

Lec 4: Designing Linear Systems

15-369/669/769: Numerical Computing

Instructor: Minchen Li

Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment

Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment

Parametric Regression

Overview

- An application from data analysis
- e.g. to understand the structure of the results from a scientific experiment:
 - Write the *independent variables* of a given trial in a vector $\mathbf{x} \in \mathbb{R}^n$
 - Think of the *dependent variable* as a function $f(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$
 - Given a few $(\mathbf{x}, f(\mathbf{x}))$ pairs, predict $f(\mathbf{x})$ for a new \mathbf{x} without carrying out the full experiment

Parametric Regression

Linear Regression

In *parametric* regression, we additionally assume that we know the structure of f ahead of time. For example, suppose we assume that f is linear:

$$f(\mathbf{x}) = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

Then, our goal becomes more concrete: to estimate the coefficients a_1, \dots, a_n .

We can carry out n experiments to reveal $y^{(k)} := f(\mathbf{x}^{(k)})$ for samples $\mathbf{x}^{(k)}$, where $k \in \{1, \dots, n\}$. For the linear example, plugging into the formula for f shows a set of statements:

$$y^{(1)} = f(\mathbf{x}^{(1)}) = a_1x_1^{(1)} + a_2x_2^{(1)} + \cdots + a_nx_n^{(1)}$$

$$y^{(2)} = f(\mathbf{x}^{(2)}) = a_1x_1^{(2)} + a_2x_2^{(2)} + \cdots + a_nx_n^{(2)}$$

⋮

Parametric Regression

Linear Regression (Matrix Form)

Contrary to our earlier notation $A\mathbf{x} = \mathbf{b}$, the unknowns here are the a_i 's, *not* the $\mathbf{x}^{(k)}$'s.

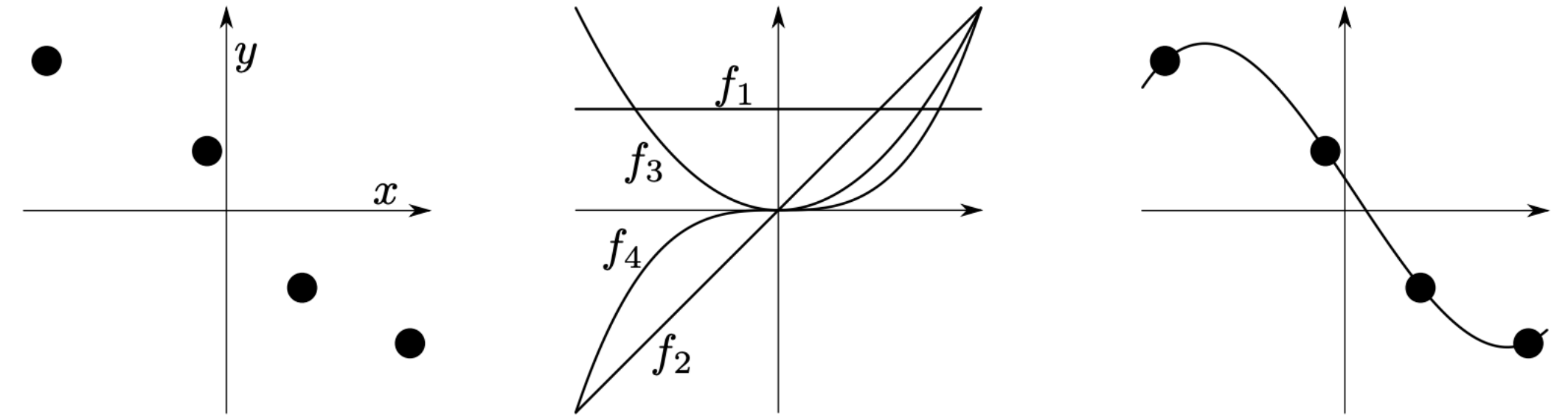
With this notational difference in mind, if we make exactly n observations we can write

$$\begin{pmatrix} - & \mathbf{x}^{(1)\top} & - \\ - & \mathbf{x}^{(2)\top} & - \\ & \vdots & \\ - & \mathbf{x}^{(n)\top} & - \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}.$$

In other words, if we carry out n trials of our experiment and write the independent variables in the columns of a matrix $X \in \mathbb{R}^{n \times n}$ and the dependent variables in a vector $\mathbf{y} \in \mathbb{R}^n$, then the coefficients \mathbf{a} can be recovered by solving the linear system $X^\top \mathbf{a} = \mathbf{y}$.

