大阪市立大学電子・物理工学特別講義

今日からできる スパースモデリング

京都大学大学院情報学研究科システム科学専攻
大関　真之

圧縮センシング実践編

スパース解推定

- L0ノルム最小化によるスパース解推定
  - 以下の最小化問題でスパース解を探索しよう！

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = A\mathbf{x}$$

    - 非零成分の個数が小さい（スパース）で方程式を満たすものを探す
    - 残念ながら計算量的に困難
- L1ノルム最小化によるスパース解推定
  - 代わりに次の最小化問題でスパース解を探索しよう！

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = A\mathbf{x}$$

    - 零が多ければ小さくなる＋大きさも小さくなりがち
    - 計算量は非常に軽い（Nの3乗程度）

ノルムについて注意

- Lpノルムの定義

$$\|\mathbf{x}\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + \cdots + |x_N|^p}$$

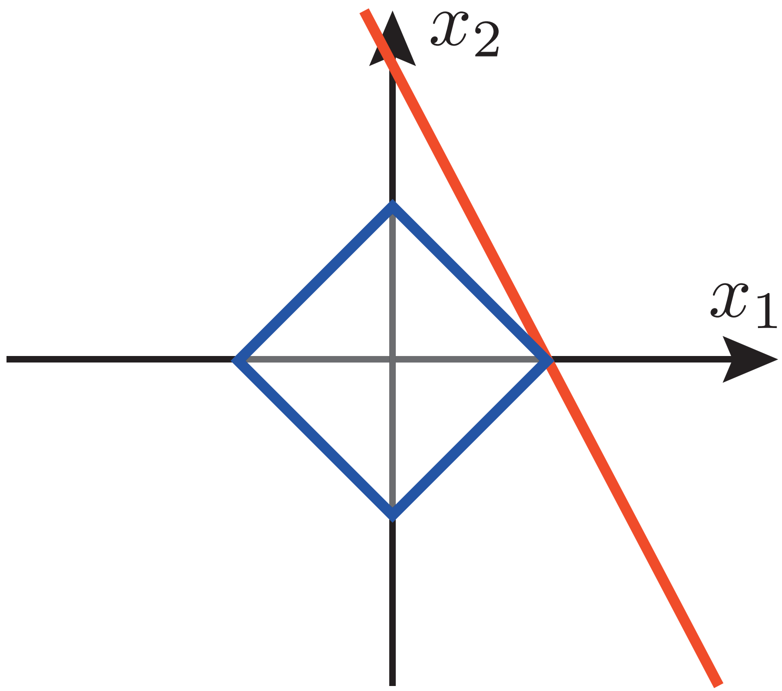
  - L0「ノルム」はちょっと変
    - p->0の極限で定義.
  - L1ノルムは絶対値の和

$$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_N|$$

  - L2ノルムはユークリッド距離

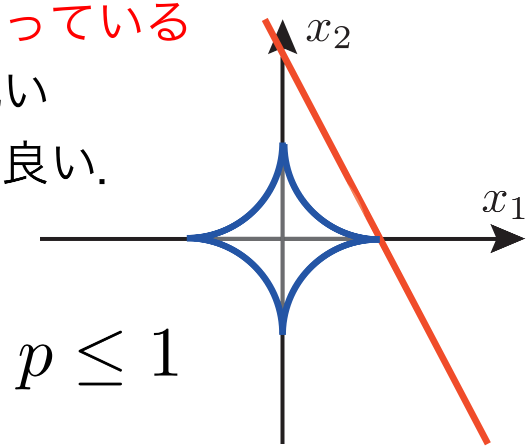
$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_N^2}$$

吟味と注意



- L1ノルム最小化で
スパース解は確かに得られる.
  - L1ノルムは尖っている
  - L2ノルムは丸い
  - Lpノルムでも良い.

$$0 < p \leq 1$$



- L2ノルム最小化でも劣決定系の
方程式の解が得られる.
  - ノルム最小化による解選択
    - L1ノルムはスパース解
    - L2ノルムは大きさが小さい解

基底追跡の性能

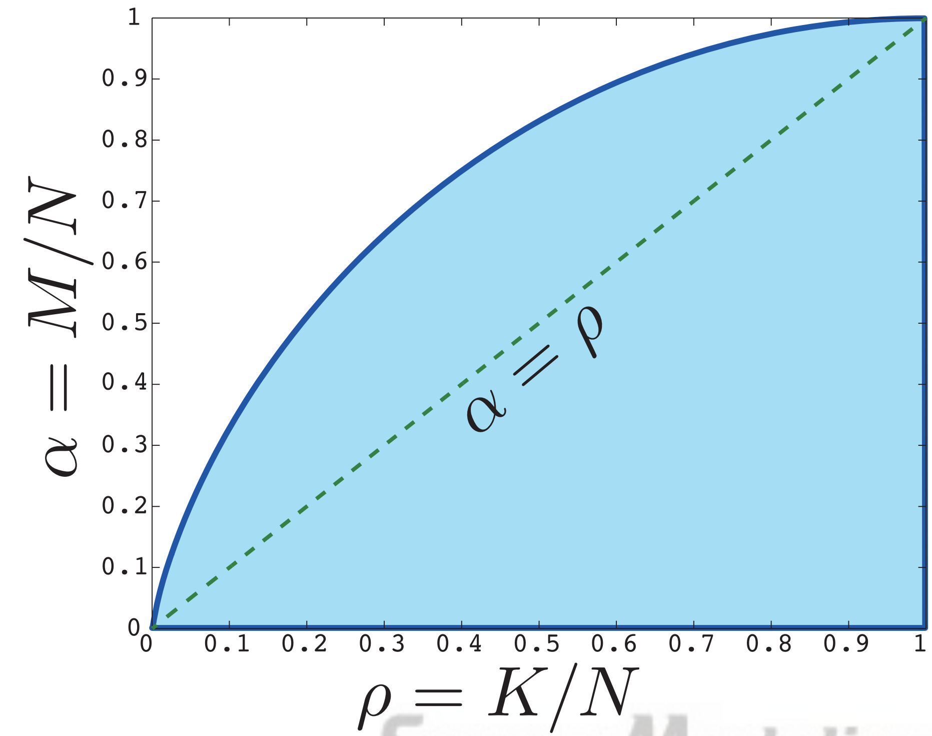
- L1ノルム最小化によるスパース解推定法
  - 基底追跡（Basis Pursuit）

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = A\mathbf{x}$$

$$\frac{1}{\alpha} = 1 + \sqrt{\frac{\pi}{2}} t e^{\frac{t^2}{2}} \{1 - 2Q(t)\}$$

$$\frac{\rho}{1-\rho} = 2\left(\frac{e^{\frac{t^2}{2}}}{t\sqrt{2\pi}} - Q(t)\right)$$

$$Q(t) = \int_t^{\infty} \frac{e^{\frac{x^2}{2}}}{\sqrt{2\pi}} dx$$



QUANTUM ANNEALING
MACHINE LEARNING
Sparse Modeling

ノイズ有り圧縮センシング

- 観測結果にはノイズがつきもの？！
  - 加法的ノイズが混入した場合

$$\mathbf{y} = A\mathbf{x} + \sigma_0 \mathbf{w}$$

  - 等式制約はもはや成立しない
    - LASSO（Least Absolute Shrinkage and Selection Operators）

$$\min_{\mathbf{x}} \left\{ \|\mathbf{y} - A\mathbf{x}\|_2^2 \right\} \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq a$$

    - ラグランジュ未定乗数法により等価な最適化問題

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

例題

以下の最小化問題を解け.

$$\min_x \left\{ |x| + \frac{1}{2\lambda}(y-x)^2 \right\}$$

ここで λ, y は適当な実数である.

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

正解

- 絶対値関数は場合分けすれば良い
  - X>0のときについて平方完成すると以下のようになる.

$$\frac{1}{2\lambda}\left\{x-(y-\lambda)\right\}^2+y-\frac{\lambda}{2}$$

  - ２次関数の最小値は頂点！？
    - 頂点の正負により最小値が変わる.

