The Noisy Trigonometric Moment Problem

Badri Narayan

ECE Dept

April 27, 2010

Trigonometric Moment Problem

Given a sequence of m complex numbers $(x_k)_{k=1}^m$ and $x_0 > 0$, does there exist a positive Borel measure μ on $[0, 1]$ such that x_k is the kth trigonometric moment with respect to μ?

In other words, does there exist $\mu > 0$ supported on $[0, 1]$ such that

$$x_k = \int_0^1 e^{-i2\pi kt}\mu(dt)$$

for all $k \in \{0, \ldots, m\}$?

Note that x_k is simply the kth Fourier coefficient of μ.

Caratheodory's theorem

There exists a measure that solves the moment problem if and only if the Hermitian Toeplitz Matrix formed by $x_0, \ldots, x_m$, *viz.*

$$X = T(x) = \begin{pmatrix} x_0 & x_1 & \cdots & x_m \\ x_1^* & x_0 & \ddots & x_{m-1} \\ \vdots & \ddots & \ddots & \vdots \\ x_m^* & & \cdots & x_0 \end{pmatrix}$$

is positive semidefinite.

Caratheodory's theorem (contd.)

Furthermore, if $n = \text{rank}(X)$, there exists a finitely atomic measure

$$\mu(dt) = \sum_{i=1}^{n} c_i \delta(t - \theta_i) dt,$$

where c_is are positive and θ_is are in $[0, 1]$, solving the moment problem if and only if $X \succeq 0$.

Caratheodory's theorem (contd.)

Furthermore, if $n = \operatorname{rank}(X)$, there exists a finitely atomic measure

$$\mu(dt) = \sum_{i=1}^{n} c_i \delta(t - \theta_i) dt,$$

where c_is are positive and θ_is are in $[0, 1]$, solving the moment problem if and only if $X \succeq 0$.

Let us verify the necessary conditions. For the given μ, note that

$$x_k = \sum_{i=1}^{n} c_i e^{i2\pi k \theta_i}.$$

Caratheodory's theorem (contd.)

Furthermore, if $n = \text{rank}(X)$, there exists a finitely atomic measure

$$\mu(dt) = \sum_{i=1}^{n} c_i \delta(t - \theta_i) dt,$$

where c_is are positive and θ_is are in $[0, 1]$, solving the moment problem if and only if $X \succeq 0$.

Let us verify the necessary conditions. For the given μ, note that

$$x_k = \sum_{i=1}^{n} c_i e^{i 2\pi k \theta_i}.$$

Our key tool will be the Vandermonde Decomposition.

Vandermonde Decomposition

Any sequence of m consecutive values can be written as

$$\begin{pmatrix} x_{m+k-1} \\ \vdots \\ x_k \end{pmatrix} = [\vec{z}_n \cdots \vec{z}_1] \begin{pmatrix} c_1 e^{i2\pi k\theta_1} \\ \vdots \\ c_n e^{i2\pi k\theta_n} \end{pmatrix}$$

where $\vec{z}_i = [e^{i(m-1)\theta_i}, \ldots, e^{i\theta_i}, 1]^T$.

Vandermonde Decomposition

Any sequence of m consecutive values can be written as

$$\begin{pmatrix} x_{m+k-1} \\ \vdots \\ x_k \end{pmatrix} = [\vec{z}_n \cdots \vec{z}_1] \begin{pmatrix} c_1 e^{i2\pi k\theta_1} \\ \vdots \\ c_n e^{i2\pi k\theta_n} \end{pmatrix}$$

where $\vec{z}_i = [e^{i(m-1)\theta_i}, \dots, e^{i\theta_i}, 1]^T$.

