THE UNIVERSITY OF TOKYO

数理情報工学特論第一
【機械学習とデータマイニング】
4章：教師なし学習③

かしま　ひさし
鹿島 久嗣
（数理 6 研）
kashima@mist.i.～

東京大学
THE UNIVERSITY OF TOKYO

DEPARTMENT OF MATHEMATICAL INFORMATICS

グラフィカルモデルについて学びます

- グラフィカルモデル
- グラフィカルラッソ
- グラフィカルラッソの推定アルゴリズム

グラフィカルモデル

教師なし学習の主要タスクは4つあるのでした

- 教師なし学習においては通常、データ上の確率分布$P(\phi(x))$を何らかの形で推定することが行われる
- 確率分布の使い道としては主に以下の4つが挙げられる：
  - **タスク1: 確率分布そのものを用いた分析** （← 今日はコレ）
  - タスク2: データの確率評価
  - タスク3: 未観測値（欠損値）の推定 （←前々回）
  - タスク4: 潜在変数の推定 （←前回）

今回は、データの特徴ベクトルの要素間の関係を調べるために、精度行列が疎であるような多次元確率分布を考えます

- ４つのタスクのひとつ「確率分布そのものを用いた分析」では、確率分布そのものを用いてデータに対する知見を得ることが目的
  1. パラメータ（もしくは確率分布$P(\phi(x))$の形）を「鑑賞」することによって、データ全体がどのような形で分布しているかを知る
  2. 2つのデータ集合から推定された2つの確率分布が異なるか、異なるとしたらどのように異なるのかを調べる
- 多次元正規分布モデルの場合、精度行列Λの各要素は、任意の2要素の積にかかる係数であるから、要素間の直接的な関係を表している

$$P(\phi(x);\boldsymbol{\mu},\boldsymbol{\Lambda}) \equiv \frac{1}{(2\pi)^{D/2}}|\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\phi(x)-\boldsymbol{\mu})^{\top}\boldsymbol{\Lambda}(\phi(x)-\boldsymbol{\mu})\right)$$

- この要素間の関係は、精度行列の非零要素の数、つまり、直接的な関係の数が少ない方が、解釈が容易になる
- そのため、今回は、精度行列が疎（多くの要素が0）で、要素間の関係がよりはっきりと示されているような、多次元正規分布を扱う

一般に、パラメータがグラフ構造（疎な構造）をもつと解釈できるようなモデルをグラフィカルモデルと呼びます

- （多次元正規分布に限らず）一般に、疎なパラメータ構造をもち、その構造がある種のグラフ構造として解釈できるような確率モデルを**グラフィカルモデル**と呼ぶ
  - 連続の場合の代表的モデル：
    - **グラフィカルラッソ**（疎な多次元正規分布）
  - 離散の場合の代表的モデル：
    - ベイジアンネットワーク
- 広い意味では、教師つき学習のところで紹介した条件付き確率場（CRF）もグラフィカルモデルの範疇に入る

グラフィカルラッソ

精度行列の最尤推定量は通常「密」になります

- 多次元正規分布の精度行列を観測されたデータからの最尤推定する方法については以前述べた
- たとえ、真の精度行列が疎である場合でも、通常は推定された精度行列は密になってしまう
- 精度行列 $\boldsymbol{\Lambda}$ を用いた多次元正規分布の密度関数：

$$P(\boldsymbol{\phi}(x); \boldsymbol{\mu}, \boldsymbol{\Lambda}) \equiv \frac{1}{(2\pi)^{D/2}} |\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\phi}(x) - \boldsymbol{\mu})^\top \boldsymbol{\Lambda} (\boldsymbol{\phi}(x) - \boldsymbol{\mu})\right)$$

- パラメータ $\boldsymbol{\mu}$ および $\boldsymbol{\Lambda}$ を、訓練データ集合 $\{x^{(i)}\}_{i=1}^{N}$ から最尤推定によって求めることにすると、その最適化問題は：

$$(\boldsymbol{\mu}^*, \boldsymbol{\Lambda}^*) \equiv \underset{(\boldsymbol{\mu}, \boldsymbol{\Lambda})}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \log P(\boldsymbol{\phi}(x^{(i)}); \boldsymbol{\mu}, \boldsymbol{\Lambda})$$

