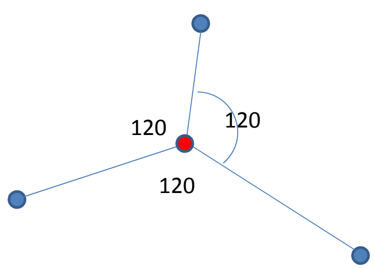
Iterative Reweighted Least Squares

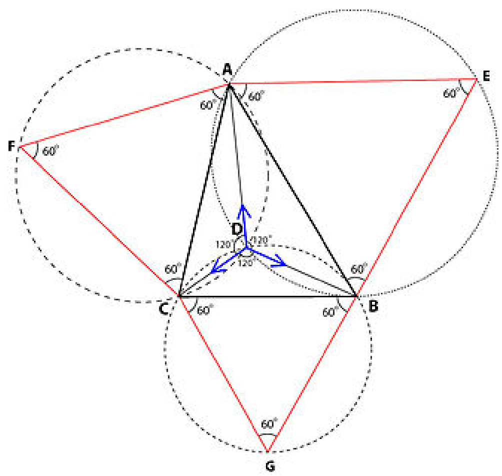
ECCV, Sept 7, 2014

Three Points in R^2
The Fermat point of the triangle

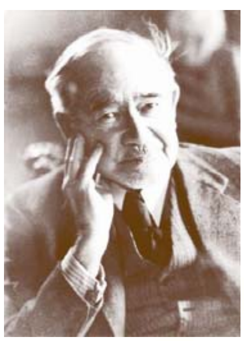


What point minimizes the distance to the three points of a triangle?

Exercise: find this point using ruler and compass construction.



Ruler and Compass Construction
to find Fermat point



Alfred Weber

ig.2 Varignon frame (Weber 1929: 229)

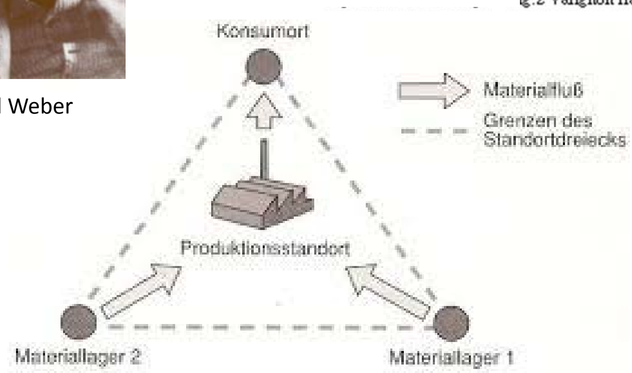
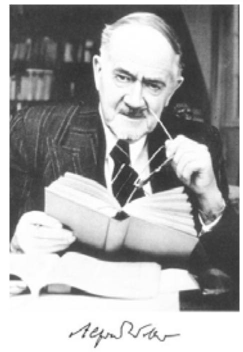


Abb. 1: Das Standortdreieck
Quelle: REICHART,, Baustein der Wirtschaftsgeographie, S. 45 Abb. II-6

Weber's Solution (1906)

- Center of Gravity



Gustav Weler was a political decoy (doppelgänger or Body-double) of Adolf Hitler. At the end of the Second World War, he was executed by a gunshot to the forehead in an attempt to confuse the Allied troops when Berlin was taken.[citation needed] He was also used "as a decoy for security reasons".[2] When his corpse was discovered in the Reich Chancellery garden by Soviet troops, it was mistakenly believed to be that of Hitler because of his identical moustache and haircut. The corpse was also photographed and filmed by the Soviets.

One servant from the bunker declared that the dead man was one of Hitler's cooks. He also believed this man "had been assassinated because of his startling likeness to Hitler, while the latter had escaped from the ruins of Berlin".[3]

Weler's body was brought to Moscow for investigations and buried in the yard at Lefortovo prison.[4]

Gustav Weler

Fermat Weber problem

1. In $\mathbb{R}^1$
   - L_2 average of several points is the mean,
   - L_1 average is the median - more rosbust to outliers. Computable in linear time.
2. In $\mathbb{R}^2$ or $\mathbb{R}^n$ the problem is a classical problem.
3. Considered by Fermat, Torricelli (1636), Weber (1906), Weiszfeld (1933).
4. More Recent Work:
   - Speed up through prediction (Ostresh 1978),
   - Banach spaces,
   - Riemannian manifolds (Fletcher 2009, Yang 2010).

14/63

Andrew Vázsonyi (1916–2003), also known as **Endre Weiszfeld** and **Zepartzatt Gozinto**) was a mathematician and operations researcher. He is known for Weiszfeld's algorithm for minimizing the sum of distances to a set of points, and for founding The Institute of Management Sciences.[1][2][3]

Weiszfeld

E. Weiszfeld, Sur le point pour lequel la somme des distances de n points donnés est minimum, Tôhoku Mathematics Journal 43 (1937), 355 - 386.

Weiszfeld Algorithm for points

NICTA

1. An iterative algorithm to find L_1 minimum point of a set of points.
2. Given a set of points $\mathbf{y}_i$, the cost function to minimize is

$$\mathbf{x}^* = \underset{\mathbf{x}}{\text{argmin}} \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i) \ ,$$

where $d(\mathbf{x}, \mathbf{y}_i)$ is the distance of $\mathbf{x}$ and $\mathbf{y}_i$.

17/63

Weiszfeld Algorithm for points

NICTA

- Given points $\mathbf{y}_i \in \mathbb{R}^N$, find the point that minimizes the L_1 cost function

$$C_1(\mathbf{x}) = \sum_{i=1}^{n} d(\mathbf{x}, \mathbf{y}_i)$$

Robust (L1) cost function

- Given a current estimate $\mathbf{x}^t$, the Weiszfeld algorithm computes the next estimate $\mathbf{x}^{t+1}$ as

$$\mathbf{x}^{t+1} = \frac{\sum_{i=1}^{n} w_i^t \mathbf{y}_i}{\sum_{i=1}^{n} w_i^t} = \text{argmin}_{\mathbf{x}} \sum_{i=1}^{n} w_i^t \, d(\mathbf{x}, \mathbf{y}_i)^2$$

where $w_i^t = 1/d(\mathbf{x}^t, \mathbf{y}_i)$.

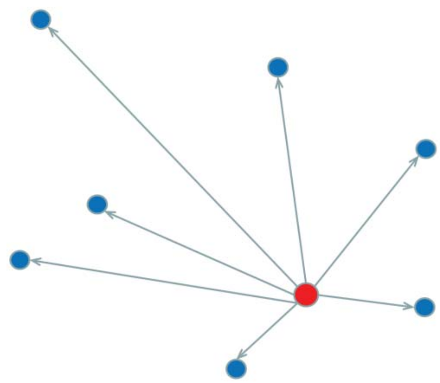
Weighted L2 cost function

- $\mathbf{x}^{t+1}$ is the centre of gravity of a configuration formed by placing a weight w_i^t at each point $\mathbf{y}_i$.

18/63

Weiszfeld Algorithm for points

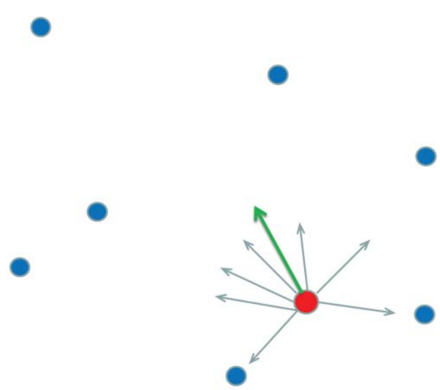
- Given a set of points in space. We start with a random initial estimation of the median,



21/63

Weiszfeld Algorithm for points

- Compute the sum of negative gradients

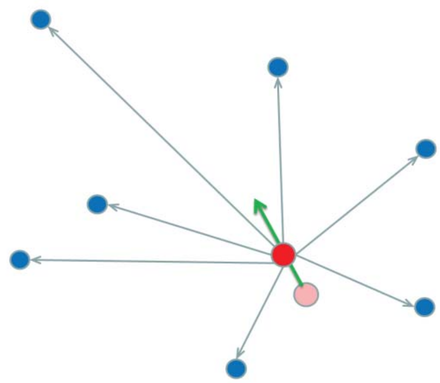


22/63

Weiszfeld Algorithm for points

NICTA

- Move the estimate in the downhill direction.



Gradient descent

- In R^n the cost function

$$C(\mathbf{y}) = \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{y}) = \sum \|x_i - y\|$$

is convex, and has a single minimum (unless all x_i are collinear).

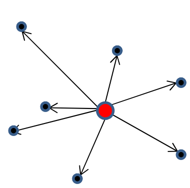
- Gradient is

$$\nabla C = \sum_{i=1}^{n} (\mathbf{y} - \mathbf{x}_i)/\|\mathbf{y} - \mathbf{x}_i\|$$

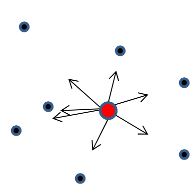
- Gradient descent algorithm:

$$\mathbf{y}^{t+1} = \mathbf{y}^t + \gamma^t \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}^t)/\|\mathbf{x}_i - \mathbf{y}^t\|$$

γ^t is the step-size.