Repeating this for $k = (1-m)$ to 1, we have

$$X = [\vec{z}_n \cdots \vec{z}_1] \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_n \end{pmatrix} \begin{pmatrix} \vec{z}_n^H \\ \vdots \\ \vec{z}_1^H \end{pmatrix} = \sum_{i=1}^{n} c_i \vec{z}_i \vec{z}_i^H$$

Clearly, $\text{rank}(X) = \min\{m, n\}$ and $X \succeq 0$

Prony's technique

Put $m = n$, and $\vec{\alpha} = [\alpha_n, \ldots, \alpha_1, 1]^T$. Then,

$$\vec{\alpha}^H X \vec{\alpha} = \sum_{i=1}^{n} c_i |\vec{\alpha}^H \vec{z}_i|^2$$

So, $\alpha^H X \vec{\alpha} = 0$ if and only if

Prony's technique

Put $m = n$, and $\vec{\alpha} = [\alpha_n, \ldots, \alpha_1, 1]^T$. Then,

$$\vec{\alpha}^H X \vec{\alpha} = \sum_{i=1}^{n} c_i |\vec{\alpha}^H \vec{z}_i|^2$$

So, $\alpha^H X \vec{\alpha} = 0$ if and only if

- $\vec{\alpha}$ is the unique null vector with last coordinate equal to unity.

Prony's technique

Put $m = n$, and $\vec{\alpha} = [\alpha_n, \dots, \alpha_1, 1]^T$. Then,

$$\vec{\alpha}^H X \vec{\alpha} = \sum_{i=1}^{n} c_i |\vec{\alpha}^H \vec{z}_i|^2$$

So, $\alpha^H X \vec{\alpha} = 0$ if and only if

- $\vec{\alpha}$ is the unique null vector with last coordinate equal to unity.
- $\vec{\alpha}$ are the coefficients of the unique monic polynomial with $e^{i2\pi\theta_1}, \dots, e^{i2\pi\theta_n}$ as roots.

Prony's technique

Put $m = n$, and $\vec{\alpha} = [\alpha_n, \ldots, \alpha_1, 1]^T$. Then,

$$\vec{\alpha}^H X \vec{\alpha} = \sum_{i=1}^{n} c_i |\vec{\alpha}^H \vec{z}_i|^2$$

So, $\alpha^H X \vec{\alpha} = 0$ if and only if

- $\vec{\alpha}$ is the unique null vector with last coordinate equal to unity.
- $\vec{\alpha}$ are the coefficients of the unique monic polynomial with $e^{i2\pi\theta_1}, \ldots, e^{i2\pi\theta_n}$ as roots.

Using the above, θ_is and hence c_is can be determined.

Prony's technique

Put $m = n$, and $\vec{\alpha} = [\alpha_n, \ldots, \alpha_1, 1]^T$. Then,

$$\vec{\alpha}^H X \vec{\alpha} = \sum_{i=1}^{n} c_i |\vec{\alpha}^H \vec{z}_i|^2$$

So, $\alpha^H X \vec{\alpha} = 0$ if and only if

- $\vec{\alpha}$ is the unique null vector with last coordinate equal to unity.
- $\vec{\alpha}$ are the coefficients of the unique monic polynomial with $e^{i2\pi\theta_1}, \ldots, e^{i2\pi\theta_n}$ as roots.

Using the above, θ_is and hence c_is can be determined.
Note: A stable reformulation is possible.

Noisy Trigonometric Moment Problem

Let's add some noise! Suppose we observe

$$x_k = \sum_{i=1}^{n} c_i e^{i2\pi k\theta_i} + \nu_k$$

for $0 \le k \le m$, where

- c_is are positive and θ_is are in $[0, 1]$.
- ν_k is *iid* Gaussian noise with variance σ^2.

If we have $m \gg n$ observations, can we alleviate the distortion due to noise?

Approach

Approach

- The Toeplitz Hermitian Matrix formed by $x_0, \ldots, x_n$ is low rank ($n \ll m$).

Approach

- The Toeplitz Hermitian Matrix formed by $x_0, \ldots, x_n$ is low rank ($n \ll m$).
- Every vector $[x_0, x_1, \ldots, x_m]^T \in \mathbb{R}_+ \times \mathbb{C}^m$ is a valid candidate for the moment problem whenever the associated hermitian toeplitz matrix $X \succeq 0$ (Caratheodory's theorem).

Approach

- The Toeplitz Hermitian Matrix formed by $x_0, \ldots, x_n$ is low rank ($n \ll m$).
- Every vector $[x_0, x_1, \ldots, x_m]^T \in \mathbb{R}_+ \times \mathbb{C}^m$ is a valid candidate for the moment problem whenever the associated hermitian toeplitz matrix $X \succeq 0$ (Caratheodory's theorem).