- このままでは精度行列の推定量 $\boldsymbol{\Lambda}^*$ が（$=\boldsymbol{\Sigma}^{*-1}$）密行列になってしまう

$$\boldsymbol{\Sigma}^* = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{\phi}(x^{(i)}) - \boldsymbol{\mu})(\boldsymbol{\phi}(x^{(i)}) - \boldsymbol{\mu})^\top$$

精度行列の推定量を「疎」にするために、L_1正則化を行ったものをグラフィカルラッソと呼びます

- パラメータ μ および Λ を、訓練データ集合$\{x^{(i)}\}_{i=1}^{N}$から最尤推定によって求めることにすると、その最適化問題は：

$$(\mu^*, \Lambda^*) \equiv \underset{(\mu,\Lambda)}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \log P(\phi(x^{(i)}); \mu, \Lambda)$$

  - このままでは、精度行列の推定量Λ^*が密行列になってしまう

- 精度行列の推定量 Λ^* を疎にするために、精度行列についてL_1正則化を行うことにすると、最適化問題は：

$$(\mu^*, \Lambda^*) \equiv \underset{(\mu,\Lambda)}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \log P(\phi(x^{(i)}); \mu, \Lambda) - \lambda \| \Lambda \|_1$$

  - $\| \Lambda \|_1$は行列のL_1-ノルムであり、以下のように各要素の絶対値の和

$$\| \Lambda \|_1 \equiv \sum_{k,\ell=1}^{D} \left| [\Lambda]_{k,\ell} \right|$$

L_1正則化を行うと、精度行列の推定量は閉じた形では求まりません

- 平均パラメータの推定量 $\boldsymbol{\mu}^*$ については正則化項が存在しないため、最尤推定の場合と同じになり、その推定量は：

$$\boldsymbol{\mu}^* = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\phi}(x^{(i)})$$

  - 全データの特徴ベクトルの平均

- 一方で、精度行列のほうはL_1正則化項の存在により、平均パラメータのように閉じた形では求まらない
- 従って、グラフィカルラッソの本質は、いかにして精度行列の推定を行うかということになる

グラフィカルラッソの目的関数はシンプルに書けます

- $\boldsymbol{\Lambda}$ についてのみ最適化することにして、多次元正規分布の密度関数の定義を代入して書き下すと：

$$\boldsymbol{\Lambda}^* \equiv \underset{\boldsymbol{\Lambda}}{\text{argmax}} \ \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{\phi}(x)-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\phi}(x)-\boldsymbol{\mu}) - \lambda \parallel \boldsymbol{\Lambda} \parallel_1$$

- この最適化問題の目的関数は以下のように書き換えられる：

$$J(\boldsymbol{\Lambda}) \equiv \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\mathrm{Tr}\boldsymbol{\Lambda}\mathbf{S} - \lambda \parallel \boldsymbol{\Lambda} \parallel_1$$

  - ここで、以下を使った：
    - $(\boldsymbol{\phi}(x)-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\phi}(x)-\boldsymbol{\mu}) = \mathrm{Tr}(\boldsymbol{\phi}(x)-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{\phi}(x)-\boldsymbol{\mu})$
      $= \mathrm{Tr}\boldsymbol{\Lambda}(\boldsymbol{\phi}(x)-\boldsymbol{\mu})(\boldsymbol{\phi}(x)-\boldsymbol{\mu})^\top$
    - 標本分散共分散行列 $\mathbf{S}$：

$$\mathbf{S} \equiv \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\phi}(x)-\boldsymbol{\mu})(\boldsymbol{\phi}(x)-\boldsymbol{\mu})^\top$$

グラフィカルラッソのアルゴリズム

まずは目的関数を精度行列で微分します

- グラフィカルラッソの目的関数を最大化する $\boldsymbol{\Lambda}$ を求めたい

$$J(\boldsymbol{\Lambda}) \equiv \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\mathrm{Tr}\boldsymbol{\Lambda}\mathbf{S} - \lambda \parallel \boldsymbol{\Lambda} \parallel_1$$

- 実は、この最適化問題は精度行列 $\boldsymbol{\Lambda}$ についてではなく、その逆行列である分散共分散行列 $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$ について解くことになる

