Polynomial system solving in applications

Julia Lindberg

University of Wisconsin-Madison

LAAS BrainPOP Seminar
Polynomial systems arise naturally in many applications

- Lagrange systems of polynomial optimization problems
- Gaussian mixture moment systems
- Power flow equations
Polynomial systems arise naturally in many applications

Solutions to these systems often have physical meaning
Motivation

- Polynomial systems arise naturally in many applications
- Solutions to these systems often have physical meaning
- This talk will outline techniques to solve these systems
Motivation

- Polynomial systems arise naturally in many applications
- Solutions to these systems often have physical meaning
- This talk will outline techniques to solve these systems
- We will consider three examples
  1. Lagrange systems of polynomial optimization problems
  2. Gaussian mixture moment systems
  3. Power flow equations
Consider the variety defined by polynomials $f_1, \ldots, f_n \in \mathbb{R}[x_1, \ldots, x_n]$

$$\mathcal{V}(F) = \{x \in \mathbb{C}^n : f_1(x) = 0, \ldots, f_n(x) = 0\}$$

where $\dim(\mathcal{V}(F)) = 0$
Consider the variety defined by polynomials $f_1, \ldots, f_n \in \mathbb{R}[x_1, \ldots, x_n]$

$$\mathcal{V}(F) = \{ x \in \mathbb{C}^n : f_1(x) = 0, \ldots, f_n(x) = 0 \}$$

where $\dim(\mathcal{V}(F)) = 0$

**Goal:** Find all points in $\mathcal{V}(F)$
<table>
<thead>
<tr>
<th></th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Homotopy Continuation</td>
</tr>
<tr>
<td>2</td>
<td>Example 1: Polynomial optimization</td>
</tr>
<tr>
<td>3</td>
<td>Example 2: Gaussian mixture moment systems</td>
</tr>
<tr>
<td>4</td>
<td>Monodromy</td>
</tr>
<tr>
<td>5</td>
<td>Example 3: Power flow equations</td>
</tr>
</tbody>
</table>
Method 1
Homotopy continuation

- Idea: Solving most polynomial systems is hard, but some are easy

\[ H_T = \begin{cases} \frac{2(x_2^2 - x_1 y_1) + 3y_1}{y_1}, \frac{2(x_1 y_2 - x_2 y_1)}{y_2}, \frac{x_2^2 + y_2^2}{x_2}, \frac{x_2^2 + y_2^2}{y_2} \end{cases} = 0 \]

\[ H_S = \begin{cases} x_2^1 = 1, x_2^2 = 1, y_2^1 = 1, y_2^2 = 1 \end{cases} \]

Can I map my solutions from \( H_S \) to \( H_T \)?

Define \( H_t := (1-t)H_S + tH_T \) and compute \( H_t \) as \( t \to 1 \)

Called following homotopy paths

Typically use predictor-corrector methods

Predict: Take step along tangent direction at a point
Correct: Use Newton's method
Method 1
Homotopy continuation

- Idea: Solving most polynomial systems is hard, but some are easy

\[ H_T = \begin{cases} 
2(x_2 y_1 - x_1 y_2) + 3 y_1 = 0 \\
2(x_1 y_2 - x_2 y_1) + 4 y_2 = 0 \\
x_1^2 + y_1^2 = 1 \\
x_2^2 + y_2^2 = 1 
\end{cases} \]

\[ H_S = \begin{cases} 
x_1^2 = 1 \\
x_2^2 = 1 \\
y_1^2 = 1 \\
y_2^2 = 1 
\end{cases} \]
Method 1
Homotopy continuation

- Idea: Solving most polynomial systems is hard, but some are easy

\[ H_T = \begin{cases} 
2(x_2y_1 - x_1y_2) + 3y_1 = 0 \\
2(x_1y_2 - x_2y_1) + 4y_2 = 0 \\
x_1^2 + y_1^2 = 1 \\
x_2^2 + y_2^2 = 1 
\end{cases} \quad H_S = \begin{cases} 
x_1^2 = 1 \\
x_2^2 = 1 \\
y_1^2 = 1 \\
y_2^2 = 1 
\end{cases} \]

- Can I map my solutions from \( H_S \) to \( H_T \)?
Method 1
Homotopy continuation

- Idea: Solving most polynomial systems is hard, but some are easy

\[ H_T = \begin{cases} 
2(x_2 y_1 - x_1 y_2) + 3y_1 = 0 \\
2(x_1 y_2 - x_2 y_1) + 4y_2 = 0 \\
x_1^2 + y_1^2 = 1 \\
x_2^2 + y_2^2 = 1 
\end{cases} \quad H_S = \begin{cases} 
x_1^2 = 1 \\
x_2^2 = 1 \\
y_1^2 = 1 \\
y_2^2 = 1 
\end{cases} \]

- Can I map my solutions from \( H_S \) to \( H_T \)?