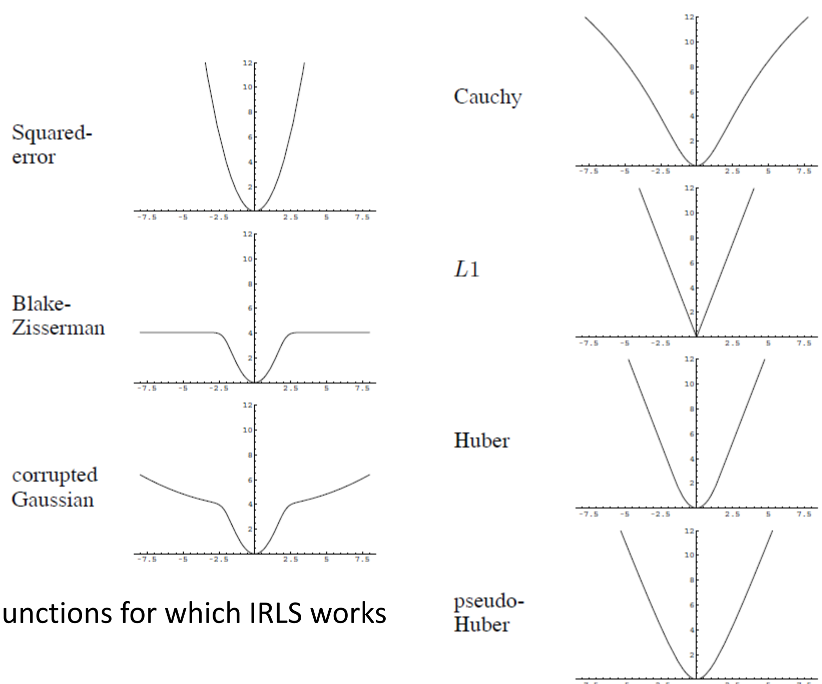


Gradient of L2 distance



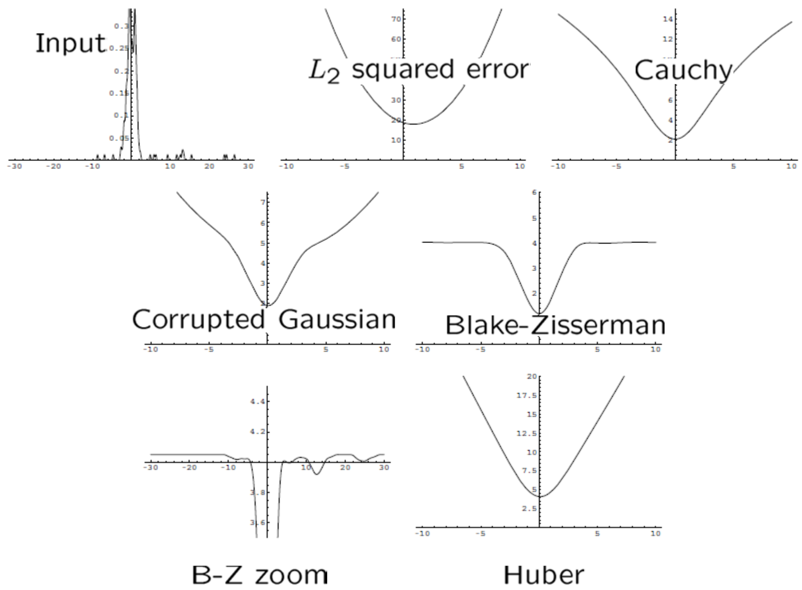
Gradient of L1 distance

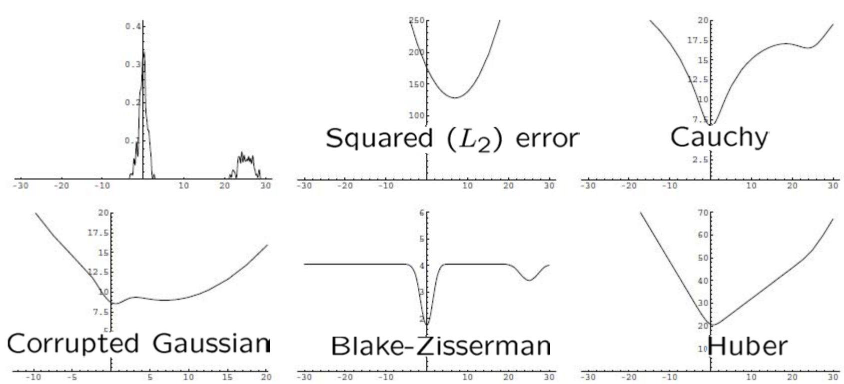
Generalizing IRLS



Functions for which IRLS works

Resistance to Outliers





Robustness to systematic outliers
(e.g. ghost edges)