- この目的関数を$\boldsymbol{\Lambda}$の各(k,l)-要素$[\boldsymbol{\Lambda}]_{k,l}$で微分すると：

$$\frac{\partial J(\boldsymbol{\Lambda})}{\partial [\boldsymbol{\Lambda}]_{k,\ell}} = \begin{cases} \frac{1}{2}[\boldsymbol{\Sigma} - \mathbf{S}]_{k,\ell} - \lambda & (\text{if } [\boldsymbol{\Lambda}]_{k,\ell} > 0) \\ \frac{1}{2}[\boldsymbol{\Sigma} - \mathbf{S}]_{k,\ell} + \lambda & (\text{if } [\boldsymbol{\Lambda}]_{k,\ell} < 0) \\ \text{undefined} & (\text{if } [\boldsymbol{\Lambda}]_{k,\ell} = 0) \end{cases}$$

- ここで行列の微分の公式をもちいた：

$$\frac{\partial \log|\boldsymbol{\Lambda}|}{\partial \boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma} \qquad \frac{\partial \mathrm{Tr}\boldsymbol{\Lambda}\mathbf{S}}{\partial \boldsymbol{\Lambda}} = \mathbf{S}$$

- $\boldsymbol{\Lambda}$と$\boldsymbol{\Sigma}$の逆行列を通じた等価性から、↑の式において($\boldsymbol{\Lambda}$についての微分)=0の成立は、対応する$\boldsymbol{\Sigma}$が最適解であることも意味する

対角成分は閉じた形で求まります

- まず、微分を $\boldsymbol{\Lambda}$ の対角成分についてみてみる
- 精度行列 $\boldsymbol{\Lambda}$ は正定行列であることに注意すると、その対角成分は必ず正、つまり$[\boldsymbol{\Lambda}]_{k,k}>0$であることから、微分の対角成分は：

$$\frac{\partial J(\boldsymbol{\Lambda})}{\partial [\boldsymbol{\Lambda}]_{k,k}} = \frac{1}{2}[\boldsymbol{\Sigma} - \mathbf{S}]_{k,k} - \lambda$$

- これを0と置くと、分散共分散行列 $\boldsymbol{\Sigma}$ の対角成分を閉じた形で得る：

$$[\boldsymbol{\Sigma}]_{k,k} = [\mathbf{S}]_{k,k} + 2\lambda$$

- これで分散共分散行列 $\boldsymbol{\Sigma}$ の対角成分については最適解が求まった
- 今後は $\boldsymbol{\Sigma}$ の非対角成分についてのみの最適化を考えればよい

非対角成分は、閉じた形で求まりませんが、実は、L_1正則化回帰（の繰り返し）に帰着されます

- 分散共分散行列 $\boldsymbol{\Sigma}$ の対角成分以外の成分の最適解を求める
  - 精度行列 $\boldsymbol{\Lambda}$ の最適解は $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ の関係によって自動的に定まる
- しかし、対角成分全てについての最適化は難しい
- そこで、分散共分散行列のある行（対称なので、ある列としても同じ）のみについての最適化を行い、これを選ぶ行を変えながら繰り返すという戦略をとることにする
- これから示すように、実は、分散共分散行列のある行（列）についての最適化は、回帰問題のときに紹介したL_1正則化回帰に帰着されることがわかる

分散共分散行列を部分的に最適化するため、各行列の分割を考えます

- 分散共分散行列を部分的に最適化するため、分散共分散行列の分割を考える：

$$\boldsymbol{\Sigma} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \tilde{\boldsymbol{\sigma}} \\ \tilde{\boldsymbol{\sigma}}^\top & \sigma_D \end{bmatrix}$$

  - $\tilde{\boldsymbol{\Sigma}}$ は $(D\text{-}1) \times (D\text{-}1)$の行列、$\tilde{\boldsymbol{\sigma}}$ は長さD-1のベクトル、σ_D はスカラー
- $\boldsymbol{\Sigma}$ の最後の行（列）について最適化するとすれば、$\tilde{\boldsymbol{\Sigma}}$ は定数であり、σ_D は対角成分なので既に最適解が求まっており、定数とみなせる
- 従って、ここでパラメータは $\tilde{\boldsymbol{\sigma}}$ となる
- $\boldsymbol{\Sigma}$と同様に、精度行列と標本分散共分散行列の分割をそれぞれ：

$$\boldsymbol{\Lambda} = \begin{bmatrix} \tilde{\boldsymbol{\Lambda}} & \tilde{\boldsymbol{\lambda}} \\ \tilde{\boldsymbol{\lambda}}^\top & \lambda_D \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \tilde{\mathbf{S}} & \tilde{\mathbf{s}} \\ \tilde{\mathbf{s}}^\top & s_D \end{bmatrix}$$

