Master ID3D - Modèles statistiques pour l'image
Model fitting

Julie Digne
julie.digne@cnrs.fr

LIRIS - CNRS

09/09/2024

Today...

- Model regression
- Outlier-robust model regression (*outliers*)
- RANSAC algorithm for model regression

What do we model in this course?

- An explicit object model (circle, line, ellipse)...

What do we model in this course?

- An explicit object model (circle, line, ellipse)...
- A *transformation* between two objects

Application example: building a panorama

Image: Kai Herng Loh

Application example: segment detection

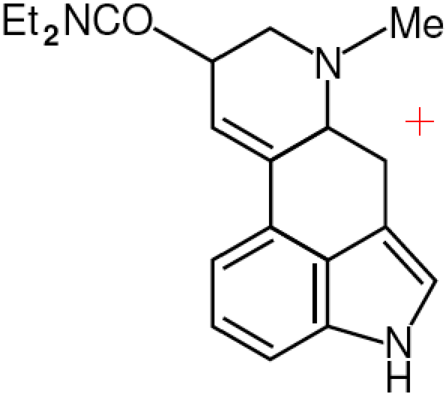


Image: Grompone von Gioi et al., IPOL, 2012, http://www.ipol.im/pub/art/2012/gjmr-lsd/

Application example: segment detection

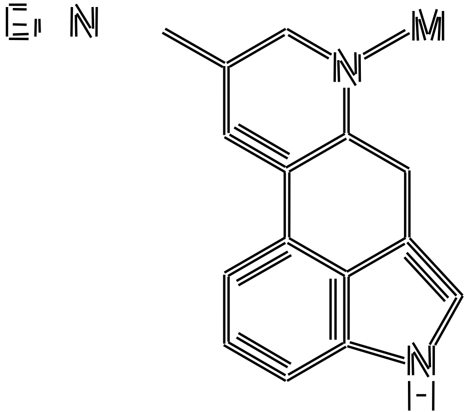


Image: **Grompone von Gioi et al.**, **IPOL**, **2012**, `http://www.ipol.im/pub/art/2012/gjmr-lsd/`

Outline

Outline

Toy example

- Let $(p_i)_{i=1 \cdots N}$ be points detected as belonging to a line in an image.

Toy example

- Let $(p_i)_{i=1 \cdots N}$ be points detected as belonging to a line in an image.
- We want to find the equations for these lines expressed as: $a \cdot x + b \cdot y + c = 0$

Toy example

- Let $(p_i)_{i=1\cdots N}$ be points detected as belonging to a line in an image.
- We want to find the equations for these lines expressed as: $a \cdot x + b \cdot y + c = 0$
- In the ideal case where all points lie exactly on the line, one needs to find a, b, c such that:

$$\forall i \in \{1 \cdots N\}\ a \cdot x_i + b \cdot y_i + c = 0$$

Toy example

- Let $(p_i)_{i=1\cdots N}$ be points detected as belonging to a line in an image.
- We want to find the equations for these lines expressed as: $a \cdot x + b \cdot y + c = 0$
- In the ideal case where all points lie exactly on the line, one needs to find a, b, c such that:

$$\forall i \in \{1 \cdots N\} \; a \cdot x_i + b \cdot y_i + c = 0$$

- Ideal case: 2 points (p_1, p_2) are enough to solve this problem.

Toy example

- Let $(p_i)_{i=1\cdots N}$ be points detected as belonging to a line in an image.
- We want to find the equations for these lines expressed as: $a \cdot x + b \cdot y + c = 0$
- In the ideal case where all points lie exactly on the line, one needs to find a, b, c such that:

$$\forall i \in \{1 \cdots N\} \ a \cdot x_i + b \cdot y_i + c = 0$$

- Ideal case: 2 points (p_1, p_2) are enough to solve this problem.

However, in practice...

We always have points that are not exactly on the line. One must find the line that best fits the points.

An example: polynomial regression

Exercise: Interpolation case

Find the parameters (a, b, c) of a parabola $y = ax^2 + bx + c$ passing through: $(-1, 1), (0, -1), (2, 7)$

An example: polynomial regression

Exercise: Interpolation case

Find the parameters (a, b, c) of a parabola $y = ax^2 + bx + c$ passing through: $(-1, 1), (0, -1), (2, 7)$

- How would you set up the problem of a polynomial interpolation of degree n?

An example: polynomial regression

Exercise: Interpolation case

Find the parameters (a, b, c) of a parabola $y = ax^2 + bx + c$ passing through: $(-1, 1), (0, -1), (2, 7)$

- How would you set up the problem of a polynomial interpolation of degree n?
- If the points are not exactly such that $f(x_i) = y_i$ but rather such that $y_i = f(x_i) + \varepsilon_i$ (where ε_i is a *noise*).

Practical case (In 2D)

- Find a line L minimizing the distance from points $p_i(x_i, y_i)$ to L

Practical case (In 2D)

- Find a line L minimizing the distance from points $p_i(x_i, y_i)$ to L
- L is given by the equation $ax + by + c = 0$

Practical case (In 2D)

- Find a line L minimizing the distance from points $p_i(x_i, y_i)$ to L
- L is given by the equation $ax + by + c = 0$
- The distance from the points to the line writes $|ax + by + c|$ if $a^2 + b^2 = 1$

Practical case (In 2D)

- Find a line L minimizing the distance from points $p_i(x_i, y_i)$ to L
- L is given by the equation $ax + by + c = 0$
- The distance from the points to the line writes $|ax + by + c|$ if $a^2 + b^2 = 1$
- A way to find the line equation is to solve for:

$$\min_{a,b,c} \sum_{i=1}^{n} (ax_i + by_i + c)^2 \text{ s.t. } a^2 + b^2 = 1$$

Regression problem formulation

Choice of a model

Let n variables $(X_i)_{i=1\cdots n}$ that *model* a variable Y through an unknown process $Y = \mathcal{F}(X_1, \cdots, X_n)$. Let $\mathcal{F}_\theta$ be a model that depends on a parameter $\theta \in \Theta$. We look for the value of θ that makes $\mathcal{F}_\theta(X_1, \cdots, X_n)$ *close to* Y:

$$\min_{\theta \in \Theta} \| Y - \mathcal{F}_\theta(X_1, \ldots, X_n) \|$$

- What is a model?