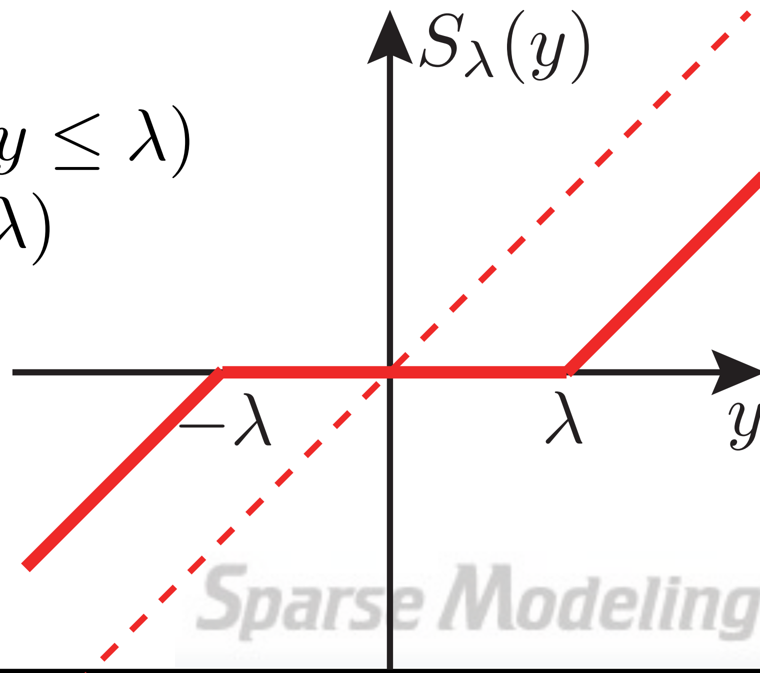
結果

- 軟判定しきい値関数の導入

$$\min_x \left\{ |x| + \frac{1}{2\lambda}(y-x)^2 \right\}$$

- この最小化問題の解は、以下の軟判定しきい値関数で与えられる.

$$S_\lambda(y) = \begin{cases} y-\lambda & (y>\lambda) \\ 0 & (-\lambda \le y \le \lambda) \\ -y-\lambda & (y<-\lambda) \end{cases}$$



QUANTUM ANNEALING MACHINE LEARNING

Sparse Modeling

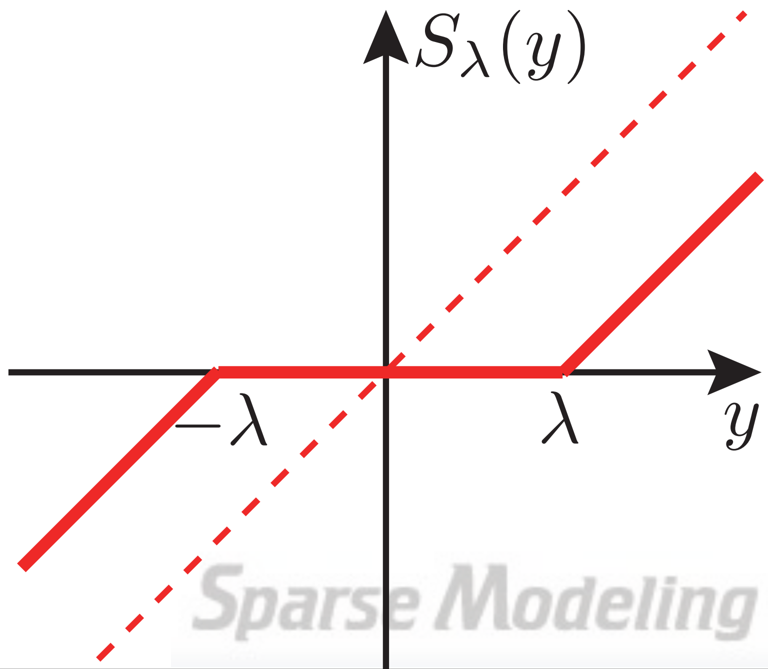
多変数であっても同様

- L1ノルム、L2ノルムの分離性により

$$\min_{\mathbf{x}} \left\{ |\mathbf{x}| + \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{x}\|_2^2 \right\}$$

  - この最小化問題の解は、軟判定しきい値関数で与えられる.

$$\mathbf{x}^* = S_\lambda(\mathbf{y})$$



多変数であっても同様

- L1ノルム、L2ノルムの分離性により

$$\min_{\mathbf{x}}\left\{|\mathbf{x}|+\frac{1}{2\lambda}\|\mathbf{y}-\mathbf{x}\|_2^2\right\}$$

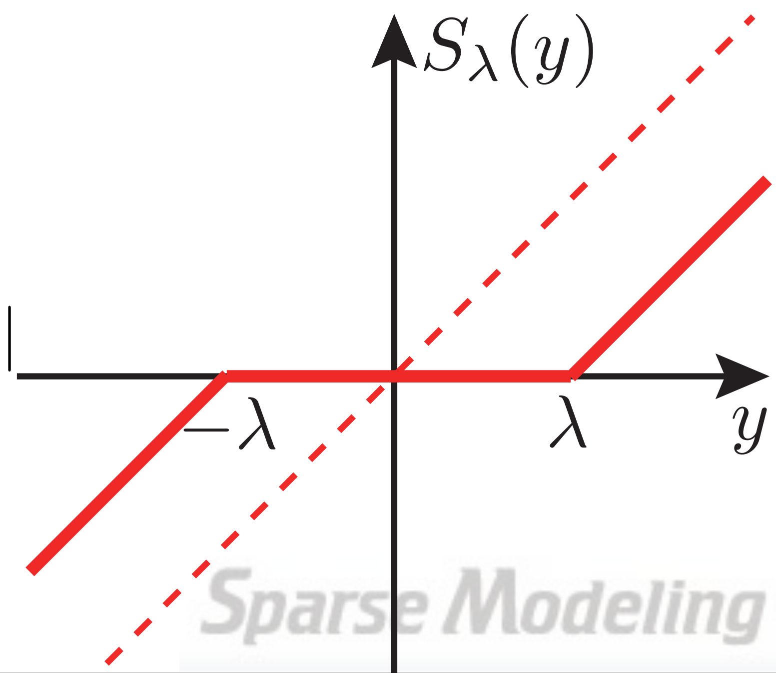
  - この最小化問題の解は、軟判定しきい値関数で与えられる.

$$\mathbf{x}^* = S_{\lambda}(\mathbf{y})$$

  - 分離性は、成分毎の和に分かれること

$$\|\mathbf{x}\|_1 = \|x_1\| + \|x_2\| + \cdots + \|x_N\|$$

$$\|\mathbf{x}\|_2^2 = x_1^2 + x_2^2 + \cdots + x_N^2$$



QUANTUM ANNEALING MACHINE LEARNING

Sparse Modeling

問題点

- 解くべき問題はLASSO型最小化問題

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

  - ちょっとそのまま軟判定しきい値関数は利用できない

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

問題点

- 解くべき問題はLASSO型最小化問題

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

  - ちょっとそのまま軟判定しきい値関数は利用できない
  - なんとかして分離性を持つ形にしたい.

$$\min_{\mathbf{x}} \left\{ |\mathbf{x}| + \frac{1}{2\lambda} \|\mathbf{v} - \mathbf{x}\|_2^2 \right\}$$

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

問題点

- 解くべき問題はLASSO型最小化問題

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

  - ちょっとそのまま軟判定しきい値関数は利用できない
  - なんとかして分離性を持つ形にしたい.

$$\min_{\mathbf{x}} \left\{ |\mathbf{x}| + \frac{1}{2\lambda} \|\mathbf{v} - \mathbf{x}\|_2^2 \right\}$$

  - そうすれば軟判定しきい値関数を適用できる

$$\mathbf{x}^* = S_\lambda(\mathbf{v})$$

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

メジャライザー最小化

- 最小化したい関数 $g(\mathbf{x})$ を上からふたするメジャライザーを導入

$$q_L(\mathbf{x}, \mathbf{v}) = g(\mathbf{v}) + (\nabla g(\mathbf{v}))^{\mathrm{T}} (\mathbf{x} - \mathbf{v}) + \frac{L}{2} \|\mathbf{x} - \mathbf{v}\|_2^2$$

$$g(\mathbf{x}) \leq q_L(\mathbf{x}, \mathbf{v})$$

メジャライザー最小化

- 最小化したい関数 $g(\mathbf{x})$ を上からふたするメジャライザーを導入

$$q_L(\mathbf{x}, \mathbf{v}) = g(\mathbf{v}) + (\nabla g(\mathbf{v}))^{\mathrm{T}} (\mathbf{x} - \mathbf{v}) + \frac{L}{2} \|\mathbf{x} - \mathbf{v}\|_2^2$$

$$g(\mathbf{x}) \leq q_L(\mathbf{x}, \mathbf{v})$$

  - メジャライザーは２次関数なので平方完成すれば最小値は即座にわかる

メジャライザー最小化

- 最小化したい関数 $g(\mathbf{x})$ を上からふたするメジャライザーを導入

$$q_L(\mathbf{x},\mathbf{v}) = g(\mathbf{v}) + (\nabla g(\mathbf{v}))^{\mathrm{T}}(\mathbf{x}-\mathbf{v}) + \frac{L}{2}\|\mathbf{x}-\mathbf{v}\|_2^2$$

$$g(\mathbf{x}) \leq q_L(\mathbf{x},\mathbf{v})$$

  - メジャライザーは２次関数なので平方完成すれば最小値は即座にわかる

- 逐次メジャライザーの最小化をすると…

$$g(\mathbf{x}[t+1]) \leq q_L(\mathbf{x}[t+1],\mathbf{x}[t]) \leq g(\mathbf{x}[t])$$