- Define \( H_t := (1 - t)H_S + tH_T \) and compute \( H_t \) as \( t \to 1 \)
  - Called following homotopy paths
**Method 1**

**Homotopy continuation**

- Idea: Solving most polynomial systems is hard, but some are easy

\[
H_T = \begin{cases} 
2(x_2y_1 - x_1y_2) + 3y_1 = 0 \\
2(x_1y_2 - x_2y_1) + 4y_2 = 0 \\
x_1^2 + y_1^2 = 1 \\
x_2^2 + y_2^2 = 1
\end{cases} \quad \text{and} \quad H_S = \begin{cases} 
x_1^2 = 1 \\
x_2^2 = 1 \\
y_1^2 = 1 \\
y_2^2 = 1
\end{cases}
\]

- Can I map my solutions from \( H_S \) to \( H_T \)?

- Define \( H_t := (1 - t)H_S + tH_T \) and compute \( H_t \) as \( t \to 1 \)
  - Called following *homotopy paths*

- Typically use predictor-corrector methods
  - Predict: Take step along tangent direction at a point
  - Correct: Use Newton’s method
Figure: The homotopy $H_t = (1 - t) H_S + t H_T$ (left) [KW14] and a predictor corrector step (right) [BT18]
Homotopy continuation

Start system

Want to pick a start system, $H_S$, such that

1. The solutions of $H_S$ are easy to find
2. The number of solutions to $H_S \approx$ the number of solutions to $H_T$
Want to pick a start system, $H_S$, such that

1. The solutions of $H_S$ are easy to find
2. The number of solutions to $H_S \approx$ the number of solutions to $H_T$

Need to know $|\mathcal{V}(F)|$
Theorem (Bezout)

\[ |\mathcal{V}(F)| \leq d_1 \cdots d_n \text{ where } d_i = \deg(f_i) \]
Theorem (Bezout)

$$|\mathcal{V}(F)| \leq d_1 \cdots d_n \text{ where } d_i = \deg(f_i)$$

- If $$|\mathcal{V}(F)| \approx d_1 \cdots d_n$$ then a **total degree** start system is suitable. i.e.

$$H_S = \langle x_1^{d_1} - 1, \ldots, x_n^{d_n} - 1 \rangle$$
Theorem (Bezout)

\[ |\mathcal{V}(F)| \leq d_1 \cdots d_n \text{ where } d_i = \deg(f_i) \]

- If \( |\mathcal{V}(F)| \approx d_1 \cdots d_n \) then a **total degree** start system is suitable. i.e.

\[
H_S = \langle x_1^{d_1} - 1, \ldots, x_n^{d_n} - 1 \rangle
\]

- This bound is **generically** tight but can be a strict upper bound when \( f_i \) are sparse
Newton Polytopes

- The monomial support of $f$ is the set of exponent vectors of the monomials of $f$
Newton Polytopes

- The *monomial support* of \( f \) is the set of exponent vectors of the monomials of \( f \)

- Ex. \( f(x, y) = 1 + 2x - 3y + xy \) has monomial support

\[
\{(0, 0), (1, 0), (0, 1), (1, 1)\}
\]
Newton Polytopes

- The *monomial support* of $f$ is the set of exponent vectors of the monomials of $f$

- Ex. $f(x, y) = 1 + 2x - 3y + xy$ has monomial support

  $$\{(0, 0), (1, 0), (0, 1), (1, 1)\}$$

- The *Newton polytope* of $f$ is the convex hull of its monomial support
Newton Polytopes

- The *monomial support* of $f$ is the set of exponent vectors of the monomials of $f$

- Ex. $f(x, y) = 1 + 2x - 3y + xy$ has monomial support

  \[ \{(0, 0), (1, 0), (0, 1), (1, 1)\} \]

