

Application of linear algebra in computer vision and bioinformatics

Ivan Markovsky

Joint work with Sasan Mahmoodi and Mahesan Niranjan

ECS, University of Southampton

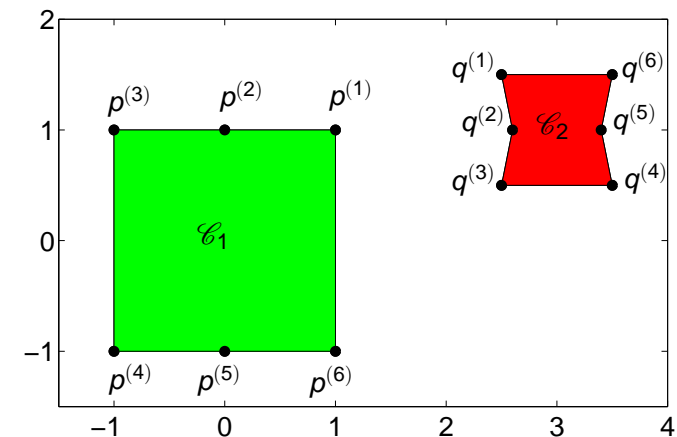
Contour alignment

Outline

- Least squares contour alignment
 - Problem formulation
 - Main result
 - Distance between contours
 - Examples
- Approx. low-rank factorization with structured factors
 - Rank estimation
 - Microarray data analysis
 - Estimation problem
 - Alternating least squares

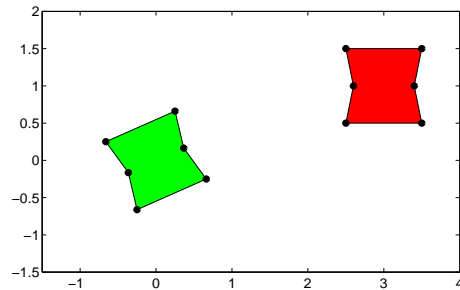
Consider contours $\mathcal{C}_1, \mathcal{C}_2$ in \mathbb{R}^2 , specified by N corresponding points

$$p^{(i)} \in \mathcal{C}_1 \subset \mathbb{R}^2 \leftrightarrow q^{(i)} \in \mathcal{C}_2 \subset \mathbb{R}^2, \quad i = 1, \dots, N$$



Aim: find how far is \mathcal{C}_1 from \mathcal{C}_2 , modulo translation, rotation, scaling

Example: The following contours are “far” from each other



but are obtained by translation and rotation of one contour.

They are the same contour “modulo translation and rotation”.

The distance between them is 0 modulo translation and rotation.

Aim: quantify closeness modulo translation, rotation, scaling.

Translation, rotation, and scaling operator

\mathcal{R}_θ — operator that rotates by $\theta \in [-\pi, \pi)$ rad its argument

$$\mathcal{R}_\theta(\mathbf{q}) := \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mathbf{q}$$

$\mathcal{A}_{\mathbf{a}, \theta, s}$ — rotates by θ , scales by $s > 0$, and translates by $\mathbf{a} \in \mathbb{R}^2$

$$\mathcal{A}_{\mathbf{a}, \theta, s}(\mathbf{q}) := s\mathcal{R}_\theta(\mathbf{q}) + \mathbf{a}.$$

define the matrices

$$\mathbf{C}_1 := [\mathbf{p}^{(1)} \quad \dots \quad \mathbf{p}^{(N)}] \quad \text{and} \quad \mathbf{C}_2 := [\mathbf{q}^{(1)} \quad \dots \quad \mathbf{q}^{(N)}]$$

acting on the matrix \mathbf{C}_2 , $\mathcal{A}_{\mathbf{a}, \theta, s}$ transforms each column $\mathbf{q}^{(i)}$ of \mathbf{C}_2

Least squares contour alignment problem

$\mathcal{A}_{\mathbf{a}, \theta, s}(\mathbf{C}_2)$ — translated, rotated, scaled contour \mathcal{C}_2

$\mathbf{E} := \mathbf{C}_1 - \mathcal{A}_{\mathbf{a}, \theta, s}(\mathbf{C}_2)$ — alignment error (a contour)

$\|\mathbf{E}\|_F := \sqrt{\sum_{i=1}^N \|\mathbf{e}^{(i)}\|_2^2}$ — alignment error (a number)

Alignment problem:

minimize over $\mathbf{a} \in \mathbb{R}^2$, $s > 0$, $\theta \in [-\pi, \pi)$ $\|\mathbf{C}_1 - \mathcal{A}_{\mathbf{a}, \theta, s}(\mathbf{C}_2)\|_F$

Comments:

- a constrained nonlinear least squares optimization problem
- can be solved by local optimization methods
- do not find a global solution and need initial approximation
- may have convergence problems

Main result: Equivalent least squares problem

Minimize over $(a_1, a_2, b_1, b_2) \in \mathbb{R}^4$

$$\left\| \begin{bmatrix} p_1^{(1)} \\ p_1^{(1)} \\ \vdots \\ p_1^{(N)} \\ p_2^{(N)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & q_1^{(1)} & -q_2^{(1)} \\ 0 & 1 & q_2^{(1)} & q_1^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & q_1^{(N)} & -q_2^{(N)} \\ 0 & 1 & q_2^{(N)} & q_1^{(N)} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \right\|_2$$