A general IRLS algorithm

1. Identify a weighted optimization problem that can be solved optimally (e.g. in closed form)

$$C(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i \, f_i(\mathbf{x})$$

Written without the squares

2. Solve iteratively: At each step, define weights (how)

$$w_i^t = w_i(\mathbf{x}^t)$$

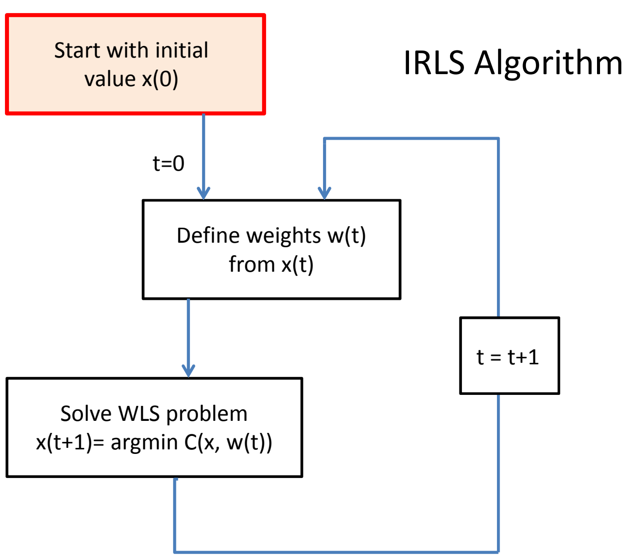
Define weights

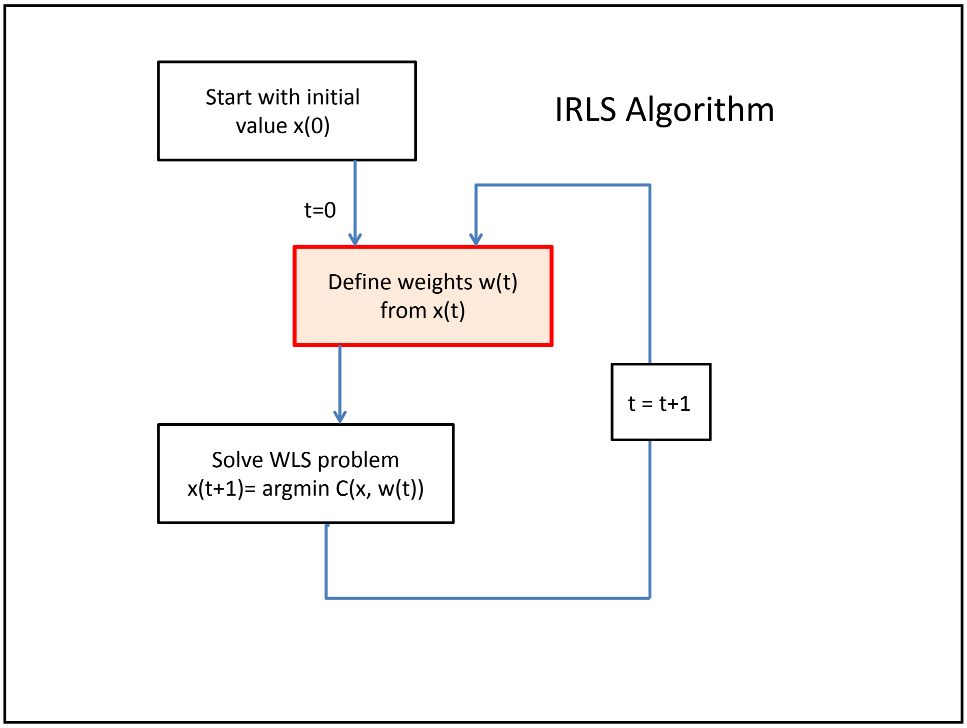
and define

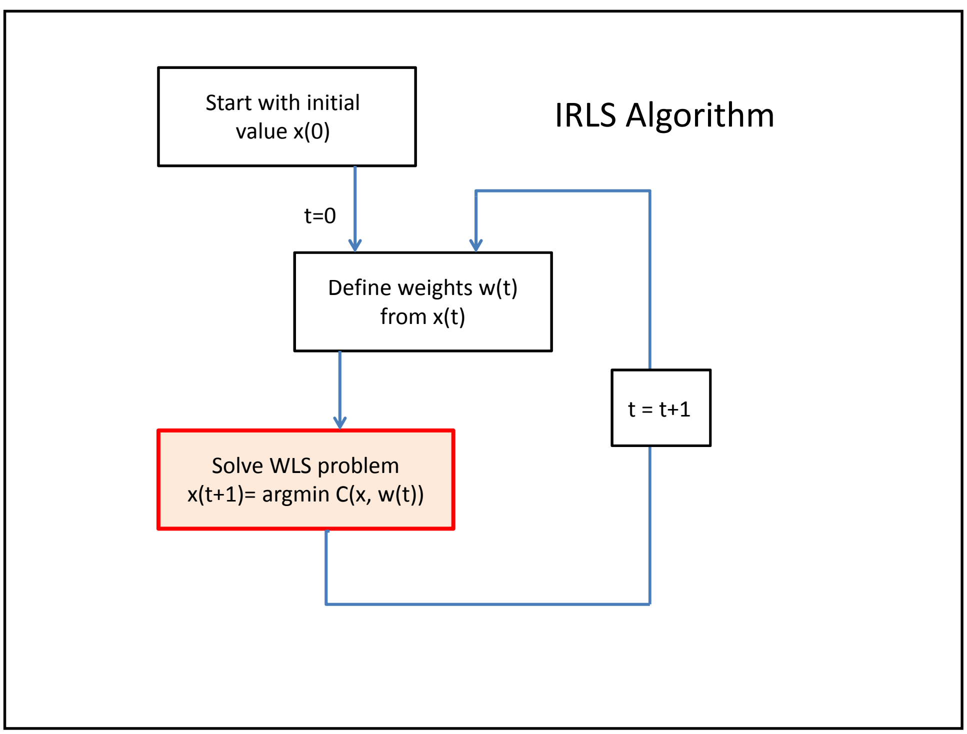
$$\begin{aligned} \mathbf{x}^{t+1} &= \text{argmin}_{\mathbf{x}} \, C(\mathbf{x}, \mathbf{w}^t) \\ &= \text{argmin}_{\mathbf{x}} \sum_{i=1}^{n} w_i^t \, f_i(\mathbf{x}) \end{aligned}$$

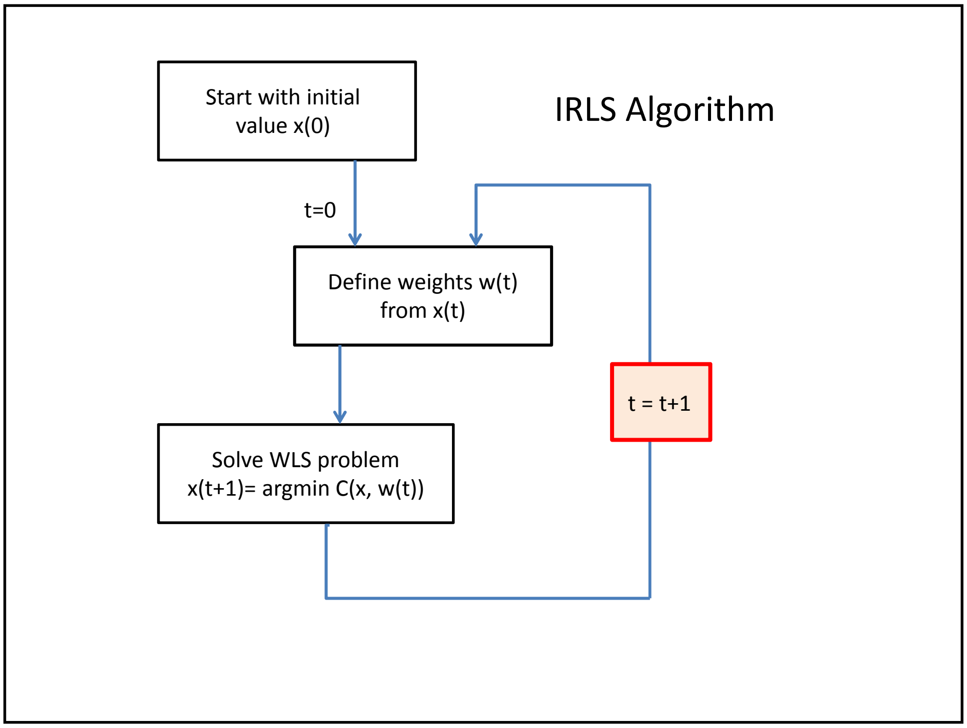
Minimize weighted cost

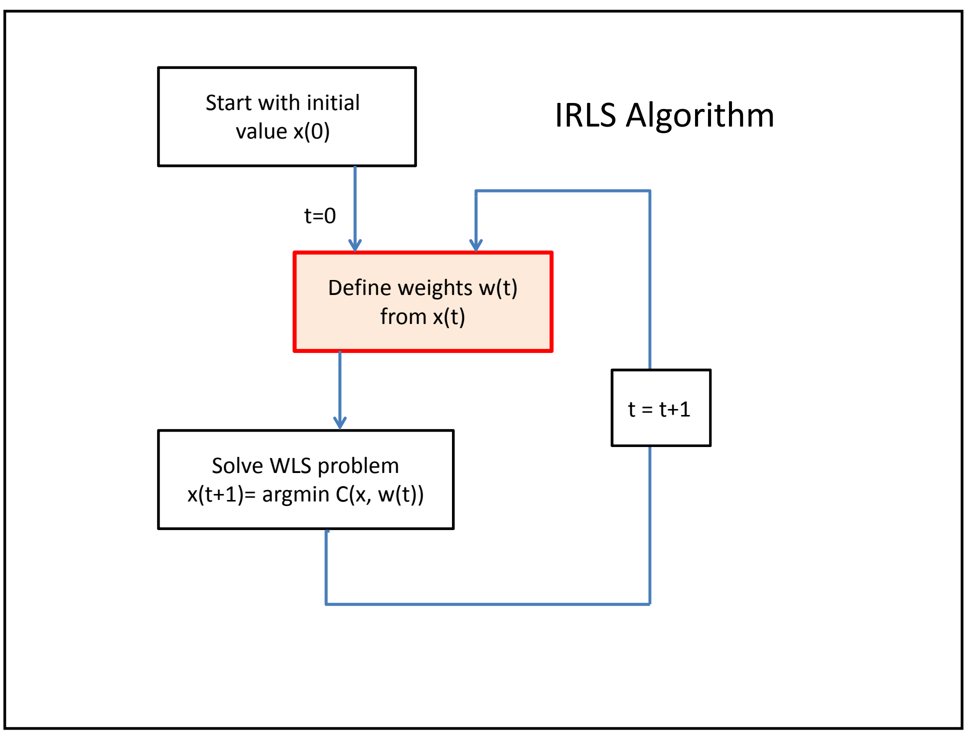
3. Hope that it converges to what you want.




Start with initial value x(0)
IRLS Algorithm
t=0
Define weights w(t) from x(t)
t = t+1
Solve WLS problem x(t+1)= argmin C(x, w(t))


Start with initial value x(0)
IRLS Algorithm
t=0
Define weights w(t) from x(t)
t = t+1
Solve WLS problem x(t+1)= argmin C(x, w(t))


Start with initial value x(0)
IRLS Algorithm
t=0
Define weights w(t) from x(t)
t = t+1
Solve WLS problem x(t+1)= argmin C(x, w(t))


Start with initial value x(0)
IRLS Algorithm
t=0
Define weights w(t) from x(t)
t = t+1
Solve WLS problem x(t+1)= argmin C(x, w(t))

How to choose the weights

- Assume we can minimize the cost

$$C(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i f_i(\mathbf{x}) \ .$$

← No square !!

