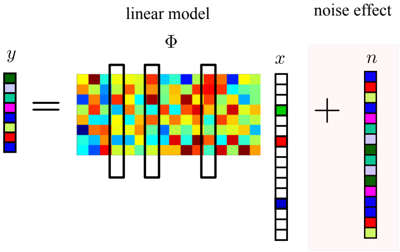
Convex and Greedy Methods for Sparse Approximations

Mehrdad Yaghoobi
Edinburgh Research Partnership in Signal and Image Processing
Institute for Digital Communications,
The University of Edinburgh

UDRC Summer School, Edinburgh, 24 June, 2014

Sparse Representation



ℓ_0 Sparse Representation

$$\underset{x}{\operatorname{argmin}} \|x\|_0 \text{ s. t. } y = \Phi x$$

ℓ_0 Sparse Approximation

$$\underset{x}{\operatorname{argmin}} \|x\|_0 \text{ s. t. } \|y - \Phi x\|_2 \leq \epsilon$$

Why Sparse Approximation Is Difficult?

Difficulties?

- Combinatorial optimisation $\rightarrow$ Non-polynomial time solvers.
- Non-Smooth objective $\rightarrow$ No (direct) Gradient Descent method.

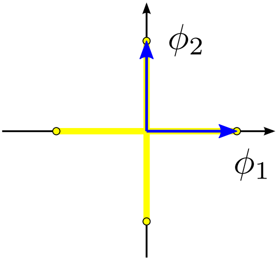
Possible Approaches?

- Relaxation of the objective, *e.g.* convex surrogate objective $\ell_p,\ 1 \leq p$
- Approximate solution finding, *e.g.* greedy and iterative methods.
- Combination of two, *e.g.* surrogate objective $\ell_p^p,\ 0 \leq p < 1$

What would be covered in this session?

1. **Convex relaxation and optimisation techniques.**
2. **Broader range sparse approximation methods.**
3. **Greedy optimisation techniques.**
4. **Iterative thresholding methods.**

Convex Relaxation



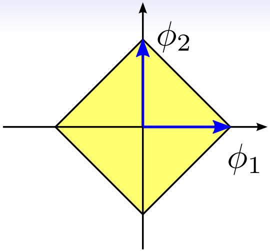
Bounded K-Sparse Vectors

$$\mathcal{X} = \{x : \|x\|_0 \leq k, \|x\|_\infty \leq 1\}$$

Assumption in This Talk

Φ has normalised columns.

Convex Relaxation



Bounded K-Sparse Vectors

$\mathcal{X} = \{x : \|x\|_0 \leq k, \|x\|_\infty \leq 1\}$

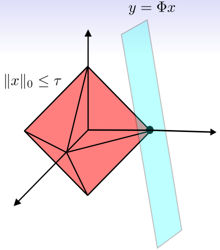
Assumption in This Talk

Φ has normalised columns.

Convex Hull of K-Sparse Vectors: ℓ_1-ball

$$\begin{aligned}\mathcal{C} =& \{\lambda x_1 + (1-\lambda)x_2 : \ 0 \leq \lambda \leq 1, \ x_1, x_2 \in \mathcal{X}\} \\ =& \{x : \|x\|_1 = \sum_i |x_i| \leq \tau\}\end{aligned}$$

ℓ_1 Convex Optimisation: A Geometric View



ℓ_1 Convex Optimisation Formulation

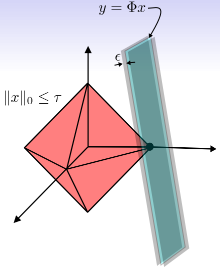
- **Basis Pursuit (BP):**[Chen et al. 98]

$$\operatorname{argmin}_x \|x\|_1 \text{ s. t. } y = \Phi x$$

Noisy ℓ_1 Convex Formulations

- **Basis Pursuit Denoising (BPDN):** [Chen et al. 98]
$$\operatorname{argmin}_x \|x\|_1 \quad \text{s. t.} \quad \|y - \Phi x\|_2^2 \leq \epsilon$$
- **Least Absolute Shrinkage/Selection Operator (LASSO):** [Tibshirani 96]
$$\operatorname{argmin}_x \|y - \Phi x\|_2^2 \quad \text{s. t.} \quad \|x\|_1 \leq \tau$$
- **Regularised Sparse Approximation:**
$$\operatorname{argmin}_x \|x\|_1 + \lambda \|y - \Phi x\|_2^2$$

