Optimization Techniques for Alphabet-Constrained Signal Design

Mojtaba Soltanalian

Department of Electrical Engineering
California Institute of Technology

Stanford EE- ISL
Mar. 2015

Outline

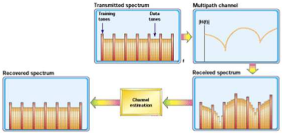
Signal Design- some applications

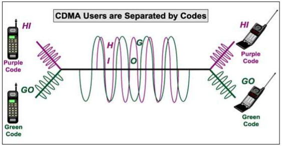
- **Signal design for active sensing.**
  Goal: To acquire (or preserve) the maximum information from the desirable sources in the environment.
- Signal is a medium to collect information.

- The research in this area is focused on the design and optimization of probing signals to improve target detection performance, as well as the target location and speed estimation accuracy.

Signal Design- some applications

- **Signal design for communications.**
  Goal: To transfer the maximum information among chosen agents in the network.



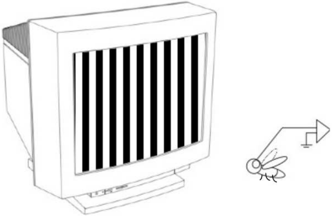


- Applications in *Channel Estimation, Code-Division Multiple-Access (CDMA) Schemes, Synchronization, Beamforming, . . .*

Signal Design- some applications

- **Signal design for life sciences.**
  Goal: To make the best identification of the living organism, usually by *maximal excitation.*

. . .



Signal Design- Keywords

- Waveform design and diversity (signal processing- communications)
- Input design (control- system identification)
- Sequence design (signal processing- information theory- communications- mathematics)
- Stimulus design, excitation design.

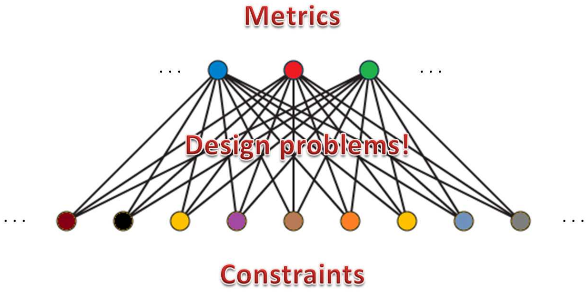
Signal Design- Metrics

- Mean-Square Error (MSE) of parameter estimation
- Signal-to-Noise Ratio (SNR) of the received data
- Information-Theoretic criteria
- Auto/Cross Correlation Sidelobe metrics
- Excitation metrics
- Stability metrics
- Secrecy metrics
- . . .

Signal Design- Constraints

- Energy
- Peak-to-Average Power Ratio (PAPR, PAR)
- Unimodularity (being Constant-Modulus)
- Finite or Discrete-Alphabet
  (integer, binary, *m*-ary constellation)
- . . .

Signal Design



Many of these problem are shown to be NP-hard;
Many others are *deemed* to be difficult!

Challenges:

How to handle signal constraints?
–and how to do it fast?

Signal Design- Methodologies

Useful design techniques:

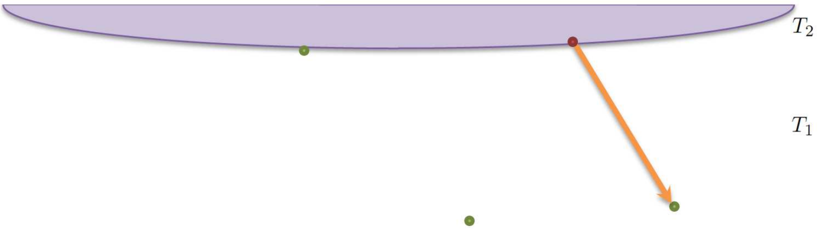
- Alternating Projections on Converging Sets (ALPS-CS)
- Power Method-Like Iterations
- MERIT

Alternating Projections on Converging Sets (ALPS-CS)

- Alternating Projections for signal design
- Alternating Projections
  convex vs non-convex, finite-alphabet

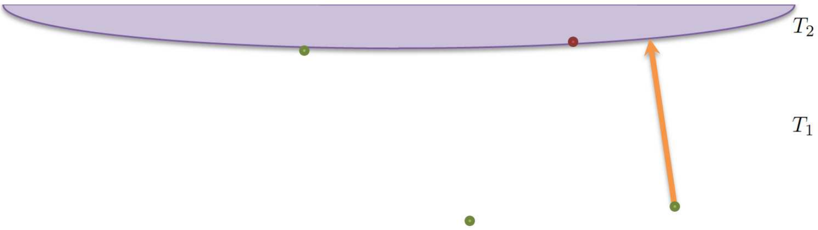
Alternating Projections on Converging Sets (ALPS-CS)

- Alternating Projections
  convex vs non-convex, finite-alphabet
- Example: T_1 a set with 3 elements (green dots); T_2 a convex set.