- We wish to minimize

$$C_h(\mathbf{x}) = \sum_{i=1}^{n} h \circ f_i(\mathbf{x}) \ .$$

Robust cost function

- We want

$$\nabla_{\mathbf{x}} C(\mathbf{x}, \mathbf{w}) = 0 \text{ if and only if } \nabla_{\mathbf{x}} C_h(\mathbf{x}) = 0 \ .$$

- So

$$\begin{aligned} \nabla_{\mathbf{x}} w_i f_i(\mathbf{x}) &= \nabla_{\mathbf{x}}(h \circ f_i(\mathbf{x})) \\ w_i \nabla_{\mathbf{x}} f_i(\mathbf{x}) &= h'(f_i(\mathbf{x})).\nabla_{\mathbf{x}} f_i \end{aligned}$$

$$\boxed{w_i^t = h'(f_i(x^t))}$$

Required weights

Example L_1

Let

$$\begin{aligned} f_i(\mathbf{x}) &= d(\mathbf{x}, \mathbf{y}_i)^2 \\ h(\mathbf{x}) &= \sqrt{\mathbf{x}} \\ C_h(\mathbf{x}) &= \sum_{i=1}^{n} h \circ f_i(\mathbf{x}) = \sum_{i=1}^{n} d(\mathbf{x}, \mathbf{y}_i) \end{aligned}$$

Sum of distances

Then

$$\begin{aligned} w_i &= h'(f(\mathbf{x})) \\ &= \frac{1}{2} f(\mathbf{x})^{-1/2} \\ &= \boxed{\frac{1}{2} d(\mathbf{x}, \mathbf{y}_i)^{-1} \ .} \end{aligned}$$

Example L_q

Let

$$f_i(\mathbf{x}) = d(\mathbf{x}, \mathbf{y}_i)^2$$
$$h(\mathbf{x}) = x^{q/2}$$
$$C_h(\mathbf{x}) = \sum_{i=1}^{n} h \circ f_i(\mathbf{x}) = \boxed{\sum_{i=1}^{n} d(\mathbf{x}, \mathbf{y}_i)^q}$$

Then

$$w_i = h'(f(\mathbf{x}))$$
$$= \frac{q}{2} f(\mathbf{x})^{(q-2)/2}$$

$$\boxed{w_i = \frac{q}{2} d(\mathbf{x}, \mathbf{y}_i)^{q-2}}$$

Descent condition for IR least sum

Lemma: Let $h : R \to R$ be a concave function and let h^s denote a supergradient of h. For $i = 1, \ldots, n$ let r_i^t and r_i^{t+1} be real numbers (residuals) such that

$$\sum_{i=1}^{n} w_i r_i^{t+1} \leq \sum_{i=1}^{n} w_i r_i^t$$

Weighted residual sum decreases

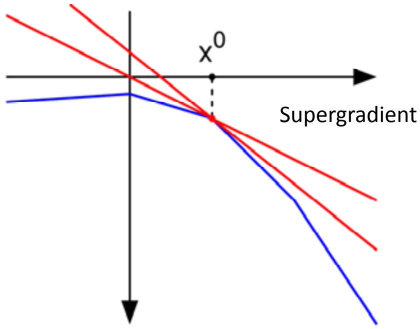
where $w_i = h^s(r_i^t)$. Then

$$\sum_{i=1}^{n} h(r_i^{t+1}) \leq \sum_{i=1}^{n} h(r_i^t)$$

Robust residual sum decreases

with equality if and only if $r_i^{t+1} = r_i^t$ for all i. //

Apply this with $r_i^t = f_i(\mathbf{x}^t)$ and $r_i^{t+1} = f_i(\mathbf{x}^{t+1})$.



A concave function always has a supergradient

Proof: Since h^s is a supergradient,

$$h(r_i^{t+1}) \le h(r_i^t) + (r_i^{t+1} - r_i^t)\, h^s(r_i^t)$$

Definition of supergradient

for all i. Summing over i gives

$$\sum_{i=1}^{n} h(r_i^{t+1}) \le \sum_{i=1}^{n} h(r_i^t) + \sum_{i=1}^{n} (r_i^{t+1} - r_i^t) h^s(r_i^t) \ .$$

The last sum is non-positive by hypothesis, completing the proof. //

General condition for descent of IRLS

Corollary: Let $h : R^+ \to R$ be a function such that $h(\sqrt{x})$ is concave. For $i = 1, \ldots, n$ let r_i^t and r_i^{t+1} be non-negative real numbers (residuals) such that

$$\sum_{i=1}^{n} w_i \ (r_i^{t+1})^2 \le \sum_{i=1}^{n} w_i \ (r_i^t)^2$$

Weighted squared residual decreases

where $w_i = h'(r_i^t)/r_i^t$. Then

$$\sum_{i=1}^{n} h(r_i^{t+1}) \le \sum_{i=1}^{n} h(r_i^t)$$

Robust cost decreases

with equality if and only if $r_i^{t+1} = r_i^t$ for all i. //

Summary

- To minimize

$$C_h(\mathbf{x}) = \sum_{i=1}^{n} h \circ f_i(\mathbf{x}) \qquad (1)$$

minimize the weighted L_2 cost

$$C_2^{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{n} w_i^t f_i(\mathbf{x})^2 \qquad (2)$$

with weights

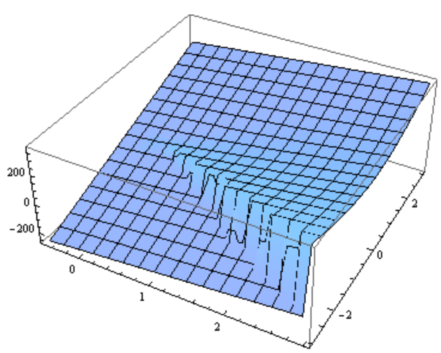
$$w_i^t = \left. \frac{h'(y)}{y} \right|_{\mathbf{y} \to f_i(\mathbf{x}^t)} .$$

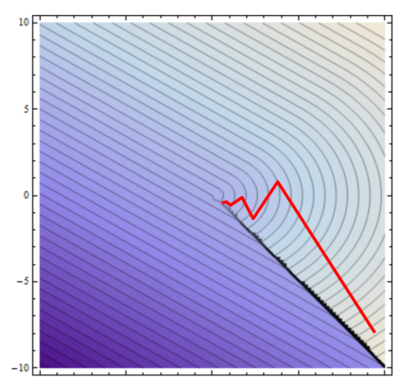
- Decrease in weighted L_2 cost guarantees a decrease in the robust cost, as long as:

$h(\sqrt{x})$ is concave.

Convergence

- **Warning:** Decrease in cost is no guarantee that the sequence of iterates converges!!





Where gradient descent does not converge to a minimum

Convergence conditions

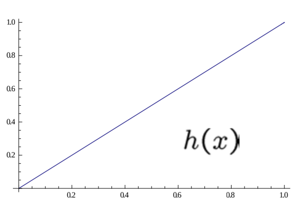
If

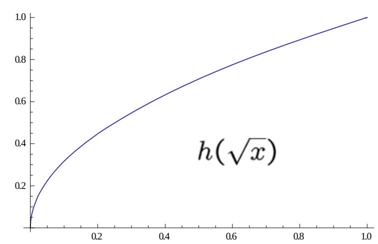
- $h(\sqrt{x})$ is concave and has continuous derivative (for $x \geq 0$);
- $f_i(x)^2$ is continuously differentiable.
- $\text{argmin}_{\mathbf{x}} C_2^{\mathbf{w}}(\mathbf{x})$ is continuous as a function of the weights w_i,

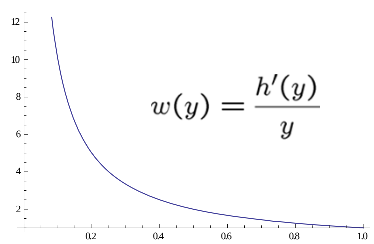
then IRLS will converge to the set of critical points of $C_h(\mathbf{x})$.

Hence of $C_h(\mathbf{x})$ is convex, then IRLS will converge to the global minimum.

L1



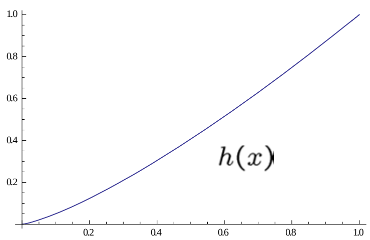


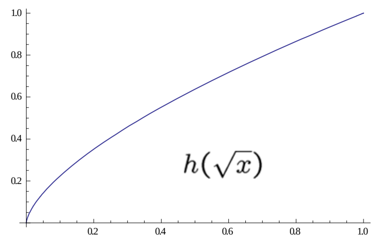


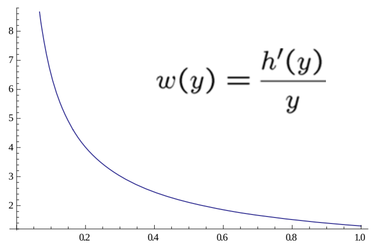
Advantage: Robust

Disadvantage:
- Function not differentiable
- Weights not defined at 0
- Can stop at non-minimum.

Lq



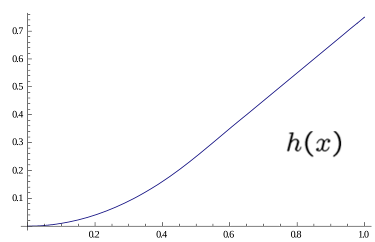


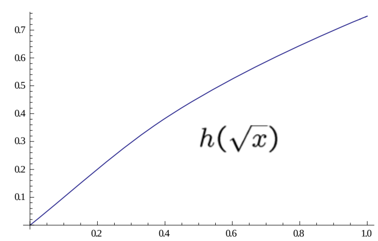


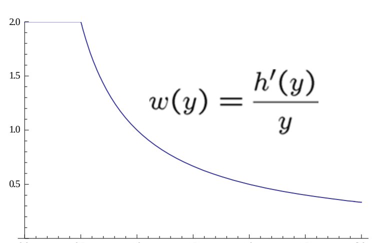
Advantage:
- Robust
- Cost function differentiable everywhere

Disadvantage:
- Weights not defined at 0
- Can stop at non-minimum.