Regression problem formulation

Choice of a model

Let n variables $(X_i)_{i=1\cdots n}$ that *model* a variable Y through an unknown process $Y = \mathcal{F}(X_1, \cdots, X_n)$. Let $\mathcal{F}_\theta$ be a model that depends on a parameter $\theta \in \Theta$. We look for the value of θ that makes $\mathcal{F}_\theta(X_1, \cdots, X_n)$ *close to* Y:

$$\min_{\theta \in \Theta} \|Y - \mathcal{F}_\theta(X_1, \ldots, X_n)\|$$

- What is a model?
- How do we find the optimal θ?

Regression problem formulation

Choice of a model

Let n variables $(X_i)_{i=1\cdots n}$ that *model* a variable Y through an unknown process $Y = \mathcal{F}(X_1, \cdots, X_n)$. Let $\mathcal{F}_\theta$ be a model that depends on a parameter $\theta \in \Theta$. We look for the value of θ that makes $\mathcal{F}_\theta(X_1, \cdots, X_n)$ *close to* Y:

$$\min_{\theta \in \Theta} \|Y - \mathcal{F}_\theta(X_1, \ldots, X_n)\|$$

- What is a model?
- How do we find the optimal θ?
- How do we measure the distance between Y and the prediction?

Classical example: least squares regression line

Exercise

Find the line given by parameters $\theta = (a, b, c)$ such that:

$$\min_{a,b,c} \sum_{i=1} (ax_i + by_i + c)^2 \text{ s.t. } a^2 + b^2 = 1$$

In general

Different minimization problems

$$\min_{\|u\|=1} \sum_i u^T x_i$$

$$\min_{\|u\|} \|Mu - b\|^2$$

$$\dots$$

A very common regression case

Solving a least squares problem

Let $M \in \mathbb{R}^{m,n}$, $b \in \mathbb{R}^m$, we look for $u \in \mathbb{R}^n$ such that $Mu = b$. If $m > n$, we relax the system as:

$$\min_{u \in \mathbb{R}^n} \|Mu - b\|_2^2$$

- $\|Mu - b\|_2^2 = u^T M^T M u - 2u^T M^T b + b^T b$

A very common regression case

Solving a least squares problem

Let $M \in \mathbb{R}^{m,n}$, $b \in \mathbb{R}^m$, we look for $u \in \mathbb{R}^n$ such that $Mu = b$. If $m > n$, we relax the system as:

$$\min_{u \in \mathbb{R}^n} \|Mu - b\|_2^2$$

- $\|Mu - b\|_2^2 = u^T M^T M u - 2u^T M^T b + b^T b$
- Zeroing the gradient: $2M^T M u - 2M^T b = 0$ and $(M^T M) \cdot u = M^T b$ (*Normal Equation*)

A very common regression case

Solving a least squares problem

Let $M \in \mathbb{R}^{m,n}$, $b \in \mathbb{R}^m$, we look for $u \in \mathbb{R}^n$ such that $Mu = b$. If $m > n$, we relax the system as:

$$\min_{u \in \mathbb{R}^n} \|Mu - b\|_2^2$$

- $\|Mu - b\|_2^2 = u^T M^T M u - 2u^T M^T b + b^T b$
- Zeroing the gradient: $2M^T Mu - 2M^T b = 0$ and $(M^T M) \cdot u = M^T b$ (*Normal Equation*)
- If $M^T M$ can be inverted: $u = (M^T M)^{-1} M^T b$

Modeling a transform between two objects

Problem setting

Let (p_i, q_i) be $\mathbb{R}^2$ points such that p_i is paired to point q_i, we look for the transform T among a family of transforms $\mathcal{T}$ such that:

$$\min_{T \in \mathcal{T}} \sum_{i=1 \cdots n} \|q_i - T(p_i)\|$$

- It is still a model choice for p_i to explain q_i.
- We need to choose a norm.
- Transforms can be rotations, translations, or an affinity ...

Rigid transform estimation example

Exercise

Let $(p_i, q_i)_{i=1\cdots n}$ n pairs of matched points in $\mathbb{R}^2$, we are looking for a rigid transform (A, b) such that $q_i \approx Ap_i + b$ ($A \in \mathbb{R}^{2,2}$, $b \in \mathbb{R}^2$).

- Objective function to minimize: $\sum_{i=1}^{n} \|q_i - Ap_i - b\|_2^2$

Rigid transform estimation example

Exercise

Let $(p_i, q_i)_{i=1\cdots n}$ n pairs of matched points in $\mathbb{R}^2$, we are looking for a rigid transform (A, b) such that $q_i \approx Ap_i + b$ ($A \in \mathbb{R}^{2,2}$, $b \in \mathbb{R}^2$).

- Objective function to minimize: $\sum_{i=1}^{n} \|q_i - Ap_i - b\|_2^2$
- Differentiation w.r.t. b:

$$\sum_{i=1}^{n} 2Ap_i + 2b - 2q_i = 0$$

Rigid transform estimation example

Exercise

Let $(p_i, q_i)_{i=1\cdots n}$ n pairs of matched points in $\mathbb{R}^2$, we are looking for a rigid transform (A, b) such that $q_i \approx Ap_i + b$ ($A \in \mathbb{R}^{2,2}, b \in \mathbb{R}^2$).

- Objective function to minimize: $\sum_{i=1}^{n} \|q_i - Ap_i - b\|_2^2$
- Differentiation w.r.t. b:

$$\sum_{i=1}^{n} 2Ap_i + 2b - 2q_i = 0$$

- Differentiation w.r.t. A?

Rigid transform estimation example

Exercise

Let $(p_i, q_i)_{i=1\cdots n}$ n pairs of matched points in $\mathbb{R}^2$, we are looking for a rigid transform (A, b) such that $q_i \approx Ap_i + b$ ($A \in \mathbb{R}^{2,2}$, $b \in \mathbb{R}^2$).

- Objective function to minimize: $\sum_{i=1}^{n} \|q_i - Ap_i - b\|_2^2$
- Differentiation w.r.t. b:

$$\sum_{i=1}^{n} 2Ap_i + 2b - 2q_i = 0$$

- Differentiation w.r.t. A?

