Computational methods in applied inverse problems

Uri Ascher

Department of Computer Science
University of British Columbia
October 2017

Four lectures

- Calibration and simulation of deformable objects
- Data manipulation and completion
- Estimating the trace of a large implicit matrix and applications
- Numerical analysis in visual computing: not too little, not too much

Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions

Here is the T-shirt

Data completion and manipulation

- The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.
- And yet it seems to be everywhere in practice!
    1. "Completing scarce data" by some interpolation/extrapolation or other approximation
    2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
    3. "Completing data" to obtain a more efficient algorithm
    4. "Completing data" to obtain a "more solid theory"
    5. Manipulating data because we don't know how to solve the problem otherwise.
- When is it OK to do this?!
- Attempt to get more insight by considering case studies.

Data completion and manipulation

- The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.
- And yet it seems to be everywhere in practice!
  1. "Completing scarce data" by some interpolation/extrapolation or other approximation
  2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
  3. "Completing data" to obtain a more efficient algorithm
  4. "Completing data" to obtain a "more solid theory"
  5. Manipulating data because we don't know how to solve the problem otherwise.
- When is it OK to do this?!
- Attempt to get more insight by considering case studies.

Data completion and manipulation

- The practice of manipulating given observed data for solving inverse problems is known to have its perils: loss of statistical relevance, danger of calibrating a model to handle our own generated errors, etc.
- And yet it seems to be everywhere in practice!
  1. "Completing scarce data" by some interpolation/extrapolation or other approximation
  2. Preferring to see data given at regular mesh nodes, or otherwise having a hidden uncertainty in the location of data values
  3. "Completing data" to obtain a more efficient algorithm
  4. "Completing data" to obtain a "more solid theory"
  5. Manipulating data because we don't know how to solve the problem otherwise.
- When is it OK to do this?!
- Attempt to get more insight by considering case studies.

Our case studies

1. Inverting Maxwell's equations and DC resistivity in exploration geophysics
   [Haber, A. & Oldenburg, 2004]
2. Recovering local volatility surface in financial mathematics
   [Albani, A., Yang & Zubelli, 2017; A., A. & Z., 2017]
3. Denoising of surface triangle mesh
   [Huang & A., 2008]
4. Calibrating and simulating soft bodies in computer graphics
   [Wang, Wu, Yin, A., Liu & Huang, 2015]
5. Obtaining union of observation locations for many data sets
   [Roosta-Khorasani, van den Doel & A., 2014]

Inverse problem setting

- Given observed data $\mathbf{d} \in \mathbb{R}^l$ and a forward operator $\mathbf{f}(m)$ which provides predicted data for each instance of distributed parameter function m, find m (discretized and reshaped into $\mathbf{m}$) such that the predicted and observed data agree to within noise:

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) + \boldsymbol{\eta}.$$

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., $\mathbf{f}(\mathbf{m}) = P\mathbf{u} = PG(\mathbf{m})\mathbf{q}$, where G is a discrete Green's function.
- Iterative algorithm on $\mathbf{m}$ to reduce objective. Assuming $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$, the maximum likelihood (ML) data misfit function is

$$\phi(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2$$

The discrepancy principle yields the stopping criterion

$$\phi(\mathbf{m}) \leq \rho, \quad \text{where} \quad \rho = \sigma^2 l$$

Inverse problem setting

- Given observed data $\mathbf{d} \in \mathbb{R}^l$ and a forward operator $\mathbf{f}(m)$ which provides predicted data for each instance of distributed parameter function m, find m (discretized and reshaped into $\mathbf{m}$) such that the predicted and observed data agree to within noise:

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) + \boldsymbol{\eta}.$$

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., $\mathbf{f}(\mathbf{m}) = P\mathbf{u} = PG(\mathbf{m})\mathbf{q}$, where G is a discrete Green's function.
- Iterative algorithm on $\mathbf{m}$ to reduce objective. Assuming $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$, the maximum likelihood (ML) data misfit function is

$$\phi(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2$$

The discrepancy principle yields the stopping criterion

$$\phi(\mathbf{m}) \le \rho, \quad \text{where} \quad \rho = \sigma^2 l$$

Inverse problem setting

- Given observed data $\mathbf{d} \in \mathbb{R}^l$ and a forward operator $\mathbf{f}(m)$ which provides predicted data for each instance of distributed parameter function m, find m (discretized and reshaped into $\mathbf{m}$) such that the predicted and observed data agree to within noise:

$$\mathbf{d} = \mathbf{f}(\mathbf{m}) + \boldsymbol{\eta}.$$

- Consider a case where a PDE must be solved to evaluate the forward operator, i.e., $\mathbf{f}(\mathbf{m}) = P\mathbf{u} = PG(\mathbf{m})\mathbf{q}$, where G is a discrete Green's function.
- Iterative algorithm on $\mathbf{m}$ to reduce objective. Assuming $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$, the maximum likelihood (ML) data misfit function is

$$\phi(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2$$

The discrepancy principle yields the stopping criterion

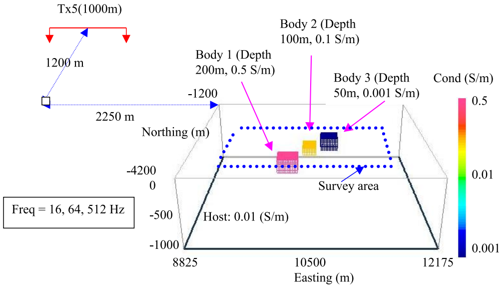
$$\phi(\mathbf{m}) \leq \rho, \quad \text{where} \quad \rho = \sigma^2 l$$

Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions

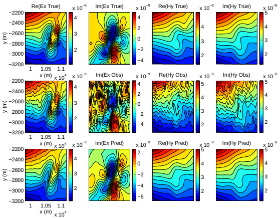
Example (CS1): scarce data in electromagnetic data inversion

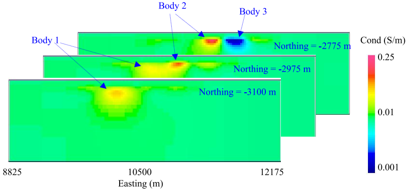
G is Green's function for Maxwell's equations in time or frequency domain, **m** is conductivity or resistivity. [Haber, A. & Oldenburg, 2004]



EM DATA INVERSION IN GEOPHYSICS

Use Tikhonov-type regularization: a prior penalizing lack of smoothness in surface function m through gradient.





Top: misfit. Bottom: recovered **m**.

Dupire's equation (CS2)

[Dupire, 1994]: replace the Black-Scholes equation for option price by a parabolic PDE of the form

$$\frac{\partial C}{\partial \tau} = \frac{1}{2}\sigma^2(\tau, K)K^2\frac{\partial^2 C}{\partial K^2} - bK\frac{\partial C}{\partial K}, \quad \tau > 0, K \geq 0,$$

s.t. initial and boundary conditions (for calls)

$$C(\tau = 0, K) = (S_0 - K)^+,$$

$$\lim_{K\to\infty} C(\tau, K) = 0, \qquad \lim_{K\to 0} C(\tau, K) = S_0.$$

Here τ is time to maturity, K is strike price, $C = C(\tau, K)$ is value of the European call option with expiration date $T = \tau$, and $\sigma(\tau, K)$ is volatility. Can write all this in operator form as

$$\tilde{L}(\sigma)C = \tilde{q}(S_0),$$

with $\tilde{L}$ a linear differential operator for a given σ.

Assume first that the stock price S_0 is a given parameter.

Calibrating the model: solve inverse problem for $\sigma(\tau, K)$ given C-data.

Changing variables: log moneyness

To simplify and make problem dimensionless, change K to $y = \log(K/S_0)$ (so $-\infty < y < \infty$), then $u(\tau, y) = C(\tau, S_0 \exp(y))/S_0$ and $m(\tau, y) = \frac{1}{2}\sigma(\tau, K(y))^2$. Obtain

$$-\frac{\partial u}{\partial \tau} + m\left(\frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial y}\right) + b\frac{\partial u}{\partial y} = 0, \quad \tau > 0, y \in \Re,$$

s.t. side conditions

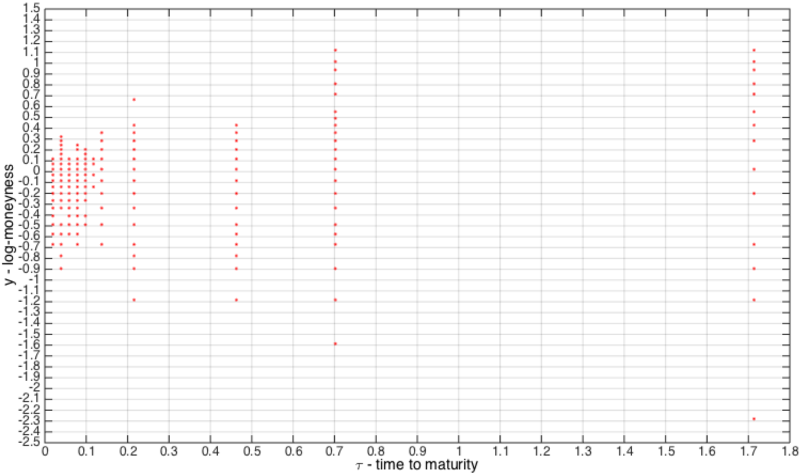
$$u(\tau = 0, y) = (1 - \exp(y))^+,$$
$$\lim_{y\to\infty} u(\tau, y) = 0, \qquad \lim_{y\to-\infty} u(\tau, y) = 1.$$