分散共分散行列と精度行列の関係式がブロックごとに導かれます

- $\boldsymbol{\Sigma}\,\boldsymbol{\Lambda} = \mathbf{I}$ であることから、分割した行列の各ブロックについて：

$$\begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \tilde{\boldsymbol{\sigma}} \\ \tilde{\boldsymbol{\sigma}}^\top & \sigma_D \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Lambda}} & \tilde{\boldsymbol{\lambda}} \\ \tilde{\boldsymbol{\lambda}}^\top & \lambda_D \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\Lambda}} + \tilde{\boldsymbol{\sigma}}\tilde{\boldsymbol{\lambda}}^\top & \tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\lambda}} + \lambda_D\tilde{\boldsymbol{\sigma}} \\ \tilde{\boldsymbol{\sigma}}^\top\tilde{\boldsymbol{\Lambda}} + \sigma_D\tilde{\boldsymbol{\lambda}}^\top & \tilde{\boldsymbol{\sigma}}^\top\tilde{\boldsymbol{\lambda}} + \sigma_D\lambda_D \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

- 特に、右上のブロックから：$\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\lambda}} + \lambda_D\tilde{\boldsymbol{\sigma}} = 0$
- すなわち：$\tilde{\boldsymbol{\lambda}} = -\lambda_D\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\sigma}}$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \tilde{\boldsymbol{\sigma}} \\ \tilde{\boldsymbol{\sigma}}^\top & \sigma_D \end{bmatrix} \qquad \boldsymbol{\Lambda} = \begin{bmatrix} \tilde{\boldsymbol{\Lambda}} & \tilde{\boldsymbol{\lambda}} \\ \tilde{\boldsymbol{\lambda}}^\top & \lambda_D \end{bmatrix}$$

分散共分散行列の現在注目しているブロックが最適解になる条件を導きます

- 先の微分の $\tilde{\boldsymbol{\lambda}}$（精度行列の現在注目しているブロック）に関連する部分だけを取り出すと：

$$\frac{\partial J(\boldsymbol{\Lambda})}{\partial [\tilde{\boldsymbol{\lambda}}]_k} = \begin{cases} \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k - \lambda & (\text{if } [\tilde{\boldsymbol{\lambda}}]_k > 0) \\ \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k + \lambda & (\text{if } [\tilde{\boldsymbol{\lambda}}]_k < 0) \\ \text{undefined} & (\text{if } [\tilde{\boldsymbol{\lambda}}]_k = 0) \end{cases}$$

- となっているので、ここから $\tilde{\boldsymbol{\lambda}}$ を $\tilde{\boldsymbol{\lambda}} = -\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}$（前頁）を用いて消去すれば：

$$\frac{\partial J(\boldsymbol{\Lambda})}{\partial [\tilde{\boldsymbol{\lambda}}]_k} = \begin{cases} \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k - \lambda & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k > 0) \\ \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k + \lambda & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k < 0) \\ \text{undefined} & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k = 0) \end{cases}$$

- 結果、微分が、$\tilde{\boldsymbol{\sigma}}$（と$\lambda_D$）によって表される
  - これが =0 になるのが最適解の条件
  - しかし、このままでは、場合分けの条件式がややこしい…

パラメータを置き換えて、最適解の条件を簡単に書きます

- この微分を簡単にしたい：$$\frac{\partial J(\mathbf{\Lambda})}{\partial [\tilde{\boldsymbol{\lambda}}]_k} = \begin{cases} \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k - \lambda & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k > 0) \\ \frac{1}{2}[\tilde{\boldsymbol{\sigma}} - \tilde{\mathbf{s}}]_k + \lambda & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k < 0) \\ \text{undefined} & (\text{if } [-\lambda_D \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}]_k = 0) \end{cases}$$
- 新たなパラメータβを導入する： $\beta \equiv \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{\sigma}}$
  - つまり： $\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\Sigma}} \beta$
- であること、また、精度行列 $\boldsymbol{\Lambda}$ は正定であるので、その対角成分である λ_D は正であることに注意すると、微分の式は：

$$\frac{\partial J(\mathbf{\Lambda})}{\partial [\tilde{\boldsymbol{\lambda}}]_k} = \begin{cases} \frac{1}{2}[\tilde{\boldsymbol{\Sigma}} \beta - \tilde{\mathbf{s}}]_k - \lambda & (\text{if } [\beta]_k < 0) \\ \frac{1}{2}[\tilde{\boldsymbol{\Sigma}} \beta - \tilde{\mathbf{s}}]_k + \lambda & (\text{if } [\beta]_k > 0) \\ \text{undefined} & (\text{if } [\beta]_k = 0) \end{cases}$$