ℓ_1 Convex Optimisation: A Geometric View



ℓ_1 Convex Optimisation Formulation

- **Basis Pursuit (BP):**[Chen et al. 98]

$$\operatorname{argmin}_x \|x\|_1 \text{ s. t. } y = \Phi x$$

Noisy ℓ_1 Convex Formulations

- **Basis Pursuit Denoising (BPDN):** [Chen et al. 98]

$$\operatorname{argmin}_x \|x\|_1 \quad \text{s. t.} \quad \|y - \Phi x\|_2^2 \leq \epsilon$$

- **Least Absolute Shrinkage/Selection Operator (LASSO):** [Tibshirani 96]

$$\operatorname{argmin}_x \|y - \Phi x\|_2^2 \quad \text{s. t.} \quad \|x\|_1 \leq \tau$$

- **Regularised Sparse Approximation:**

$$\operatorname{argmin}_x \|x\|_1 + \lambda \|y - \Phi x\|_2^2$$

Convex Optimisation Techniques for Sparse Representation

Properties

- Smooth but non-differentiable objectives. (instead of LASSO)
- Often, no need to solve it with the machine precision.
- Medium to large scale problems.

Optimisation techniques

- Interior point methods.
- First order methods, *i.e.* Gradient descent methods.
- Forward-Backward techniques.
- Augmented Lagrangian method.
- .... many more!!!

Iterative Soft Thresholding: Motivation

The unconstrained optimisation:

$$\text{argmin}_x \|x\|_1 + \lambda \|y - \Phi x\|_2^2$$

- Non-differentiable objective: subgradient descent method.
- Convergence is very slow.

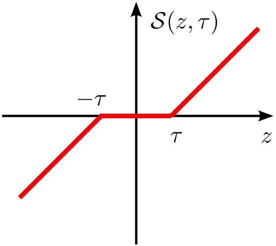
Idea

We know how to solve,

$$x^* = \text{argmin}_x \|x\|_1 + \lambda \|z - x\|_2^2$$

$$x_i^* = \mathcal{S}(z_i, \frac{1}{2\lambda}) = \begin{cases} 0 & |z_i| \le \frac{1}{2\lambda} \\ z_i - \frac{1}{2\lambda} & z_i > \frac{1}{2\lambda} \\ z_i + \frac{1}{2\lambda} & z_i < \frac{-1}{2\lambda} \end{cases}$$

The solution is called Soft-thresholding.



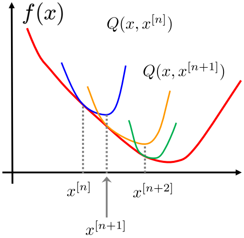
Majorization Minimisation

Decoupling the Minimisation Objective

- The objective has some coupling between elements of $x \Rightarrow$ a single soft-thresholding does not find the solution.
- Majorization Minimisation (MM) can be used to decouple the optimisation problem around current solution.

Majorization Minimisation

- Problem: $x^* = \text{argmin}_x f(x)$
- $Q(x, x^{[n]})$ majorizes $f(x)$ at $x^{[n]}$ if, $Q(x, x^{[n]}) \geq f(x)$, $Q(x^{[n]}, x^{[n]}) = f(x^{[n]})$.
- Majorization Minimisation Technique: $x^{[n+1]} = \text{argmin}_x Q(x, x^{[n]})$
- MM monotonically decreases the original objective value.



Iterative Soft Thresholding: Algorithm

- **Taylor's approximation** for deriving majorizing function.

$$
\begin{aligned}
\|y - \Phi x\|_2^2 \leq & \left\| y - \Phi x^{[n]} \right\|_2^2 \\
& + 2(x - x^{[n]})^T \Phi^T (\Phi x - y) + L/2 \left\| x - x^{[n]} \right\|_2^2
\end{aligned}
$$

- We derive a new optimisation problem which can be solved using soft thresholding [Daubechies et al. 03],

$$
\begin{aligned}
x^{[n+1]} =& \underset{x}{\operatorname{argmin}} \|x\|_1 + \frac{L}{2} \left\| x - (x^{[n]} - \frac{2}{L} \Phi^T (\Phi x^{[n]} - y)) \right\|_2^2 \\
=& \mathcal{S}(x^{[n]} - \frac{2}{L} \Phi^T (\Phi x^{[n]} - y), \frac{1}{2\lambda})
\end{aligned}
$$

Convergence

- Iterate to achieve ϵ residual error or for K times.
- Converges linearly $\mathcal{O}(\frac{1}{n}) \rightarrow$ slow convergence!