メジャライザー最小化

- 最小化したい関数 $g(\mathbf{x})$ を上からふたするメジャライザーを導入

$$q_L(\mathbf{x},\mathbf{v}) = g(\mathbf{v}) + (\nabla g(\mathbf{v}))^{\mathrm{T}}(\mathbf{x}-\mathbf{v}) + \frac{L}{2}\|\mathbf{x}-\mathbf{v}\|_2^2$$

$$g(\mathbf{x}) \leq q_L(\mathbf{x},\mathbf{v})$$

  - メジャライザーは２次関数なので平方完成すれば最小値は即座にわかる

- 逐次メジャライザーの最小化をすると…

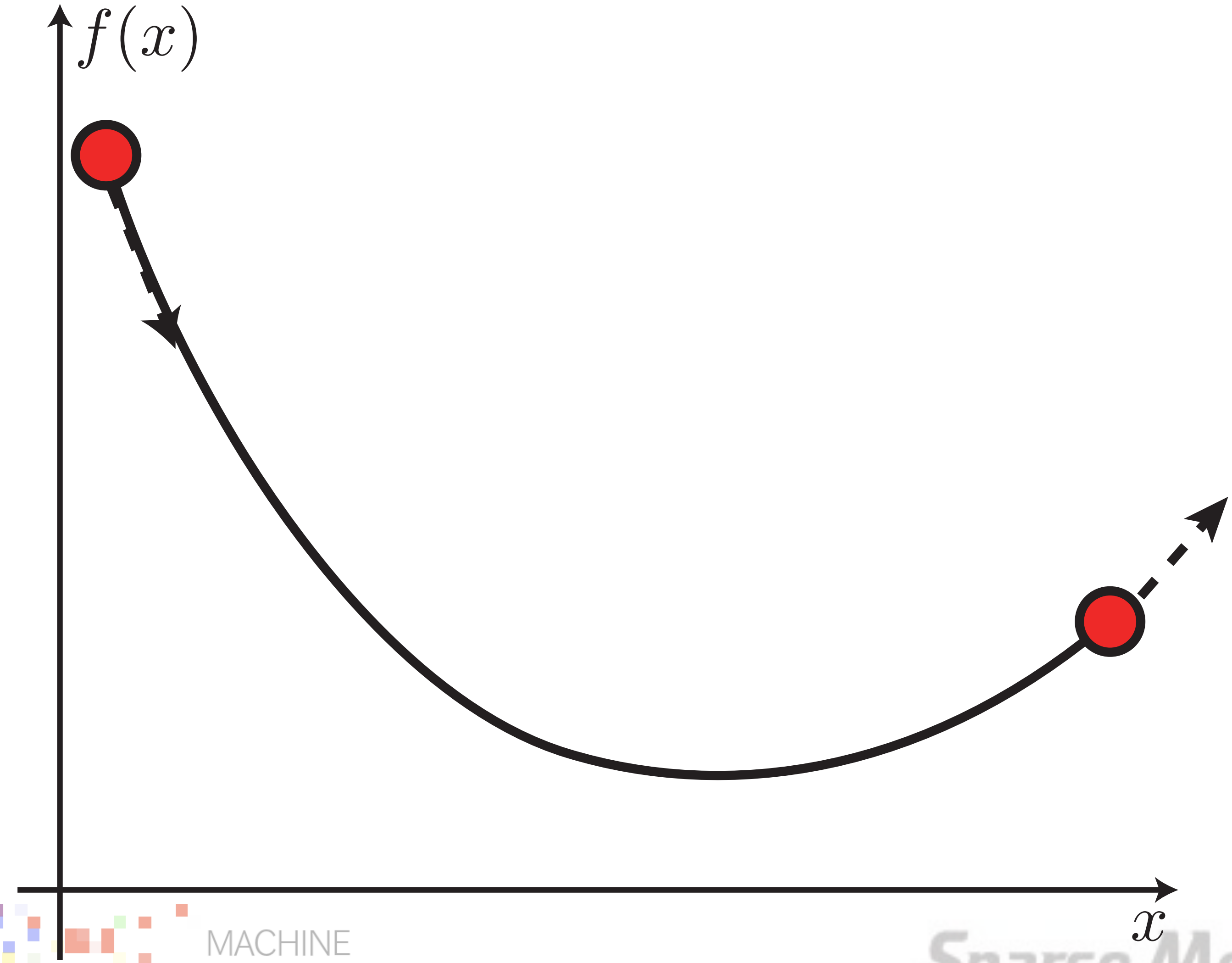
$$g(\mathbf{x}[t+1]) \leq q_L(\mathbf{x}[t+1],\mathbf{x}[t]) \leq g(\mathbf{x}[t])$$

