볼록 군집분석] 볼록 군집분석 (convex clustering) 방법을 살펴보자. 자료행렬을 $X=(x_{ij}), i=1,2,\cdots,n, j=1,\cdots,p$로, 모수행렬을 $\Theta=(\theta_1^T;\theta_2^T;\cdots;\theta_n^T), \theta_i=(\theta_{ij})^T$로 두자. Chi

and Lan (2015)는 쌍별 규합벌점함수 (pairwise fused penalty)를 활용한 다음의 볼록 군집문제에서 ADMM을 활용하여 최적화하는 방법을 제안하였다.

$$\min_{\Theta\in\mathbb{R}^{n\times p}} \frac{1}{2}\|X-\Theta\|_F^2 + \lambda\sum_{i<j}\|\theta_i-\theta_j\|_q, q=1,2,\infty, \tag{3.5}$$

여기서 $\theta_i$는 모수 행렬 $\Theta$의 $i$번째 행벡터이다. 예를 들어, $q=2$에 대해서 살펴보면 $\|\theta_i-\theta_j\|_2$는 두 관측치 $i$와 $j$의 모수벡터들간의 차이에 대한 $l_2$ 노음 벌점, 즉, group lasso (Yuan과 Lin, 2006) 벌점이 된다. 이 경우 $n$차원의 보조변수 벡터를 $\gamma_{ij}=\theta_i-\theta_j$로 두면 ADMM을 적용할 수 있다. 다변량 이진 (multivariate binary) 자료나 박스그림 (box-plot), 히스토그램 (histogram) 등의 심볼릭 (symbolic) 형태의 자료에도 볼록 군집방법을 적용할 수 있다 (Choi와 Park, 2016; Choi와 Lee, 2017).

***예제 3.6*** [overlapping 벌점] 회귀계수들에 중복벌점을 부여한 경우를 살펴보자. 예를 들어, 두 회귀계수 $\beta_1$과 $\beta_2$에 대해 복합벌점 (composite absolute penalty) $\sqrt{\beta_1^2+\beta_2^2}+|\beta_2|$을 고려해보자 (Zhao 등, 2009). 이 벌점은 그래디언트의 구조적인 성질에 의해 $\beta_2\neq 0$이면 $\beta_1\neq 0$이 되도록 유도한다. 교호작용항을 고려한 회귀분석을 예로 들면, 일반적으로 교호작용항이 유의하면 그에 상응하는 주효과항은 개별 유의성 검정에서 유의하지 않아도 모형에 포함시킨다 (strong heredity principle). 만약 $\beta_2$를 교호작용항, $\beta_1$을 주효과항의 회귀계수라고 하면 전술한 복합벌점은 '$\beta_2\neq 0$이면 $\beta_1\neq 0$' 원칙을 최적화 과정에 반영할 수 있게 된다. 이를 일반화한 다음의 벌점함수를 살펴보자.

$$\sum_v w_v\|\beta_{\mathcal{G}_v}\|_2,$$

여기서, $\mathcal{G}_v$를 변수의 $v$번째 그룹에 속하는 첨자집합이라 하고 $w_v$를 가중치라 놓자. 따라서, 한 변수가 중복된 그룹에 속할 수 있으므로 이러한 $l_2$ 벌점을 overlapping group 벌점이라 부른다. 보다 일반적으로는 구조화된 성김성 유도 노음 (structured sparsity-inducing norm)으로 불리운다. 예를 들어, 변수들의 위계적 관계를 반영한, 예를 들면 계통수 (phylogeny tree)에서는 $\mathcal{G}_v$는 하위 계통에서 상위 계통으로 올라갈수록 그룹 멤버쉽이 중복되는 횟수가 커진다 (Kim 등, 2012). Yu와 Liu (2016)는 잠재요인 (latent factor)에도 overlapping group 벌점을 부여하는 방법을 제안하였다. Yan과 Bien (2015)은 overlapping group 벌점을 효과적으로 연산하는 ADMM을 제안하고 기존 방법의 계산복잡도를 개선하였다.