Accelerated First Order Methods

Linear convergence $\mathcal{O}(\frac{1}{n})$ v.s. $\mathcal{O}(\frac{1}{n^2})$

- As the objective is not differentiable, we can not expect quadratic convergence rate, *i.e.* similar to Newton's method.
- We can still accelerate using optimal first order methods, *i.e.* $\mathcal{O}(\frac{1}{n^2})$.

- Idea: using the information of two recent iterations.
- Different approach to achieve such a goal, *e.g.* FISTA [Beck and Teboulle 09], NESTA [Becker et al. 11], Nesterov's method [Nestrov 83].

FISTA

1. $t^{[n+1]} = \frac{1+\sqrt{1+(2t^{[n]})^2}}{2}$
2. $z^{[n+1]} = x^{[n]} + \frac{t^{[n]}-1}{t^{[n+1]}}(x^{[n]} - x^{[n-1]})$
3. $x^{[n+1]} = \mathcal{S}(z^{[n]} - \beta\Phi^T(\Phi z^{[n]} - y), \beta/\lambda)$

Gradient Projection for LASSO Problem

$$\mathrm{argmin}_x \|y - \Phi x\|_2^2 \quad \text{s. t.} \quad \|x\|_1 \leq \tau$$

- LASSO problem has a smooth objective and convex constraint.
- Projection onto the ℓ_1 ball, $\mathcal{P}_{\ell_1}(x, \tau)$, can be done efficiently, *i.e.* $\mathcal{O}(n \log n)$.
- Projected Gradient method is suitable for this problem, *e.g.* SPGl1 [Van Den Berg and Friedlander 08].

SPGl1

1. $G(x^{[n+1]}) = -2\Phi^T(\Phi x^{[n]} - y)$
2. $x^{[n+1]} = \mathcal{P}_{\ell_1}(x^{[n]} - \beta G(x^{[n+1]}), \tau)$

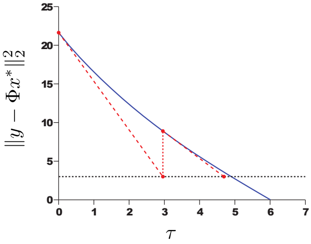
- Converges linearly, but much faster than IST.
- $\mathcal{O}(n)$ projection onto ℓ_1 ball for large scale problems.

Gradient Projection for Basis Pursuit and Basis Pursuit Denoising Problems

LASSO solution for a given τ

$$x^*(\tau) = \operatorname{argmin}_x \|y - \Phi x\|_2^2 \text{ s. t. } \|x\|_1 \le \tau$$

- $\mathbf{x}^*(\tau)$ is a differentiable convex function.
- $\mathbf{x}^*(\tau)$ is the solution of BP or BPDN if $\|y - \Phi x^*(\tau)\|_2 = 0$ or $\sqrt{\epsilon}$ respectively.
- A root finding problem $\Rightarrow$ Newton's method to solve $x^*(\tau) = 0$ or $x^*(\tau) - \epsilon = 0$



Gradient Projection for BP(DN)

1. $x^*(\tau^{[n]})$ by solving LASSO problem.
2. $\tau^{[n+1]}$ from Newton's update step.

Broader Range Sparse Approximation Methods

- **Analysis sparsity:** Ωx is sparse or compressible with a linear operator Ω,
$$\operatorname{argmin}_x \|\Omega x\|_1 + \lambda \|y - \Phi x\|_2^2$$
- **Total Variation (TV) norm:** sparsity in the gradient domain,
$$\operatorname{argmin}_x \|\nabla x\|_1 + \lambda \|y - \Phi x\|_2^2$$
- **Weighted ℓ_1 norm:** non-normalised dictionaries and iterative re-weighting
$$\operatorname{argmin}_x \sum_i w_i |x_i| + \lambda \|y - \Phi x\|_2^2$$

Augmented Lagrangian Method

- Problem: $\text{argmin}_x\, f(x)$ s. t. $y = \Phi x$

Augmented Lagrangian (AL)

- Surrogate objective:
$$L_\mu(x,\lambda) = f(x) + \lambda^T(\Phi x - y) + \frac{\mu}{2}\|\Phi x - y\|_2^2$$
- The aim is to solve $\text{argmin}_x \max_\lambda L_\mu(x,\lambda)$

- **Augmented Lagrangian Method** is a practical approach to find such a saddle point.
- The convergence of the iterative method is guaranteed for some f's.