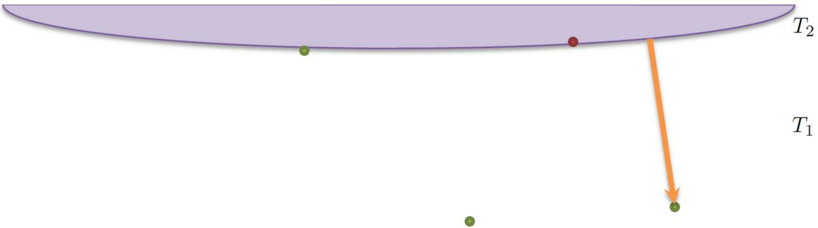
Alternating Projections on Converging Sets (ALPS-CS)

- Alternating Projections
  convex vs non-convex, finite-alphabet
- Example: T_1 a set with 3 elements (green dots); T_2 a convex set.



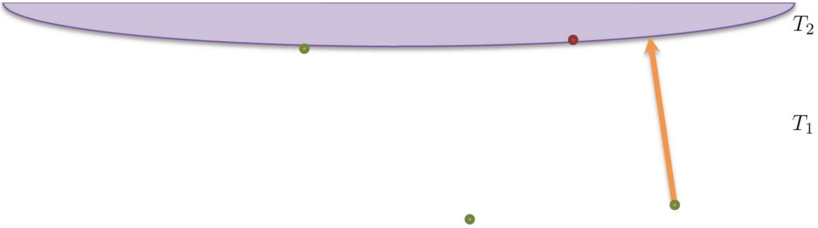
Alternating Projections on Converging Sets (ALPS-CS)

- Alternating Projections
  convex vs non-convex, finite-alphabet
- Example: T_1 a set with 3 elements (green dots); T_2 a convex set.



Alternating Projections on Converging Sets (ALPS-CS)

- Alternating Projections
  convex vs non-convex, finite-alphabet
- Example: T_1 a set with 3 elements (green dots); T_2 a convex set.

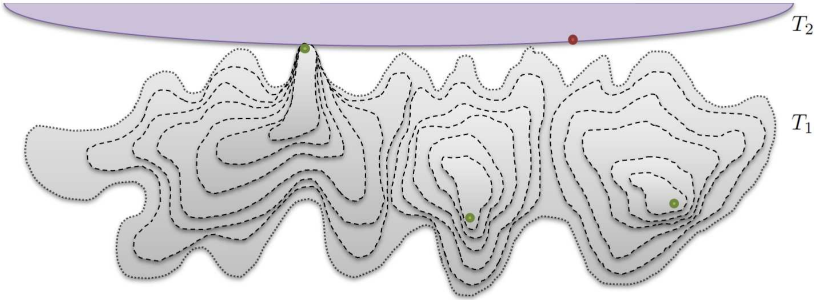


- Significant possibility of getting stuck in a poor "solution".

Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the "tricky" set with a well-behaved (perhaps compact/convex) set that in limit converges to the "tricky" set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the "tricky" set.

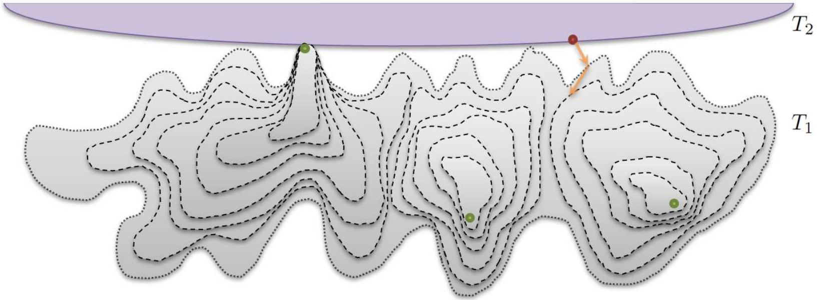
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the “tricky” set with a well-behaved (perhaps compact/convex) set that in limit converges to the “tricky” set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the “tricky” set.

Example: similar to the one before!

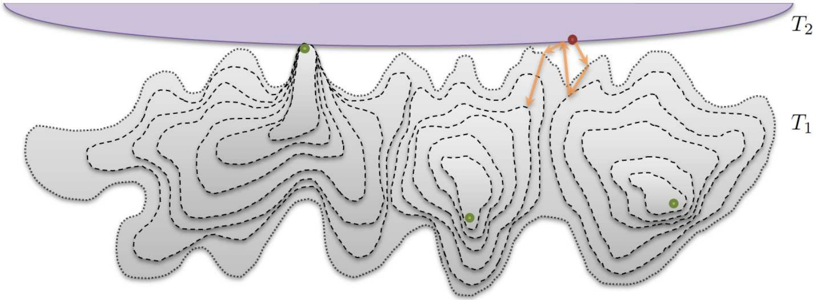


Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:

To replace the "tricky" set with a well-behaved (perhaps compact/convex) set that in limit converges to the "tricky" set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the "tricky" set.