Differentiation formulas with respect to a vector or a matrix: http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf

Image example

(a) (b) Input images with keypoints (c) Aligned images

Image: Justin Solomon

Generalization

Problem

$(y_i) \in \mathbb{R}^n$, $(x_i) \in \mathbb{R}^m$, Find $A \in \mathbb{R}^{n \times m}$ minimizing:

$$\sum_{i=1}^{n} \|y_i - Ax_i\|_2^2 = \|Y - AX\|_F^2$$

with $Y = \begin{pmatrix} y_1 & y_2 & \cdots & y_p \end{pmatrix}$ and $X = \begin{pmatrix} x_1 & x_2 & \cdots & x_p \end{pmatrix}$ (the first norm is the ℓ^2 *norm*, the second norm is the Frobenius norm)

Regularization

Regularization

Adding a *regularization term* permits to favor some solutions for which the problem might be easier to solve.

Regularization

Regularization

Adding a *regularization term* permits to favor some solutions for which the problem might be easier to solve.

- Tikhonov Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_2$

Regularization

Regularization

Adding a *regularization term* permits to favor some solutions for which the problem might be easier to solve.

- Tikhonov Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_2$
- L^0 Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_0$

Regularization

Regularization

Adding a *regularization term* permits to favor some solutions for which the problem might be easier to solve.

- Tikhonov Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_2$
- L^0 Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_0$
- L^1 Regularization: $\min \|y - Ax\|_2^2 + \lambda \|x\|_1$

Outline

Estimating a rotation

- 2D: A point and an angle: unitary complex number.
- 3D: An axis and an angle:

Estimating a rotation

- 2D: A point and an angle: unitary complex number.
- 3D: An axis and an angle: Unitary quaternion

Estimating a rotation in 3D: Quaternions

- Can be seen as a generalization of the complex numbers to higher dimension.
- $\dot{q} = q_0 + q_1 \mathrm{i} + q_2 \mathrm{j} + q_3 \mathrm{k}$
- Conjugate of a quaternion $\dot{q}^* = q_0 - q_1 \mathrm{i} - q_2 \mathrm{j} - q_3 \mathrm{k}$
- Unitary quaternion $\|\dot{q}\|^2 = \dot{q} \cdot \dot{q}^* = 1$
- A rotation of axis (w_x, w_y, w_z) and angle θ can be seen as the quaternion:

$$\cos \frac{\theta}{2} + \sin \frac{\theta}{2} (w_x \mathrm{i} + w_y \mathrm{j} + w_z \mathrm{k})$$

Manipulating quaternions as matrices

- Vector in space correspond to imaginary quaternions ($q_0 = 0$)
- Advantage: easier to work with than rotation matrices
- The translation can be deduced [Horn 87]

Better: rotation estimation using SVD (Procrustes problem)

Let $\mathcal{P} = (p_i)_{i=1\cdots n}$ and $\mathcal{Q} = (q_i)_{i=1\cdots n}$ such that (p_i, q_i) is a matched pair.
Goal: Find R, t minimizing

$$F(R,T) = \sum_{i=1}^{n} \|Rp_i + T - q_i\|_2^2.$$

1. Centering $\tilde{p}_i = p_i - \frac{1}{n}\sum_{i=1}^{n} p_i$; $\tilde{q}_i = q_i - \frac{1}{n}\sum_{i=1}^{n} q_i$.
2. Compute $M = P \cdot Q^T$ and its svd $M = USV^T$
3. Compute

$$R = V \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & det(VU^T) \end{pmatrix} U^T$$

4. ... and

$$T = \frac{1}{n}\sum_{i=1}^{n} q_i - R(\frac{1}{n}\sum_{i=1}^{n} p_i)$$

Outline

A brief reminder on norms

Norm definition

Let E be a vector space over a subfield K, a norm on E is an application with nonnegative values $\|\| : E \rightarrow R$ such that for all $\alpha \in K$ and $u, v \in E$:

- $\|\alpha v\| = |\alpha|\|v\|$ (positive homogeneity)
- $\|u + v\| \leq \|u\| + \|v\|$ (subadditivity)
- $\|u\| = 0_K \Leftrightarrow u = 0_E$ (separation)

A brief reminder on norms

Norm definition

Let E be a vector space over a subfield K, a norm on E is an application with nonnegative values $\|\|: E \rightarrow R$ such that for all $\alpha \in K$ and $u, v \in E$:

- $\|\alpha v\| = |\alpha|\|v\|$ (positive homogeneity)
- $\|u + v\| \leq \|u\| + \|v\|$ (subadditivity)
- $\|u\| = 0_K \Leftrightarrow u = 0_E$ (separation)

- The ℓ^2 norm is also called the euclidean norm. Let x be a vector in $\mathbb{R}^n$ with coordinates $(x_1, \cdots, x_n)$ in the canonical basis, the ℓ^2 norm writes:

$$\|x\|_2 = \sqrt{x \cdot x^T} = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$$

Norm Examples on vectors of $\mathbb{R}^n$

- ℓ^1 Norm (Manhattan)

$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$

Norm Examples on vectors of $\mathbb{R}^n$

- ℓ^1 Norm (Manhattan)

$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$

- ℓ^3

$$\|x\|_3 = (\sum_{i=1}^{n} |x_i|^3)^{\frac{1}{3}}$$

Norm Examples on vectors of $\mathbb{R}^n$

- ℓ^1 Norm (Manhattan)

$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$

- ℓ^3

$$\|x\|_3 = (\sum_{i=1}^{n} |x_i|^3)^{\frac{1}{3}}$$

- $\ell^{2.1}$:

$$\|x\|_{2.1} = (\sum_{i=1}^{n} x_i^{2.1})^{\frac{1}{2.1}}$$

Norm Examples on vectors of $\mathbb{R}^n$

- ℓ^1 Norm (Manhattan)

$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$

- ℓ^3

$$\|x\|_3 = (\sum_{i=1}^{n} |x_i|^3)^{\frac{1}{3}}$$

- $\ell^{2.1}$:

$$\|x\|_{2.1} = (\sum_{i=1}^{n} x_i^{2.1})^{\frac{1}{2.1}}$$

- ℓ^p pour $p \geq 1$

$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

Norm Examples on vectors of $\mathbb{R}^n$

- ℓ^1 Norm (Manhattan)