- The *Newton polytope* of $f$ is the convex hull of its monomial support

- Ex. If $f(x, y) = 1 + 2x - 3y + xy$, then $\text{Newt}(f) = \text{Conv}\{(0, 0), (1, 0), (0, 1), (1, 1)\}$
Given $k$ polytopes, $P_1, \ldots, P_k$ in $\mathbb{R}^n$ their mixed volume is

$$\text{MVol}(P_1, \ldots, P_k) = \sum_{J \subseteq [k]} (-1)^{k-|J|} \cdot \text{Vol}_n(P_J)$$

where $P_J = P_{i_1} + \ldots + P_{i_{|J|}}$ and $J = (i_1, \ldots, i_{|J|})$
Given $k$ polytopes, $P_1, \ldots, P_k$ in $\mathbb{R}^n$ their mixed volume is

$$
\text{MVol}(P_1, \ldots, P_k) = \sum_{J \subseteq [k]} (-1)^{k-|J|} \cdot \text{Vol}_n(P_J)
$$

where $P_J = P_{i_1} + \ldots + P_{i_{|J|}}$ and $J = (i_1, \ldots, i_{|J|})$

Ex. $k = n = 2$,

$$
\text{MVol}(P_1, P_2) = \text{Area}(P_1 + P_2) - \text{Area}(P_1) - \text{Area}(P_2)
$$
Start system
Polyhedral

Theorem (BKK Bound [Ber75, Kho78, Kou76])

\[ |\mathcal{V}(F)| \leq \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \]
Theorem (BKK Bound [Ber75, Kho78, Kou76])

\[ |\mathcal{V}(F)| \leq \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \]

- If \( \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \ll d_1 \cdots d_n \) then a polyhedral start system is suitable.
**Theorem (BKK Bound [Ber75, Kho78, Kou76])**

\[ |\mathcal{V}(F)| \leq \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \]

- If \( \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \ll d_1 \cdots d_n \) then a **polyhedral** start system is suitable
- There exists an algorithm that finds this binomial start system [HS95]
Start system
Polyhedral

Theorem (BKK Bound [Ber75, Kho78, Kou76])
\[ |\mathcal{V}(F)| \leq \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \]

- If \( \text{MVol}(\text{Newt}(f_1), \ldots, \text{Newt}(f_n)) \ll d_1 \cdots d_n \) then a polyhedral start system is suitable.
- There exists an algorithm that finds this binomial start system [HS95]
- In general, not easy to compute the mixed volume (\#P hard)
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Homotopy Continuation</td>
</tr>
<tr>
<td>2</td>
<td>Example 1: Polynomial optimization</td>
</tr>
<tr>
<td>3</td>
<td>Example 2: Gaussian mixture moment systems</td>
</tr>
<tr>
<td>4</td>
<td>Monodromy</td>
</tr>
<tr>
<td>5</td>
<td>Example 3: Power flow equations</td>
</tr>
</tbody>
</table>
Example 1
Critical points of polynomial optimization problems

- Want to solve polynomial optimization problem of the form

\[
\min_{x \in \mathbb{R}^n} g(x) \quad \text{subject to} \quad f_i(x) = 0, \ i \in [m]
\]
Example 1
Critical points of polynomial optimization problems

- Want to solve polynomial optimization problem of the form

  $\min_{x \in \mathbb{R}^n} g(x) \text{ subject to } f_i(x) = 0, \ i \in [m]$ 

- Consider Lagrangian

  \[ \mathcal{L}(x, \lambda) = g(x) - \sum_{i=1}^{m} \lambda_i f_i(x) \]
Example 1
Critical points of polynomial optimization problems

- Want to solve polynomial optimization problem of the form

$$\min_{x \in \mathbb{R}^n} \ g(x) \ \text{subject to} \ f_i(x) = 0, \ i \in [m]$$

- Consider Lagrangian

$$\mathcal{L}(x, \lambda) = g(x) - \sum_{i=1}^{m} \lambda_i f_i(x)$$

- Smooth critical points = solutions to polynomial system

$$\ell_i = \frac{\partial \mathcal{L}}{\partial x_i} = 0$$

$$f_i = \frac{\partial \mathcal{L}}{\partial \lambda_i} = 0$$
Example 1
Critical points of polynomial optimization problems

Theorem ([L., Nicholson, Rodriguez, Wang [LNRW21]])

For generic $f_i, g$ the number of smooth critical points to

$$\min_{x \in \mathbb{R}^n} g(x) \quad \text{subject to} \quad f_i(x) = 0, \ i \in [m]$$

is equal to $\text{MVol}(\ell_1, \ldots, \ell_n, f_1, \ldots, f_m)$. Moreover, the number of critical points is dictated by $\text{Newt}(f_1), \ldots, \text{Newt}(f_m), \text{Newt}(g)$. 

Implication 1: Polyhedral start systems will efficiently find all complex critical points for polynomial optimization problems

Implication 2: Only consider monomials corresponding to vertices of Newton polytopes
Example 1
Critical points of polynomial optimization problems

Theorem (L., Nicholson, Rodriguez, Wang [LNRW21])

For generic $f_i, g$ the number of smooth critical points to

$$\min_{x \in \mathbb{R}^n} g(x) \text{ subject to } f_i(x) = 0, \ i \in [m]$$

is equal to $\text{MVol}(\ell_1, \ldots, \ell_n, f_1, \ldots, f_m)$. Moreover, the number of critical points is dictated by $\text{Newt}(f_1), \ldots, \text{Newt}(f_m), \text{Newt}(g)$.