  - 本来の最小化したい関数も小さくできる！

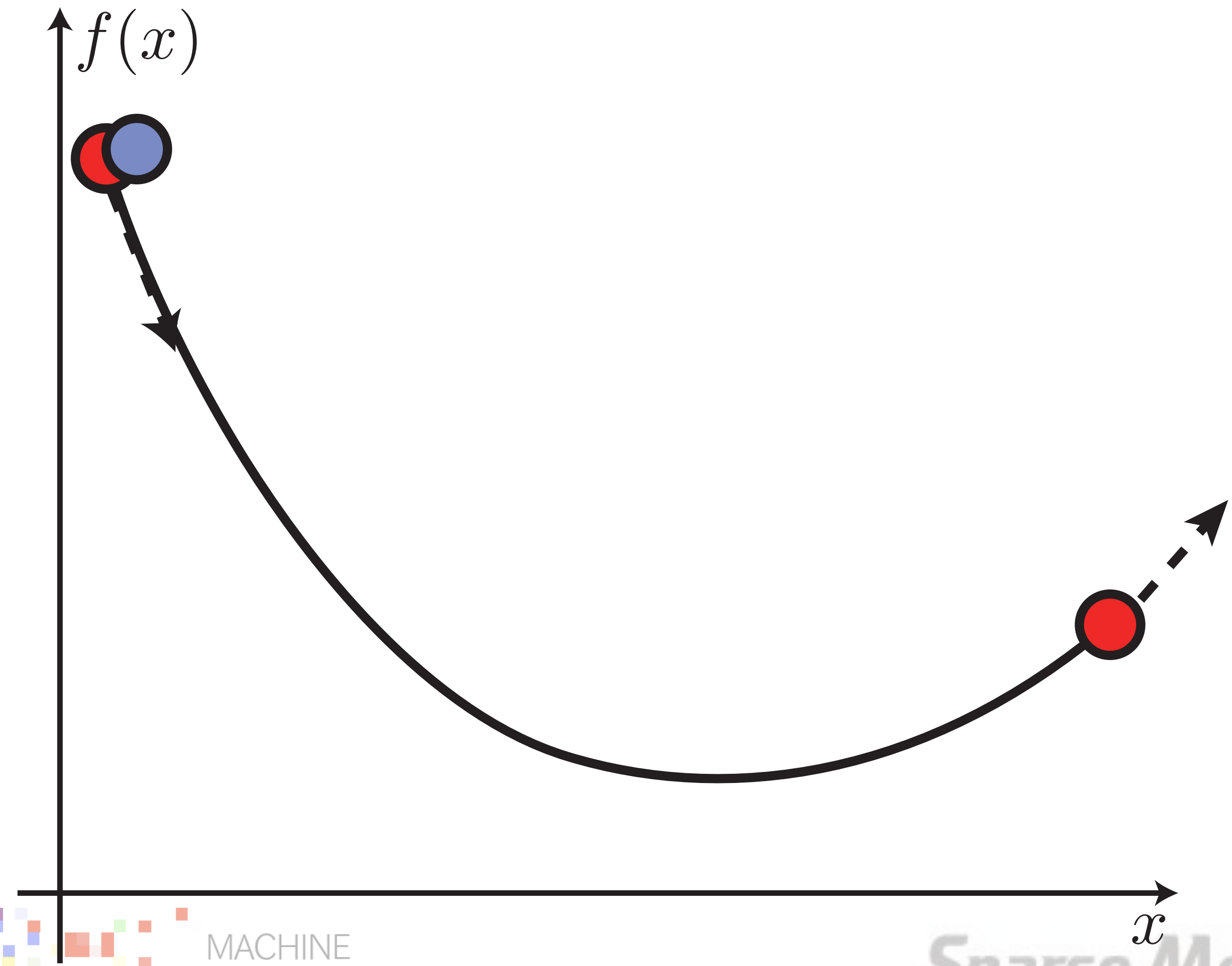
QUANTUM ANNEALING MACHINE LEARNING

Sparse Modeling

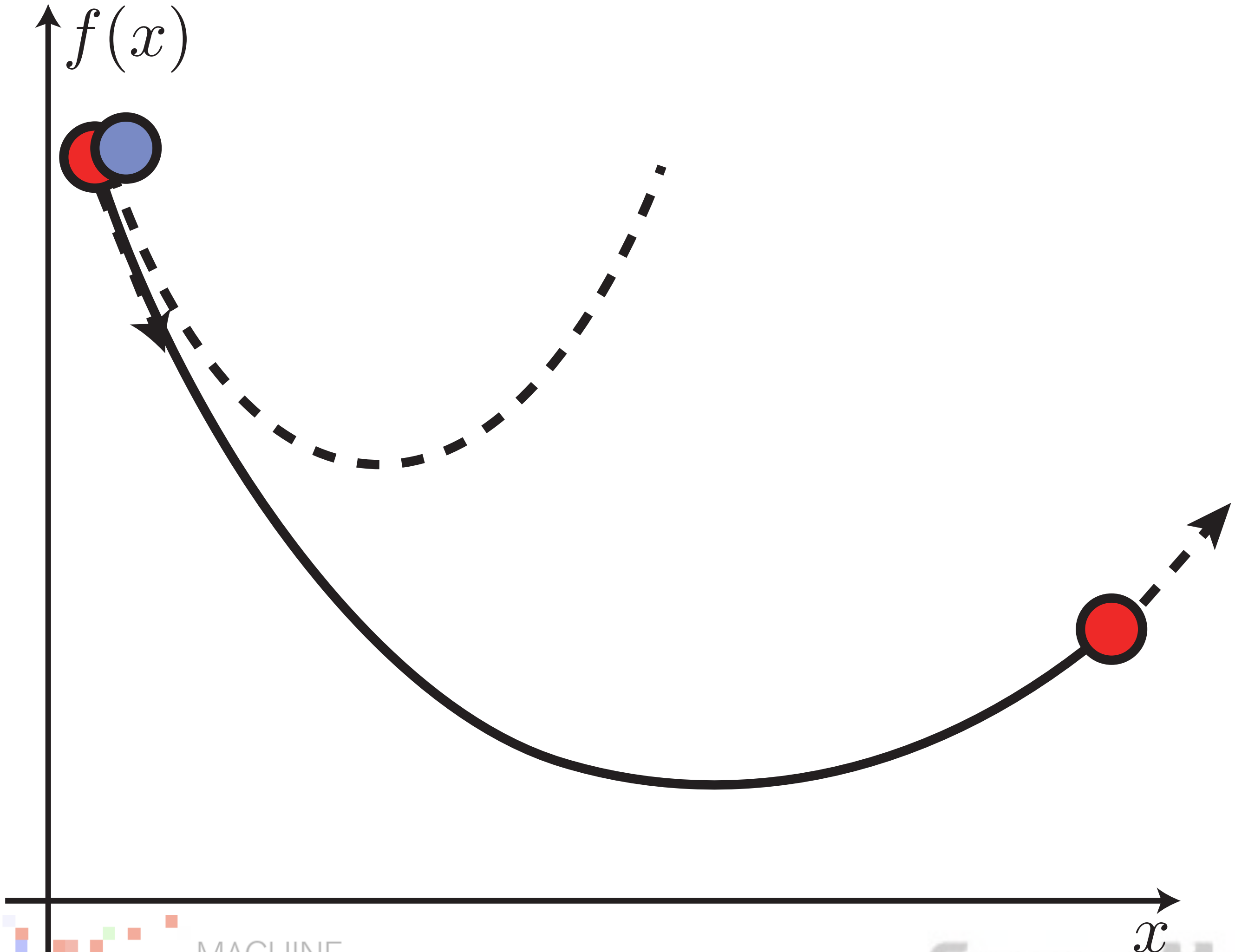
メジャライザー逐次最小化のイメージ



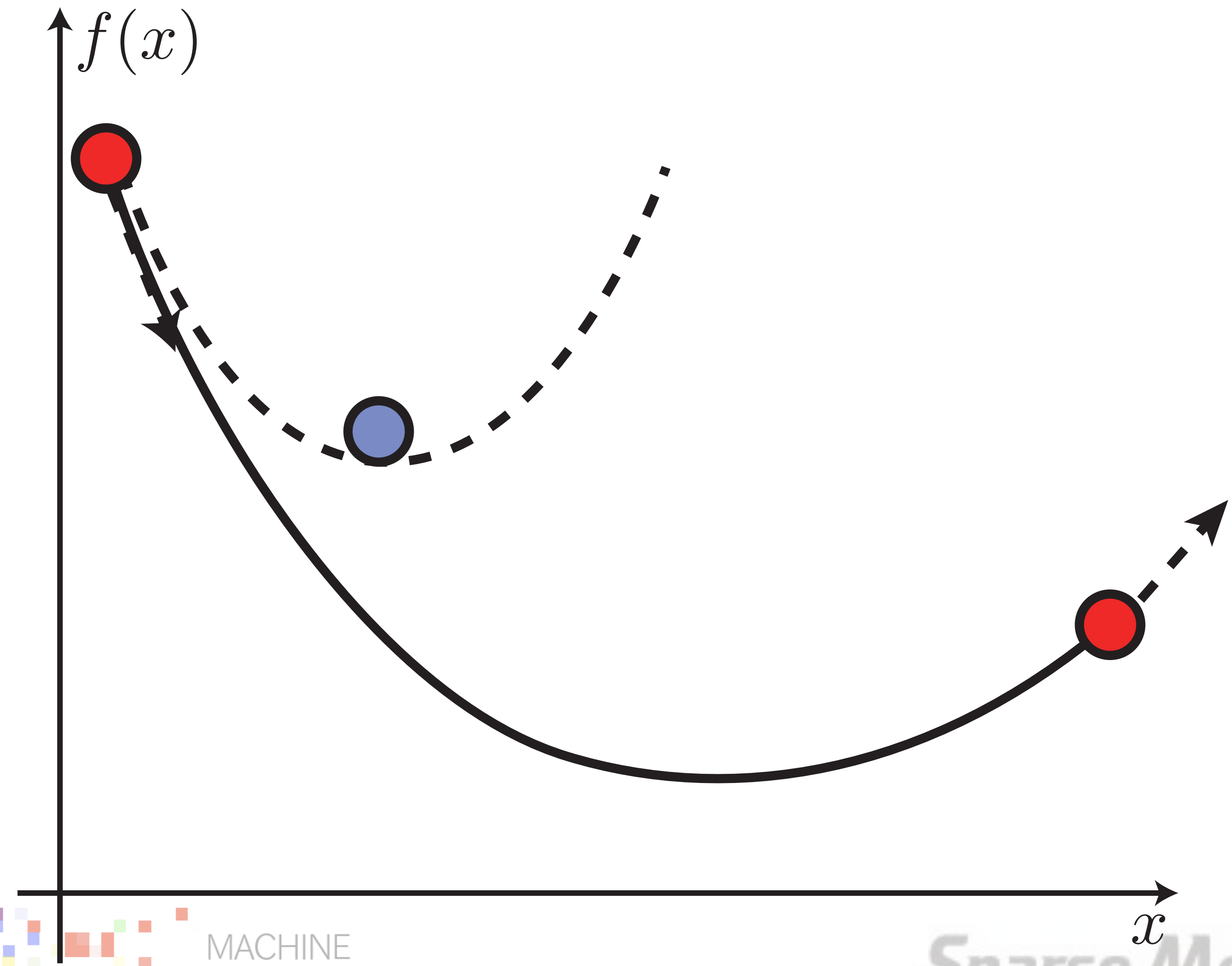
メジャライザー逐次最小化のイメージ



メジャライザー逐次最小化のイメージ

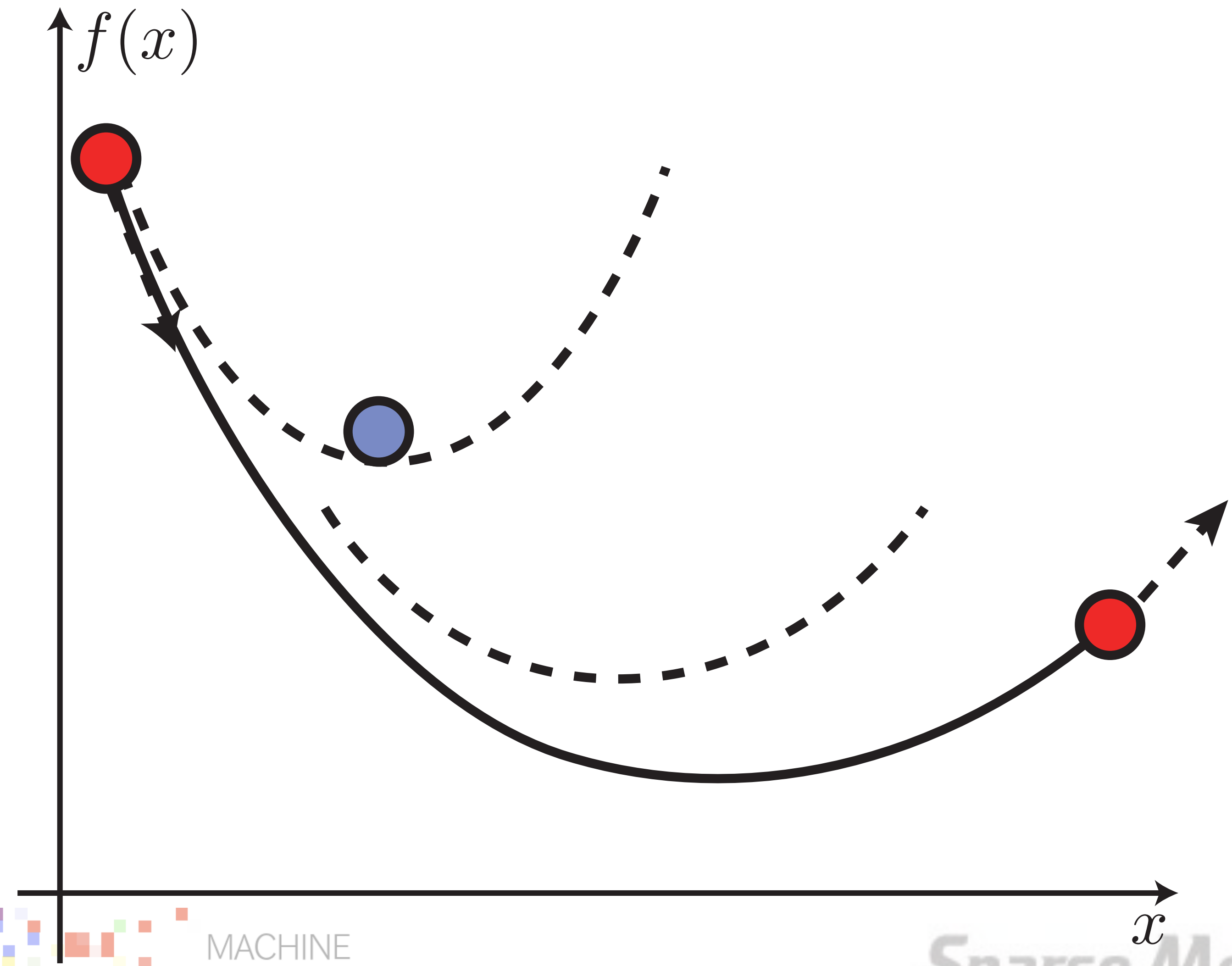


メジャライザー逐次最小化のイメージ

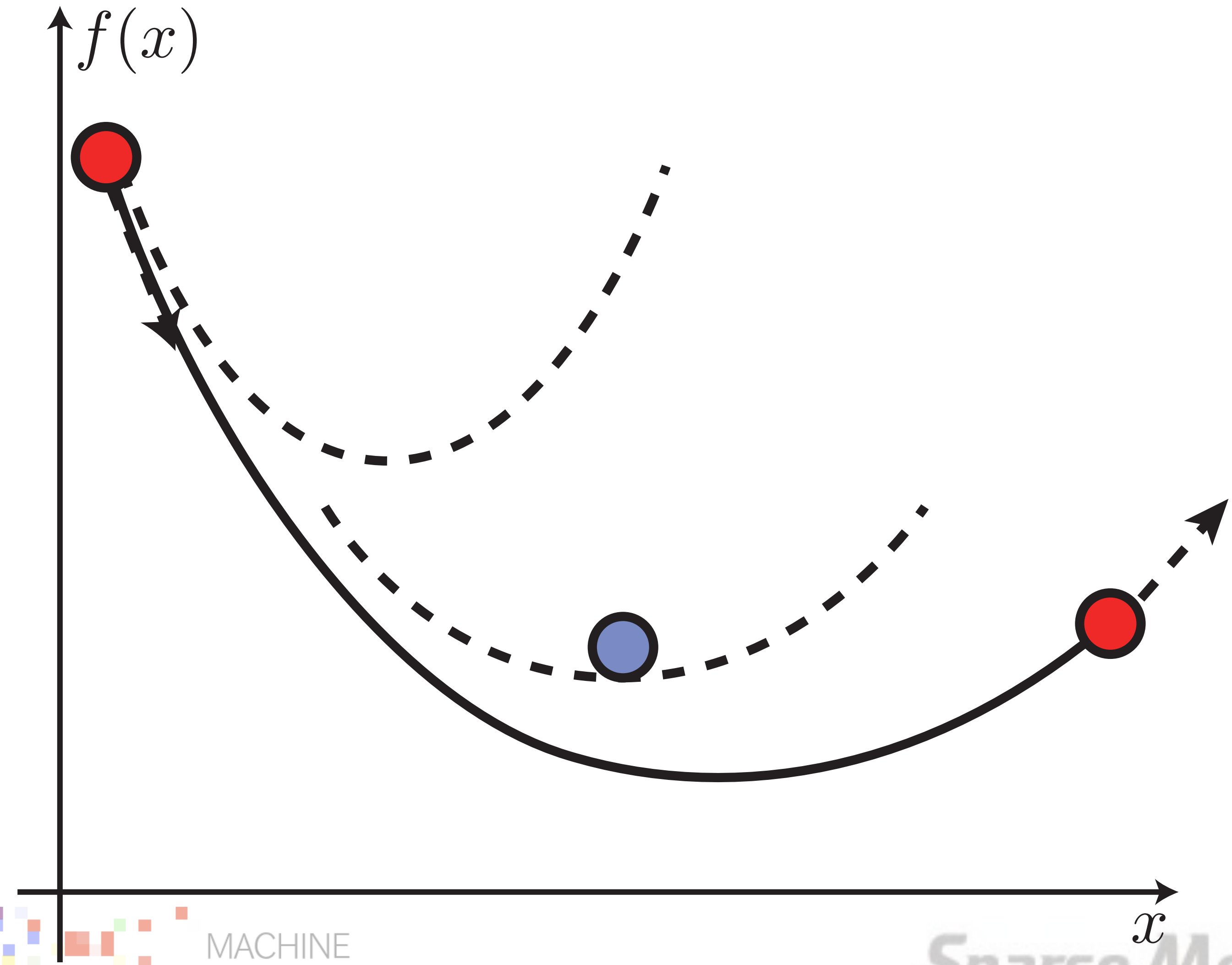


QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

メジャライザー逐次最小化のイメージ



メジャライザー逐次最小化のイメージ

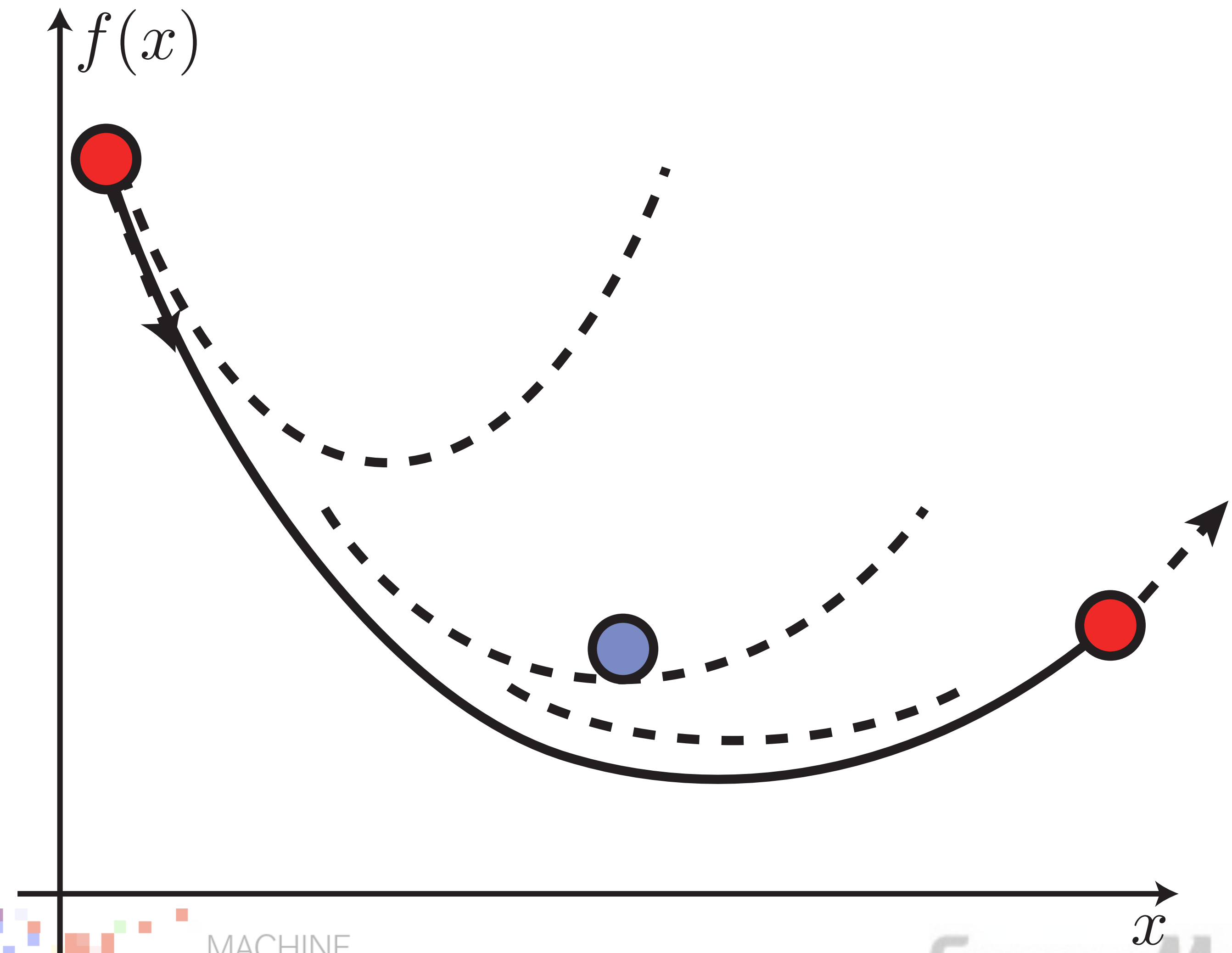


QUANTUM
ANNEALING

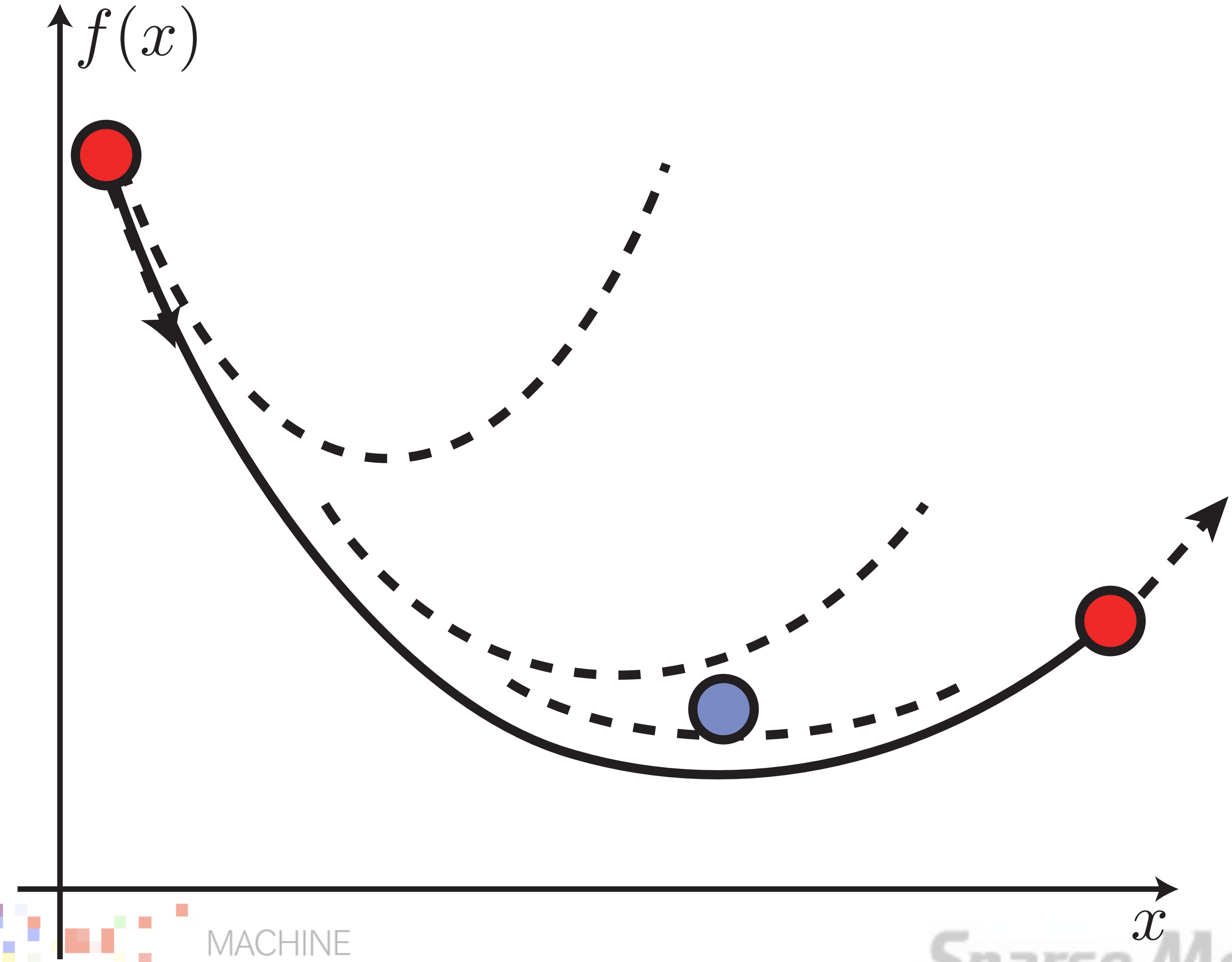
MACHINE
LEARNING

Sparse Modeling

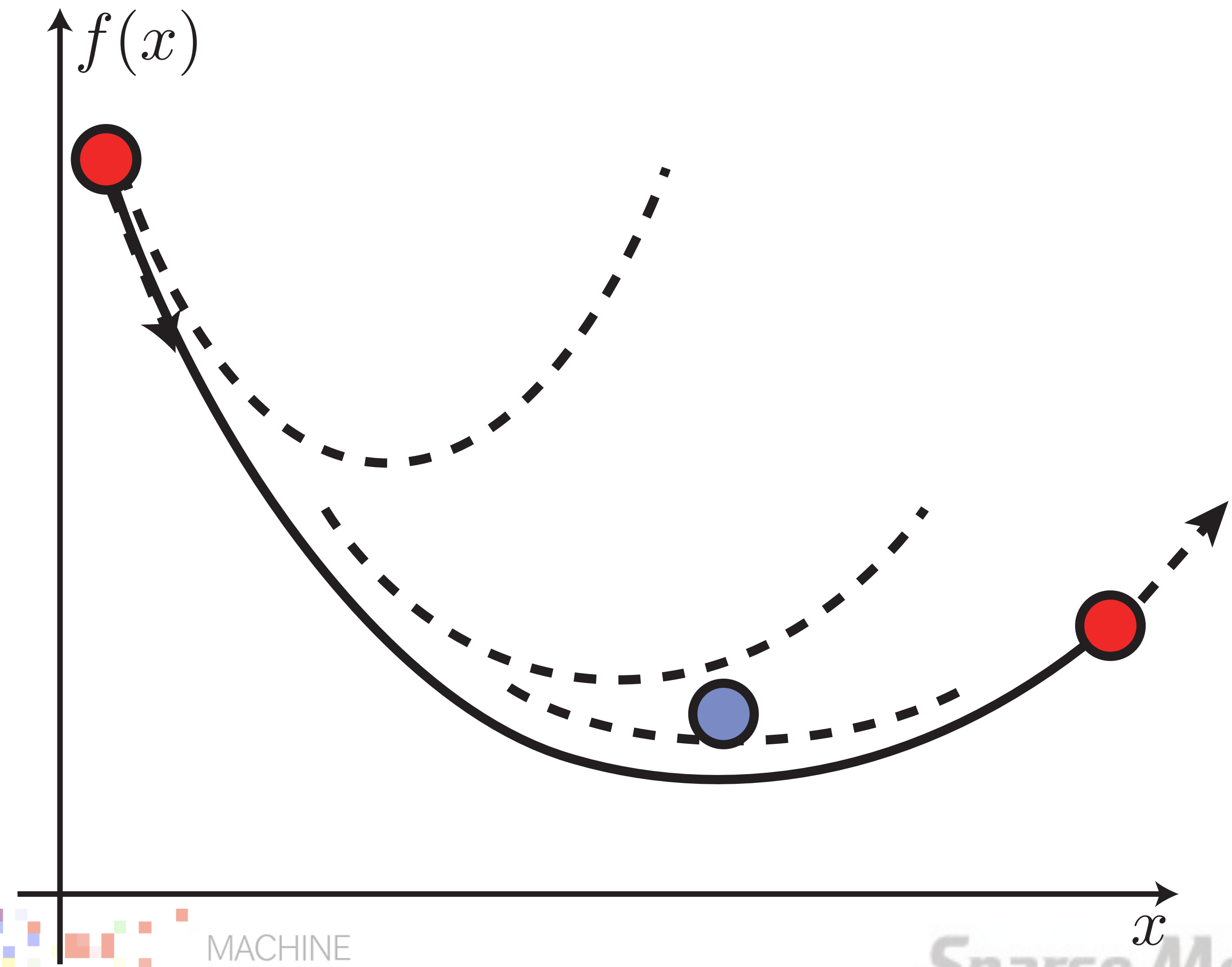
メジャライザー逐次最小化のイメージ



メジャライザー逐次最小化のイメージ



メジャライザー逐次最小化のイメージ



LASSOにメジャライザー最小化

- L1ノルムが追加されてもメジャライザーの性質は崩れない.

$$g(\mathbf{x}) + \|\mathbf{x}\|_1 \leq q_L(\mathbf{x}, \mathbf{v}) + \|\mathbf{x}\|_1$$

- メジャライザー逐次最小化を用いて、LASSOを解こう！

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

QUANTUM ANNEALING MACHINE LEARNING Sparse Modeling

LASSOにメジャライザー最小化

- L1ノルムが追加されてもメジャライザーの性質は崩れない.