$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$

- ℓ^3

$$\|x\|_3 = (\sum_{i=1}^{n} |x_i|^3)^{\frac{1}{3}}$$

- $\ell^{2.1}$:

$$\|x\|_{2.1} = (\sum_{i=1}^{n} x_i^{2.1})^{\frac{1}{2.1}}$$

- ℓ^p pour $p \geq 1$

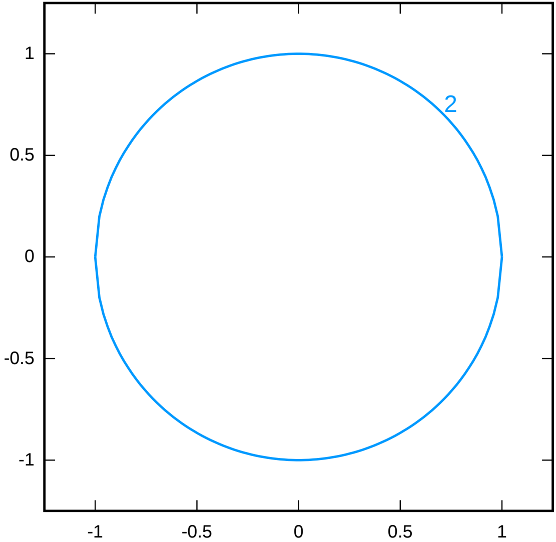
$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

- ℓ^∞

$$\|x\|_\infty = \max_{i=1\cdots n} |x_i|$$

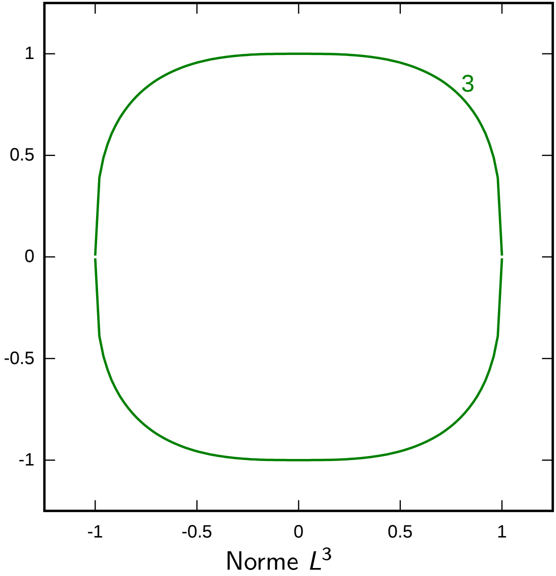
Exercice: Prove that ℓ^∞ is indeed a norm?

The ball of radius 1 for norms ℓ^p with $p \geq 2$

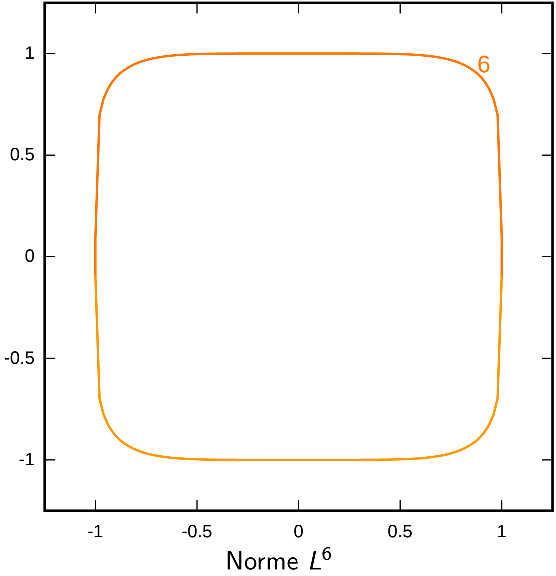


Norme L^2

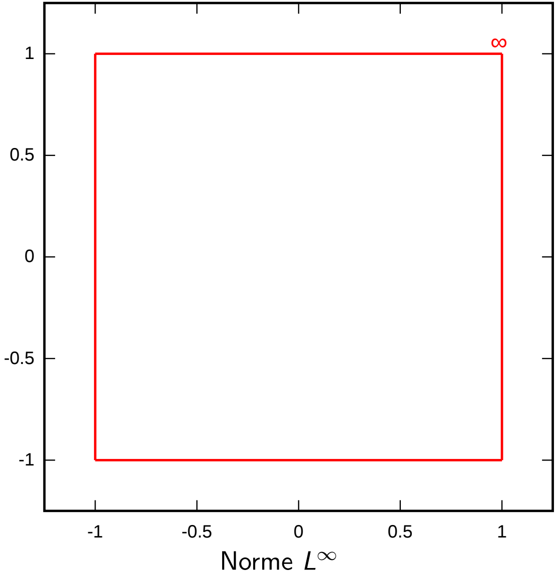
The ball of radius 1 for norms ℓ^p with $p \geq 2$



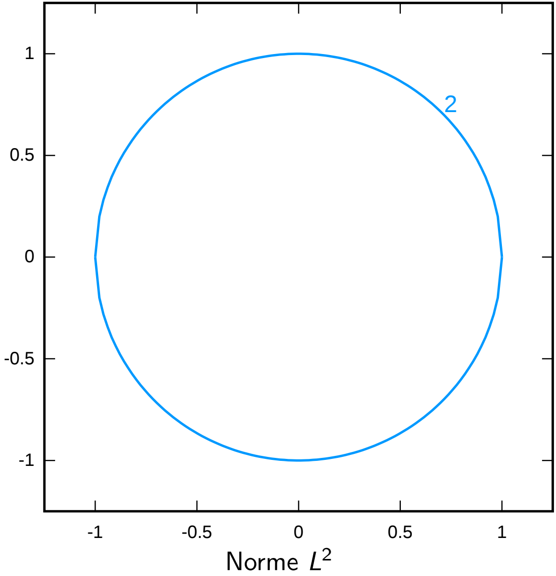
The ball of radius 1 for norms ℓ^p with $p \geq 2$



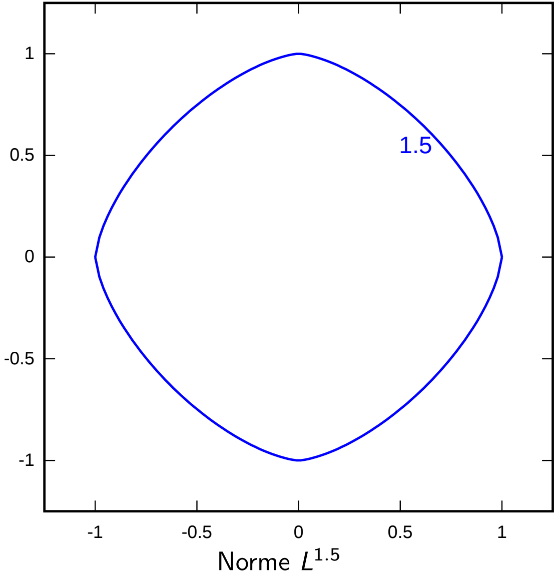
The ball of radius 1 for norms ℓ^p with $p \geq 2$