### 3.2. 모의실험

이 절에서는 Boyd 등 (2010)에서 활용한 모의 생성 자료를 통하여 lasso 알고리즘의 성능을 간략히 살펴보기로 한다. 데이터의 수는 $n=1500, 3000, 4500$이고 변수의 수 $p=5000, 10000, 15000, 20000$이다. 각 변수 $x_j, j=1,\cdots,p$는 표준정규분포를 따른다. 발생후 $x_j^2$이 1이 되게 척도화 (normalize)하였다. 참회귀계수는 $\beta=(\beta_1,\cdots,\beta_p)^T$이고, $p$개의 $\beta_j$들 중에서 100개의 난수를 N(0,1)로부터 임의 추출하여 0이 아닌 값을 부여한다. 종속변수에 대한 선형모형은

$$y_i=\mathbf{x}_i^T\beta+\epsilon_i, i=1,\cdots,n$$

이다. 여기서 $\epsilon_i\sim N(0,10^{-3})$으로 설정한다. 이 모의 실험의 잡음대비 신호의 크기 (signal to noise ratio)는 대략 60으로 매우 크다. $X_j$를 자료행렬 $X$의 $j$번째 열벡터로, 반응변수 벡터를 $y=(y_1,\cdots,y_n)^T$로 이의 평균값을 $\bar{y}=1/n\sum_{i=1}^n y_i$로 놓자. Boyd *et al.*, 2010의 실험과 동일한 환경을 만들기 위해 조율모수값을 $\lambda=0.1\lambda_{\max}$로 고정하였다. 여기서 $\lambda_{\max}=\max_{j=1,\cdots,p}|X_j^T(y-\bar{y})|$로 모든 회귀계수가 0이 될 때의 조율모수 값이다.

모의실험에서는 https://github.com/afbujan/admm_lasso에서 배포하는 python 코드를 기반으로 복잡도가 커지는 상황 ($n, p$가 모두 커지는 상황)을 고려하여 GPGPU (general purpose graphic processing unit) 연산이 가능하도록 재프로그래밍한 코드를 이용하였다. 재프로그래밍의 주요 내용은 주 코드에서 사용한 scipy 패키지의 spsolve 함수와 numpy 패키지의 cholesky 함수를 theano tensor에 의한 연산이 가능하도록 theano의 linear algebra ops인 tensor.slinalg의 solve와 cholesky 함수를 사용할 수 있도록 수정하였다. 이렇게 수정된 코드를 이용하여 여러가지 표본크기에 대하여 모수의 수가 비교적 큰 네가지 상황에 대한 모의실험을 수행하였으며, 실행시간 결과는 Table 3.1에 정리하였다. $p$가 커져도 0이 아닌 회귀계수의 수는 100개로 변하지 않기 때문에 $p$에 대한 복잡도는 선형적으로 증가한다. 반면에 $p$가 상대적으로 클 때, $n$이 증가할수록 복잡도는 더 커짐을 확인할 수 있다.

**Table 3.1** Result of elasped time (in second)

| | $p$ | | | |
|---|---|---|---|---|
| $n$ | 5000 | 10000 | 15000 | 20000 |
| 1500 | 10.2196 | 18.6746 | 29.2402 | 41.6025 |
| 3000 | 24.2095 | 38.1463 | 79.9708 | 102.7723 |
| 4500 | 46.6089 | 62.7049 | 141.4850 | 184.3950 |

수정 코드에서 사용한 코드는 CPU 연산과 병렬연산을 전제로 하지 않은 코드였기 때문에 GPGPU를 사용하는 코드와 연산 속도에 관한 비교는 직접할 수 없다는 문제가 있다. 따라서 결론 및 제언에서 다시 논의하게될 GPGPU 병렬연산이 가능한 프로그래밍을 통해 CPU 병렬처리와의 비교를 수행하는 것이 필요할 것이다.

## 4. 결론 및 제언

분할성 (separability)를 만족하는 문제에서, ADMM은 알고리즘은 원 문제를 작은 부문제로 나누고, proximal 작용소와 gradient 작용소를 활용하여 원시와 쌍대 문제를 교대로 최적화하는 알고리즘이라 할 수 있다. 본고에서는 Boyd 등 (2010), Parikh과 Boyd (2013), Polson 등 (2015) 등의 논문을 기초로 하여 ADMM 알고리즘을 리뷰하고, 조정화방법론 분야에서 개발된 다양한 방법론들을 살펴보았다.