The relation between b_1, b_2 and θ, s is given by

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = s \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad \begin{bmatrix} \theta \\ s \end{bmatrix} = \begin{bmatrix} \sin^{-1}(b_2 / \sqrt{b_1^2 + b_2^2}) \\ \sqrt{b_1^2 + b_2^2} \end{bmatrix}$$

Derivation of the equivalent problem

Consider a point $q \in \mathbb{R}^2$ and define

$$q_r := \mathcal{R}_{\pi/2}(q)$$

The key observation is that

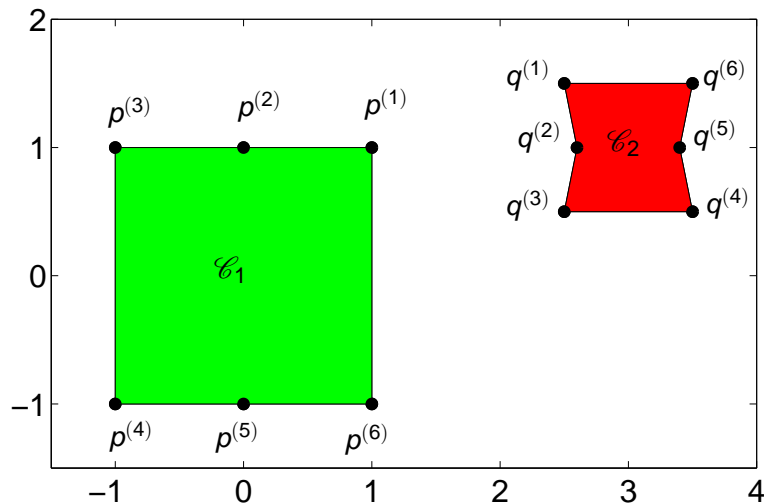
$$\mathcal{A}_{a,\theta,s}(q) = \begin{bmatrix} q & q_r \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + a$$

where b_1, b_2 and θ, s are in 1-to-1 relation, derived by solving

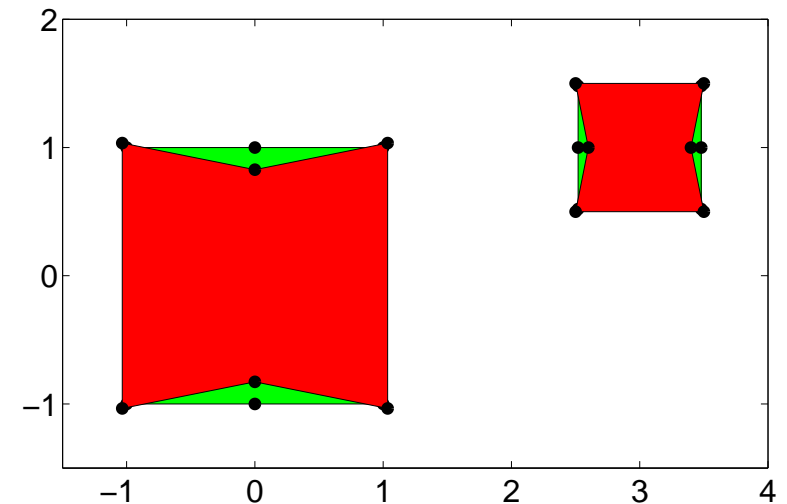
$$\begin{bmatrix} q & q_r \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = s \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} q$$

In terms of a_1, a_2, b_1, b_2 , the original problem is a LS problem.