Augmented Lagrangian Method

1. $x^{[n+1]} = \text{argmin}_x\, L_\mu(x, \lambda^{[n]})$
2. $\lambda^{[n+1]} = \lambda^{[n]} + \mu(\Phi x^{[n+1]} - y)$

Augmented Lagrangian for Variable Splitting

$$\min_x \|\Omega x\|_1 + \lambda \|y - \Phi x\|_2^2 \Longleftrightarrow \min_{x, z = \Phi x} \|z\|_1 + \lambda \|y - \Phi x\|_2^2$$

- Introducing auxiliary parameter $z = \Phi x$ and solving the constrained problem.
- The technique is also called Alternating Directions Method of Multipliers (ADMM) [Eckstein and Bertsekas 92].
- For parameter update:
  $x^{[n+1]}, z^{[n+1]} = \operatorname{argmin}_{x,z} \|z\|_1 + \lambda \|y - \Phi x\|_2^2 + \frac{\mu}{2} \left\|\Phi x - z - d^{[n]}\right\|_2^2$

ADMM

1. $x^{[n+1]} = \operatorname{argmin}_x \|\Phi x - y\|_2^2 + \frac{\mu}{2} \left\|\Phi x - z^{[n]} - d^{[n]}\right\|_2^2$
2. $z^{[n+1]} = \operatorname{argmin}_z \|z\|_1 + \frac{\mu}{2} \left\|\Phi x^{[n+1]} - z - d^{[n]}\right\|_2^2$
3. $d^{[n+1]} = d^{[n]} - (\Phi x^{[n+1]} - z^{[n+1]})$

Iterative Re-weighting For Sparse Approximation

Problem: $\operatorname{argmin}_x \|x\|_p^p + \lambda \|y - \Phi x\|_2^2, \quad \|x\|_p^p = \sum_i |x_i|^p,\ 0 < p < 1$

- Non-convex and hard to exactly solve it.
- It is sometimes practically preferred as an alternative to the convex formulation.
- A local minimum can be found by MM method using Taylor's first order approximation, $|\alpha|^p \leq |\frac{1}{(|\alpha^{[n]}|+\epsilon)^{1-p}}\alpha|$ with small positive ϵ [Candes et al. 08].

Iterative Re-weighted ℓ_1

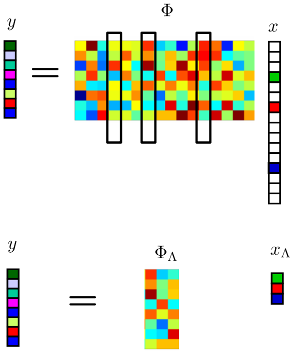
1. $x^{[n+1]} = \operatorname{argmin}_x \sum_i w_i^{[n]} |x_i| + \lambda \|y - \Phi x\|_2^2$
2. $w_i^{[n]} = \frac{1}{(|x_i^{[n+1]}|+\epsilon)^{1-p}}$

Greedy Methods for Sparse Approximations

- Finding a support Λ by iteratively adding one or more new atoms to the support.

$$x_\Lambda = \Phi_\Lambda^\dagger y$$

- Computationally cheaper than convex optimisation methods.
- It can be easily modified to consider extra structures in the representations.



Matching Pursuit

- Adding the atom which is the most fit to the remaining signal $r^{[n]} = y - \Phi x_{\Lambda^{[n]}}$ [Mallat and Zhang 93].
- It is guaranteed to reduce the energy of the remaining signal energy.
- Iterates until the energy of $r^{[n]}$ becomes small or for certain number of iterations.
- MP converges exponentially, with the incoherent dictionaries.

Matching Pursuit (MP)

1. $\Lambda^{[n+1]} = \Lambda^{[n]} \cup \{j^*\}, \qquad j^* = \mathrm{argmax}_j \, |(\Phi^T(y - \Phi x^{[n]}))_j|$
2. $x^{[n+1]} = x^{[n+1]} + (\Phi^T(y - \Phi x^{[n]}))_{j^*} e_{j^*}$

- e_{j^*} is the canonical basis for the j^*th coordinate.