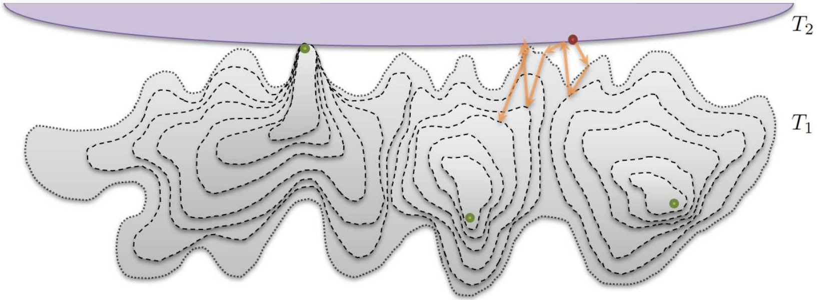
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the “tricky” set with a well-behaved (perhaps compact/convex) set that in limit converges to the “tricky” set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the “tricky” set.

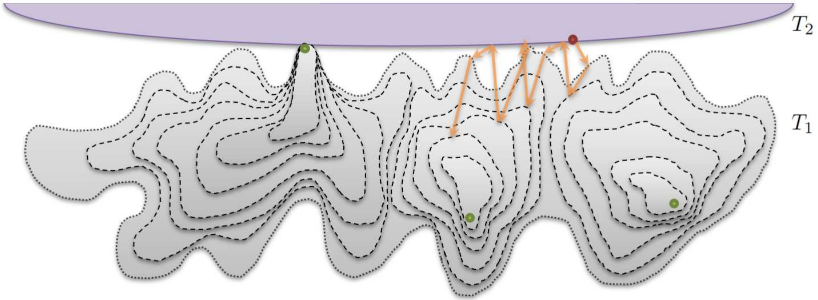
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the “tricky” set with a well-behaved (perhaps compact/convex) set that in limit converges to the “tricky” set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the “tricky” set.

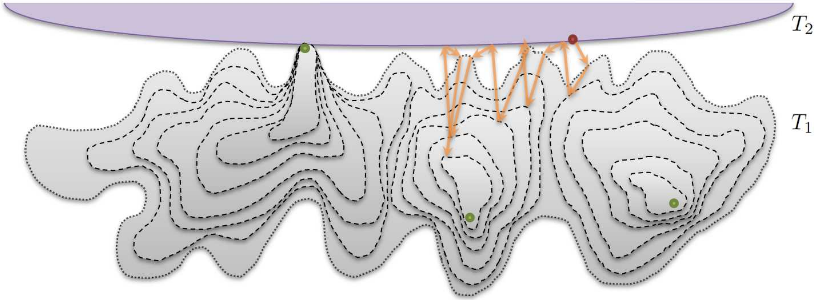
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the "tricky" set with a well-behaved (perhaps compact/convex) set that in limit converges to the "tricky" set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the "tricky" set.

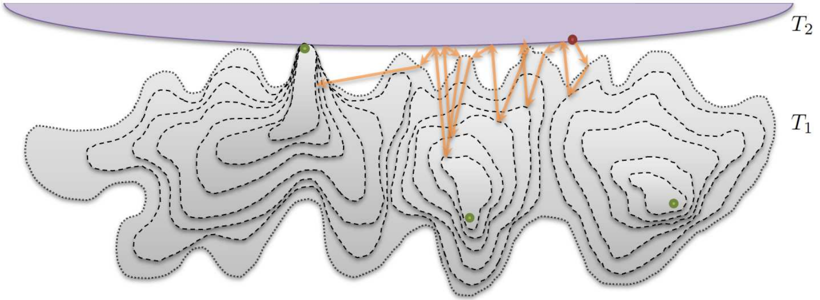
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Central Idea:
To replace the "tricky" set with a well-behaved (perhaps compact/convex) set that in limit converges to the "tricky" set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the "tricky" set.