$$g(\mathbf{x}) + \|\mathbf{x}\|_1 \leq q_L(\mathbf{x}, \mathbf{v}) + \|\mathbf{x}\|_1$$

- メジャライザー逐次最小化を用いて、LASSOを解こう！

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

  - 方針
    - メジャライザー+L1ノルムを用意する.

$$g(\mathbf{x}) + \|\mathbf{x}\|_1 \leq q_L(\mathbf{x}, \mathbf{x}[t]) + \|\mathbf{x}\|_1$$

    - メジャライザー+L1ノルムの最小解を得る.

$$\mathbf{x}[t+1] = \arg\min_{\mathbf{x}} \left\{ q_L(\mathbf{x}, \mathbf{x}[t]) + \|\mathbf{x}\|_1 \right\}$$

    - メジャライザーを更新する.

$$q_L(\mathbf{x}, \mathbf{x}[t+1])$$

LASSOにメジャライザー最小化

- LASSOの場合　$g(\mathbf{x}) = \frac{1}{2\lambda}\|\mathbf{y} - A\mathbf{x}\|_2^2$

$$q_L(\mathbf{x}, \mathbf{v}) = g(\mathbf{v}) + \frac{A^{\mathrm{T}}A}{\lambda}(\mathbf{y} - A\mathbf{v})(\mathbf{x} - \mathbf{v}) + \frac{L}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$$

LASSOにメジャライザー最小化

- LASSOの場合　$g(\mathbf{x}) = \frac{1}{2\lambda}\|\mathbf{y} - A\mathbf{x}\|_2^2$

$$q_L(\mathbf{x}, \mathbf{v}) = g(\mathbf{v}) + \frac{A^{\mathrm{T}}A}{\lambda}(\mathbf{y} - A\mathbf{v})(\mathbf{x} - \mathbf{v}) + \frac{L}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$$