Can write this as $L(m)u = q$.
Discretize over a mesh with step sizes $\Delta\tau$ and Δy; denote the corresponding discretization as

$$L_h(m)u_h = q_h.$$

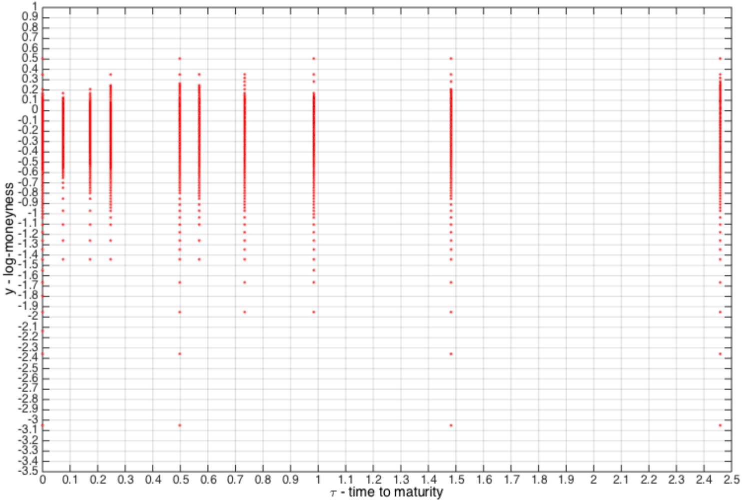
Example: locations of (real) data

Locations of u-data values for the PBR data set.



Example: locations of (real) data

Locations of u-data values for the SPX data set.



Interpolating/extrapolating the data

- Several researchers have applied interpolation/extrapolation to this type of data, followed by assimilation of the resulting data set with the discretized Dupire PDE problem.
- Use [Kahale, 2005] for this purpose. This algorithm applies data completion with a "financial prior", insisting that the resulting data surface reproduce the "smile" effect.
- An obvious objection, however, is that the resulting data surface does not satisfy the discretized differential problem for any $m(\tau, y)$, and vice versa. The assimilation of these two pieces of information may be more difficult.
- Compare this to not modifying the given data, using for both cases a Tikhonov-type regularization as well as EnKF.

Interpolating/extrapolating the data

- Several researchers have applied interpolation/extrapolation to this type of data, followed by assimilation of the resulting data set with the discretized Dupire PDE problem.
- Use [Kahale, 2005] for this purpose. This algorithm applies data completion with a "financial prior", insisting that the resulting data surface reproduce the "smile" effect.
- An obvious objection, however, is that the resulting data surface does not satisfy the discretized differential problem for any $m(\tau, y)$, and vice versa. The assimilation of these two pieces of information may be more difficult.
- Compare this to not modifying the given data, using for both cases a Tikhonov-type regularization as well as EnKF.

Regularized inverse problem

- Maximum likelihood for simplest case of white noise:

$$\phi(m_h, u_h) = \|Pu_h - \mathbf{d}\|^2 = \|PL_h(m_h)^{-1}q_h - \mathbf{d}\|^2$$

where matrix P projects to data locations: P has many more columns than rows for original data, whereas $P = I$ for completed data.

- Regularize the problem: minimize the maximum a posteriori (MAP) *merit function*

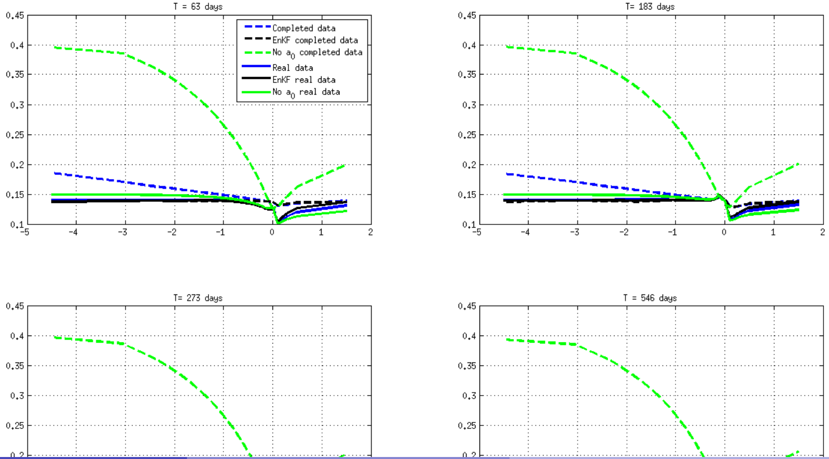
$$\phi_R(m_h, u_h) = \phi(m_h, u_h) + R(m_h).$$