Huber

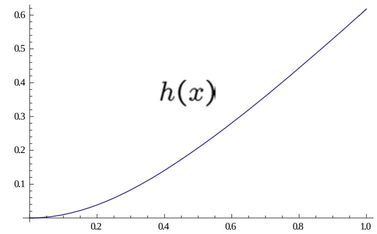


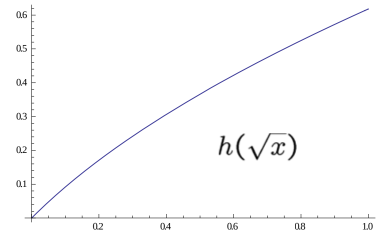


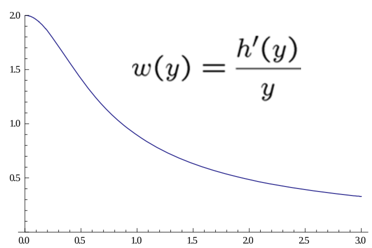


Advantage:
- Robust
- Cost function differentiable
- Weights defined at zero
- Convex
- Guaranteed to converge (at least to local minimum)

Pseudo Huber

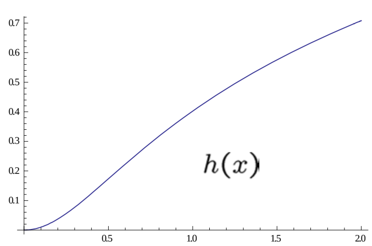


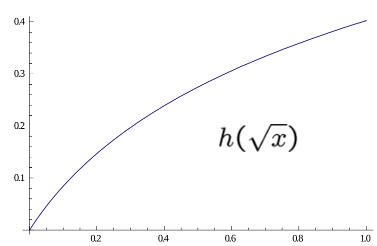


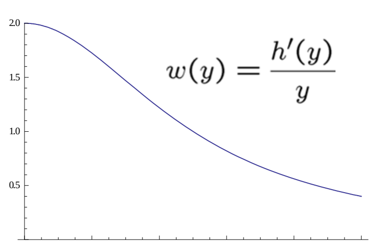


Advantage:
- Robust
- Cost function differentiable
- Weights defined at zero
- Convex
- Guaranteed to converge (at least to local minimum)

Cauchy



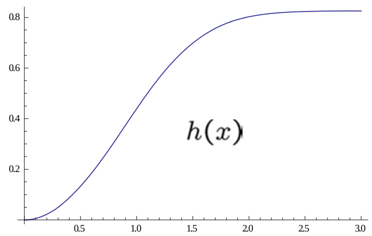


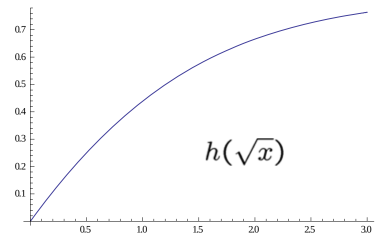


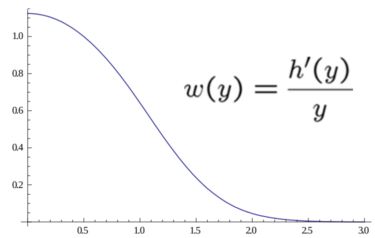
Advantage:
- Robust
- Cost function differentiable
- Weights defined at zero
- Guaranteed to converge (at least to local minimum)

Disadvantage:
- Non-convex
- Increased number of local minima

Blake Zisserman



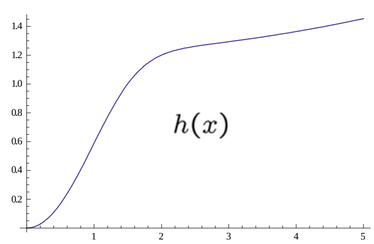


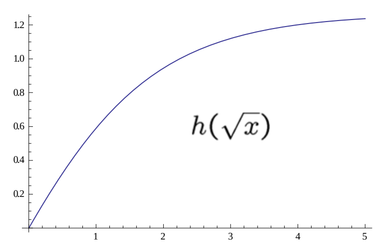


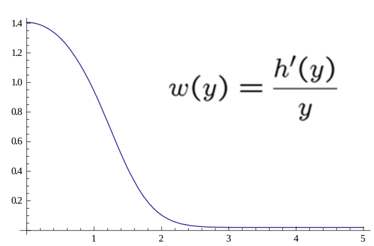
Advantage:
- Very robust to outliers
- Cost function differentiable
- Weights defined at zero
- Guaranteed to converge (at least to local minimum)

Disadvantage:
- Non-convex
- Increased number of local minima

Corrupted Gaussian







Advantage:
- Robust
- Cost function differentiable
- Weights defined at zero
- Guaranteed to converge (at least to local minimum)

Disadvantage:
- Non-convex
- Increased number of local minima

Problems for which IRLS works

1. Any problem that you can solve the least-squares solution for exactly.
2. Point averaging.
3. Alignment of point sets (Horn's absolute orientation problem)
4. Regression
5. Rotation averaging
6. Bundle adjustment (to local minimum)
7. ...

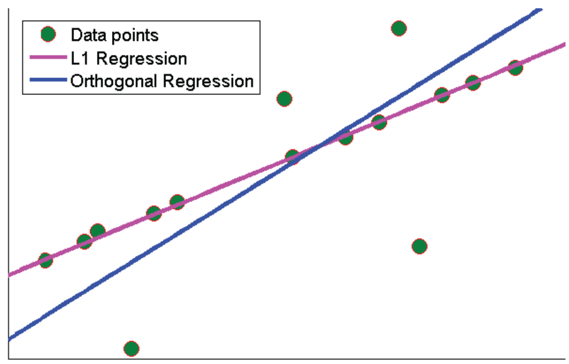
Example. L1 Regression

Results: L_1 Regression

NICTA

Line Fitting:

- L_1 regression is compared with L_2 regression.

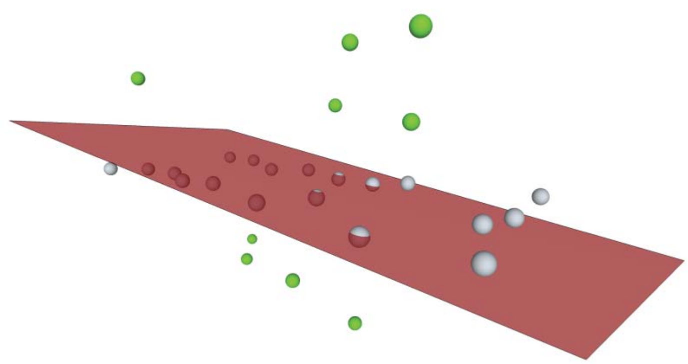


58/63

Example: Regression

NICTA

- Squared distance does not work in the presence of outliers.

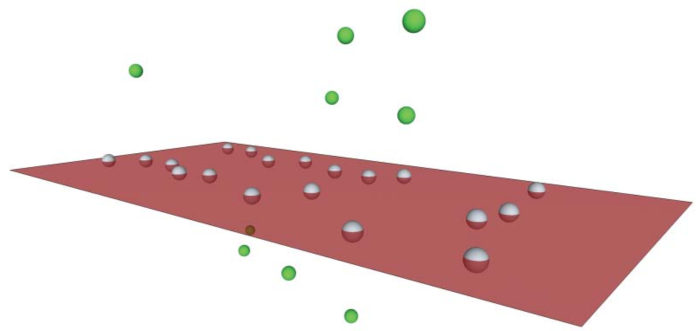


11/63

Example: Regression

NICTA

- The ideal model should be like this



12/63

Generalized Weiszfeld Algorithm

NICTA

Gradient Descent:

- Given a set of subspaces $\mathcal{S}_i$, the cost function to find L_1 closest point to subspaces is

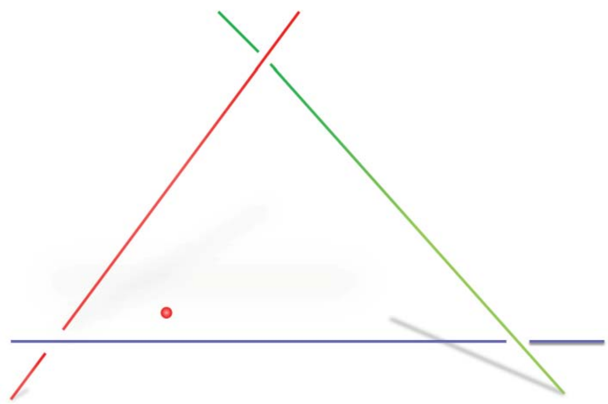
$$\widetilde{C}_2^t(\mathbf{X}) = \sum_{i=1}^{n} w_i^t \, \|\mathbf{X} - \mathcal{P}_{\mathcal{S}_i}(\mathbf{X}^t)\|^2 \ .$$

Then, let $\mathbf{X}^{t+1} = \mathrm{argmin}_{\mathbf{X}} \ C_2^t(\mathbf{X})$.