  - β のみを使って書くことができる。
  - 場合分けの不等式の向きが変わったことに注意する

同じ微分をもつ、別の最適化問題に書き換えます

- 最終的に得られた微分：

$$\frac{\partial J(\mathbf{\Lambda})}{\partial [\tilde{\lambda}]_k} = \begin{cases} \frac{1}{2}[\tilde{\Sigma}\beta - \tilde{\mathrm{s}}]_k - \lambda & (\text{if } [\beta]_k < 0) \\ \frac{1}{2}[\tilde{\Sigma}\beta - \tilde{\mathrm{s}}]_k + \lambda & (\text{if } [\beta]_k > 0) \\ \text{undefined} & (\text{if } [\beta]_k = 0) \end{cases}$$

- これは、以下の目的関数の β についての微分と一致する：

$$\tilde{J}(\beta) \equiv \frac{1}{4}\beta^\top \tilde{\Sigma}\beta - \frac{1}{2}\tilde{\mathrm{s}}^\top \beta + \lambda \parallel \beta \parallel_1$$

- もともとの最大化問題の目的関数を β の関数としてみたものと、この最小化問題の目的関数は同一
  - 連続で、微分不可能な点が同じで、それ以外の場所では微分が一致
- つまり、もともとの目的関数のかわりに、新たな目的関数を用いて β を求めても差し支えないということを意味する

新しい目的関数を、変数ごとに最適化することを考えます

- さらに、目的関数：

$$\tilde{J}(\beta) \equiv \frac{1}{4}\beta^\top \tilde{\Sigma}\beta - \frac{1}{2}\tilde{\mathrm{s}}^\top \beta + \lambda \parallel \beta \parallel_1$$

を、 β の第k成分$[\beta]_k$についてのみ最適化することを考える

- 目的関数 $\tilde{J}(\beta)$ を $[\beta]_k$についての関数だと思って整理すると：

$$\begin{aligned}
\tilde{J}([\beta]_k) \equiv & \frac{1}{4}[\tilde{\Sigma}]_{k,k}[\beta]_k^2 + \frac{1}{2}\sum_{j\neq k}[\beta]_j[\tilde{\Sigma}]_{j,k}[\beta]_k - \frac{1}{2}[\tilde{\mathrm{s}}]_k[\beta]_k + \lambda|[\beta]_k| + const. \\
= & \frac{1}{4}[\tilde{\Sigma}]_{k,k}[\beta]_k^2 - \frac{1}{2}\left([\tilde{\mathrm{s}}]_k - \sum_{j\neq k}[\beta]_j[\tilde{\Sigma}]_{j,k}\right)[\beta]_k + \lambda|[\beta]_k| + const. \\
= & \frac{1}{4}[\tilde{\Sigma}]_{k,k}\left([\beta]_k - \frac{[\tilde{\mathrm{s}}]_k - \sum_{j\neq k}[\beta]_j[\tilde{\Sigma}]_{j,k}}{[\tilde{\Sigma}]_{k,k}}\right)^2 + \lambda|[\beta]_k| + const.
\end{aligned}$$

ちょうどL_1正則化回帰のときに出てきた目的関数と同じ形であることに気付きます

- 結局、$[\beta]_k$の最適解$[\beta]_k^*$は、以下の最適化問題を解くことで求まる

$$[\beta]_k^* \equiv \underset{[\beta]_k}{\operatorname{argmin}} \ \frac{1}{2}\left([\beta]_k - [\tilde{\beta}]_k\right)^2 + \gamma|[\beta]_k|$$

$$\text{where } \gamma \equiv \frac{2\lambda}{[\tilde{\Sigma}]_{k,k}} \text{ and } [\tilde{\beta}]_k \equiv \frac{[\tilde{\mathbf{s}}]_k - \sum_{j \neq k}[\beta]_j[\tilde{\Sigma}]_{j,k}}{[\tilde{\Sigma}]_{k,k}}$$