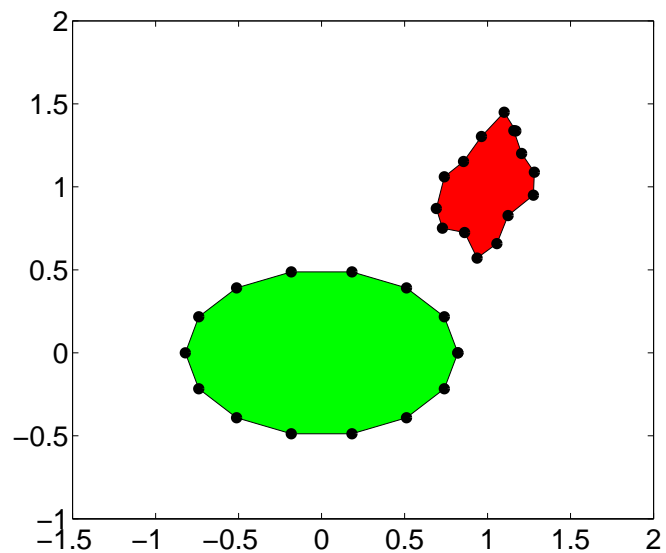
Academic example



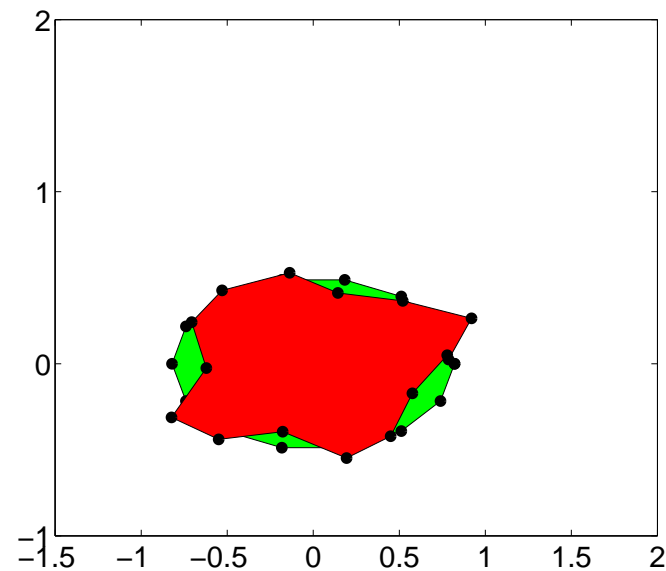
Results for the academic example



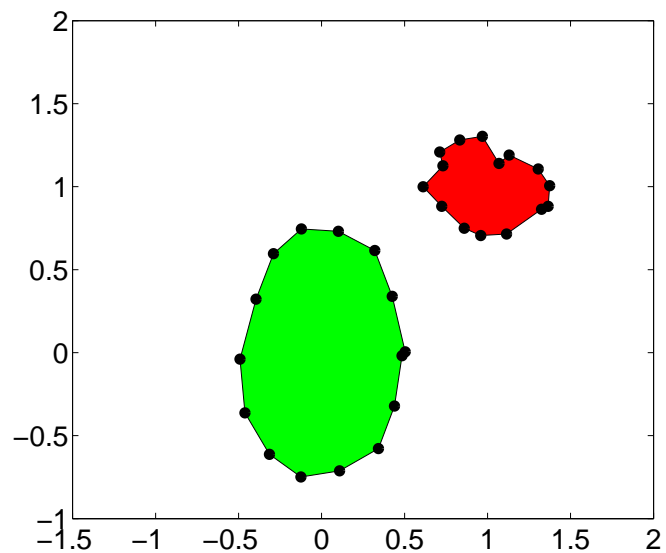
Less academic example: given contours



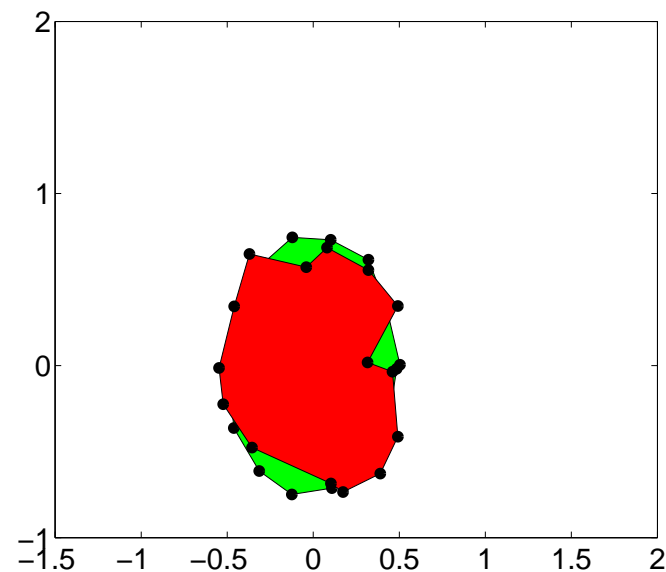
Less academic example: aligned contours



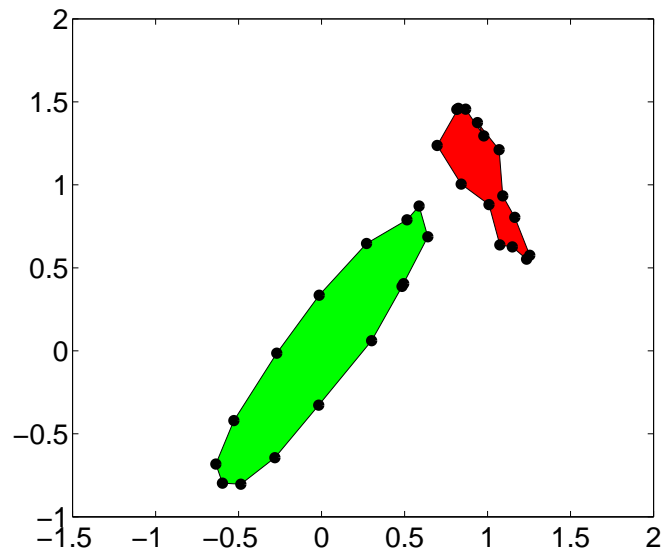
Another example: given contours



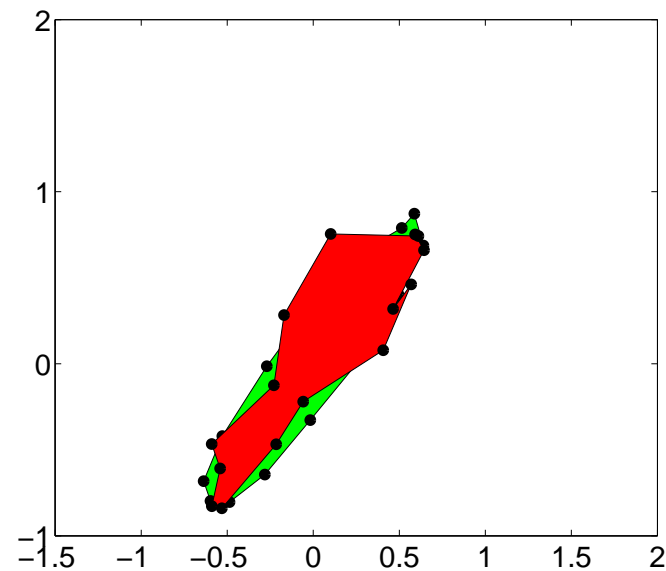
Another example: aligned contours



Yet another example: given contours



Yet another example: aligned contours



Distance between contours

The minimum value

$$\text{dist}'(C_1, C_2) := \min_{a \in \mathbb{R}^2, s > 0, \theta \in [-\pi, \pi)} \|C_1 - \mathcal{A}_{a, \theta, s}(C_2)\|_F$$

is not a suitable distance measure, because, in general,

$$\text{dist}'(C_1, C_2) \neq \text{dist}'(C_2, C_1)$$

and $\text{dist}'(C_1, C_2)$ is not invariant to affine transformation of $\mathcal{C}_1, \mathcal{C}_2$.