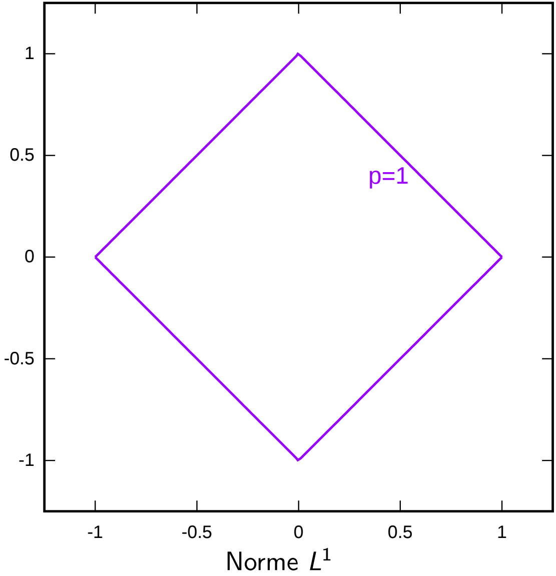
The ball of radius 1 for norms ℓ^p with $p \leq 2$



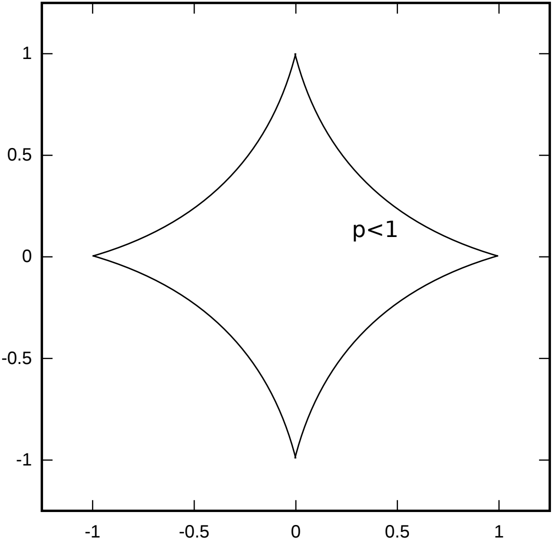
The ball of radius 1 for norms ℓ^p with $p \leq 2$



The ball of radius 1 for norms ℓ^p with $p \leq 2$



The ball of radius 1 for norms ℓ^p with $p \leq 2$



Quasi-Norm ℓ^{p}, $p < 1$

The ball of radius 1 with norms and quasi-norms ℓ^p

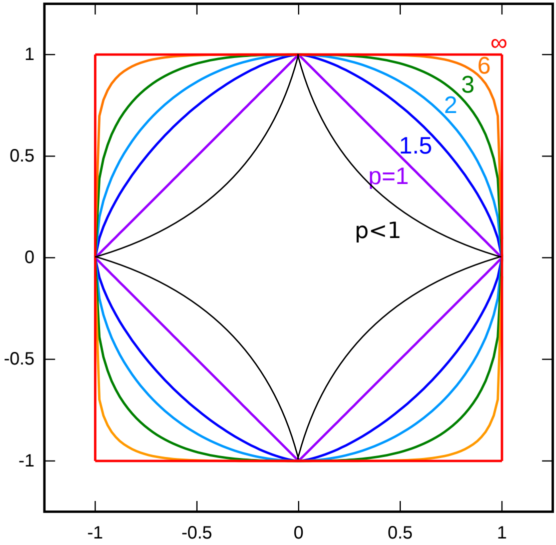


Image wikipedia (modified)

Norm and sparsity

Sparsity definition

A vector $x \in \mathbb{R}^N$ is said to be s-sparse if at most s of its entries are non zero, i.e.

$$\text{card}\, support(x) \leq s$$

where $support(x) = \{i | x_i \neq 0\}$.
We note $\|x\|_0 = \text{card}\, support(x)$ and call it ℓ^0.

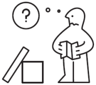
Norm and sparsity

Sparsity definition

A vector $x \in \mathbb{R}^N$ is said to be *s*-sparse if at most s of its entries are non zero, i.e.

$$\text{card}\, support(x) \leq s$$

where $support(x) = \{i | x_i \neq 0\}$.
We note $\|x\|_0 = \text{card}\, support(x)$ and call it ℓ^0.

- Is ℓ^0 a norm?

Norm and sparsity

Sparsity definition

A vector $x \in \mathbb{R}^N$ is said to be s-sparse if at most s of its entries are non zero, i.e.

$$\text{card}\, support(x) \leq s$$

where $support(x) = \{i | x_i \neq 0\}$.
We note $\|x\|_0 = \text{card}\, support(x)$ and call it ℓ^0.

-  Is ℓ^0 a norm?
- $\|x\|_0$ is the limit of $\|x\|_p^p$ for $p \to 0$

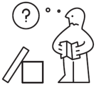
Norm and sparsity

Sparsity definition

A vector $x \in \mathbb{R}^N$ is said to be s-sparse if at most s of its entries are non zero, i.e.

$$\text{card}\, support(x) \leq s$$

where $support(x) = \{i | x_i \neq 0\}$.
We note $\|x\|_0 = \text{card}\, support(x)$ and call it ℓ^0.

- 

Is ℓ^0 a norm?
- $\|x\|_0$ is the limit of $\|x\|_p^p$ for $p \to 0$

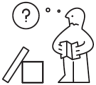
Norm and sparsity

Sparsity definition

A vector $x \in \mathbb{R}^N$ is said to be s-sparse if at most s of its entries are non zero, i.e.

$$\text{card}\, support(x) \leq s$$

where $support(x) = \{i | x_i \neq 0\}$.
We note $\|x\|_0 = \text{card}\, support(x)$ and call it ℓ^0.

- Is ℓ^0 a norm?
- $\|x\|_0$ is the limit of $\|x\|_p^p$ for $p \to 0$
- Optimization with L^0 constraints: nonconvex problems $\Rightarrow$ very hard to solve!