  - 平方完成してメジャライザー+L1ノルム最小解を求める

$$\mathbf{x}[t+1] = \arg\min_{\mathbf{x}}\left\{\frac{L}{2}\left\{\mathbf{x} - \left(\mathbf{x}[t] + \frac{1}{L\lambda}A^{\mathrm{T}}(\mathbf{y} - A\mathbf{x}[t])\right)\right\}^2 + \|\mathbf{x}\|_1\right\}$$

LASSOにメジャライザー最小化

- LASSOの場合 $g(\mathbf{x}) = \frac{1}{2\lambda}\|\mathbf{y} - A\mathbf{x}\|_2^2$

$$q_L(\mathbf{x}, \mathbf{v}) = g(\mathbf{v}) + \frac{A^{\mathrm{T}}A}{\lambda}(\mathbf{y} - A\mathbf{v})(\mathbf{x} - \mathbf{v}) + \frac{L}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$$

  - 平方完成してメジャライザー+L1ノルム最小解を求める

$$\mathbf{x}[t+1] = \arg\min_{\mathbf{x}} \left\{ \frac{L}{2}\left\{ \mathbf{x} - \left(\mathbf{x}[t] + \frac{1}{L\lambda}A^{\mathrm{T}}(\mathbf{y} - A\mathbf{x}[t]) \right) \right\}^2 + \|\mathbf{x}\|_1 \right\}$$

  - 分離性を持つ形であるので軟判定しきい値関数を適用するだけ.

$$\mathbf{x}[t+1] = S_{1/L}\left(\mathbf{x}[t] + \frac{1}{L\lambda}A^{\mathrm{T}}(\mathbf{y} - A\mathbf{x}[t]) \right)$$

QUANTUM ANNEALING MACHINE LEARNING