45/63

L_1 closest point to subspaces

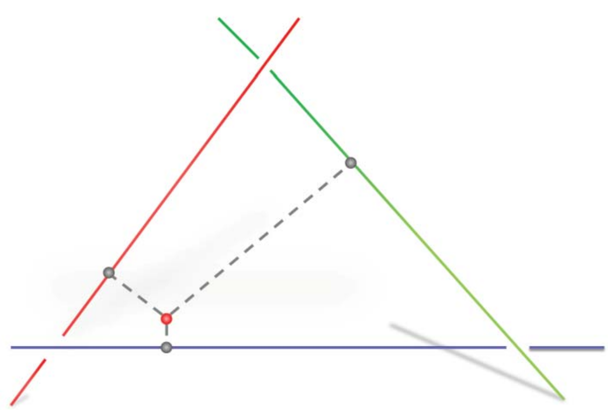
NICTA



47/63

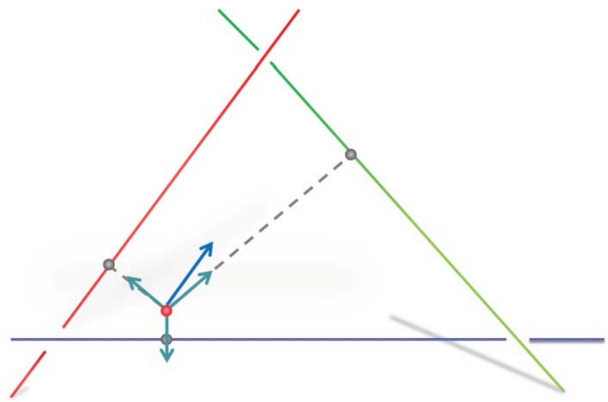
L_1 closest point to subspaces

NICTA

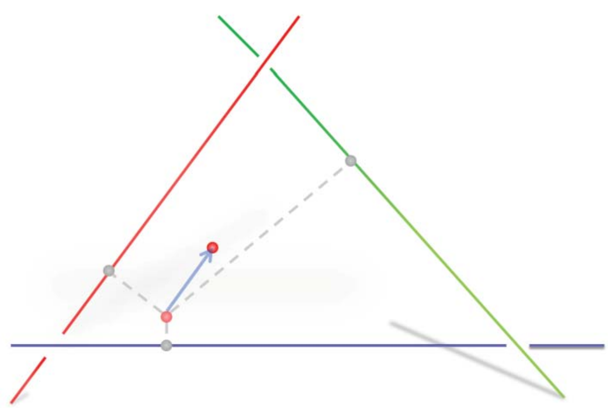


48/63

L_1 closest point to subspaces

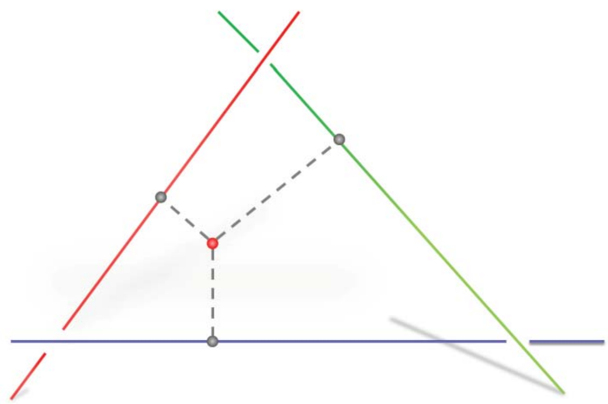


L_1 closest point to subspaces



L_1 closest point to subspaces

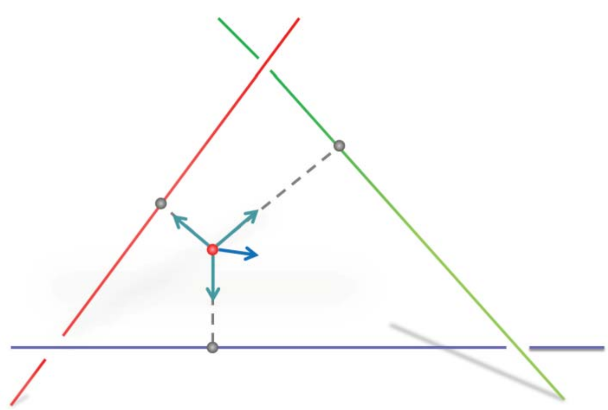
NICTA



51/63

L_1 closest point to subspaces

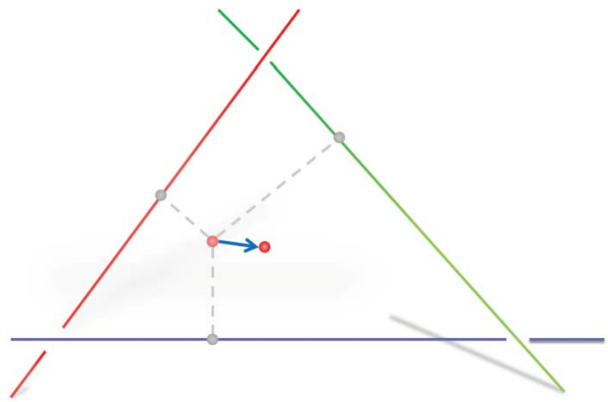
NICTA



52/63

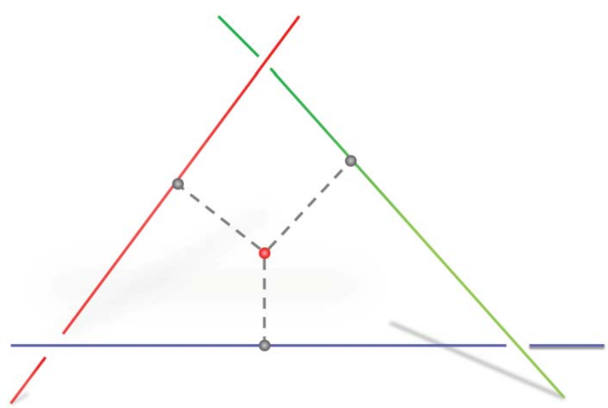
L_1 closest point to subspaces

NICTA



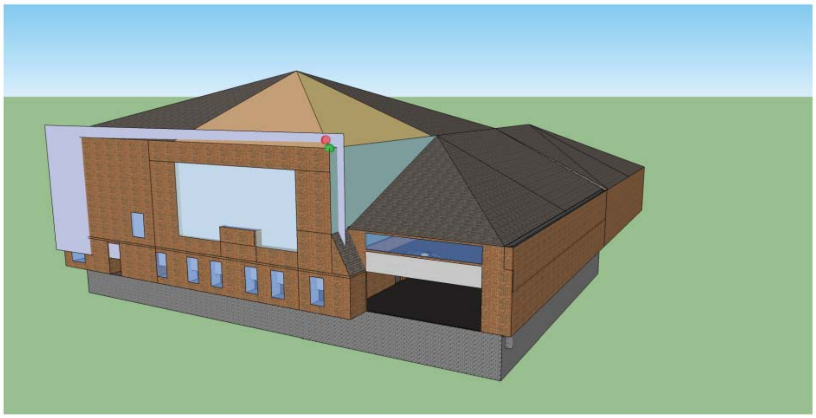
L_1 closest point to subspaces

NICTA

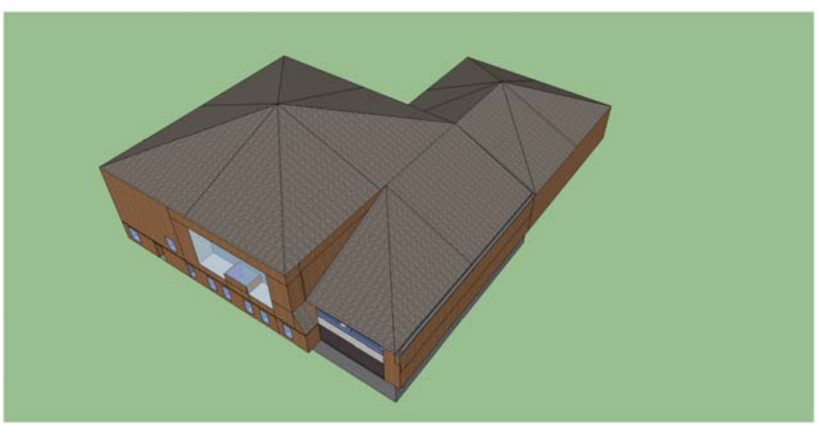


Possible application

L_1 optimal point of Intersection of Planes,

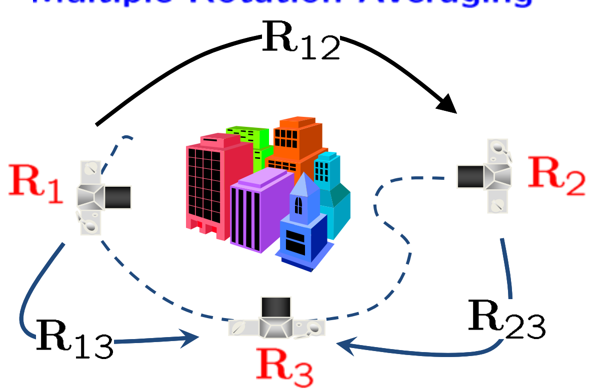


L_1 optimal point of Intersection of Planes in Aerial view,



Example. Averaging Rotations

Multiple Rotation Averaging



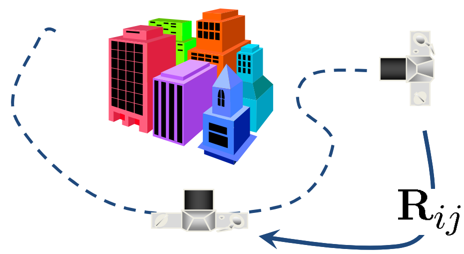
For consistency require

$$\mathbf{R}_{13} = \mathbf{R}_{23}\mathbf{R}_{12}.$$

Define absolute rotations $\mathbf{R}_i$ satisfying

$$\mathbf{R}_{ij} = \mathbf{R}_j\mathbf{R}_i^{-1}.$$

Single Rotation Averaging for Relative Orientation of Cameras



- Five corresponding points between the two images allow a computation of relative rotation (and translation).
- Very fast (about 35 μs).
- Take many different sets of 5 points and average the computed rotations.
- Individual estimates can be noisy, so we need robust method of rotation averaging.