- これはまさにL_1正則化回帰の次元ごとの逐次解法のときに出てきた最適化問題↓と同じ形をしている

$$w_d^* = \underset{w_d}{\operatorname{argmin}} \ \tilde{L}(w_d)$$

$$\tilde{L}(w_d) \equiv \frac{1}{2}(w_d - \tilde{w}_d)^2 + \gamma|w_d|$$

L_1正則化回帰のときの解を利用して、解を導きます

- L_1正則化回帰の次元ごとの逐次解法のときに出てきた最適化問題：

$$w_d^* = \underset{w_d}{\text{argmin}}\ \tilde{L}(w_d) \text{ where } \tilde{L}(w_d) \equiv \frac{1}{2}(w_d - \tilde{w}_d)^2 + \gamma|w_d|$$

- この解は：

$$w_d^* = \begin{cases} \tilde{w}_d - \gamma & (\text{if } \tilde{w}_d > \gamma) \\ \tilde{w}_d + \gamma & (\text{if } \tilde{w}_d < -\gamma) \\ 0 & (\text{if } -\gamma \leq \tilde{w}_d \leq \gamma) \end{cases}$$

- これを利用すると、いま解きたい最適化問題：

$$[\beta]_k^* \equiv \underset{[\beta]_k}{\text{argmin}}\ \frac{1}{2}\left([\beta]_k - [\tilde{\beta}]_k\right)^2 + \gamma|[\beta]_k|$$

の最適解$[\beta]_k^*$は：

$$[\beta]_k^* = \begin{cases} [\tilde{\beta}]_k - \gamma & (\text{if } [\tilde{\beta}]_k > \gamma) \\ [\tilde{\beta}]_k + \gamma & (\text{if } [\tilde{\beta}]_k < -\gamma) \\ 0 & \text{if}(-\gamma \leq [\tilde{\beta}]_k \leq \gamma) \end{cases}$$

- これをkを変えながら繰り返すことによって、βが求まる

L_1正則化回帰を繰り返すことによって、グラフィカルラッソの推定ができます

- 一旦 β が得られると、$\tilde{\sigma} = \tilde{\Sigma}\beta$ を用いて、$\tilde{\sigma}$ を求めることができる
- この一連の手続きによって Σ の最後の行（列）を求めるのがグラフィカルラッソのアルゴリズムにおける1ステップである
- このステップを、行（列）の順番を入れ替えて最後にくるものを適当に変えながら繰り返し行い、収束するまで繰り返す
  - つまり、L_1正則化回帰問題を繰り返し解く
- グラフィカルラッソのアルゴリズムは：
  - ブロック分割で最後にもってくる行（列）を変えながら、最後の行（列）についての最適化を繰り返す外側のループ
  - 最後の行（列）についての最適化を行うために、β のある次元についての最適化を、選ぶ次元を変えながら繰り返す内側のループ

の2つのループで構成されている

このアルゴリズムは逆行列の計算を必要としないため効率的です

- このアルゴリズムの効率的なところは、$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$ という関係をもつ２つの行列$\boldsymbol{\Sigma}$ と$\boldsymbol{\Lambda}$を扱っていながら、逆行列の計算がまったく出てこないところである
  - $\boldsymbol{\Sigma}\,\boldsymbol{\Lambda} = \mathbf{I}$ をブロック分割した式（右上と右下）および $\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\Sigma}}\beta$ の連立方程式を解くことで $\tilde{\boldsymbol{\sigma}}$ から$\boldsymbol{\Lambda}$ の最後の行（列）が求まる：

$$\lambda_D = \frac{1}{\sigma_D - \tilde{\boldsymbol{\sigma}}^\top \beta} \qquad \tilde{\lambda} = -\frac{\beta}{\sigma_D - \tilde{\boldsymbol{\sigma}}^\top \beta}$$

- また、得られた解は条件$\boldsymbol{\Sigma}\,\boldsymbol{\Lambda} = \mathbf{I}$ を満たすように作られるため、得られる $\boldsymbol{\Sigma}$ と $\boldsymbol{\Lambda}$ は正定であることが保証される