Parametric Regression

Nonlinear Regression



We can generalize this method to certain nonlinear forms for the function f using an approach illustrated in Figure 4.1. The key is to write f as a linear combination of *basis functions*. Suppose $f(\mathbf{x})$ takes the form

$$f(\mathbf{x}) = a_1 f_1(\mathbf{x}) + a_2 f_2(\mathbf{x}) + \cdots + a_m f_m(\mathbf{x}),$$

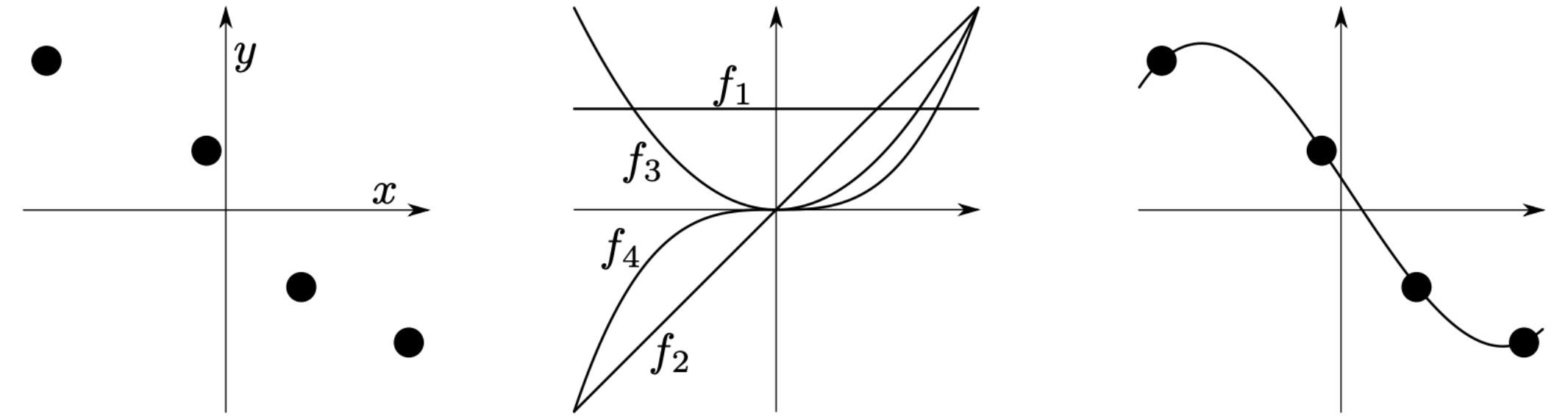
where $f_k : \mathbb{R}^n \rightarrow \mathbb{R}$ and we wish to estimate the parameters a_k . Then, by a parallel derivation given m observations of the form $\mathbf{x}^{(k)} \mapsto y^{(k)}$ we can find the parameters by solving:

$$\begin{pmatrix} f_1(\mathbf{x}^{(1)}) & f_2(\mathbf{x}^{(1)}) & \cdots & f_m(\mathbf{x}^{(1)}) \\ f_1(\mathbf{x}^{(2)}) & f_2(\mathbf{x}^{(2)}) & \cdots & f_m(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \cdots & \vdots \\ f_1(\mathbf{x}^{(m)}) & f_2(\mathbf{x}^{(m)}) & \cdots & f_m(\mathbf{x}^{(m)}) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}.$$

That is, even if the f 's are nonlinear, we can learn weights a_k using purely linear techniques.

Parametric Regression

Example



Example 4.3 (Polynomial regression). As in Figure 4.1, suppose that we observe a function of a single variable $f(x)$ and wish to write it as an $(n - 1)$ -st degree polynomial

$$f(x) := a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}.$$

Given n pairs $x^{(k)} \mapsto y^{(k)}$, we can solve for the parameters \mathbf{a} via the system

$$\begin{pmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{n-1} \\ 1 & x^{(2)} & (x^{(2)})^2 & \cdots & (x^{(2)})^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x^{(n)} & (x^{(n)})^2 & \cdots & (x^{(n)})^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}.$$

The design matrix of polynomial regression

In other words, we take $f_k(x) = x^{k-1}$ in the general form above. Incidentally, the matrix on the left-hand side of this relationship is known as a Vandermonde matrix.