- **Implication 1:** Polyhedral start systems will efficiently find all complex critical points for polynomial optimization problems
Example 1
Critical points of polynomial optimization problems

Theorem (L., Nicholson, Rodriguez, Wang [LNRW21]):

For generic $f_i, g$ the number of smooth critical points to

$$\min_{x \in \mathbb{R}^n} g(x) \quad \text{subject to} \quad f_i(x) = 0, \ i \in [m]$$

is equal to $\text{MVol}(\ell_1, \ldots, \ell_n, f_1, \ldots, f_m)$. Moreover, the number of critical points is dictated by $\text{Newt}(f_1), \ldots, \text{Newt}(f_m), \text{Newt}(g)$.

- **Implication 1**: Polyhedral start systems will efficiently find all complex critical points for polynomial optimization problems
- **Implication 2**: Only consider monomials corresponding to vertices of Newton polytopes
Example 1
Critical points of polynomial optimization problems

Consider

$$\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - x_1 x_2 + x_1 x_3 - 2x_2^2 + 3x_2 x_3 + 4x_3^2 - x_1 + 2x_2 - x_3 - 1 = 0$$
Example 1
Critical points of polynomial optimization problems

Consider

\[
\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - x_1x_2 + x_1x_3 - 2x_2^2 + 3x_2x_3 + 4x_3^2 - x_1 + 2x_2 - x_3 - 1 = 0
\]

The Lagrange system is

\[
\begin{align*}
\ell_1 &= 3 - \lambda(2x_1 - x_2 + x_3 - 1) \\
\ell_2 &= -1 - \lambda(-x_1 - 4x_2 + 3x_3 + 2) \\
\ell_3 &= 2 - \lambda(x_1 + 3x_2 + 8x_3 - 1) \\
f &= x_1^2 - x_1x_2 + x_1x_3 - 2x_2^2 + 3x_2x_3 + 4x_3^2 - x_1 + 2x_2 - x_3 - 1
\end{align*}
\]

Using a total degree start system, need to track 16 paths
Example 1
Critical points of polynomial optimization problems

Consider

$$\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - x_1x_2 + x_1x_3 - 2x_2^2 + 3x_2x_3 + 4x_3^2 - x_1 + 2x_2 - x_3 - 1 = 0$$

The Lagrange system is

$$\ell_1 = 3 - \lambda(2x_1 - x_2 + x_3 - 1)$$
$$\ell_2 = -1 - \lambda(-x_1 - 4x_2 + 3x_3 + 2)$$
$$\ell_3 = 2 - \lambda(x_1 + 3x_2 + 8x_3 - 1)$$
$$f = x_1^2 - x_1x_2 + x_1x_3 - 2x_2^2 + 3x_2x_3 + 4x_3^2 - x_1 + 2x_2 - x_3 - 1$$

Using a total degree start system, need to track 16 paths

I claim there are only 2 critical points
Example 1
Critical points of polynomial optimization problems

Consider the same optimization problem restricted to the monomials corresponding to vertices of each Newton polytope:

\[
\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - 2x_2^2 + 4x_3^2 - 1 = 0
\]
Example 1

Critical points of polynomial optimization problems

- Consider the same optimization problem restricted to the monomials corresponding to vertices of each Newton polytope:

\[
\begin{align*}
\min_{x \in \mathbb{R}^3} & \quad 3x_1 - x_2 + 2x_3 \\
\text{subject to} & \quad x_1^2 - 2x_2^2 + 4x_3^2 - 1 = 0
\end{align*}
\]

- The Lagrange system is

\[
\begin{align*}
\ell_1 &= 3 - 2\lambda x_1 \\
\ell_2 &= -1 + 4\lambda x_2 \\
\ell_3 &= 2 - 8\lambda x_3 \\
f &= x_1^2 - 2x_2^2 + 4x_3^2 - 1
\end{align*}
\]

Can explicitly solve this system and get 2 complex solutions, then use this as a start system for the general problem.