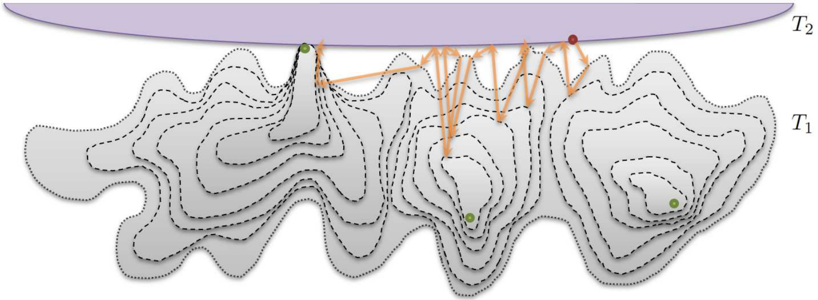
Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

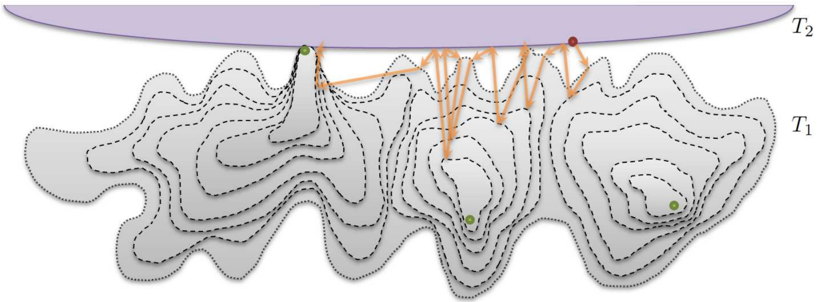
Central Idea:
To replace the "tricky" set with a well-behaved (perhaps compact/convex) set that in limit converges to the "tricky" set of interest! Then we employ the typical alternating projections, while the replaced set, at each iteration, gets closer to the "tricky" set.

Example: similar to the one before!



Alternating Projections on Converging Sets (ALPS-CS)

Why should this work?



Alternating Projections on Converging Sets (ALPS-CS)

- Selection of the converging sets can be done by choosing a converging function. Example ($\nu > 0$)

(a) $T = \mathbb{R} - \{0\}, \ T^{\dagger} = \{-1, 1\}$:

$$f(t,s) = \mathrm{sgn}(t) \cdot |t|^{e^{-\nu s}}; \tag{1}$$

(b) $T = \mathbb{C} - \{0\}, \ T^{\dagger} = \{\zeta \in \mathbb{C} \mid |\zeta| = 1\}$:

$$f(t,s) = |t|^{e^{-\nu s}} \cdot e^{j\arg(t)}. \tag{2}$$

Alternating Projections on Converging Sets (ALPS-CS)

- If the associated function f is **monotonic** and **identity**, we can show the convergence.
- How to choose f "optimally"? (open problem)
- For more details, see

  "Computational Design of Sequences with Good Correlation Properties," IEEE Transactions on Signal Processing, vol. 60, no. 5, pp. 2180-2193, 2012.

- **A Numerical Example**

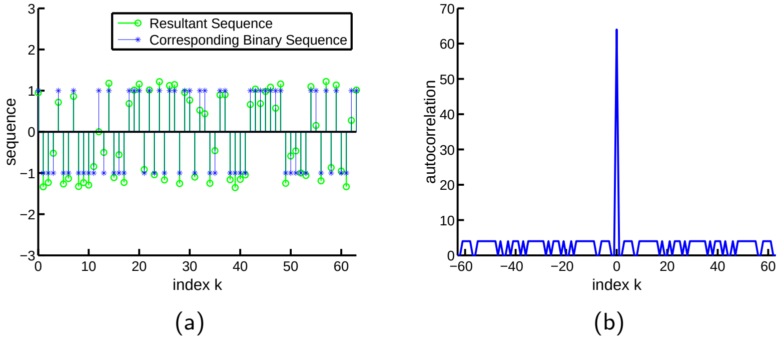


Figure: Design of a binary sequence of length 64 with good periodic auto-correlation using ALPS-CS. (a) the sequence provided by ALPS-CS when stopped, along with the corresponding binary sequence (obtained by clipping). The autocorrelation of the binary sequence is shown in (b).

ALPS-CS requires a design of the alternating projections as well as a suitable choice of converging function.

Let's see a simpler method!

Power Method-Like Iterations

- Many signal design problems can be formulated as (a sequence of) quadratic programs (QPs): SNR maximization, CRLB minimization, MSE minimization, beam-pattern matching, optimization of information-theoretic criteria, low-rank recovery, maximum-likelihood.

Power Method-Like Iterations

- Many signal design problems can be formulated as (a sequence of) quadratic programs (QPs): SNR maximization, CRLB minimization, MSE minimization, beam-pattern matching, optimization of information-theoretic criteria, low-rank recovery, maximum-likelihood.
- Some may need more sophisticated ideas for transformation to QP: fractional programming, MM algorithm, cyclic optimization, over-parametrization, etc.