A naïve first attempt:

$$\begin{aligned} \text{minimize}_{z\in\Omega} \; & \|z - x\|^2 \\ \text{subject to } \; & \text{rank}(Z) \leq n \\ & Z \succeq 0 \end{aligned} \tag{P}$$

where $\Omega = \mathbb{R}_+ \times \mathbb{C}^m$ and $Z = T(z)$.

Approach

- The Toeplitz Hermitian Matrix formed by $x_0, \ldots, x_n$ is low rank ($n \ll m$).
- Every vector $[x_0, x_1, \ldots, x_m]^T \in \mathbb{R}_+ \times \mathbb{C}^m$ is a valid candidate for the moment problem whenever the associated hermitian toeplitz matrix $X \succeq 0$ (Caratheodory's theorem).

A naïve first attempt:

$$\begin{aligned} \text{minimize}_{z \in \Omega} \; & \|z - x\|^2 \\ \text{subject to} \; & \text{rank}(Z) \leq n \\ & Z \succeq 0 \end{aligned} \tag{P}$$

where $\Omega = \mathbb{R}_+ \times \mathbb{C}^m$ and $Z = T(z)$.
Our constraint set is not convex.

Reformulation

Let's use Lagrange Multipliers:

$$\begin{aligned} \text{minimize}_{z \in \Omega} \; & \|z - x\|^2 + \lambda \, \text{rank}(Z) \\ \text{subject to } \; & Z \succeq 0 \end{aligned} \tag{L}$$

Reformulation

Let's use Lagrange Multipliers:

$$\begin{aligned} &\text{minimize}_{z \in \Omega} \|z - x\|^2 + \lambda \operatorname{rank}(Z) \\ &\text{subject to } Z \succeq 0 \end{aligned} \tag{L}$$

Finally, relax and minimize the convex envelope of the non-convex objective above:

$$\begin{aligned} &\text{minimize}_{z \in \Omega} \|z - x\|^2 + \lambda \|Z\|_* \\ &\text{subject to } Z \succeq 0 \end{aligned} \tag{R}$$

Reformulation

Let's use Lagrange Multipliers:

$$\begin{aligned} \text{minimize}_{z \in \Omega} \; & \|z - x\|^2 + \lambda \, \text{rank}(Z) \\ \text{subject to } & Z \succeq 0 \end{aligned} \tag{L}$$

Finally, relax and minimize the convex envelope of the non-convex objective above:

$$\begin{aligned} \text{minimize}_{z \in \Omega} \; & \|z - x\|^2 + \lambda \|Z\|_* \\ \text{subject to } & Z \succeq 0 \end{aligned} \tag{R}$$

What λ? We'll try a range of values and pick the best. Let's code it up!

Interlude – Alternating Projections

Here's a competing algorithm.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

A few remarks are in order:

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

A few remarks are in order:

- This method is proposed by Cadzow based on Von Neumman's method to find the closest point in the intersection of two subspaces.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

A few remarks are in order:

- This method is proposed by Cadzow based on Von Neumman's method to find the closest point in the intersection of two subspaces.
- With two convex sets, we only get some point in the intersection, not the closest.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

A few remarks are in order:

- This method is proposed by Cadzow based on Von Neumman's method to find the closest point in the intersection of two subspaces.
- With two convex sets, we only get some point in the intersection, not the closest.
- But the space of low rank matrix is pretty hairy (certainly not convex). We don't have theoretical guarantees that this is even a good idea.

Interlude – Alternating Projections

Here's a competing algorithm.

- Use SVD to find the best rank-n approximation.
- Find the closest Toeplitz Hermitian approximation of the rank-n approximation.
- Alternate between the above two steps and repeat until convergence.

A few remarks are in order:

- This method is proposed by Cadzow based on Von Neumman's method to find the closest point in the intersection of two subspaces.
- With two convex sets, we only get some point in the intersection, not the closest.
- But the space of low rank matrix is pretty hairy (certainly not convex). We don't have theoretical guarantees that this is even a good idea.

Let's code this up too!