ADMM방법은 일반적인 가정에서 수렴하는 것으로 알려져 있다. 그러나 최적화가 어려운 문제를 비교적 쉽게 처리할 수 있으나, 수렴속도는 $O(1/t)$로 다소 느리다. 이 속도를 개선하기 위한 다양한 방안이 연구되고 있다. 예를 들면, 선형화 ADMM (linearized ADMM; Zhang 등, 2010; Zhang 등, 2011) 혹은 증강 ADMM (augmented ADMM; Zhu, 2017)은 primal 업데이트에서 MM 알고리즘 (Lange, 2016)을 활용한다. 이를 통해 $p > n$ 데이터구조의 일반화된 lasso의 학습에서 효율적임을 실증하였다. 실제 문제에 적용함에 있어, ADMM이 가장 큰 문제점은 $\rho$값에 대한 해의 민감성 문제이다. 최근의 연구동향은 $\rho$값의 의존성을 줄이는 방안으로 고정된 상수보다는 데이터에 적응하여, 반복중에 변동함으로써 알고리즘의 효율성을 높이기 위한 연구가 진행되고 있다 (Xu 등, 2017). 한편, 응용분야로, 지리 등의 공간적인 정보 (Hwang과 Shim, 2017)나 기능성 자기 공명 영상 (functional magnetic resonance imaging) 데이터 등 시공간 인접정보로부터 함수를 추정하는 분야에도 활용될 수 있으로 기대된다.

본고에서는 다루지 못하였으나 theano에서 멀티 GPU를 사용하여 병렬처리를 하기 위해서는, 주어진 문제를 분할하고 각각의 GPU에서 분할된 문제를 계산한 후에, 다시 합치는 과정을 별도로 프로그램해야 한다. Figure 4.1은 이러한 과정을 보여준다. 먼저 CPU와 두 개의 GPU에서 공유해야 하는 데이터 (shared data)를 결정한다. 이 데이터는 일반적으로 CPU 및 각각의 GPU에서 이루어지는 계산에 공통적으로 필요한 데이터이다. 병렬처리의 첫 단계 (①)로 Figure 4.1과 같이 CPU 영역, 즉 주메모리로

부터 각 GPU의 메모리로 이 공유 데이터들을 전송한다. theano에서 이와 같은 공유 데이터는 공유 변수 (shared variable)로 처리된다.