Single Rotation Averaging

NICTA

Given rotations $\mathtt{R}_i \in \mathrm{SO}(3)$, the L_p mean is equal to

$$\mathtt{S}^* = \underset{\mathtt{S}\in\mathrm{SO}(3)}{\text{argmin}} \sum_{i=1}^{n} d(\mathtt{R}_i, \mathtt{S})^p \ .$$

- $p = 2$: Least-squares L_2 averaging. Usually simpler, not robust to outliers.
- $p = 1$: L_1 averaging. More robust to outliers.

32/63

So how do we average rotations?

Average of rotations is the rotation that minimizes

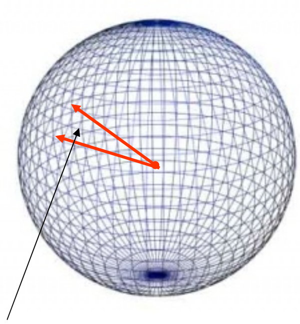
$$C(\mathbf{S}) = \sum_i d(\mathbf{R}_i, \mathbf{S})^p$$

- $p = 2$ – Least-squares (L_2) averaging. Usually simpler, not robust to outliers.
- $p = 1$ – L_1 averaging. More robust to outliers.

What is meant by $d(\mathbf{R}_i, \mathbf{S})$?

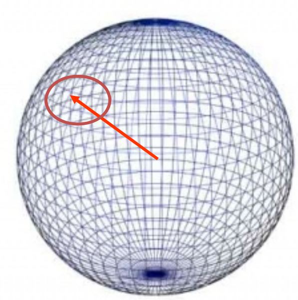
1. Angular distance $d_{\text{ang}}(\mathbf{R}, \mathbf{S})$
2. Quaternion distance $\min(\|\mathbf{r} - \mathbf{s}\|, \|\mathbf{r} + \mathbf{s}\|)$
3. Chordal distance $\|\mathbf{R} - \mathbf{S}\|_F$.

Isometry of Rotations and Quaternions

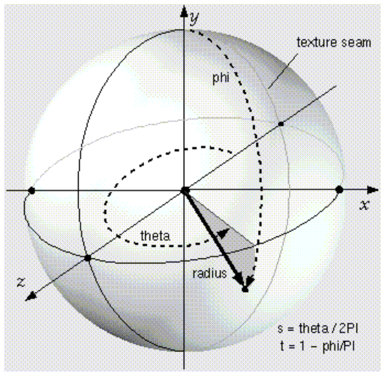


Angle between two quaternions is half the angle between the corresponding rotations, defined by

$$\text{angle}(\mathbf{r}_1, \mathbf{r}_2) = \text{angle}(R_1 R_2^{-1})/2$$

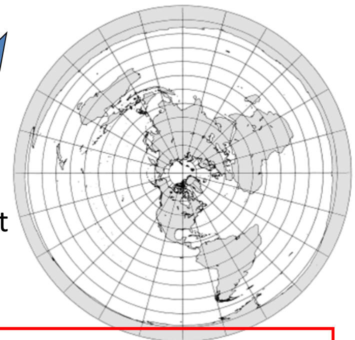


All rotations within a delta-neighbourhood of a reference rotation form a circle on the quaternion sphere.



Angle-axis representation of Rotations

Flatten out the meridians (longitude lines)

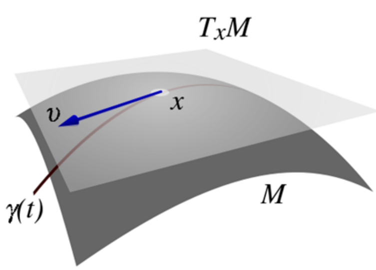


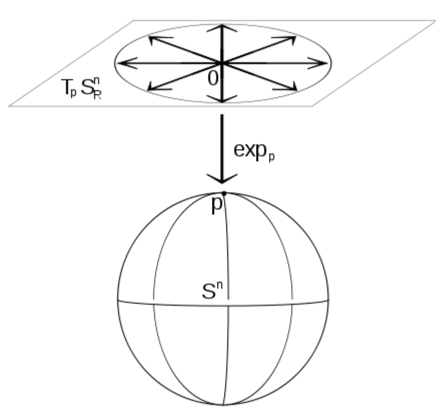
Azimuthal Equidistant Projection

Rotations are represented by a ball of radius pi in 3-Dimensional space.

IRLS Algorithm on a Manifold

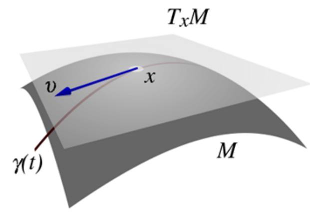
- Map back and forth from the manifold to the tangent space using the exponential and logarithm maps.



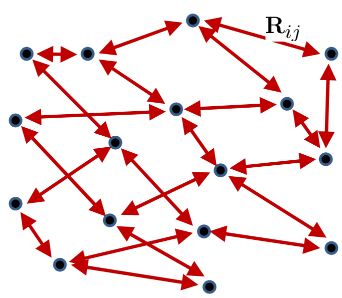


Steps of the Weiszfeld Algorithm on $SO3$

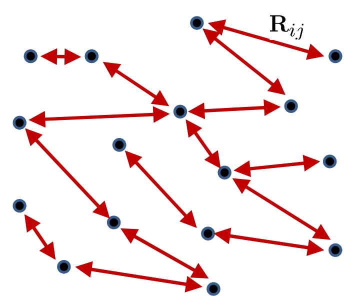
1. Find an initial estimate $\mathbf{S}^0$ for the median.
2. At any time $t = 0, 1, \ldots$ apply the logarithm map centred at $\mathbf{S}^t$ to compute
$$\mathbf{v}_i = \log_{\mathbf{S}^t}(\mathbf{R}_i).$$
3. (Weiszfeld step): Set
$$\delta = \frac{\sum_{i=1}^n \mathbf{v}_i/\|\mathbf{v}_i\|}{\sum_{i=1}^n 1/\|\mathbf{v}_i\|}$$
4. Set
$$\mathbf{S}^{t+1} = \exp(\delta)\mathbf{S}^t.$$
5. Repeat steps 1 to 3 until convergence.



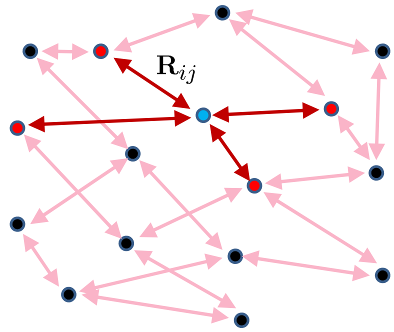
Averaging over a graph



Relative rotations are computed between some nodes in the graph.



Initialization: Propagate rotations estimates across a tree.



Orientation of each node in turn is recomputed, given the known orientation of its neigbours, and the relative orientation.

This involves “single-rotation averaging”.

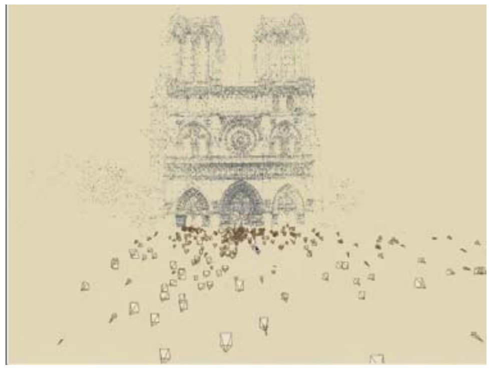
Experimental Setup

NICTA

Nore Dame Set:

- No. of Images = 595.
- No. of $3D$ points = 280, 000.
- No. of images pairs with >= 30 matched points = 42000.

39/63



- o 569 Images
- o 280,000 points
- o 42000 pairs of overlapping images (more than 30 points in common)

Task: Find the orientations of all cameras.