Orthogonal Matching Pursuit

- No mechanism to not reselect already selected atom in MP.
- The convergence rate can be very slow with coherent dictionaries.
- Orthogonal MP [Pati et al 93] finds the best signal representation, given the support, at each iteration.

Orthogonal Matching Pursuit (OMP)

1. $\Lambda^{[n+1]} = \Lambda^{[n]} \cup \{j^*\}, \qquad j^* = \mathrm{argmax}_j \left|(\Phi^T(y - \Phi x^{[n]}))_j\right|$
2. $x^{[n+1]} = \mathrm{argmin}_z \|y - \Phi z\|_2^2\,, \text{ s. t. } \mathrm{supp}(z) = \Lambda^{[n+1]}$

- The minimisation step can be done using the pseudo-inverse of $\Phi_{\Lambda^{[n+1]}}$
- Matrix inversion can be done with efficient matrix factorisation techniques, *e.g.* QR, Cholesky factorsation.

Compressive Sampling Matching Pursuit

- No deselection strategy in MP or OMP.
- **Compressive Sampling MP (CoSaMP)** [Tropp and Needell 09] is a variant of MP which has a backward deselection step.
- It relies on the best K-term approximation operator $\mathcal{H}_K(\cdot)$, which selects the largest K coefficients and lets the rest be zero.

CoSaMP

1. $\widehat{\Lambda}^{[n+1]} = \Lambda^{[n]} \cup \mathrm{supp}(\mathcal{H}_{2K}(\Phi^T(y - \Phi x^{[n]})))$
2. $\widehat{x}^{[n+1]} = \mathrm{argmin}_z \|y - \Phi z\|_2^2, \text{ s.t. } \mathrm{supp}(z) = \widehat{\Lambda}^{[n+1]}$
3. $x^{[n+1]} = \mathcal{H}_K(\widehat{x}^{[n+1]})$
4. $\Lambda^{[n+1]} = \mathrm{supp}(x^{[n+1]})$

Iterative Hard Thresholding: K-sparse Approximation

$$\operatorname{argmin}_x \|y - \Phi x\|_2^2 \text{ s.t. } \|x\|_0 \leq K$$

- **Differentiable objective**
- **Projection** onto the K sparse set is easy, *i.e.* $\mathcal{H}_K(\cdot)$.
- Projected Gradient technique for finding a good solution.
- The quality of solution depends on the initial point and gradient step.

IHT(K) [Blumensath and Davies 08]

1. $G(x^{[n+1]}) = -2\Phi^T(\Phi x^{[n]} - y)$
2. $x^{[n+1]} = \mathcal{H}_K(x^{[n]} - \beta G(x^{[n+1]}))$

- β can be fixed, *e.g.* $\beta \leq \frac{1}{\|\Phi\|}$, or be adaptively selected.

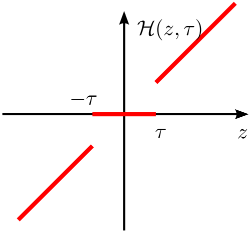
Iterative Hard Thresholding: Lagrangian Formulation

$$\text{argmin}_x \, \|x\|_0 + \lambda \, \|y - \Phi x\|_2^2$$

- We know the solution of the decoupled problem,

$$x^* = \underset{x}{\text{argmin}} \, \|x\|_0 + \lambda \, \|z - x\|_2^2$$

$$\Rightarrow x_i^* = \mathcal{H}(z_i, \frac{1}{\sqrt{\lambda}}) = \begin{cases} 0 & |z_i| \leq \frac{1}{\sqrt{\lambda}} \\ z_i & |z_i| > \frac{1}{\sqrt{\lambda}} \end{cases}$$



- MM technique for decoupling the parameters.

IHT(λ) [Blumensath and Davies 08]

$$x^{[n+1]} = \underset{x}{\text{argmin}} \, \|x\|_0 + \frac{L}{2} \left\| x - (x^{[n]} - \frac{2}{L}\Phi^T(\Phi x^{[n]} - y)) \right\|_2^2$$

$$= \mathcal{H}(x^{[n]} - \frac{2}{L}\Phi^T(\Phi x^{[n]} - y), \frac{1}{\sqrt{\lambda}})$$

Algorithm Selection

Convex optimisation $\leftrightsquigarrow$ Greedy/Iterative

- **Computational power?**
- **Online v.s Offline computation?**
- **Sparsity v.s. Compressibility?**
- **Accuracy of the solution?**
- **Guarantee of the recovery?**