Parametric Regression

Example (Cont.)

As an example, suppose we wish to find a parabola $y = ax^2 + bx + c$ going through $(-1, 1)$, $(0, -1)$, and $(2, 7)$. We can write the Vandermonde system in two ways:

$$\left\{ \begin{array}{l} a(-1)^2 + b(-1) + c = 1 \\ a(0)^2 + b(0) + c = -1 \\ a(2)^2 + b(2) + c = 7 \end{array} \right\} \iff \begin{pmatrix} 1 & -1 & (-1)^2 \\ 1 & 0 & 0^2 \\ 1 & 2 & 2^2 \end{pmatrix} \begin{pmatrix} c \\ b \\ a \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 7 \end{pmatrix}.$$

Gaussian elimination on this system shows $(a, b, c) = (2, 0, -1)$, corresponding to the polynomial $y = 2x^2 - 1$.

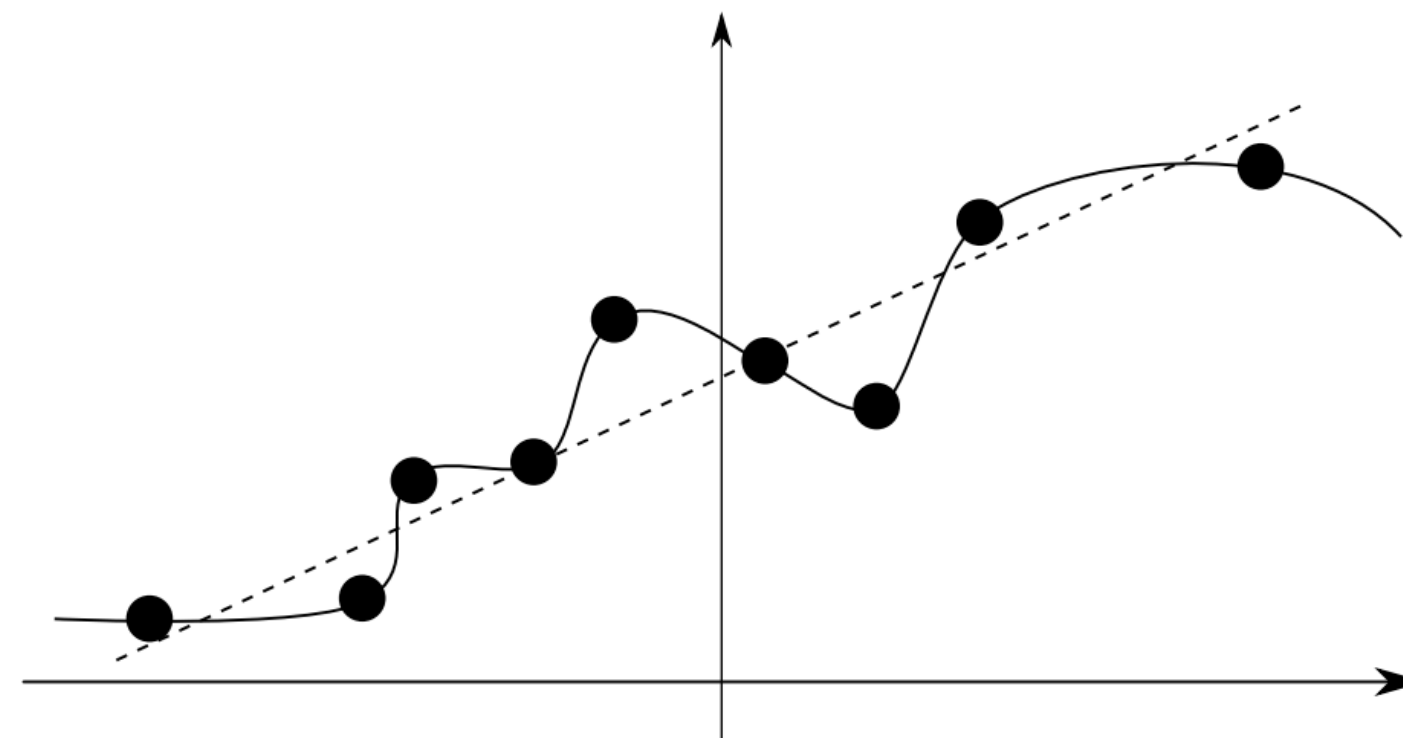
Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment

Least-Squares

Drawbacks of Interpolation

- Interpolation: the fitted function passes through all data points.
- However, noisy data might be better represented by a simple function:



(a) Overfitting

Least-Squares

Formulation

More broadly, suppose we wish to solve the linear system $A\mathbf{x} = \mathbf{b}$ for \mathbf{x} . If we denote row k of A as \mathbf{r}_k^\top , then the system looks like

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} - & \mathbf{r}_1^\top & - \\ - & \mathbf{r}_2^\top & - \\ & \vdots & \\ - & \mathbf{r}_n^\top & - \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ by expanding } A\mathbf{x}$$
$$= \begin{pmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \mathbf{r}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{r}_n \cdot \mathbf{x} \end{pmatrix} \text{ by definition of matrix multiplication.}$$

- When using simpler models: # data points (n_b) > # model parameters (n_x) — **No solution!**
- Walk around: find x so that $Ax \approx b$, e.g. using **least-squares**: $\min_x \|Ax - b\|^2$ **When $Ax = b$ is solvable, the solutions are the same**

Least-Squares

Normal Equation

$$\|\mathbf{b} - A\mathbf{x}\|_2^2 = \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{b}^\top A\mathbf{x} + \|\mathbf{b}\|_2^2.$$

The gradient of this expression with respect to \mathbf{x} must be zero at its minimum, yielding the following system:

$$\mathbf{0} = 2A^\top A\mathbf{x} - 2A^\top \mathbf{b},$$

or equivalently, $A^\top A\mathbf{x} = A^\top \mathbf{b}.$

Theorem 4.1 (Normal equations). Minima of the residual norm $\|\mathbf{b} - A\mathbf{x}\|_2$ for $A \in \mathbb{R}^{m \times n}$ (with no restriction on m or n) satisfy $A^\top A\mathbf{x} = A^\top \mathbf{b}.$

The matrix $A^\top A$ is sometimes called a *Gram matrix*. If at least n rows of A are linearly independent, then $A^\top A \in \mathbb{R}^{n \times n}$ is invertible. In this case, the minimum residual occurs uniquely at $(A^\top A)^{-1}A^\top \mathbf{b}.$ Put another way:

In the overdetermined case, solving the least-squares problem $A\mathbf{x} \approx \mathbf{b}$ is equivalent to solving the *square* system $A^\top A\mathbf{x} = A^\top \mathbf{b}.$

Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment

Tikhonov Regularization

Motivation

- When A is a wide matrix ($m < n$), $Ax = b$ may have an infinite number of solutions.
- To choose between the possible solutions, we must make an additional assumption on x .
- The particular choice may be application-dependent, a common assumption is $\|x\|$ is small:
- For fixed $\alpha > 0$, we introduce an additional term to the minimization problem:

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \alpha \|\mathbf{x}\|_2^2$$

Tikhonov regularizer The formulation is also called Ridge regression.

- When α increases, finding x with small norm is prioritized more

Tikhonov Regularization

Derivation

To minimize this new objective, we take the derivative with respect to \mathbf{x} and set it equal to zero:

$$\mathbf{0} = 2A^\top A\mathbf{x} - 2A^\top \mathbf{b} + 2\alpha\mathbf{x},$$

or equivalently

$$(A^\top A + \alpha I_{n \times n})\mathbf{x} = A^\top \mathbf{b}.$$

When $A\mathbf{x} = \mathbf{b}$ is underdetermined, the matrix $A^\top A$ is not invertible. The new Tikhonov term resolves this issue, since for any $\mathbf{x} \neq \mathbf{0}$,

$$\mathbf{x}^\top (A^\top A + \alpha I_{n \times n})\mathbf{x} = \|A\mathbf{x}\|_2^2 + \alpha\|\mathbf{x}\|_2^2 > 0.$$

The strict $>$ holds because $\mathbf{x} \neq \mathbf{0}$; this inequality implies that $A^\top A + \alpha I_{n \times n}$ cannot have a null space vector \mathbf{x} . Hence, regardless of A , the Tikhonov-regularized system of equations is invertible. In the language we will introduce in §4.2.1, it is positive definite.