Power Method-Like Iterations

- Formulation:

$$\max_{\boldsymbol{s}\in\mathbb{C}^n} . \quad \boldsymbol{s}^H \boldsymbol{R} \boldsymbol{s} \qquad (3)$$
$$\text{s. t.} \quad \boldsymbol{s} \in \Omega$$
$$(\Omega : \text{ search space})$$

- We can usually assume that the signal power is fixed: (why?)

$$\max_{\boldsymbol{s}\in\mathbb{C}^n} . \quad \boldsymbol{s}^H \boldsymbol{R} \boldsymbol{s} \qquad (4)$$
$$\text{s. t.} \quad \boldsymbol{s} \in \Omega$$
$$\|\boldsymbol{s}\|_2^2 = n.$$

Power Method-Like Iterations

Central Idea

- Assume $\boldsymbol{R}$ is positive definite (or make it so).

Power Method-Like Iterations

Central Idea

- Assume $\boldsymbol{R}$ is positive definite (or make it so).
- Start from some feasible $\boldsymbol{s} = \boldsymbol{s}^{(0)}$, and form the sequence:

$$\boldsymbol{s}^{(t+1)} = \texttt{Proj}_{\Omega}\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right) \tag{5}$$

where $\texttt{Proj}_{\Omega}(\boldsymbol{x}) = \arg\min_{\boldsymbol{s}\in\Omega,\, \|\boldsymbol{s}\|_2^2=n} \|\boldsymbol{s}-\boldsymbol{x}\|_2$ denotes the nearest vector in the search space (l_2-norm sense).

Power Method-Like Iterations

Central Idea

- Assume $\boldsymbol{R}$ is positive definite (or make it so).
- Start from some feasible $\boldsymbol{s} = \boldsymbol{s}^{(0)}$, and form the sequence:

$$\boldsymbol{s}^{(t+1)} = \texttt{Proj}_{\Omega}\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right) \tag{5}$$

where $\texttt{Proj}_{\Omega}(\boldsymbol{x}) = \arg\min_{\boldsymbol{s}\in\Omega,\, \|\boldsymbol{s}\|_2^2=n} \|\boldsymbol{s}-\boldsymbol{x}\|_2$ denotes the nearest vector in the search space (l_2-norm sense).

- **The above power method-like iterations lead to a monotonic increase of the QP objective. $\rightarrow$ convergence!**

Power Method-Like Iterations

Central Idea

- Assume $\boldsymbol{R}$ is positive definite (or make it so).
- Start from some feasible $\boldsymbol{s} = \boldsymbol{s}^{(0)}$, and form the sequence:

$$\boldsymbol{s}^{(t+1)} = \mathtt{Proj}_{\Omega}\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right) \tag{5}$$

  where $\mathtt{Proj}_{\Omega}(\boldsymbol{x}) = \arg\min_{\boldsymbol{s}\in\Omega,\ \|\boldsymbol{s}\|_2^2=n} \|\boldsymbol{s}-\boldsymbol{x}\|_2$ denotes the nearest vector in the search space (l_2-norm sense).

- **The above power method-like iterations lead to a monotonic increase of the QP objective. $\rightarrow$ convergence!**

$\rightsquigarrow$ This is very fast! (No matrix inversion needed.)

Power Method-Like Iterations

Let's see some examples– Constraints:

- Unimodular $\boldsymbol{s}$ ($\Omega = \{s: \ |s| = 1\}^n$):

$$\boldsymbol{s}^{(t+1)} = \exp\left(j \arg\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right)\right) \tag{6}$$

. . . just keep the phase.

Power Method-Like Iterations

Let's see some examples– Constraints:

- Unimodular $\boldsymbol{s}$ ($\Omega = \{s : \ |s| = 1\}^n$):

$$\boldsymbol{s}^{(t+1)} = \exp\left(j \arg\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right)\right) \tag{6}$$

. . . just keep the phase.

- Binary $\boldsymbol{s}$ ($\Omega = \{-1, +1\}^n$):

$$\boldsymbol{s}^{(t+1)} = \operatorname{sgn}\left(\Re\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right)\right) \tag{7}$$

. . . just keep the sign.

Power Method-Like Iterations

Let's see some examples– Constraints:

- Unimodular $\boldsymbol{s}$ ($\Omega = \{s : \ |s| = 1\}^n$):

$$\boldsymbol{s}^{(t+1)} = \exp\left(j \arg\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right)\right) \tag{6}$$

. . . just keep the phase.

- Binary $\boldsymbol{s}$ ($\Omega = \{-1, +1\}^n$):

$$\boldsymbol{s}^{(t+1)} = \operatorname{sgn}\left(\Re\left(\boldsymbol{R}\boldsymbol{s}^{(t)}\right)\right) \tag{7}$$

. . . just keep the sign.