Distance between contours

Can be shown that

$$\text{dist}(C_1, C_2) := \frac{1}{\|C_1 - \frac{1}{N}C_1\mathbf{1}_N\mathbf{1}_N^\top\|_F} \text{dist}'(C_1, C_2)$$

is symmetric and affine invariant, *i.e.*,

$$\text{dist}(C_1, C_2) = \text{dist}(C_2, C_1) = \text{dist}(\mathcal{A}_{a, \theta, s}(C_1), \mathcal{A}_{a, \theta, s}(C_2)),$$

for all $a \in \mathbb{R}^2$, $\theta \in [-\pi, \pi)$, and $s > 0$.

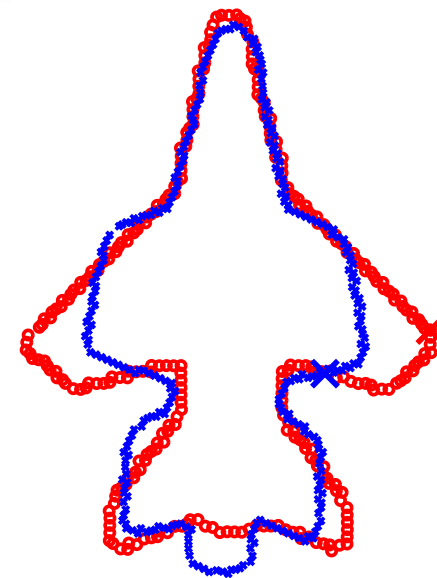
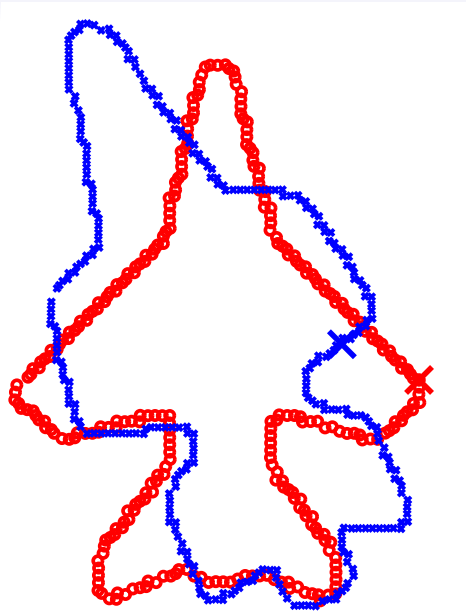
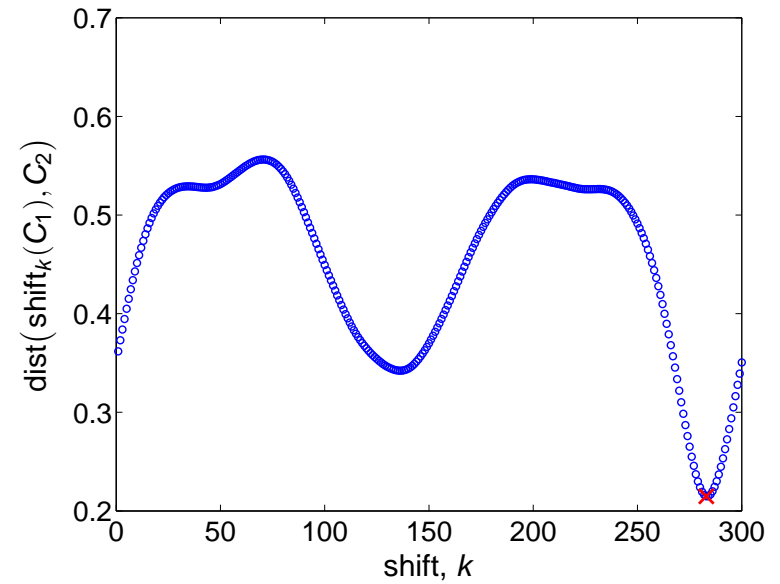
$$C_{1, \text{cent}} := C_1 - \frac{1}{N}C_1\mathbf{1}_N\mathbf{1}_N^\top \text{ — centered contour } \mathcal{C}_1$$

Complete example

Given images



1. detect edges (Prewitt algorithm)
2. collect points from the edges (edge follower)
3. up/down sample to equal number of points (cubic spline)
4. compute $\text{dist}(\text{shift}_k(C_1), C_2)$ for $k = 0, 1, \dots, N$
5. choose the minimum value



Conclusions for the first part of the talk

- Using the “right” problem formulation and parameterisation, makes the solution easy
- The point correspondence assumption is a strong one
- For sequential points, however, finding point correspondence reduces to solving N alignment problems
- Current work: alignment of implicitly represented contours

For more information:

<http://eprints.ecs.soton.ac.uk/16829>

Rank estimation

Rank revealing factorization of $X_0 \in \mathbb{R}^{m \times n}$, $\text{rank}(X_0) = r_0$

$$X_0 = C_0 P_0, \quad \text{where } C_0 \in \mathbb{R}^{m \times r_0} \quad \text{and} \quad P_0 \in \mathbb{R}^{r_0 \times n}$$

Suppose that

$$X := X_0 + E, \quad \text{where } E \text{ is a perturbation.}$$

“small” $E \implies X$ is “close” to rank- r_0 , i.e., $\text{dist}(X, r_0) < \|E\|_F$

$$\text{dist}(X, r_0) := \min_{\hat{X}} \|X - \hat{X}\|_F \quad \text{subject to} \quad \text{rank}(\hat{X}) = r_0 \quad (\text{LRA})$$

\implies if $\|E\|_F$ is known, $\text{dist}(X, r)$ can be used to estimate $\text{rank}(X_0)$

Approximate factorization

(LRA) has analytic solution in terms of the singular values

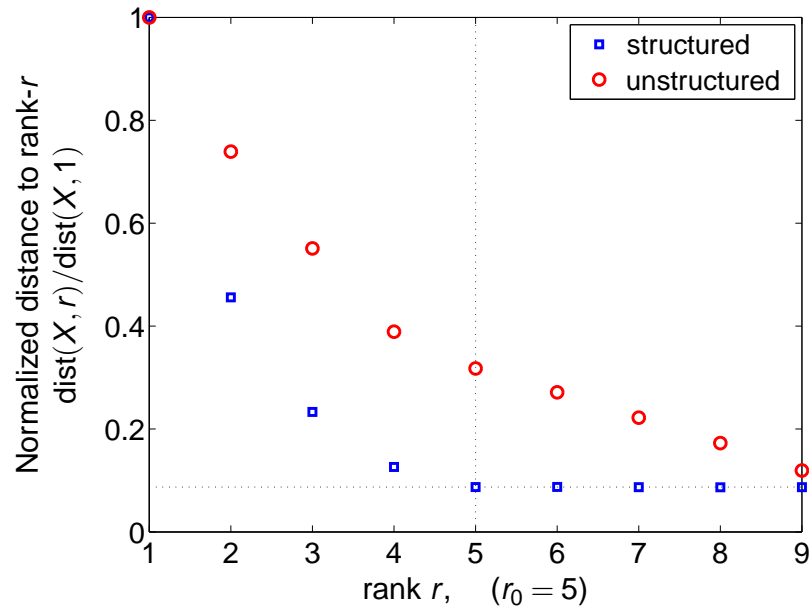
$\sigma_1, \dots, \sigma_{\min(m,n)}$ of X

$$\text{dist}(X, r_0) = \sqrt{\sigma_{r_0+1}^2 + \dots + \sigma_{\min(m,n)}^2},$$

$\implies \text{rank}(X_0)$ can be estimation from the decay of $\sigma_1, \dots, \sigma_{\min(m,n)}$

Aim: estimate more accurately the rank of X_0
 exploiting prior knowledge about X_0

Rank estimation example



Aim: discover what transcription factors regulate particular gene and what is the time evaluation of the transcription factors

The problem is equivalent to finding an approximate revealing rank factorization $X \approx CP$.

The need of approximation comes from:

1. inability to account for all relevant transcription factors
2. measurement errors in the collection of the data

Microarray data analysis

x_{ij} — **expression level** of i th gene at j th moment of time
(rows correspond to genes, columns correspond to time instants)

$\text{rank}(X)$ — number of **transcription factors** regulating the gene
expression levels

Consider a rank revealing factorization $X = CP$.

$P(:, j)$ — intensities vector of transcription factors at time j

$C(i, :)$ — sensitivities vector of i th gene to transcription factors

$c_{ij} = 0$ — j th transcription factor does not regulate i th gene

Prior knowledge

- Some transcription factors do not regulate some genes.

This amounts to having zero elements in C .

- The transcription factor intensities are nonnegative, smooth, periodic functions

- In order to make $X = CP$ unique, we impose the normalization $C = \begin{bmatrix} I_r \\ C' \end{bmatrix}$

Meaning: assume that r genes regulate single transcription factors that are different and the corresponding sensitivities are $= 1$.

Measurement errors model

X — observed data matrix

X_0 — “true” data matrix

E — measurement noise

$$X = X_0 + E$$

where

$$X_0 = C_0 P_0, \quad C_0 \in \mathbb{R}^{m \times r}, \quad P_0 \in \mathbb{R}^{r \times n}$$

and

$$\text{vec}(E) \sim \mathcal{N}(0, \sigma^2 \text{diag}(v)) \quad (v \text{ known})$$

Define the **element-wise weighting matrix**

$$W := \text{vec}^{-1}(v_1^{-1/2}, \dots, v_{mn}^{-1/2}) \in \mathbb{R}^{m \times n}$$

Estimation problem

$$\text{minimize}_{C', P', \hat{X}} \quad \|X - \hat{X}\|_W^2 \quad \text{subject to} \quad (\text{P})$$

$$\hat{X} = CP \quad (\text{weighted low-rank appr.})$$

$$C = \begin{bmatrix} I_r \\ C' \end{bmatrix} \quad (\text{normalization of } C) \quad (\text{C1})$$

$$\text{Svec}(C') = 0 \quad (\text{zero elements of } C') \quad (\text{C2})$$

$$P = \mathbf{1}_l^\top \otimes P' \quad (\text{periodicity of } P) \quad (\text{C3})$$

$$P' \geq 0 \quad (\text{nonnegativity of } P) \quad (\text{C4})$$

$$\|P'D\|_F^2 \leq d \quad (\text{smoothness of } P) \quad (\text{C5})$$

(D is the 1st order difference matrix)

Improved estimation performance on simulated data

Consider example with $m = 100$ rows and $n = 6$ columns.