Sparse Modeling

ISTA: Iterative Shrinkage Thresholding Algorithm

1. $t=0$ とする．初期化 $\mathbf{x}[0]$.（例えば $\mathbf{x}[0]=A^{\mathrm{T}}\mathbf{y}$）
2. 平方完成により $g(\mathbf{x})$ の 2 次関数近似の頂点を求める．

$$\mathbf{v}[t]=\mathbf{x}[t]+\frac{1}{L\lambda}A^{\mathrm{T}}(\mathbf{y}-A\mathbf{x}[t])$$

3. 軟判定しきい値関数を適用する．

$$\mathbf{x}[t+1]=S_{1/L}\left(\mathbf{v}[t]\right)$$

4. 終了基準を満たすまでステップ 2-4 を繰り返す．

FISTA: Fast Iterative Shrinkage Thresholding Algorithm

1. $t = 1$ とする．初期化 $\mathbf{x}[0]$、$\beta[1] = 0$、$\mathbf{w}[1] = \mathbf{x}[0]$
2. 平方完成により $g(\mathbf{x})$ の 2 次関数近似の頂点を求める．

$$\mathbf{v}[t] = \mathbf{w}[t] + \frac{1}{L\lambda} A^{\mathrm{T}} (\mathbf{y} - A\mathbf{w}[t])$$

3. 軟判定しきい値関数を適用する．

$$\mathbf{x}[t+1] = S_{1/L}\left(\mathbf{v}[t]\right)$$

4. **[高速化部分]**$\beta[t]$ を更新する．

$$\beta[t] = \frac{1}{2}\left(1 + \sqrt{1 + 4\left(\beta[t-1]\right)^2}\right)$$

5. **[高速化部分]**$\mathbf{w}[t]$ を更新する．

$$\mathbf{w}[t+1] = \mathbf{x}[t+1] + \frac{\beta[t] - 1}{\beta[t+1]} \left(\mathbf{x}[t+1] - \mathbf{x}[t]\right)$$