This generalizes to minimizing a linear function over any hypersurface.
Example 1
Critical points of polynomial optimization problems

- Consider the same optimization problem restricted to the monomials corresponding to vertices of each Newton polytope:

\[
\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - 2x_2^2 + 4x_3^2 - 1 = 0
\]

- The Lagrange system is

\[
\begin{align*}
\ell_1 &= 3 - 2\lambda x_1 \\
\ell_2 &= -1 + 4\lambda x_2 \\
\ell_3 &= 2 - 8\lambda x_3 \\
f &= x_1^2 - 2x_2^2 + 4x_3^2 - 1
\end{align*}
\]

- Can explicitly solve this system and get 2 complex solutions, then use this as a start system for the general problem
Example 1
Critical points of polynomial optimization problems

- Consider the same optimization problem restricted to the monomials corresponding to vertices of each Newton polytope:

\[
\min_{x \in \mathbb{R}^3} 3x_1 - x_2 + 2x_3 \quad \text{subject to} \quad x_1^2 - 2x_2^2 + 4x_3^2 - 1 = 0
\]

- The Lagrange system is

\[
\ell_1 = 3 - 2\lambda x_1 \\
\ell_2 = -1 + 4\lambda x_2 \\
\ell_3 = 2 - 8\lambda x_3 \\
f = x_1^2 - 2x_2^2 + 4x_3^2 - 1
\]

- Can explicitly solve this system and get 2 complex solutions, then use this as a start system for the general problem

- This generalizes to minimizing a linear function over any hypersurface
Table of Contents

1 Homotopy Continuation

2 Example 1: Polynomial optimization

3 Example 2: Gaussian mixture moment systems

4 Monodromy

5 Example 3: Power flow equations
Example 2
Gaussian mixture moment systems

- Classical problem in statistics is density estimation: “Given $N$ samples from a density $p$, can I estimate $p$?”
Example 2
Gaussian mixture moment systems

- Classical problem in statistics is density estimation: “Given $N$ samples from a density $p$, can I estimate $p$?”
- Gaussian mixture models are a popular family to consider since they universally approximate smooth densities
Classical problem in statistics is density estimation: “Given $N$ samples from a density $p$, can I estimate $p$?”

Gaussian mixture models are a popular family to consider since they universally approximate smooth densities.

**Problem:** Given $N$ samples distributed as mixture of $k$ Gaussians, recover mean $\mu_i$, variance, $\sigma_i^2$ and mixing coefficient, $\lambda_i$ of each component.
Classical problem in statistics is density estimation: “Given $N$ samples from a density $p$, can I estimate $p$?”

Gaussian mixture models are a popular family to consider since they universally approximate smooth densities.

**Problem:** Given $N$ samples distributed as mixture of $k$ Gaussians, recover mean $\mu_i$, variance, $\sigma^2_i$ and mixing coefficient, $\lambda_i$ of each component.

Many techniques to do this, consider **method of moments**.
Moments

### Definition

Let \( f : \mathbb{R} \to \mathbb{R} \) be the probability density function of a random variable \( X \). For \( i \geq 0 \), the \( i \)-th moment of \( X \) is

\[
m_i = \mathbb{E}[X^i] = \int_{\mathbb{R}} x^i f(x) \, dx.
\]
Moments

Definition

Let $f : \mathbb{R} \to \mathbb{R}$ be the probability density function of a random variable $X$. For $i \geq 0$, the $i-$th moment of $X$ is

$$m_i = \mathbb{E}[X^i] = \int_{\mathbb{R}} x^i f(x) \, dx.$$ 

- For parameterized distributions, moments are functions of parameters.
Moments

Definition

Let $f : \mathbb{R} \to \mathbb{R}$ be the probability density function of a random variable $X$. For $i \geq 0$, the $i^{th}$ moment of $X$ is

$$m_i = \mathbb{E}[X^i] = \int_{\mathbb{R}} x^i f(x) \, dx.$$ 

- For parameterized distributions, moments are functions of parameters.
- Ex. Suppose $X \sim \text{Unif}[a, b]$ where $-\infty < a < b < \infty$
Moments

Definition

Let \( f : \mathbb{R} \to \mathbb{R} \) be the probability density function of a random variable \( X \). For \( i \geq 0 \), the \( i \)-th moment of \( X \) is

\[
m_i = \mathbb{E}[X^i] = \int_{\mathbb{R}} x^i f(x) \, dx.
\]