The true data matrix X_0 is random with rank $r_0 = 2$.

We do $N = 100$ Monte Carlo repetitions and define the

$$\text{average relative estimation error:} \quad e := \frac{1}{N} \sum_{i=1}^N \frac{\|X_0 - \hat{X}^i\|_F^2}{\|X_0\|_F^2}$$

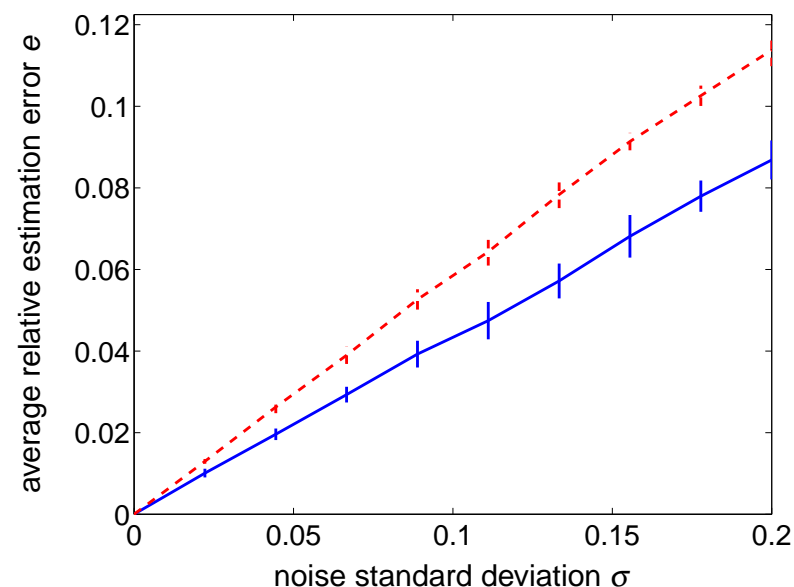
where \hat{X}^i is the estimate in the i th repetition.