Tikhonov Regularization

Remarks

- Effective for dealing with null spaces and numerical issues
- When A is poorly conditioned, adding it can improve conditioning even if the original system was solvable
- Two drawbacks:
 - The solution x of the Tikhonov-regularized system no longer satisfies $Ax = b$ exactly.
 - When α is small, the matrix $A^T A + \alpha I_{n \times n}$ is invertible but may be poorly conditioned. Increasing α solves this problem at the cost of less accurate solutions to $Ax = b$.

Tikhonov Regularization

Example

Example 4.5 (Tikhonov regularization). Suppose we pose the following linear system:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 1 \\ 0.99 \end{pmatrix}.$$

This system is solved by $\mathbf{x} = (1001, -1000)$.

The scale of this $\mathbf{x} \in \mathbb{R}^2$, however, is much larger than that of any values in the original problem. We can use Tikhonov regularization to encourage smaller values in \mathbf{x} that still solve the linear system approximately. In this case, the Tikhonov system is

$$\left[\begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix}^\top \begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix} + \alpha I_{2 \times 2} \right] \mathbf{x} = \begin{pmatrix} 1 & 1 \\ 1 & 1.00001 \end{pmatrix}^\top \begin{pmatrix} 1 \\ 0.99 \end{pmatrix},$$

or equivalently,

$$\begin{pmatrix} 2 + \alpha & 2.00001 \\ 2.00001 & 2.0000200001 + \alpha \end{pmatrix} \mathbf{x} = \begin{pmatrix} 1.99 \\ 1.9900099 \end{pmatrix}.$$

| |
|----------------------------------------------------------------------------|
| $\alpha = 0.00001 \longrightarrow \mathbf{x} \approx (0.499998, 0.494998)$ |
| $\alpha = 0.001 \longrightarrow \mathbf{x} \approx (0.497398, 0.497351)$ |
| $\alpha = 0.1 \longrightarrow \mathbf{x} \approx (0.485364, 0.485366)$. |

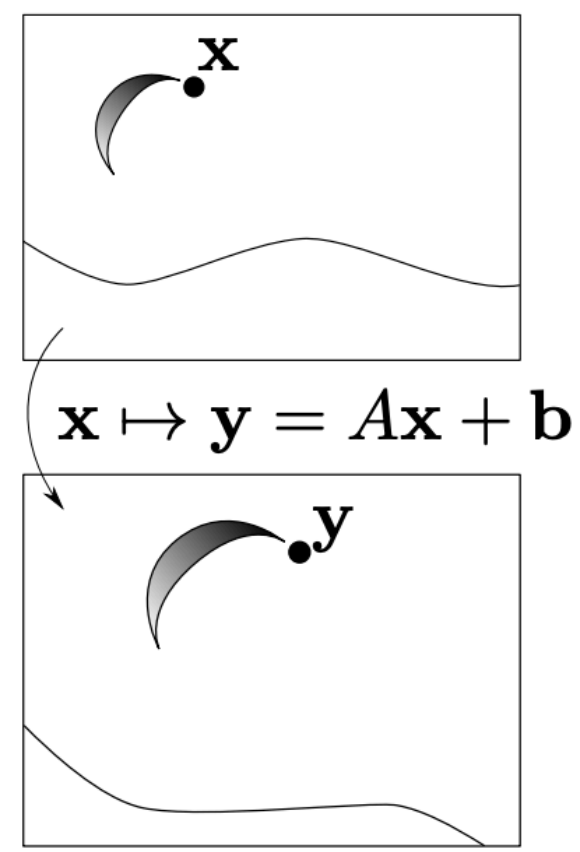
Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment

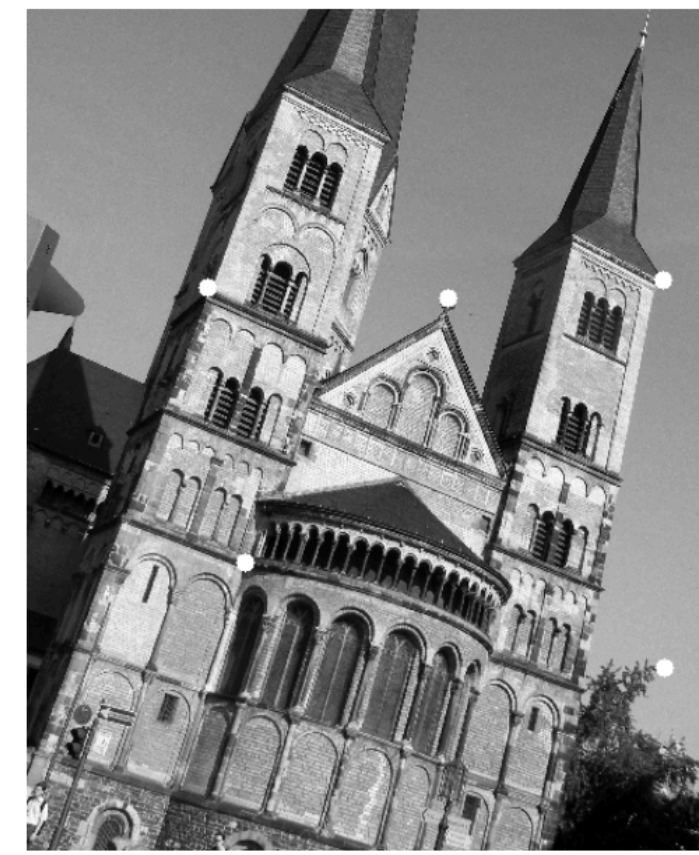
Image Alignment

Background

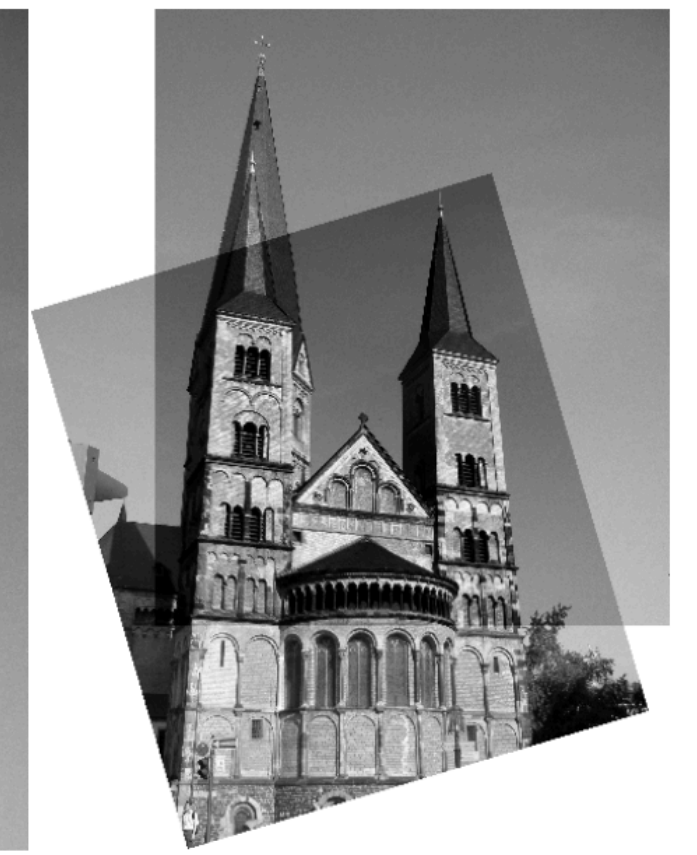
- To stitch 2 photos of the same scene taken from different angles,



(a)



(b) Input images with keypoints



(c) Aligned images

- the user (or an automatic system) marks p pairs of points $\mathbf{x}_k, \mathbf{y}_k \in \mathbb{R}^2$ such that $\forall k$ the location \mathbf{x}_k in image one corresponds to the location \mathbf{y}_k in image two.

When the camera makes a small motion, a reasonable assumption is that there exists some transformation matrix $A \in \mathbb{R}^{2 \times 2}$ and a translation vector $\mathbf{b} \in \mathbb{R}^2$ such that for all k ,

$$\mathbf{y}_k \approx A\mathbf{x}_k + \mathbf{b}.$$

That is, position \mathbf{x} on image one should correspond to position $A\mathbf{x} + \mathbf{b}$ on image two.