- Our Tikhonov-like regularization operator is (a_0 a known constant)

$$\begin{aligned} R(m_h) &= \alpha_1 \sum_i \sum_j (m_{i,j} - a_0)^2 \\ &+ \frac{\alpha_2}{\Delta\tau^2} \sum_j \sum_{i=1}^{M_\tau} (m_{i,j} - m_{i-1,j})^2 + \frac{\alpha_3}{\Delta y^2} \sum_i \sum_{j=1}^{M_y} (m_{i,j} - m_{i,j-1})^2 \end{aligned}$$

Results for SPX data

Set $\alpha_1 > 0$ and compare working with the given sparse data vs using data completion by the Kahale algorithm.



Conclusion for case study CS2

- These results clearly show that the data completion approach has not delivered.
- Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and *different α-weights* in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.
- Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using a_0 and first derivatives.
- After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.

Conclusion for case study CS2

- These results clearly show that the data completion approach has not delivered.
- Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and *different* α*-weights* in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.
- Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using a_0 and first derivatives.
- After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.

Conclusion for case study CS2

- These results clearly show that the data completion approach has not delivered.
- Additional tests for Henry Hub and WTI prices, using bilinear interpolation for the data completion and *different* α*-weights* in the Tikhonov-type priors, also clearly indicate that it is better to avoid the extensive data completion required here: the market implied smile, which has an important relationship with market risk, is better fitted upon using just the original data.
- Both EnKF algorithms we tried [Iglesias, Law & Stuart, 2013; Calvetti, Ernst & Somersalo 2014] were trivially (and significantly) improved by adding additional regularization using a_0 and first derivatives.
- After this improvement the EnKF algorithms were comparable to but not better than the Tikhonov-type regularization. Big plus: no ad hoc parameter search was required.

Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions

Examples

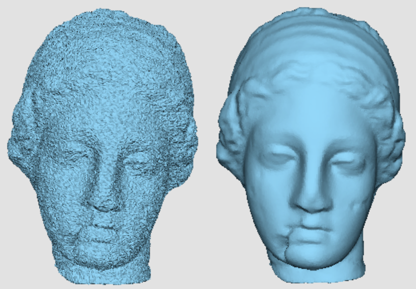
The notion that a (potentially noisy) data value d_i is given at a known, deterministic location $\mathbf{x}_i$ is often violated in practice. Here are some examples:

- Surface triangle mesh denoising
  [Huang & A., 2008]
- Minimal prospectivity mapping
  [Granek & Haber, 2014]
  (not considered further)
- Local volatility surface with uncertainty in price S_0
  [Albani, A., Yang & Zubelli, 2015]

Typical image denoising (CS3)

Left: noisy image: noisy data at precisely prescribed pixel locations.
Right: exact (ideally denoised?) image.

Surface triangle mesh denoising

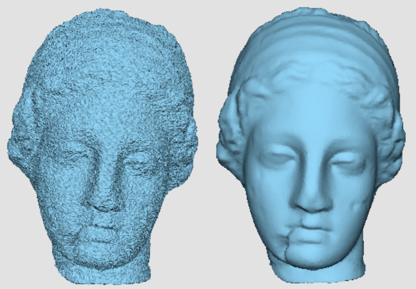


Left: noisy triangle mesh: the data are nodal values (x_i, y_i, z_i). No distinction between data value and location! Uncertainty in higher dimension.

Right: Our denoised triangle mesh.

[We had set out to generalize multiscale techniques for image denoising and ended up devising a completely different multiscale method for the surface mesh.]

Surface triangle mesh denoising

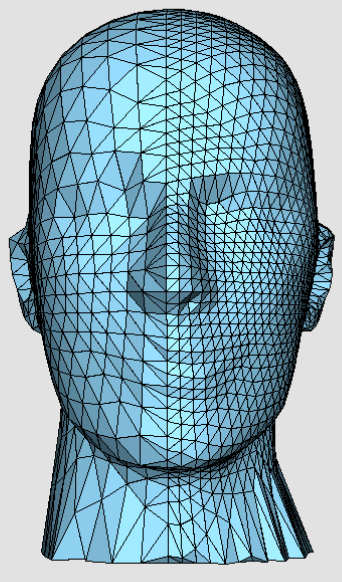


Left: noisy triangle mesh: the data are nodal values (x_i, y_i, z_i). No distinction between data value and location! Uncertainty in higher dimension.