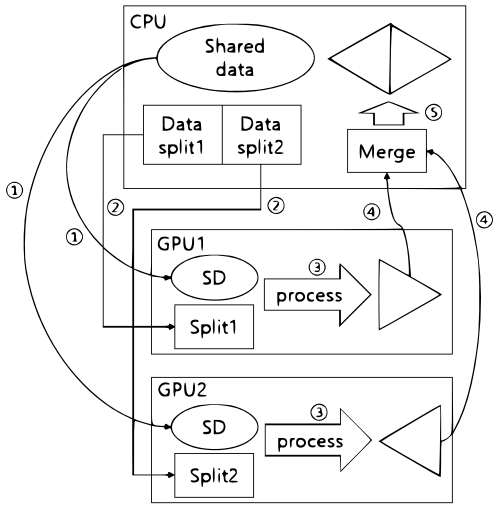


**Figure 4.1** Process for GPU computation using theano

동시에 주어진 데이터로부터 각각의 GPU에서 개별적으로 계산될 데이터 (data split)들을 분할한다. 이 데이터는 서로 의존적이지 않아야 하며 계산 후에 병합을 통해 완전한 결과를 만들어낼 수 있어야 한다. Figure 4.1에서 두번째 단계 (②)로 이러한 분할 데이터들을 CPU로부터 각 GPU로 전송한다. 각 GPU에서는 분할 데이터에 대해 공유 데이터를 기반으로 계산을 수행한다 (③). 이와 같이 각 GPU에서 분할 데이터에 대해 연산을 수행하는 부분은, 쓰레드를 이용하여 하위프로세스 (sub-process)로 구현한다. 즉 멀티쓰레드를 이용하여 두 하위프로세스가 병렬로 동시에 수행되게 함으로써 수행시간을 절감하게 된다. 각 GPU에서 계산된 결과는 병합을 위해 CPU 영역으로 전송되고 (④), CPU 연산에 의해 병합 (⑤)되어 완전한 결과가 된다. theano에서 이를 구현하기 위해서는 CPU 모드를 기본으로 하고, 각 GPU를 각각의 디바이스로 설정한 후, 하위프로세스 별로 디바이스를 선택하여 연산을 수행하도록 프로그램하여야 한다. 병렬처리가 병렬처리를 사용하지 않는 경우보다 효과적이 되기 위해서는, CPU에서의 병합에 소요되는 시간 혹은 자원과 CPU와 GPU 간의 데이터 전송에 소모되는 시간 혹은 자원에 비해 병렬처리를 통해 절약되는 시간 혹은 자원이 더 많아야만 하기 때문에, 이를 면밀하게 설계할 필요가 있으며, 이는 향후 과제로 남기고자 한다.

## References


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

Bertsekas, D. P. (2003). *Nonlinear Programming.* 2nd edition, Athena Scientific.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Optimization*, **3**, 1-122.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization.* Cambridge University Press.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *Annals of Applied Statistics*, **12**, 719-752.

Chi, E. C. and Lange, K. (2015). Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, **24**, 994-1013.

Choi, H., Koo, J.-Y., and Park, C. (2015). Fused least absolute shrinkage and selection operator for credit scoring. *Journal of Statistical Computation and Simulation*, **85**, 2135-2147.

Choi, H. and Lee, S. (2017). *Convex clustering for binary data.* Technical Report.

Choi, H. and Park, C. (2016). Clustering analysis of particulate matter data using shrinkage boxplot. *Journal of the Korean Data Analysis Society*, **18**, 2435-2443.

Choi, H., Park, H., and Park, C. (2013). Support vector machines for big data analysis. *Journal of the Korean data & Information Science Society*, **24**, 989-998.

Davis, D. and Wotao, Y. (2016). Convergence rate analysis of several splitting schemes, *Splitting methods in communication, imaging, science, and engineering*, Springer International Publishing, 115-163.

Fang, E. X., He, B., Liu, H., and Yuan, X. (2015). Generalized alteranting direction method of multipliers: new theoretical insights and applications. *Mathematical Programming Computation*, **7**, 149-187.

Forero, P. A., Cano, A., and Giannakis, G. B. (2010). Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, **6**, 2873-2898.

Hastie, T., Tibshirani, and R. Wainwright, M. (2016). *Statistical learning with sparsity: The lasso and generalizations*, CRC Press.

Hallac, D., Leskovec, J., and Boyd, S. (2015). Network lasso: Clustering and optimization in large graphs. *Proceeding KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387-396.

Hwang, C. and Shim, J. (2017). Geographically weighted least squares-support vector machine. *Journal of the Korean Data & Information Science Society*, **28**, 227-235.

He, B. and Yuan, X. (2012). On the o(1/n) convergence rate of teh Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, **50**, 700-709.

Jeon, J. and Choi, H. (2016). The sparse Luce model. *Applied Intelligence*, `https://doi.org/10.1007/s10489-016-0861-4`, In press.

Jeon, J., Kwon, S., and Choi, H. (2017). Homogeneity detection for the high-dimensional generalized linear model. *Computational Statistics and Data Analysis*, **114**, 61-74.

Kim, S. and Xing, E. P. (2012). Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping. *The Annals of Applied Statistics*, **6**, 1095-1117.

Lange, K. (2016). *MM optimization algorithms.* SIAM-Society for Industrial and Applied Mathematics.

Parekh, A. and Selesnick, I. W. (2017). Improved sparse low-rank matrix estimation. *arXiv:1605.00042v2.*

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization.* **1**, 123-231.