- Sparse $\boldsymbol{s}$ ($\|\boldsymbol{s}\|_0 \leq k$):

. . . just keep the k largest values of $\boldsymbol{R}\boldsymbol{s}^{(t)}$ (and scale). (8)

Power Method-Like Iterations

Transformations to QP

Example (beam-pattern matching, low-coherence sensing for radar): Given positive-definite $\{\boldsymbol{R}_k\}_{k=1}^{t}$ and non-negative $\{d_k\}_{k=1}^{t}$,

$$\begin{aligned} \min_{\boldsymbol{s}\in\mathbb{C}^n} . \quad & \textstyle\sum_{k=1}^{t} |\boldsymbol{s}^H \boldsymbol{R}_k \boldsymbol{s} - d_k|^2 \\ \text{s. t.} \quad & \boldsymbol{s} \in \Omega \\ & \|\boldsymbol{s}\|_2^2 = n. \end{aligned} \tag{9}$$

Power Method-Like Iterations

Transformations to QP

Example (beam-pattern matching, low-coherence sensing for radar): Given positive-definite $\{\boldsymbol{R}_k\}_{k=1}^t$ and non-negative $\{d_k\}_{k=1}^t$,

$$\begin{aligned} \min_{\boldsymbol{s}\in\mathbb{C}^n} . \quad & \textstyle\sum_{k=1}^t \left|\boldsymbol{s}^H \boldsymbol{R}_k \boldsymbol{s} - d_k\right|^2 \\ \text{s. t.} \quad & \boldsymbol{s} \in \Omega \\ & \|\boldsymbol{s}\|_2^2 = n. \end{aligned} \tag{9}$$

- Over-parametrized "almost-equivalent" form:

$$\begin{aligned} \min_{\boldsymbol{s},\{\boldsymbol{u}_k\}} . \quad & \textstyle\sum_{k=1}^t \left\|\boldsymbol{R}_k^{1/2}\boldsymbol{s} - \sqrt{d_k}\boldsymbol{u}_k\right\|^2 \\ \text{s. t.} \quad & \boldsymbol{s} \in \Omega, \ \|\boldsymbol{s}\|_2^2 = n; \\ & \|\boldsymbol{u}_k\|_2 = 1, \ 1 \le k \le t. \end{aligned} \tag{10}$$

Power Method-Like Iterations

Transformations to QP

Example (beam-pattern matching, low-coherence sensing for radar):

- Over-parametrized "almost-equivalent" form:

$$\begin{aligned} \min_{\boldsymbol{s}, \{\boldsymbol{u}_k\}} . \quad & \textstyle\sum_{k=1}^{t} \left\| \boldsymbol{R}_k^{1/2} \boldsymbol{s} - \sqrt{d_k} \boldsymbol{u}_k \right\|^2 \\ \text{s. t.} \quad & \boldsymbol{s} \in \Omega, \ \|\boldsymbol{s}\|_2^2 = n; \\ & \|\boldsymbol{u}_k\|_2 = 1, \ 1 \leq k \leq t. \end{aligned}$$

- Minimization with respect to $\boldsymbol{s}$ boils down to

$$\begin{aligned} \min_{\boldsymbol{s} \in \mathbb{C}^n} . \quad & \begin{pmatrix} \boldsymbol{s} \\ 1 \end{pmatrix}^H \begin{pmatrix} \sum_{k=1}^{t} \boldsymbol{R}_k & \sum_{k=1}^{t} \sqrt{d_k} \boldsymbol{R}_k^{1/2} \boldsymbol{u}_k \\ \sum_{k=1}^{t} \sqrt{d_k} \boldsymbol{u}_k^H \boldsymbol{R}_k^{1/2} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{s} \\ 1 \end{pmatrix} \\ \text{s. t.} \quad & \boldsymbol{s} \in \Omega \\ & \|\boldsymbol{s}\|_2^2 = n. \end{aligned}$$

Transformations to QP

For other examples, see

- Information-theoretic metrics:
  ** "Unified Optimization Framework for Multi-Static Radar Code Design Using Information-Theoretic Criteria," IEEE Transactions on Signal Processing, vol. 61, no. 21, pp. 5401-5416, 2013.*

- MSE:
  ** "Optimized Transmission for Centralized Estimation in Wireless Sensor Networks," Preprint.*

  ** "Training Signal Design for Massive MIMO Channel Estimation," Preprint.*

Power Method-Like Iterations

- For more details about power method-like iterations, see

  ** "Designing Unimodular Codes Via Quadratic Optimization," IEEE Transactions on Signal Processing, vol. 62, no. 5, pp. 1221-1234, 2014.*

  ** "Joint Design of the Receive Filter and Transmit Sequence for Active Sensing," IEEE Signal Processing Letters, vol. 20, no. 5, pp. 423-426, 2013.*

Power method is fast, but doesn't
reveal any information on where the signal quality stands
with regard to the *optimal* value of the design problem . . .