6. 終了基準を満たすまでステップ 2-6 を繰り返す．

メジャライザーは存在するか？

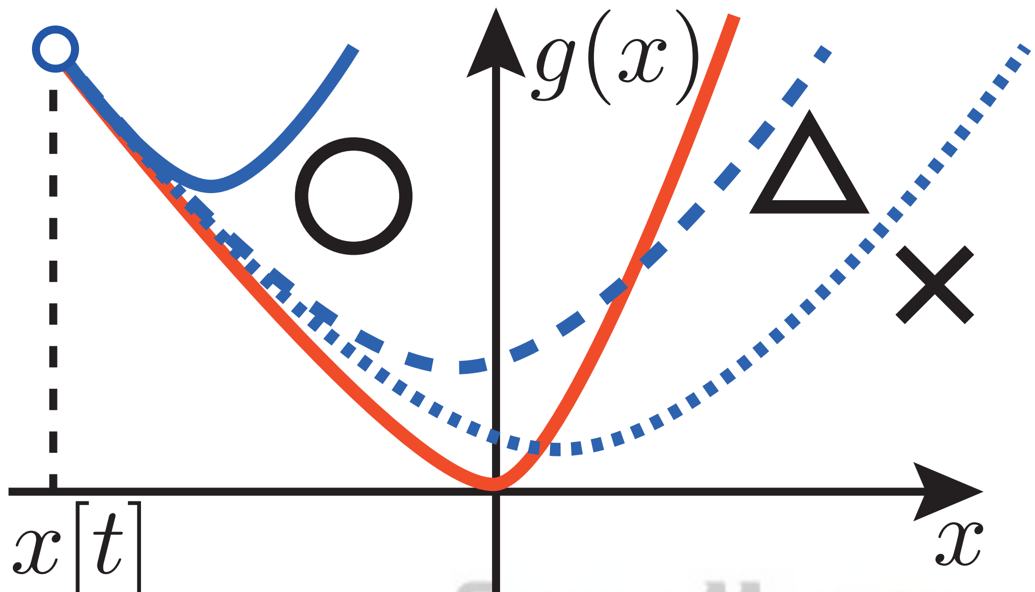
- そんな都合のいい関数あるのか？
  - メジャライザーの存在条件
    - 微分がリプシッツ定数Lのリプシッツ連続であればよい.

$$\left\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{v})\right\|_2 \leq L \left\|\mathbf{x} - \mathbf{v}\right\|_2$$

- LASSOの場合、全領域で

$L = A^{\mathrm{T}}A/\lambda$ が示される

- 区分的に上から押さえてもよい



ISTAの別解釈

- 罰金法
  - 古典的制約つき最適化問題の解法

$$\min_{\mathbf{x}}\left\{\|\mathbf{x}\|_1 + g(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}[t]\|_2^2\right\}$$

  - 暫定解に近いところで最適解を探索させる.
  - 罰金項の係数μが大きいと、制約条件に忠実

ISTAの別解釈

- 罰金法
  - 古典的制約つき最適化問題の解法

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + g(\mathbf{x}) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{x}[t]\|_2^2 \right\}$$

    - 暫定解に近いところで最適解を探索させる.
    - 罰金項の係数μが大きいと、制約条件に忠実

- ISTA＝罰金法＋線形近似
  - コスト関数を暫定解付近で展開

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + g(\mathbf{x}[t]) + (\nabla g(\mathbf{x}[t]))^{\mathrm{T}} (\mathbf{x} - \mathbf{x}[t]) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{x}[t]\|_2^2 \right\}$$

基底追跡の解法

- ノイズなし圧縮センシング
  - 基底追跡

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = A\mathbf{x}_0$$

  - 制約条件ありの最適化問題
    - ラグランジュ未定乗数

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + \left(\mathbf{h}[t]\right)^{\mathrm{T}} \left(\mathbf{y} - A\mathbf{x}\right) \right\}$$

      - 最小化のときは正の乗数、最大化のときは負の乗数
      - 未定乗数の更新と最適化にかかる変数の更新でやや不安定
    - 罰金法

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + \frac{\mu}{2} \|\mathbf{y} - A\mathbf{x}\|_2^2 \right\}$$

      - 罰金係数を徐々に大きくしていかなければならない

基底追跡の解法

- 拡張ラグランジュ法
  - ラグランジュ未定乗数と罰金法の組み合わせ

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + \left(\mathbf{h}^{\mathrm{T}}\right)\left(\mathbf{y} - A\mathbf{x}\right) + \frac{\mu}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 \right\}$$

  - ラグランジュ未定乗数の更新則

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu[t]\left(\mathbf{y} - A\mathbf{x}\right)$$

    - 罰金係数をそこまで大きくしなくてもよい.
    - 未定乗数の計算が安定化する.

QUANTUM ANNEALING　MACHINE LEARNING　Sparse Modeling

基底追跡の解法

- 拡張ラグランジュ法
  - ラグランジュ未定乗数と罰金法の組み合わせ

$$\min_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 + \color{blue}{\left(\mathbf{h}^{\mathrm{T}}\right)\left(\mathbf{y} - A\mathbf{x}\right)} + \color{red}{\frac{\mu}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2} \right\}$$

  - ラグランジュ未定乗数の更新則

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu[t]\left(\mathbf{y} - A\mathbf{x}\right)$$

    - 罰金係数をそこまで大きくしなくてもよい
    - 未定乗数の計算が安定化する

- Bregman反復法と等価である

最近の進展

- ADMM(Alternating Direction Method of Multipliers)
  - ふたつのコスト関数の最適化問題を解く方法

$$\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + g(\mathbf{x}) \right\}$$

  - 制約条件ありの最適化問題へ敢えて変更する

$$\min_{\mathbf{x}} \left\{ f(\mathbf{z}) + g(\mathbf{x}) \right\} \quad \text{s.t.} \quad \mathbf{x} - \mathbf{z} = \mathbf{0}$$

  - LASSOであれば、

$$f(\mathbf{x}) = \|\mathbf{x}\|_1 \quad g(\mathbf{x}) = \frac{1}{2\lambda}\|\mathbf{y} - A\mathbf{x}\|_2^2$$

QUANTUM ANNEALING MACHINE LEARNING

Sparse Modeling

ADMMの実際

- 拡張ラグランジュ法をちょっと適当にやる

$$\min_{\mathbf{x},\mathbf{z}} \left\{ f(\mathbf{z}) + g(\mathbf{x}) + \left(\mathbf{h}^{\mathrm{T}}\right)(\mathbf{x}-\mathbf{z}) + \frac{\mu}{2}\|\mathbf{x}-\mathbf{z}\|_2^2 \right\}$$

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu(\mathbf{x}-\mathbf{z})$$

- 拡張ラグランジュ法との違い
  - 罰金係数は適当な値で固定
  - ふたつの変数についての最適化を交互にやる

$$\mathbf{x}[t+1] = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{z}[t]) + g(\mathbf{x}) + (\mathbf{h}[t])^{\mathrm{T}}(\mathbf{x}-\mathbf{z}[t]) + \frac{\mu}{2}\|\mathbf{x}-\mathbf{z}[t]\|_2^2 \right\}$$

$$\mathbf{z}[t+1] = \arg\min_{\mathbf{z}} \left\{ f(\mathbf{z}) + g(\mathbf{x}[t+1]) + (\mathbf{h}[t])^{\mathrm{T}}(\mathbf{x}[t+1]-\mathbf{z}) + \frac{\mu}{2}\|\mathbf{x}[t+1]-\mathbf{z}\|_2^2 \right\}$$

ADMM for LASSO

- それぞれ解を代入するだけで求められる
  - Xについては二次関数に過ぎない＝平方完成、微分

$$\mathbf{x}[t+1] = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + (\mathbf{h}[t])^{\mathrm{T}}(\mathbf{x} - \mathbf{z}[t]) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z}[t]\|_2^2 \right\}$$

    - 単純更新

$$\mathbf{x}[t+1] = \left(\mu + \frac{1}{\lambda} A^{\mathrm{T}} A \right)^{-1} \left(A^{\mathrm{T}}\mathbf{y} + \mu \mathbf{z}[t] - \mathbf{h}[t] \right)$$

  - Zについては分離性がきく＝軟判定しきい値関数

$$\mathbf{z}[t+1] = \arg\min_{\mathbf{z}} \left\{ \|\mathbf{z}\|_1 + (\mathbf{h}[t])^{\mathrm{T}}(\mathbf{x}[t+1] - \mathbf{z}) + \frac{\mu}{2} \|\mathbf{x}[t+1] - \mathbf{z}\|_2^2 \right\}$$

    - 単純更新

$$\mathbf{z}[t+1] = S_{1/\mu} \left(\mathbf{x}[t+1] - \frac{\mathbf{h}[t]}{\mu} \right)$$

    - Hも単純更新

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu(\mathbf{x}[t] - \mathbf{z}[t])$$

レポート問題

- どれかのアルゴリズムを実際に実装して動作確認せよ
  - FISTAまたはISTA
    - LASSO型最適化問題なのでノイズ有り圧縮センシング
  - 拡張ラグランジュ法またはBregman反復法
    - BP型最適化問題なのでノイズなし圧縮センシング
  - ADMM
    - LASSO型最適化問題なのでノイズ有り圧縮センシング
- 実験手法（N=1000, K=20, M=100とする）
  - X0の成分をK個だけ1を、それ以外0とする.
  - Aの各成分を正規分布から生成させる.
  - Y=A*X0を計算する. （ノイズ有りならN(0,<0.1)に従うノイズを印可）
  - この条件のもと、それぞれのアルゴリズムでXを計算する.