- For parameterized distributions, moments are functions of parameters
- Ex. Suppose \( X \sim \text{Unif}[a, b] \) where \(-\infty < a < b < \infty\)
- The first few moments are:

\[
\begin{align*}
    m_1 &= \frac{1}{2} (a + b) \\
    m_2 &= \frac{1}{3} (a^2 + 2ab + b^2) \\
    m_3 &= \frac{1}{4} (b^3 + ab^2 + a^2b + a^3)
\end{align*}
\]
Consider a statistical model with $n$ unknown parameters, $\theta = (\theta_1, \ldots, \theta_n)$ and the moments up to order $n$ as functions of $\theta$

$$m_1 = g_1(\theta), \ldots, m_n = g_n(\theta)$$

and samples $y_1, \ldots, y_N$
Consider a statistical model with $n$ unknown parameters, $\theta = (\theta_1, \ldots, \theta_n)$ and the moments up to order $n$ as functions of $\theta$

$$m_1 = g_1(\theta), \ldots, m_n = g_n(\theta)$$

and samples $y_1, \ldots, y_N$

**Method of Moments:**

1. Compute sample moments

$$\bar{m}_i = \frac{1}{N} \sum_{j=1}^{N} y_j^i$$
Method of Moments

Procedure

- Consider a statistical model with $n$ unknown parameters, $\theta = (\theta_1, \ldots, \theta_n)$ and the moments up to order $n$ as functions of $\theta$

$$m_1 = g_1(\theta), \ldots, m_n = g_n(\theta)$$

and samples $y_1, \ldots, y_N$

Method of Moments:

1. Compute sample moments

$$\overline{m}_i = \frac{1}{N} \sum_{j=1}^{N} y_j^i$$

2. Solve $g_i(\theta) = \overline{m}_i$ for $i = 1, \ldots, n$ to recover parameters
The moments of the Gaussian distributions are $M_0(\mu, \sigma^2) = 1$, $M_1(\mu, \sigma^2) = \mu$, and for $\ell \geq 2$

\[ M_\ell(\mu, \sigma^2) = \mu M_{\ell-1} + (\ell - 1)\sigma^2 M_{\ell-2}, \quad \ell \geq 2 \]
The moments of the Gaussian distributions are $M_0(\mu, \sigma^2) = 1$, $M_1(\mu, \sigma^2) = \mu$,

$$M_\ell(\mu, \sigma^2) = \mu M_{\ell-1} + (\ell - 1)\sigma^2 M_{\ell-2}, \quad \ell \geq 2$$

The moments of mixtures of $k$ Gaussians are

$$m_\ell = \sum_{i=1}^{k} \lambda_i M_\ell(\mu_i, \sigma_i^2), \quad \ell \geq 0$$
Method of Moments

Gaussian Mixture Models

- The moments of the Gaussian distributions are $M_0(\mu, \sigma^2) = 1$, $M_1(\mu, \sigma^2) = \mu$,

$$M_\ell(\mu, \sigma^2) = \mu M_{\ell-1} + (\ell - 1)\sigma^2 M_{\ell-2}, \quad \ell \geq 2$$

- The moments of mixtures of $k$ Gaussians are

$$m_\ell = \sum_{i=1}^{k} \lambda_i M_\ell(\mu_i, \sigma_i^2), \quad \ell \geq 0$$

- There are three special cases of Gaussian mixture models commonly studied in the statistics literature:
  1. The mixing coefficients are known
  2. The mixing coefficients are known and the variances are equal
  3. Only the means are unknown
Gaussian Mixture Models

Theorem (L. Améndola, Rodriguez [LAR21])

In all cases, Gaussian mixture models are algebraically identifiable using moment equations of lowest degree. Moreover, the mixed volume of each of set of equations is given below.

<table>
<thead>
<tr>
<th>Moment equations</th>
<th>Known mixing coefficients</th>
<th>Known mixing coefficients + equal variances</th>
<th>Unknown means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ununknowns</td>
<td>$m_1, \ldots, m_{2k}$</td>
<td>$m_1, \ldots, m_{k+1}$</td>
<td>$m_1, \ldots, m_k$</td>
</tr>
<tr>
<td>Mixed volume</td>
<td>$\mu_i, \sigma_i^2$</td>
<td>$\mu_i, \sigma_{(k+1)!}^2$</td>
<td>$\mu_i$</td>
</tr>
<tr>
<td>Mixed volume tight</td>
<td>$(2k - 1)!!k!$</td>
<td>$\frac{(k+1)!}{2}$</td>
<td>$k!$</td>
</tr>
<tr>
<td></td>
<td>Yes for $k \leq 8$</td>
<td>Yes for $k \leq 8$</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Classes of Gaussian Mixture Models

Solving the Polynomial Systems

<table>
<thead>
<tr>
<th>Known mixing coefficients</th>
<th>Mixed Volume</th>
<th>Bezout Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known mixing coefficients + equal variances</td>
<td>((2k - 1)!!k!)</td>
<td>((2k)!)</td>
</tr>
<tr>
<td>Unknown means</td>
<td>(\frac{(k+1)!}{k!})</td>
<td>((k + 1)!)</td>
</tr>
</tbody>
</table>