MERIT

- **MERIT** stands for a
  M*onotonically* **ER***ror-Bound* **I***mproving* **T***echnique*
  for Mathematical Optimization.
- It's a computational framework to obtain sub-optimality guarantees along with the approximate solutions.
- You want to know how much the solution can be trusted . . .

The Central Idea

Let $\mathcal{P}(v, x)$ be an optimization problem *structure* with given and optimization variables partitioned as (v, x).

Example

$$\mathbf{X} = \arg\max \quad \mathrm{tr}(\mathbf{RX}) \quad \text{s.t.} \quad \mathrm{tr}(\mathbf{QX}) \leq t \qquad \underset{\Longrightarrow}{\text{variable partitioning}} \qquad \begin{array}{l} \mathbf{R}, \mathbf{Q}, t \rightarrow v \\ \mathbf{X} \rightarrow x \end{array}$$

The Central Idea

Now suppose $\mathcal{P}(v, x)$ is a "difficult" optimization problem; however,

- A sequence $v_1, v_2, v_3, \cdots$ of v can be constructed such that the associated global optima of the problem, viz. $x_k = \arg\max_x \mathcal{P}(v_k, x)$ are known for any v_k, and the "distance" between v and v_k, is decreasing with k;

- A sub-optimality guarantee of the obtained solutions x_k can be efficiently computed using the distance between v and v_k.

The Central Idea

Then, **computational sub-optimality guarantees** is obtained along with the approximate solutions, that might

- outperform existing analytically derived sub-optimality guarantees, or
- be the only class of sub-optimality guarantees in cases where no *a priori* known analytical guarantees are available for the given problem.

An example:

Unimodular Quadratic Programming (UQP)

$$\text{UQP:} \quad \max_{\boldsymbol{s} \in \Omega^n} \boldsymbol{s}^H \boldsymbol{R} \boldsymbol{s} \tag{11}$$

where $\boldsymbol{R} \in \mathbb{C}^{n \times n}$ is a given Hermitian matrix, and
Ω represents the unit circle, i.e. $\Omega = \{s \in \mathbb{C} : \ |s| = 1\}$.

- UQP is NP-hard.

Application

$$\text{UQP:} \quad \max_{\boldsymbol{s} \in \Omega^n} \boldsymbol{s}^H \boldsymbol{R} \boldsymbol{s}$$

MERIT:

Build a sequence of matrices
(for which the UQP global optima are known)
whose distance from the given matrix $\boldsymbol{R}$ is decreasing.

Application

$$\text{UQP:} \quad \max_{\boldsymbol{s}\in\Omega^n} \boldsymbol{s}^H \boldsymbol{R} \boldsymbol{s}$$

Theorem

Let $\mathcal{K}(\boldsymbol{s})$ represent the set of matrices $\boldsymbol{R}$ for which a given $\boldsymbol{s} \in \Omega^n$ is the global optimizer of UQP. Then

1. *$\mathcal{K}(\boldsymbol{s})$ is a convex cone.*
2. *For any two vectors $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \Omega^n$, the one-to-one mapping (where $\boldsymbol{s}_0 = \boldsymbol{s}_1^* \odot \boldsymbol{s}_2$)*

$$\boldsymbol{R} \in \mathcal{K}(\boldsymbol{s}_1) \Longleftrightarrow \boldsymbol{R} \odot (\boldsymbol{s}_0 \boldsymbol{s}_0^H) \in \mathcal{K}(\boldsymbol{s}_2) \tag{12}$$

   holds among the matrices in $\mathcal{K}(\boldsymbol{s}_1)$ and $\mathcal{K}(\boldsymbol{s}_2)$.

UQP: Approximation of $\mathcal{K}(\boldsymbol{s})$

Theorem

For any given $\boldsymbol{s} = (e^{j\phi_1}, \cdots, e^{j\phi_n})^T \in \Omega^n$, *let* $\mathcal{C}(\boldsymbol{V}_{\boldsymbol{s}})$ *represent the convex cone of matrices* $\boldsymbol{V}_{\boldsymbol{s}} = \boldsymbol{D} \odot (\boldsymbol{s}\boldsymbol{s}^H)$ *where* $\boldsymbol{D}$ *is any real-valued symmetric matrix with non-negative off-diagonal entries. Also let* $\mathcal{C}_{\boldsymbol{s}}$ *represent the convex cone of matrices with* $\boldsymbol{s}$ *being their dominant eigenvector (i.e the eigenvector corresponding to the maximal eigenvalue). Then for any* $\boldsymbol{R} \in \mathcal{K}(\boldsymbol{s})$, *there exists* $\alpha_0 \geq 0$ *such that for all* $\alpha \geq \alpha_0$,