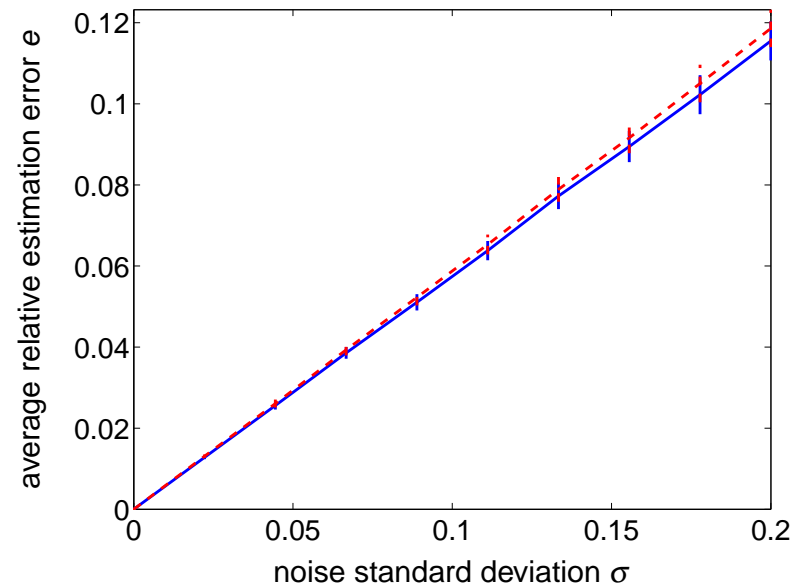
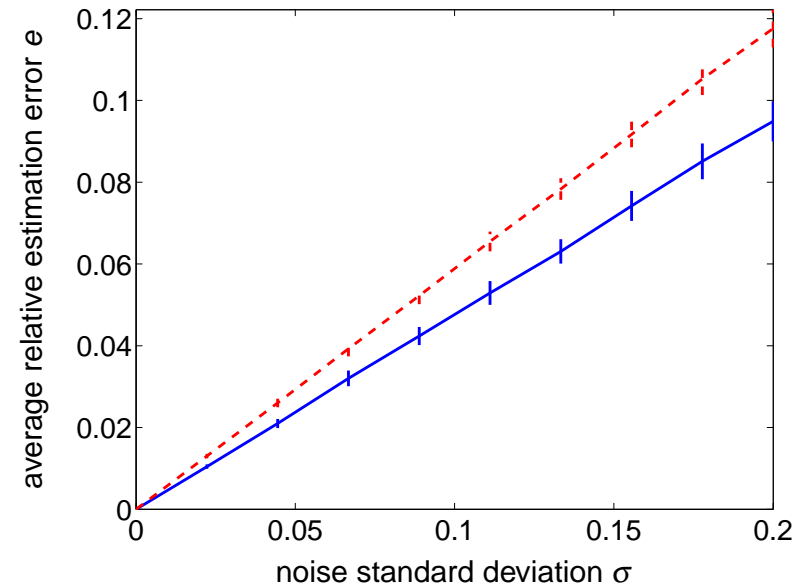
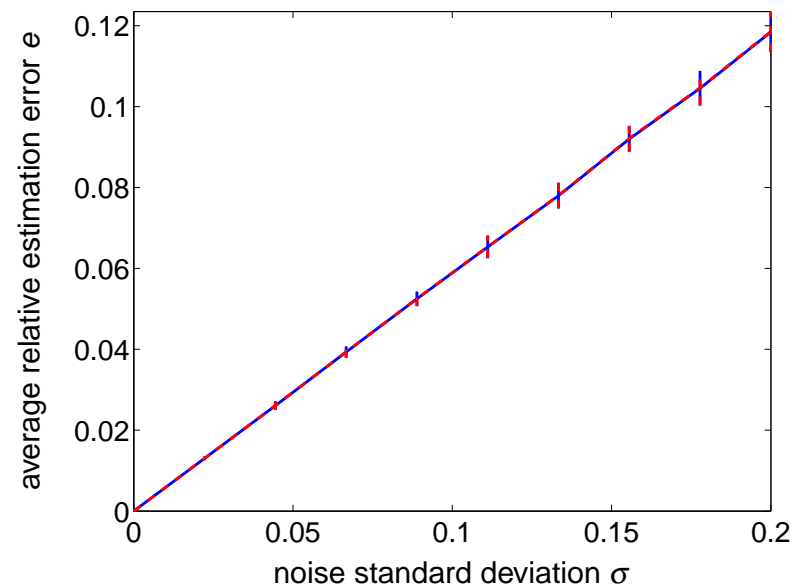
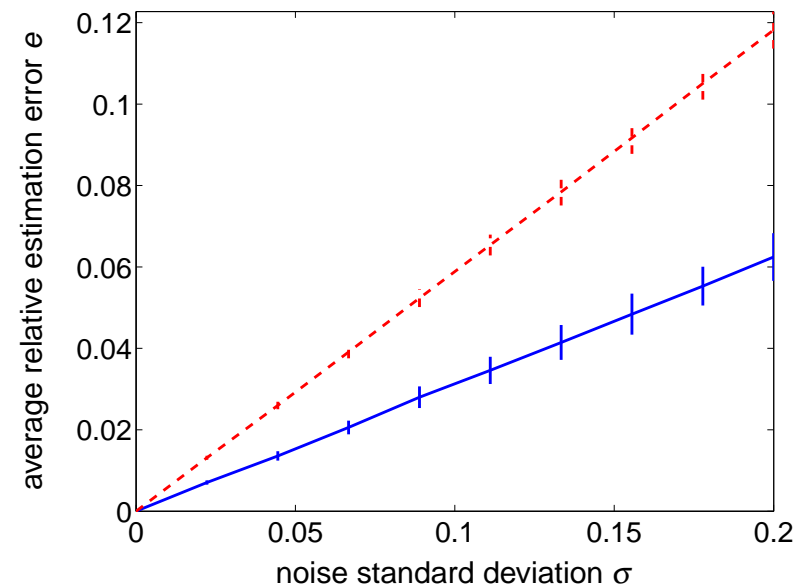
We compare the average relative error of the estimates that

- do not exploit prior knowledge (dashed line)
- exploit prior knowledge (solid line)

for different noise levels σ

Exploiting the knowledge of W only



Exploiting the periodicity of P onlyExploiting the zero elements in C onlyExploiting the nonnegativity of P onlyExploiting all prior knowledge W and (C1–C4)

Alternating projections algorithm

- Find an initial approximation $(C^{(0)}, P^{(0)})$.
- For $k = 0, 1, \dots$ till convergence do
 - $C^{(k+1)} := \arg \min_{C'} \|X - CP\|_W^2$ subject to (C1–C2)
with $P' = P^{(k)}$
 - $P^{(k+1)} := \arg \min_{P'} \|X - CP\|_W^2$ subject to (C3–C5)
with $C' = C^{(k+1)}$

Main point: The problems in steps 1 and 2 are easy to solve
(nonnegative weighted least squares)

Convergence properties

The alternating proj. algorithm is globally and monotonically convergent in the $\|\cdot\|_W$ norm to a stationary point of (P).

$\|X - \hat{X}^{(k)}\|_W^2 \geq 0$ is monotonically nonincreasing \implies convergent

Let \hat{X}^* be limit point of $\hat{X}^{(k)}$ and let

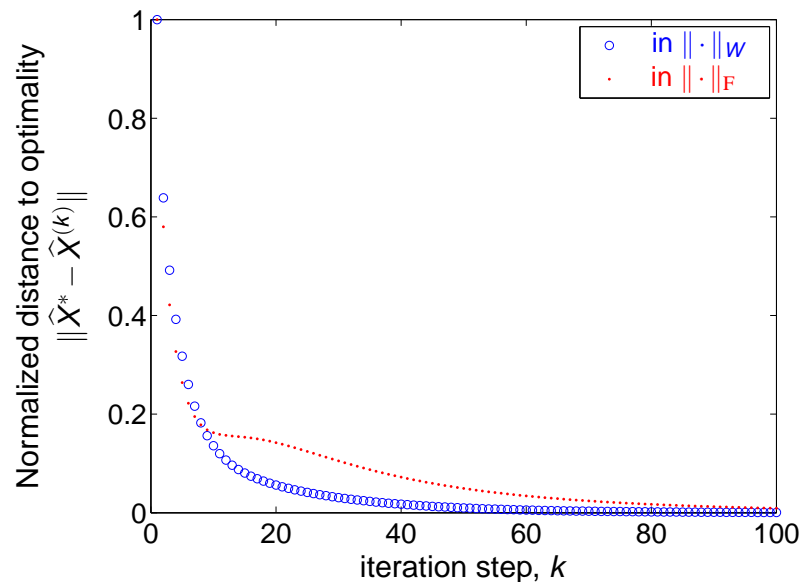
$$\hat{X}^* = C^* P^*, \quad \text{with } C^* \in \mathbb{R}^{m \times r}, \quad C^* = \begin{bmatrix} I_r \\ C_{*}^* \end{bmatrix}$$