Our proofs of the mixed volume in the first two cases give a start system that tracks the mixed volume number of paths. In the final case if \(\lambda_i = 1\) and \(\sigma_i^2\) are equal, there is a unique solution up to symmetry.
<table>
<thead>
<tr>
<th>Known mixing coefficients</th>
<th>Mixed Volume</th>
<th>Bezout Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known mixing coefficients + equal variances</td>
<td>$(2k - 1)!!k!$</td>
<td>$(2k)!$</td>
</tr>
<tr>
<td>Unknown means</td>
<td>$(k+1)!k!$</td>
<td>$(k + 1)!k!$</td>
</tr>
</tbody>
</table>

- Our proofs of the mixed volume in the first two cases give a start system that tracks mixed volume number of paths.
### Classes of Gaussian Mixture Models

Solving the Polynomial Systems

<table>
<thead>
<tr>
<th></th>
<th>Mixed Volume</th>
<th>Bezout Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known mixing coefficients</td>
<td>$(2k - 1)!!k!$</td>
<td>$(2k)!$</td>
</tr>
<tr>
<td>Known mixing coefficients + equal variances</td>
<td>$(k+1)!$</td>
<td>$(k + 1)!$</td>
</tr>
<tr>
<td>Unknown means</td>
<td>$\frac{(k+1)!}{k!}$</td>
<td>$k!$</td>
</tr>
</tbody>
</table>

- Our proofs of the mixed volume in the first two cases give a start system that tracks mixed volume number of paths.

- In the final case if $\lambda_i = \frac{1}{k}$ and $\sigma_i^2$ are equal, there is a unique solution up to symmetry.
Table of Contents

1. Homotopy Continuation

2. Example 1: Polynomial optimization

3. Example 2: Gaussian mixture moment systems

4. Monodromy

5. Example 3: Power flow equations
Consider a parameterized polynomial system $F_b$

- i.e. $F_b(x) = b_1 x^2 + b_2 x - 1$
Consider a parameterized polynomial system $F_b$

- i.e. $F_b(x) = b_1x^2 + b_2x - 1$

Monodromy works by moving $b$ in a loop while tracking $x$ along that loop
Consider a parameterized polynomial system $F_b$
- i.e. $F_b(x) = b_1x^2 + b_2x - 1$

Monodromy works by moving $b$ in a loop while tracking $x$ along that loop

After one loop, usually find a new solution
Consider a parameterized polynomial system $F_b$

- i.e. $F_b(x) = b_1 x^2 + b_2 x - 1$

Monodromy works by moving $b$ in a loop while tracking $x$ along that loop

After one loop, usually find a new solution

This process then is repeated until some stopping criterion is fulfilled
Monodromy example

- Consider $F_b(x) = b_1x^2 + b_2x - 1$ where $B = \mathbb{C}^2$
Monodromy example

- Consider $F_b(x) = b_1 x^2 + b_2 x - 1$ where $B = \mathbb{C}^2$

- For almost all $(b_1, b_2) \in \mathbb{C}^2$, $\{ x \in \mathbb{C} : F_b(x) = 0 \}$ has two elements
Monodromy example

- Consider $F_b(x) = b_1x^2 + b_2x - 1$ where $B = \mathbb{C}^2$

- For almost all $(b_1, b_2) \in \mathbb{C}^2$, $\{x \in \mathbb{C} : F_b(x) = 0\}$ has two elements

- Consider $(2, 1) \in \mathbb{C}^2$ and $x = -1$, then $F_{(2,1)}(-1) = 0$
Method 2

Monodromy

- Monodromy methods work by taking one solution $\hat{x}$ to the system of equations $F_b$ and finding other elements via the monodromy action.
Monodromy methods work by taking one solution $\hat{x}$ to the system of equations $F_b$ and finding other elements via the monodromy action.

This action is transitive if and only if the solution variety is irreducible.
Monodromy methods work by taking one solution \( \hat{x} \) to the system of equations \( F_b \) and finding other elements via the monodromy action.

This action is transitive if and only if the solution variety is irreducible.

**Major benefit:** Using monodromy we can solve up to symmetry [ALR21].
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Homotopy Continuation</td>
</tr>
<tr>
<td>2</td>
<td>Example 1: Polynomial optimization</td>
</tr>
<tr>
<td>3</td>
<td>Example 2: Gaussian mixture moment systems</td>
</tr>
<tr>
<td>4</td>
<td>Monodromy</td>
</tr>
<tr>
<td>5</td>
<td>Example 3: Power flow equations</td>
</tr>
</tbody>
</table>
Example 3

Power flow equations

- Consider electric power network on \( n \) nodes, \( v_0, \ldots, v_{n-1} \)
Example 3
Power flow equations