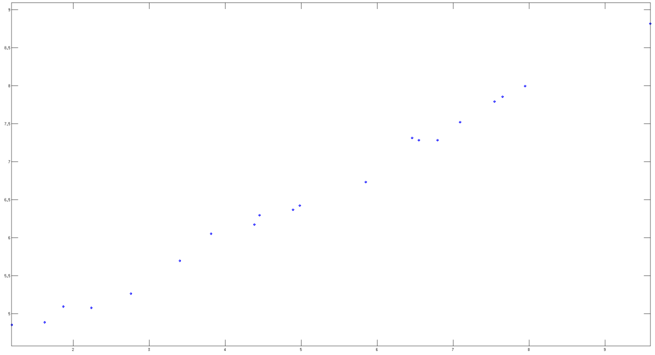
L1 regression

Least Absolute Deviation

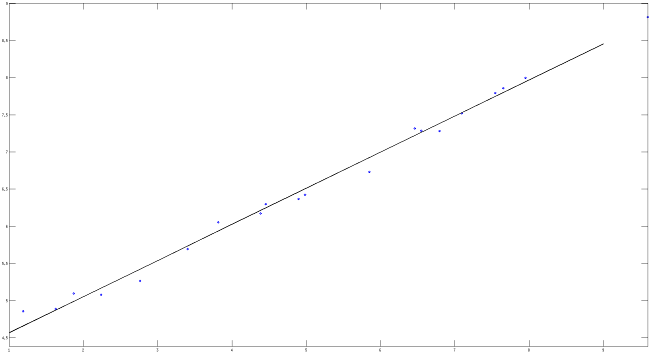
Let (x_i, y_i) be points of $\mathbb{R}^2$. We look for θ minimizing:

$$\sum_{i=1}^{n} |f_\theta(x_i) - y_i|$$

Example

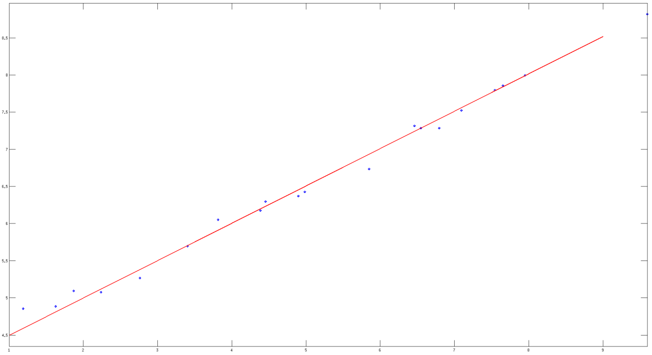


L1 vs L2



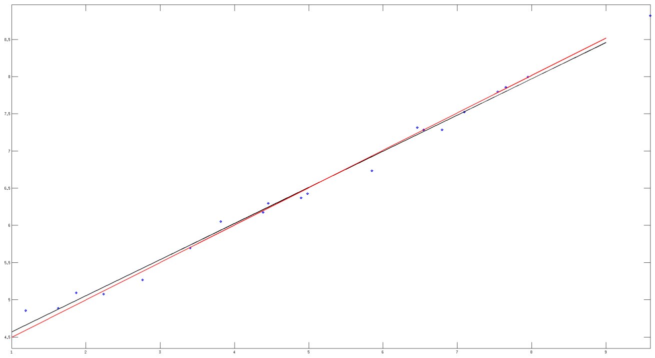
L^2

L1 vs L2



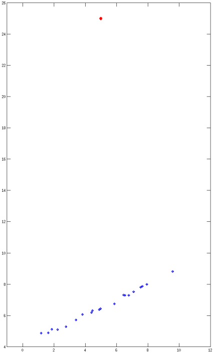
L^1

L1 vs L2



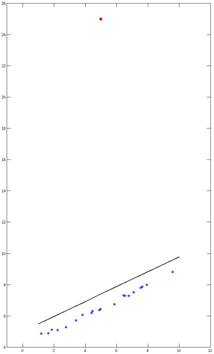
L^1 (red) and L^2 (black)

With outliers



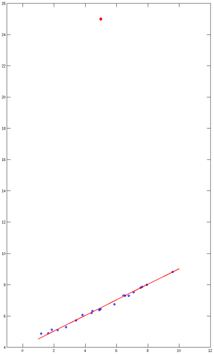
A single outlier

With outliers



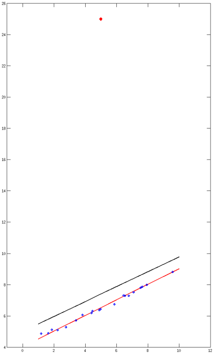
L^2

With outliers



L^1

With outliers



L^1 (red) and L^2 (black)

L^1 vs L^2

L^1 norm

- Robust to outliers

L^2 norm

L^1 vs L^2

L^1 norm

- Robust to outliers

L^2 norm

- Very sensitive to outliers

L^1 vs L^2

<u>L^1 norm</u>

- Robust to outliers
- Non differentiable, instable

<u>L^2 norm</u>

- Very sensitive to outliers

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable

L^2 norm

- Very sensitive to outliers
- Easy to optimize

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*

L^2 norm

- Very sensitive to outliers
- Easy to optimize

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*

L^2 norm

- Very sensitive to outliers
- Easy to optimize
- No penalty for vectors with small nonzero coefficients

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*
- slow optimization

L^2 norm

- Very sensitive to outliers
- Easy to optimize
- No penalty for vectors with small nonzero coefficients

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*
- slow optimization

L^2 norm

- Very sensitive to outliers
- Easy to optimize
- No penalty for vectors with small nonzero coefficients
- Fast!

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*
- slow optimization

L^2 norm

- Very sensitive to outliers
- Easy to optimize
- No penalty for vectors with small nonzero coefficients
- Fast!

Exemple

$x = \begin{pmatrix} 0.01 & 0.5 & 1 & 2 & 0.009 & 0.000012 \end{pmatrix}$; $y = \begin{pmatrix} 0 & 0 & 1 & 2 & 0 & 0 \end{pmatrix}$

L^1 vs L^2

L^1 norm

- Robust to outliers
- Non differentiable, instable
- *sparsity inducing*
- slow optimization

L^2 norm

- Very sensitive to outliers
- Easy to optimize
- No penalty for vectors with small nonzero coefficients
- Fast!

Exemple

$x = \begin{pmatrix} 0.01 & 0.5 & 1 & 2 & 0.009 & 0.000012 \end{pmatrix}$; $y = \begin{pmatrix} 0 & 0 & 1 & 2 & 0 & 0 \end{pmatrix}$

$$\|x\|_1 = 3.519$$

$$\|y\|_1 = 3$$

$$\|x\|_2 = 2.2913$$

$$\|y\|_2 = 2.2361$$

L^2: Adding a confidence value

- Transform estimation setting: n pairs of matched points (p_i, q_i).