Alternating Projections

```
function X = cadzow_denoise(X, r)
% Find the closest PSD Toeplitz matrix that to X which
% has only r prominent eigen values.
tol = 0.001;
ratio = tol;
[n,~] = size(X);
  while ratio >= tol
    [U,S,V] = svd(X); %#ok<ASGLU,NASGU>
    ratio = S(r+1,r+1)/S(r,r);
    S((r+1):end,(r+1):end) = 0;
    cvx_begin quiet
        variable X(n,n) hermitian toeplitz;
        X == hermitian_semidefinite(n); %#ok<EQEFF>
        minimize norm(X-U*S*V');
    cvx_end
  end
end
```

Nuclear Norm Relaxation

```
clc; clear; N = 5;
theta = [0.2 0.5 0.7]; c = [3 5 7];
x = fliplr([c zeros(1,N)]*vander([exp(1i*2*pi*theta) zeros(1,N)]));
y = awgn(x,5); y(1) = abs(y(1)); n = length(y);
gamma = logspace(-2.5,0,20);
sos = zeros(size(gamma));
fprintf('X_* = %.2f, (y-x) = %.2f\n\n',norm_nuc(toeplitz(x)),sum_square_abs(y-x));
fprintf('.-------------------------------------------------------.\n')
fprintf('|  Status   |   gamma  |    (z-x)    |    (z-y)    |    Z_*    |\n')
fprintf('|-------------------------------------------------------|\n')
for k = 1:length(gamma)
    cvx_begin quiet
      variable Z(n,n) toeplitz hermitian;
      z = Z(1,:);
      minimize sum_square_abs(z-y)  + gamma(k)*norm_nuc(Z);
      subject to
        Z(1) == x(1);
    cvx_end
    z=Z(1,:);sos(k) = sum_square_abs(z-x);
    fprintf('| %8s | %8.4f |  %8.2f |  %8.2f | %6.2f  |\n', ...
    cvx_status, gamma(k), sos(k), sum_square_abs(y-z), norm_nuc(Z));
end
fprintf('.-------------------------------------------------------.\n')
fprintf('\nAfter Nuclear Norm minimization...\n');
fprintf('Best distance from true x = %.2f\n\n',min(sos));
W = cadzow_denoise(toeplitz(y),3); w = W(1,:);
fprintf('\nAfter Cadzow denoising...\n');
fprintf('Best distance from true x = %.2f\n\n',sum_square_abs(w-x));
```

Simulation Time

```
X_* = 120.00, (y-x) = 2.21
```

Status	gamma	(z-x)	(z-y)	Z_*
Solved	0.0043	1.78	0.42	124.18
Solved	0.0058	1.78	0.42	124.16
Solved	0.0078	1.77	0.42	124.15
Solved	0.0106	1.76	0.42	124.12
Solved	0.0144	1.75	0.42	124.08
Solved	0.0195	1.74	0.42	124.04
Solved	0.0264	1.72	0.43	123.98
Solved	0.0357	1.70	0.44	123.90
Solved	0.0483	1.68	0.45	123.81
Solved	0.0654	1.65	0.47	123.71
Solved	0.0886	1.62	0.51	123.60
Solved	0.1199	1.61	0.58	123.47
Solved	0.1624	1.62	0.70	123.32
Solved	0.2198	1.68	0.89	123.13
Solved	0.2976	1.84	1.20	122.89
Solved	0.4030	2.18	1.75	122.64
Solved	0.5456	2.88	2.73	122.41
Solved	0.7386	4.22	4.49	122.21

```
After Nuclear Norm minimization...
Best distance from true x = 1.61

After Cadzow denoising...
Best distance from true x = 0.25
```

Conclusions and Future Work

Conclusions and Future Work

- The Nuclear Norm Minimization is too sensitive to λ.

Conclusions and Future Work

- The Nuclear Norm Minimization is too sensitive to λ.
- With the positive semidefinite constraint, cvx failed to return a solution.

Conclusions and Future Work

- The Nuclear Norm Minimization is too sensitive to λ.
- With the positive semidefinite constraint, cvx failed to return a solution.
- Penalize shrinkage?

Conclusions and Future Work

- The Nuclear Norm Minimization is too sensitive to λ.
- With the positive semidefinite constraint, cvx failed to return a solution.
- Penalize shrinkage?
- Reformulate the problem as an SDP and try again?

Conclusions and Future Work

- The Nuclear Norm Minimization is too sensitive to λ.
- With the positive semidefinite constraint, cvx failed to return a solution.
- Penalize shrinkage?
- Reformulate the problem as an SDP and try again?
- Must find out why Cadzow does so well.

Thank You!

Questions?