- Consider electric power network on \( n \) nodes, \( v_0, \ldots, v_{n-1} \)

- At \( v_0 \) we have \( x_0 = 1, y_0 = 0 \)
Consider electric power network on \( n \) nodes, \( v_0, \ldots, v_{n-1} \)

At \( v_0 \) we have \( x_0 = 1, y_0 = 0 \)

At each node \( v_i, i > 0 \) we have the equation for voltage magnitude:

\[
    x_i^2 + y_i^2 = 1
\]
Example 3

Power flow equations

- Consider electric power network on $n$ nodes, $v_0, \ldots, v_{n-1}$

- At $v_0$ we have $x_0 = 1$, $y_0 = 0$

- At each node $v_i$, $i > 0$ we have the equation for voltage magnitude:

  $$x_i^2 + y_i^2 = 1$$

- The active power injections are described by:

  $$\sum_{k=0}^{n-1} b_{ik}(x_i y_k - x_k y_i) = P_i$$
Example 3

Power flow equations

- Consider electric power network on \( n \) nodes, \( v_0, \ldots, v_{n-1} \)

- At \( v_0 \) we have \( x_0 = 1, y_0 = 0 \)

- At each node \( v_i, i > 0 \) we have the equation for voltage magnitude:
  \[
  x_i^2 + y_i^2 = 1
  \]

- The active power injections are described by:
  \[
  \sum_{k=0}^{n-1} b_{ik}(x_i y_k - x_k y_i) = P_i
  \]

- These are the power flow equations
Proposition \((\mathbf{L.}, \text{Boston, Lesieutre 2020 [LBL20]})\)

- The power flow equations with \(P_i \neq 0\) has a solution variety that is irreducible.

- The power flow equations with zero active power injections has a solution variety that decomposes into \(2^{n-1} + 1\) irreducible components.
  - \(2^{n-1}\) consist of a single point of the form \((x_i, y_i) = (\pm 1, 0)\) (trivial).
  - The remaining component consists of all nontrivial solutions to the power flow equations.
Method 2
Monodromy

Proposition (L., Boston, Lesieutre 2020 [LBL20])

- The power flow equations with $P_i \neq 0$ has a solution variety that is irreducible.
- The power flow equations with zero active power injections has a solution variety that decomposes into $2^{n-1} + 1$ irreducible components.
  - $2^{n-1}$ consist of a single point of the form $(x_i, y_i) = (\pm 1, 0)$ (trivial).
  - The remaining component consists of all nontrivial solutions to the power flow equations.

Lemma (L., Zachariah, Boston, Lesieutre [LZBL20])

- For the power flow equations with zero active power injections, solutions come in pairs.
- For the power flow equations with zero active power injections on bipartite graphs, solutions come in sets of four.
- For the power flow equations with nonzero active power injections on bipartite graphs, solutions come in pairs.
Comparing Monodromy

<table>
<thead>
<tr>
<th></th>
<th>$K_5$</th>
<th>$K_6$</th>
<th>$K_7$</th>
<th>$K_8$</th>
<th>$K_9$</th>
<th>$K_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (s): monodromy</td>
<td>0.05</td>
<td>0.37</td>
<td>1.97</td>
<td>16.39</td>
<td>65.33</td>
<td>357.926</td>
</tr>
<tr>
<td>time (s): polyhedral</td>
<td>0.37</td>
<td>2.53</td>
<td>17.10</td>
<td>112.43</td>
<td>609.49</td>
<td>2637.22</td>
</tr>
<tr>
<td>time (s): total degree</td>
<td>0.21</td>
<td>1.45</td>
<td>8.17</td>
<td>48.78</td>
<td>329.60</td>
<td>1510.01</td>
</tr>
</tbody>
</table>

**Table:** Numerical results to find all solutions for complete networks

<table>
<thead>
<tr>
<th></th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
<th>$C_9$</th>
<th>$C_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (s): monodromy</td>
<td>0.13</td>
<td>0.158</td>
<td>1.10</td>
<td>1.46</td>
<td>2.48</td>
<td>2.60</td>
</tr>
<tr>
<td>time (s): polyhedral</td>
<td>2.7</td>
<td>3.03</td>
<td>5.37</td>
<td>14.8</td>
<td>56.36</td>
<td>211.24</td>
</tr>
<tr>
<td>time (s): total degree</td>
<td>2.11</td>
<td>3.40</td>
<td>9.76</td>
<td>31.91</td>
<td>200.41</td>
<td>862.50</td>
</tr>
</tbody>
</table>

**Table:** Numerical results to find all solutions for cyclic networks
Conclusion

- Outlined methods for polynomial system solving
- Considered three applications

Questions!?
References