- Our goal is to compute the A and \mathbf{b} matching these points as closely as possible.

Image Alignment

Formulation

- Possible sources of error:
 - locating the corresponding points (input error),
 - slightly nonlinear camera projection of real-world lenses (model / approximation error)
- Thus, we ask that they are matched in a least-squares sense, and formulate:

$$\min_{A, \mathbf{b}} \sum_{k=1}^p \|(A\mathbf{x}_k + \mathbf{b}) - \mathbf{y}_k\|_2^2.$$

This problem has six unknowns total, the four elements of A and the two elements of \mathbf{b} .

Image Alignment

Derivation

$$f(A, \mathbf{b}) := \sum_k \|(A\mathbf{x}_k + \mathbf{b}) - \mathbf{y}_k\|_2^2.$$

We can simplify f as follows:

$$\begin{aligned} f(A, \mathbf{b}) &= \sum_k (A\mathbf{x}_k + \mathbf{b} - \mathbf{y}_k)^\top (A\mathbf{x}_k + \mathbf{b} - \mathbf{y}_k) \text{ since } \|\mathbf{v}\|_2^2 = \mathbf{v}^\top \mathbf{v} \\ &= \sum_k \left[\mathbf{x}_k^\top A^\top A \mathbf{x}_k + 2\mathbf{x}_k^\top A^\top \mathbf{b} - 2\mathbf{x}_k^\top A^\top \mathbf{y}_k + \mathbf{b}^\top \mathbf{b} - 2\mathbf{b}^\top \mathbf{y}_k + \mathbf{y}_k^\top \mathbf{y}_k \right] \end{aligned}$$

where terms with leading 2 apply the fact $\mathbf{a}^\top \mathbf{b} = \mathbf{b}^\top \mathbf{a}$.

Image Alignment

Derivation (cont.)

To find where f is minimized, we differentiate it with respect to \mathbf{b} and with respect to the elements of A , and set these derivatives equal to zero. This leads to the following system:

$$0 = \nabla_{\mathbf{b}} f(A, \mathbf{b}) = \sum_k [2A\mathbf{x}_k + 2\mathbf{b} - 2\mathbf{y}_k]$$

$$0 = \nabla_A f(A, \mathbf{b}) = \sum_k [2A\mathbf{x}_k\mathbf{x}_k^\top + 2\mathbf{b}\mathbf{x}_k^\top - 2\mathbf{y}_k\mathbf{x}_k^\top] \text{ by the identities in §1.4.3.}$$

The second equation is one of the first times we have encountered matrix calculus (§1.4.3) in our calculations. Simplifying somewhat, if we define $X := \sum_k \mathbf{x}_k\mathbf{x}_k^\top$, $\mathbf{x}_{\text{sum}} := \sum_k \mathbf{x}_k$, $\mathbf{y}_{\text{sum}} := \sum_k \mathbf{y}_k$, and $C := \sum_k \mathbf{y}_k\mathbf{x}_k^\top$, then the optimal A and \mathbf{b} satisfy the following linear system of equations:

$$\begin{aligned} A\mathbf{x}_{\text{sum}} + p\mathbf{b} &= \mathbf{y}_{\text{sum}} \\ AX + \mathbf{b}\mathbf{x}_{\text{sum}}^\top &= C. \end{aligned}$$

Image Alignment

Remarks

- A general pattern in modeling using least-squares:
 - Start by defining a desirable relationship between the unknowns, i.e. $(A\mathbf{x} + \mathbf{b}) - \mathbf{y} \approx 0$.
 - Given k data points $(\mathbf{x}_k, \mathbf{y}_k)$, design an objective function f measuring the quality of potential values for the unknowns A and \mathbf{b} by summing up the squared norms of expressions we wished to equal 0:

$$\sum_k \|A\mathbf{x}_k + \mathbf{b} - \mathbf{y}_k\|^2.$$

- Differentiating this sum gives a linear system of equations to solve for the best choice.
- This pattern is a common source of optimization problems that can be solved linearly and essentially is a subtle application of the normal equations.

Table of Content

- Parametric Regression
- Least-Squares
- Tikhonov Regularization
- Image Alignment