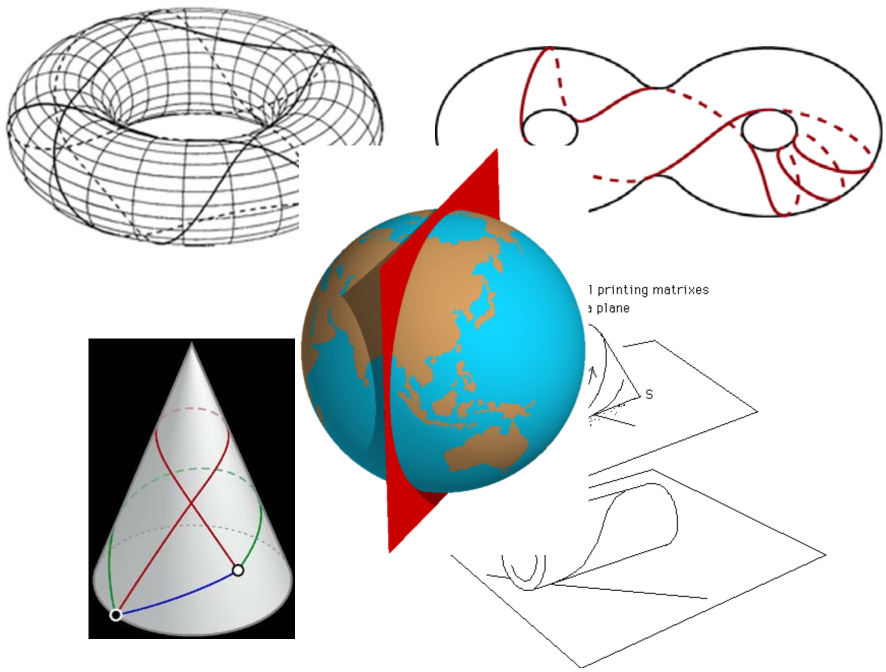
Extension: Optimization on Riemannian Manifolds

What is a manifold, anyways?

- Think of a manifold as a smooth surface in R^n
- Every point on the manifold (surface) has a neighbourhood that is the same as (homeomorphic to) a ball in $\mathbb{R}^n$.

Examples of manifolds

- $\mathbb{R}^n$
- Sphere S^n
- Rotation space $SO(3)$ – used in rotation averaging
- Positive definite matrices – "covariance features"
- Grassman Manifolds – used to model sets of images
- Essential manifold – structure and motion
- Shape manifolds – capture the shape of an object
- Essential manifold, trifocal manifold

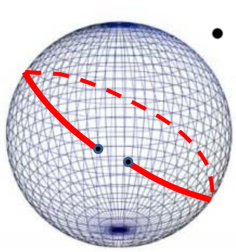


What is a geodesic?

A curve is a mapping γ from an interval $[a, b]$ to the manifold M.

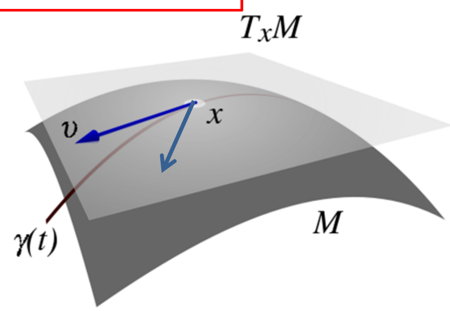
A geodesic has several descriptions:

- A locally distance-minimizing curve. The curve can be broken up into sections $[a_i, b_i]$ so that γ is the shortest curve from $\gamma(a_i)$ to $\gamma(b_i)$.
- A curve on a surface whose acceleration is always normal to the surface.
- A taut piece of elastic band on the surface.



Riemannian Manifold

- A manifold with an inner product defined in the tangent space at each point.
- Allows us to measure angles at a point
- Define the length of curves.
- Define "geodesic distance" on the manifold
- Find curves of shortest distance.
- Define "geodesics" or locally shortest curves



Geodesics and the exponential map

- Exponential map wraps vector in tangent space onto the manifold.
- Constant velocity.
- Acceleration always normal to the surface.

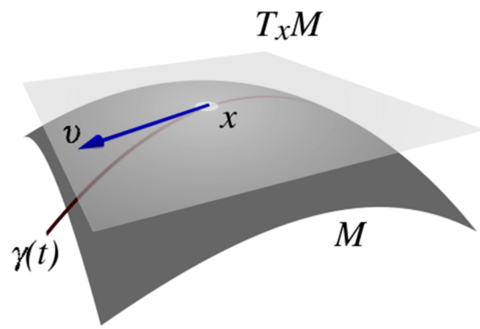
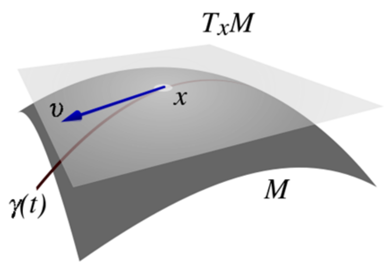
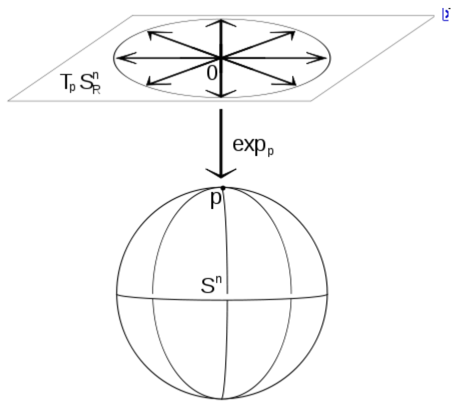


Image from
http://en.wikipedia.org/wiki/File:Tangentialvektor.svg

IRLS Algorithm on a Manifold

- Map back and forth from the manifold to the tangent space using the exponential and logarithm maps.



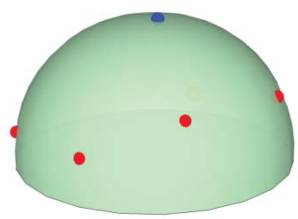


Wesiszfeld Algorithm on Manifold

Wesiszfeld algorithm on Manifold.

NICTA

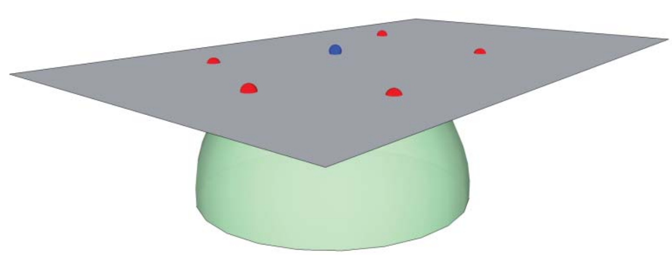
- Start with a random point on manifold



Wesiszfeld algorithm on Manifold.

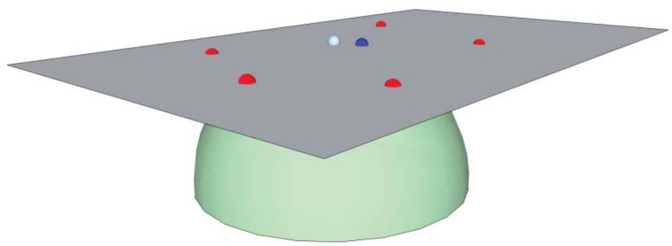
NICTA

- Project the points to the tangent space



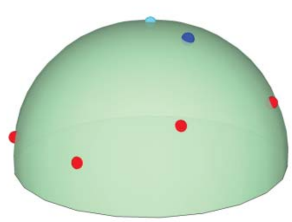
Wesiszfeld algorithm on Manifold.

- Apply the Weiszfeld algorithm



Wesiszfeld algorithm on Manifold.

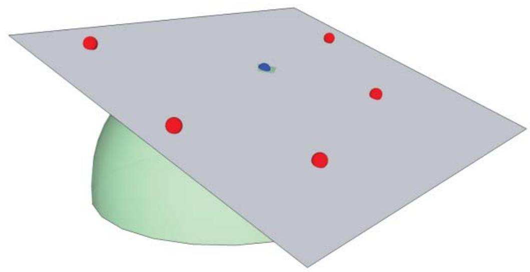
- Project the updated point to the manifold



Wesiszfeld algorithm on Manifold.

NICTA

- Again, project all the points on manifold to the tangent space and repeat the same procedure until convergence



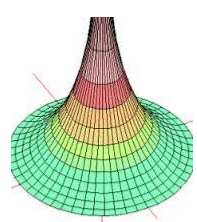
29/63

Convergence of IRLS on Manifolds

Weiszfeld algorithm will converge on a manifold of non-negative curvature (Fletcher 2009, Aftab 2011, 2014)

Why positive curvature? Toponogov's Theorem.

- With non-negative (sectional) curvature, geodesics converge.
- Distance in the tangent space is always greater than distance on the manifold.
- If iteration causes distances to decrease in the tangent space, they decrease even more in the manifold.



The top half of a pseudo-sphere

Hyperbolic (negative curvature) manifold