$$\boldsymbol{R} + \alpha \boldsymbol{s}\boldsymbol{s}^H \in \mathcal{C}(\boldsymbol{V}_{\boldsymbol{s}}) \oplus \mathcal{C}_{\boldsymbol{s}} \tag{13}$$

where $\oplus$ *stands for the Minkowski sum of the two sets.*

UQP: Approximation of $\mathcal{K}(\boldsymbol{s})$

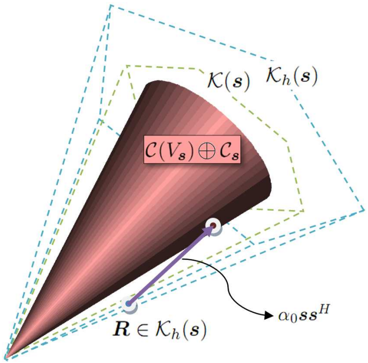


Figure: An illustration of the cone approximation technique used for MERIT's derivation in unimodular quadratic programming.

Application

UQP: MERIT Objective

- Using the previous results, we build a sequence of matrices (for which the UQP global optima are known) whose distance from the given matrix $\boldsymbol{R}$ is decreasing.

UQP: MERIT Objective

- Using the previous results, we build a sequence of matrices (for which the UQP global optima are known) whose distance from the given matrix $\boldsymbol{R}$ is decreasing.
- Instead of the original UQP, we consider the optimization problem:

$$\min_{\boldsymbol{s}\in\Omega^n, \boldsymbol{Q}_1\in\mathcal{C}_1, \boldsymbol{P}_1\in\mathcal{C}(\boldsymbol{V}_1)} \|\boldsymbol{R} - (\boldsymbol{Q}_1 + \boldsymbol{P}_1) \odot (\boldsymbol{s}\boldsymbol{s}^H)\|_F \quad (14)$$

UQP: MERIT Objective

- Using the previous results, we build a sequence of matrices (for which the UQP global optima are known) whose distance from the given matrix $\boldsymbol{R}$ is decreasing.
- Instead of the original UQP, we consider the optimization problem:

$$\min_{\boldsymbol{s}\in\Omega^n, \boldsymbol{Q}_1\in\mathcal{C}_1, \boldsymbol{P}_1\in\mathcal{C}(\boldsymbol{V}_1)} \|\boldsymbol{R} - (\boldsymbol{Q}_1 + \boldsymbol{P}_1) \odot (\boldsymbol{s}\boldsymbol{s}^H)\|_F \tag{14}$$

- $(\boldsymbol{Q}_1 + \boldsymbol{P}_1) \odot (\boldsymbol{s}\boldsymbol{s}^H)$ will get close to $\boldsymbol{R}$.

Application

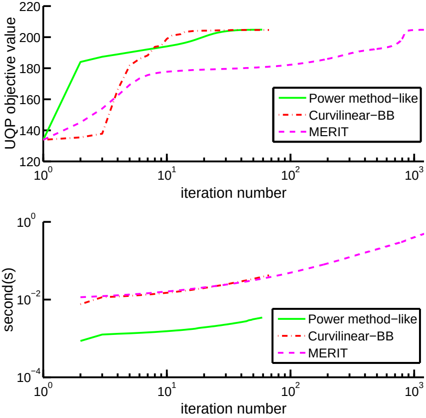


Figure: A comparison of power method-like iterations, the curvilinear search with Barzilai-Borwein step size, and MERIT: (top) the UQP objective; (bottom) the required time for approximating UQP solution ($n = 10$) with same initialization.

Application

n	Rank (d)	#problems for which $\gamma = 1$	Average γ	$\frac{\text{Average SDR time}}{\text{Average MERIT time}}$
8	*2*	17	0.9841	1.08
	8	16	0.9912	0.81
16	*2*	15	0.9789	2.08
	4	13	0.9773	0.95
	16	4	0.9610	0.92

Table: Comparison of the performance of MERIT and SDR when solving UQP for 20 random positive definite matrices of different sizes n and ranks d.

MERIT

- For more details on MERIT, see

  ** "Designing Unimodular Codes Via Quadratic Optimization," IEEE Transactions on Signal Processing, vol. 62, no. 5, pp. 1221-1234, 2014.*

  ** "Beyond Semidefinite Relaxtion: Basis Banks and Computationally Enhanced Guarantees," Submitted to IEEE International Symposium on Information Theory (ISIT), Hong Kong, 2015.*

Summary- what we discussed?

- Various signal design problems arise in practice.

- Signal design methodologies:

- Alternating Projections on Converging Sets (ALPS-CS)
- Power Method-Like Iterations
- MERIT