Park, C., Kim, Y., Kim, J., Song, J., and Choi, H. (2015). *Data mining using R.* 2nd edition, Kyowoosa.

Polson, N. G., Scott, J. G., and Willard, B. T. (2015). Proximal algorithms in statistics and machine learning. *Statistical Science*, **30**, 559-581.

Ramdas, A. and Tibshirani, R. (2016). Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, **25**, 839-858.

Taylor, G., Burmeister, R., Xu, Z., Singh, B., Patel, A., Goldstein, T. (2016). Training neural networks without gradients: A scalable ADMM approach. *CoRR, arXiv1605.02026.*

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. (2011). Nearly-Isotonic regression. *Technometrics*, **53**, 54-61.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, **67**, 91-108.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, **39**, 1335-1371.

Tibshirani, R. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, **42**, 285-323.

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016). Learning structured sparsity in deep neural networks. *In Neural Information Processing Systems*, 2074-2082.

Xu, Z., Taylor, G., Li, H., Figueiredo, M., Yuan, X., and Goldstein, T. (2017). Adaptive consensus ADMM for distributed optimization. *arXiv:1706.02869v2*.

Xu, Y., Yin, W., Wen, Z., and Zhang, Y. (2012). An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, **7**, 365-384.

Yang, Y., Sun, J., Li, H., and Xu, Z. (2017). ADMM-Net: A deep learning approach for compressive sensing MRI. *CoRR, arXiv:1705.06869*.

Yin, W., Osher, S., Goldfarb, D., and Darbon, J. (2008). Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, **1**, 143-168.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, **68**, 49-67.

Yan, X. and Bien, J. (2015). *Hierarchical sparse modeling: A choice of two group lasso formulations*. Technical Report.

Yu, G. and Liu, Y. (2016). Sparse regression incorporating graphical structure among predictors. *Journal of the American Statistical Association*, **111**, 707-720.

Zhang, X., Burger, M., Bresson, X., and Osher, S. (2010). Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal of Imaging Science*, **3**, 253-276.

Zhang, X., Burger, M., and Osher, S. (2011). A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, **46**, 20-46.

Zhao, P., Rocha, G., and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, **37**, 3468-3497.

Zhu, Y. (2017). An augmented ADMM algorithm with application to the generalized lasso Problem. *Journal of Computational and Graphical Statistics*, **26**, 195-204.

*Journal of the Korean Data & Information Science Society*
*2017,* **28**(6), 1229–1244

http://dx.doi.org/10.7465/jkdi.2017.28.6.1229
한국데이터정보과학회지

# ADMM algorithms in statistics and machine learning

Hosik Choi[1] · Hyunjip Choi[2] · Sangun Park[3]

[12]Department of Applied Statistics, Kyonggi University
[3]Department of Management Information System, Kyonggi University

Received 31 October 2017, revised 14 November 2017, accepted 21 November 2017

**Abstract**

In recent years, as demand for data-based analytical methodologies increases in various fields, optimization methods have been developed to handle them. In particular, various constraints required for problems in statistics and machine learning can be solved by convex optimization. Alternating direction method of multipliers (ADMM) can effectively deal with linear constraints, and it can be effectively used as a parallel optimization algorithm. ADMM is an approximation algorithm that solves complex original problems by dividing and combining the partial problems that are easier to optimize than original problems. It is useful for optimizing non-smooth or composite objective functions. It is widely used in statistical and machine learning because it can systematically construct algorithms based on dual theory and proximal operator. In this paper, we will examine applications of ADMM algorithm in various fields related to statistics, and focus on two major points: (1) splitting strategy of objective function, and (2) role of the proximal operator in explaining the Lagrangian method and its dual problem. In this case, we introduce methodologies that utilize regularization. Simulation results are presented to demonstrate effectiveness of the lasso.

*Keywords*: Constraint, optimization, parallel computing, penalty function, regularization.

[1] Corresponding author: Assistant professor, Department of Applied Statistics, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon 16227, Korea. E-mail: choi.hosik@gmail.com
[2] Professor, Department of Applied Statistics, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon 16227, Korea.
[3] Associate professor, Department of Management Information System, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon 16227, Korea.