$$\begin{aligned} \|\hat{X}^* - \hat{X}^{(k)}\|_W &= \|C^* P^* - C^{(k)} P^{(k)}\|_W \\ &\leq \|C^* P^* - C^{(k)} P^*\|_W = \|(C^* - C^{(k)}) P^*\|_W \end{aligned}$$

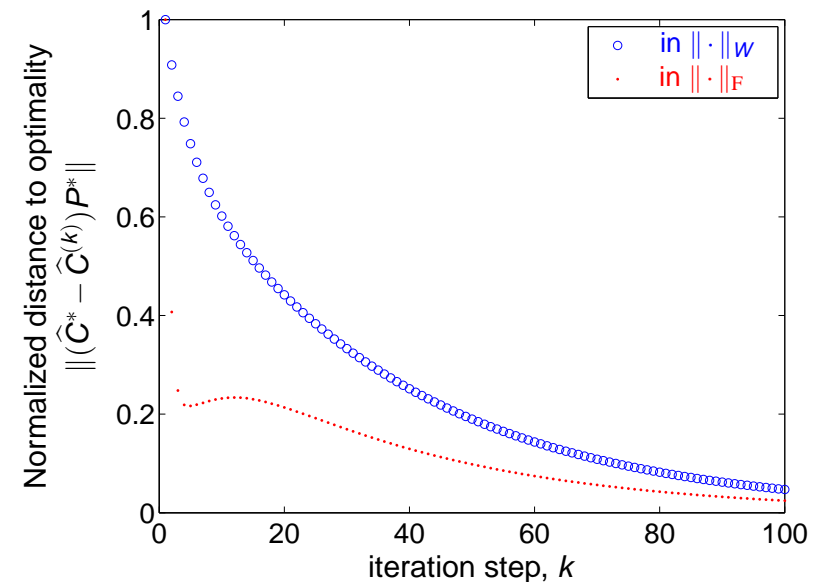
$\implies \hat{X}^* - \hat{X}^{(k)}, (C^* - C^{(k)}) P^*, C^* (P^* - P^{(k)})$
converge monotonically to 0 in $\|\cdot\|_W$ norm

This proves the global monotonic convergence of the algorithm.

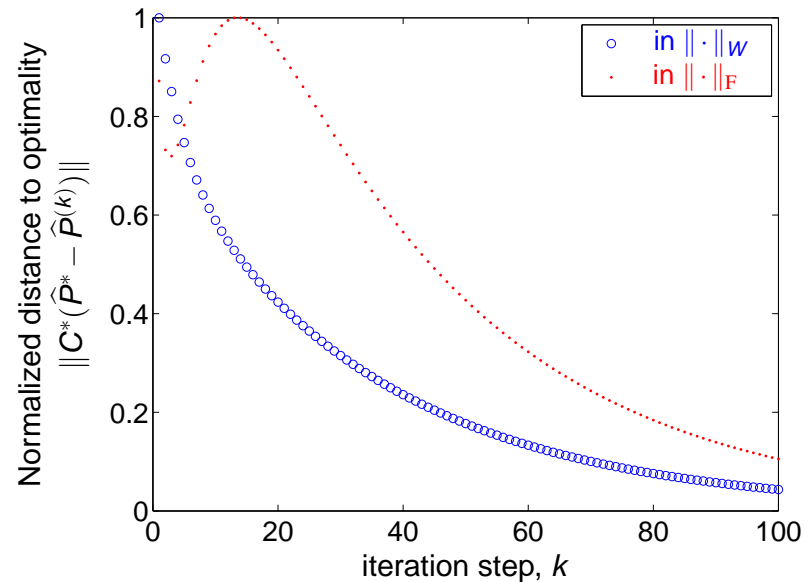
Convergence of $\hat{X}^* - \hat{X}^{(k)}$



Convergence of $(\hat{C}^* - \hat{C}^{(k)}) P^*$



Convergence of $C^*(\hat{P}^* - \hat{P}^{(k)})$



Convergence to a minimum point of (P)

We need to verify that the first order optimality conditions (OC) of (P) are satisfied at a cluster point of the algorithm.

Upon convergence

$$P^{(k-1)} = P^{(k)} =: P^* \quad \text{and} \quad C^{(k-1)} = C^{(k)} =: C^*.$$

These and the OC of the problems on steps 1 and 2, give conditions for a cluster point of the algorithm.

Can be verified that they coincide with the OC of (P).

Conclusions

- Using prior knowledge improves the estimation accuracy
- but makes the problem harder to solve
- Alternating least squares is a simple and flexible solution
- it allow us to impose various types of prior knowledge and has guaranteed convergence
- the price we pay for the simplicity is slow convergence (compared to sophisticated optimization methods)

For more information:

<http://eprints.ecs.soton.ac.uk/15959/>

Thank you