L^2: Adding a confidence value

- Transform estimation setting: n pairs of matched points (p_i, q_i).
- Assume we can measure if the pair is trustworthy or not and encode it as f_i

L^2: Adding a confidence value

- Transform estimation setting: n pairs of matched points (p_i, q_i).
- Assume we can measure if the pair is trustworthy or not and encode it as f_i
- Let $w_i = \frac{f_i}{\sum_k f_k}$

L^2: Adding a confidence value

- Transform estimation setting: n pairs of matched points (p_i, q_i).
- Assume we can measure if the pair is trustworthy or not and encode it as f_i
- Let $w_i = \frac{f_i}{\sum_k f_k}$

Weighted regression

$$\min_{T \in \mathcal{T}} \sum_{i=1 \cdots n} w_i \|q_i - T(p_i)\|_2^2$$

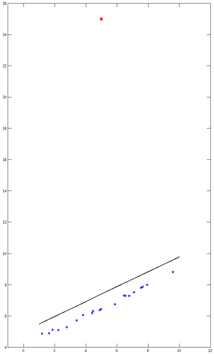
L^2: Adding a confidence value

- Transform estimation setting: n pairs of matched points (p_i, q_i).
- Assume we can measure if the pair is trustworthy or not and encode it as f_i
- Let $w_i = \frac{f_i}{\sum_k f_k}$

Weighted regression

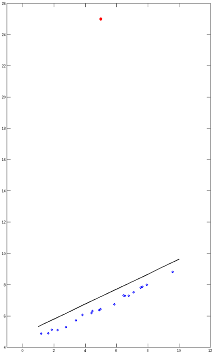
$$\min_{T \in \mathcal{T}} \sum_{i=1 \cdots n} w_i \|q_i - T(p_i)\|_2^2$$

L^2: Adding a confidence value



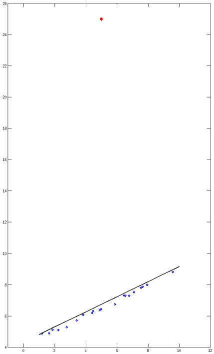
$$\forall i, f_i = 1 \;\; w_i = \frac{1}{N} \Rightarrow \text{classical Least Squares.}$$

L^2: Adding a confidence value



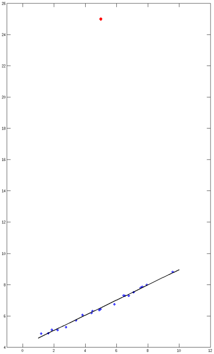
$f_i = 1$, except for the outlier: $f_{i_0} = 0.9$

L^2: Adding a confidence value



$f_i = 1$, except for the outlier: $f_{i_0} = 0.5$

L^2: Adding a confidence value



$f_i = 1$, except for the outlier: $f_{i_0} = 0.1$

Algorithm for L^1 regression:

- We will use weighted L^2 regression to approximate a L^1 regression

Algorithm for L^1 regression:

- We will use weighted L^2 regression to approximate a L^1 regression
- We do not know which points are outliers!

Algorithm for L^1 regression:

- We will use weighted L^2 regression to approximate a L^1 regression
- We do not know which points are outliers!
- Iterate two steps: 1) regression and 2) weights recomputation

Algorithm for L^1 regression:

- We will use weighted L^2 regression to approximate a L^1 regression
- We do not know which points are outliers!
- Iterate two steps: 1) regression and 2) weights recomputation
- Iteratively Reweighted least squares (IRLS)

Iterative Reweighted Least Squares

Algorithm 1: IRLS

Input: Data x_i, y_i

Output: parameter θ of the model

Set $w_i = 1/n$;

do

  Find the parameter θ minimizing $\sum_{i=1}^{n} w_i \|f_\theta(x_i) - y_i\|_2^2$;

  Update the weights $w_i = \frac{1}{|y_i - f_\theta(x_i)|}$;

Until *Convergence*;

Iterative Reweighted Least Squares

Algorithm 2: IRLS

Input: Data x_i, y_i

Output: parameter θ of the model

Set $w_i = 1/n$;

do

    Find the parameter θ minimizing $\sum_{i=1}^{n} w_i \|f_\theta(x_i) - y_i\|_2^2$;

    Update the weights $w_i = \frac{1}{|y_i - f_\theta(x_i)|}$;

Until *Convergence*;

- Pro: doable for large-scale problems.

Iterative Reweighted Least Squares

Algorithm 3: IRLS

Input: Data x_i, y_i
Output: parameter θ of the model
Set $w_i = 1/n$;
do
    Find the parameter θ minimizing $\sum_{i=1}^{n} w_i \|f_\theta(x_i) - y_i\|_2^2$;
    Update the weights $w_i = \frac{1}{|y_i - f_\theta(x_i)|}$;
Until *Convergence*;

- Pro: doable for large-scale problems.
- Con: Iterative solve.

Iterative Reweighted Least Squares

Algorithm 4: IRLS

Input: Data x_i, y_i
Output: parameter θ of the model
Set $w_i = 1/n$;
do
    Find the parameter θ minimizing $\sum_{i=1}^{n} w_i \|f_\theta(x_i) - y_i\|_2^2$;
    Update the weights $w_i = \frac{1}{|y_i - f_\theta(x_i)|}$;
Until *Convergence*;

- Pro: doable for large-scale problems.
- Con: Iterative solve.
- Safety: Avoid divisions by 0 in the weights update:

$$w_i = \frac{1}{max(\delta, |y_i - f_\theta(x_i)|)} \text{ where } \delta \text{ is small}$$

Iteratively Reweighted Least Squares

Algorithm 5: IRLS

Input: Data x_i, y_i
Output: parameter θ of the model

Set $w_i = 1/n$;
do
  Find the parameter θ minimizing $\sum_{i=1}^{n} w_i \|f_\theta(x_i) - y_i\|_2^2$;
  Update the weights $w_i = \frac{1}{|y_i - f_\theta(x_i)|}$;
Until *Convergence*;

- Pro: doable for large-scale problems.
- Con: Iterative solve.
- Safety: Avoid divisions by 0 in the weights update:

$$w_i = \frac{1}{max(\delta, |y_i - f_\theta(x_i)|)} \text{ where } \delta \text{ is small}$$

- This algorithm can be adapted for all quasi-norms ℓ^p with $p < 1$

Exact Solution: linear programming

Definition

A linear problem is a problem where the objective function and the equality or inequality constraints are linear with respect to the variables.

Exact Solution: linear programming

Definition

A linear problem is a problem where the objective function and the equality or inequality constraints are linear with respect to the variables.

- Example: Minimize $x + 2y + 3z$ s.t. $x + y = 1$, $x - z \leq 2$, $z \geq 0$
- Valid if $f_\theta(x_i) = \theta^T \cdot x_i$

L^1 regression as a linear program

Least Absolute Deviation as a linear program

$$\underset{m_i,\theta}{Minimize} \sum_i m_i$$

$$\text{s.t.} \forall i, m_i \geq y_i - \theta^T \cdot x_i$$

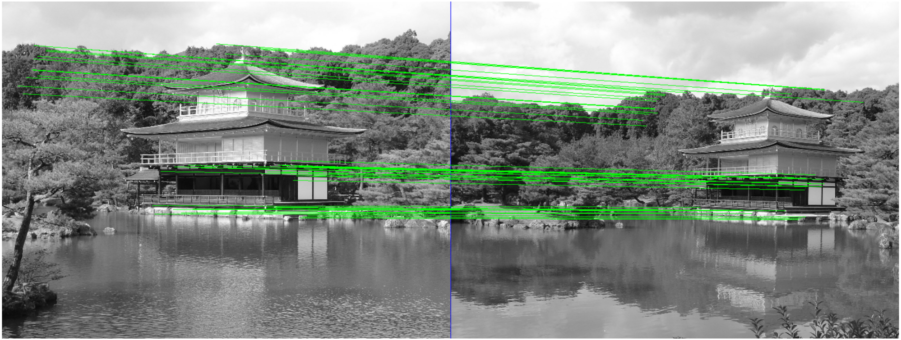
$$\text{and} \forall i, m_i \geq -(y_i - \theta^T \cdot x_i)$$

Can be solved with the *Simplex Algorithm*.

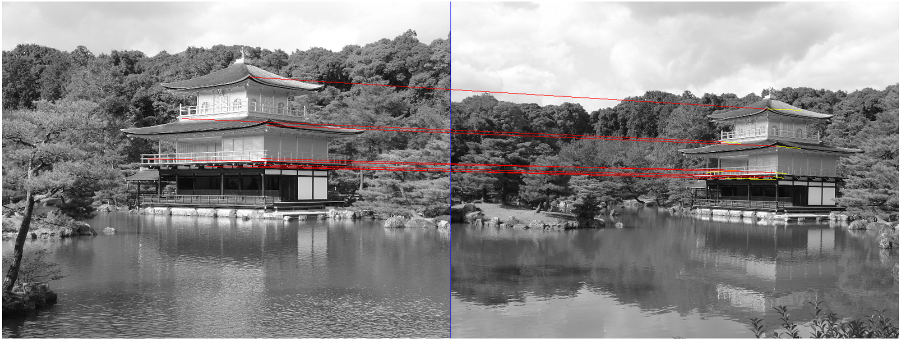
Outline

Outliers

Outliers



Outliers



Outliers

Estimation of models or transforms...

Problem statement

n pairs of matched points (p_i, q_i) that should be such that:

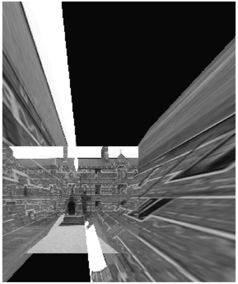
$$q_i = \mathcal{T}(p_i)$$

where $\mathcal{T}$ is an arbitrary model that can be estimated with m pairs of points. The goal is to find the best model $\mathcal{T}$ and a subset of pairs that have a consensus on the model. This consensus subset is the set of *inliers*.

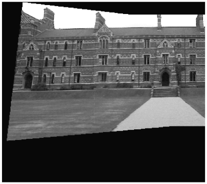
- Transform linking two pictures of the same scene: homography H
- If two points (x_1, y_1) in image 1 and (x_2, y_2) in image 2 are matched, then:

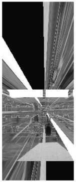
$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$$

Estimating only from all pairs



Estimating only from m random pairs





RANSAC for model estimation

- Input: n matched pairs (p_i, q_i) possibly containing false matches
- Repeat k times:
  - Select m pairs and estimate $\mathcal{T}$
  - Compute the number of pairs *who agree with* $\mathcal{T}$
  - If this score is the highest yet, store $\mathcal{T}$ and the consensus set

Estimating with RANSAC

Why is RANSAC efficient?

- $\mathcal{T}$ is estimated on a very small set of points ($\Rightarrow$ fast)

Why is RANSAC efficient?

- $\mathcal{T}$ is estimated on a very small set of points ($\Rightarrow$ fast)
- It is easy to compute the score of a transform.

Why is RANSAC efficient?

- $\mathcal{T}$ is estimated on a very small set of points ($\Rightarrow$ fast)
- It is easy to compute the score of a transform.
- We can give statistical guarantees for RANSAC

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs:

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Result

- $p = 0.99$, $w \approx 0.99$ yields $k \approx 2$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Result

- $p = 0.99$, $w \approx 0.99$ yields $k \approx 2$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Result

- $p = 0.99$, $w \approx 0.99$ yields $k \approx 2$
- $p = 0.99$, $w \approx 0.7$ yields $k \approx 11$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Result

- $p = 0.99$, $w \approx 0.99$ yields $k \approx 2$
- $p = 0.99$, $w \approx 0.7$ yields $k \approx 11$

Some guarantees for RANSAC

- $w = \frac{\#inliers}{\#pairs}$ probability to pick an inlier among the n pairs
- Probability to pick at least one outlier $1 - w^m$
- Probability to always pick an outlier among the m pairs: $(1 - w^m)^k$
- Success probability p:

$$1 - p = (1 - w^m)^k$$

Result

- $p = 0.99$, $w \approx 0.99$ yields $k \approx 2$
- $p = 0.99$, $w \approx 0.7$ yields $k \approx 11$
- $p = 0.99$, $w \approx 0.6$ yields $k \approx 19$

Conclusion

- Model regression or transform estimates are found in a vast variety of image problems.
- One must choose the model, the norm and the right algorithm
- Registration of images but also of 3D shapes!

Images from the "david laser scanner" website.