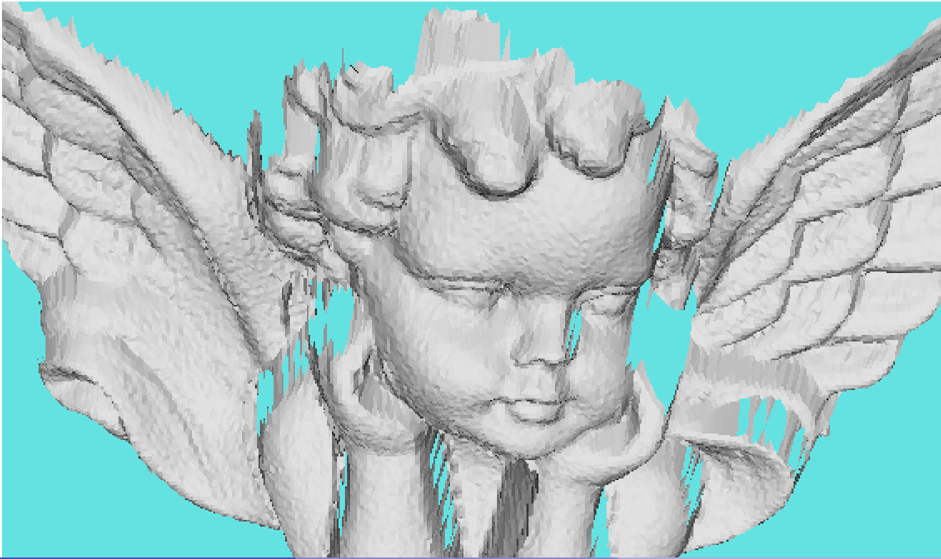
Right: Our denoised triangle mesh.

[We had set out to generalize multiscale techniques for image denoising and ended up devising a completely different multiscale method for the surface mesh.]

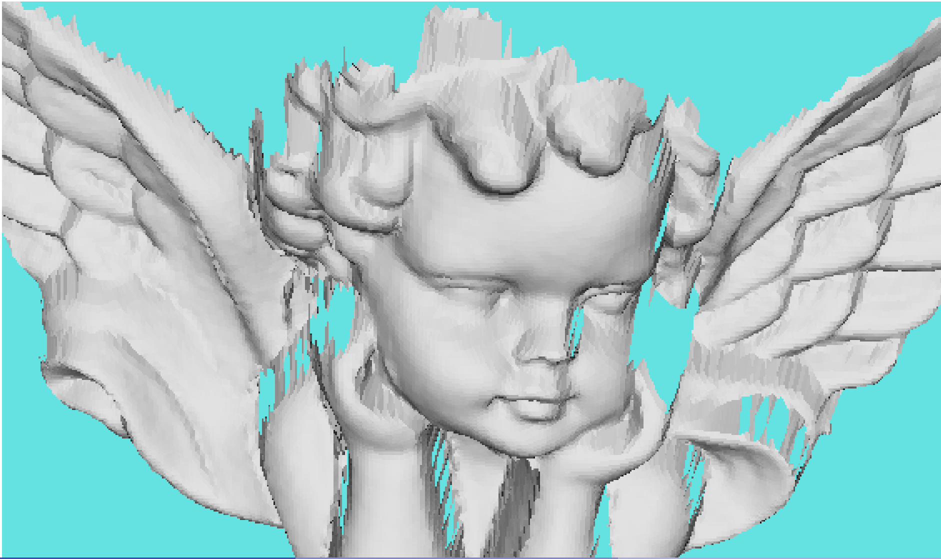
3D surface triangle mesh



Scanned noisy model (25K verts)



Smoothed by MSAL



Uncertainty in S_0 (CS2)

- Recall Dupire's equation

$$\frac{\partial C}{\partial \tau} = \frac{1}{2}\sigma^2(\tau, K)K^2\frac{\partial^2 C}{\partial K^2} - bK\frac{\partial C}{\partial K}, \quad \tau > 0, K \geq 0,$$

$$C(\tau = 0, K) = (S_0 - K)^+,$$

$$\lim_{K\to\infty} C(\tau, K) = 0, \qquad \lim_{K\to 0} C(\tau, K) = S_0.$$

  But now, stock price S_0 has (well-quantified) uncertainty, as it is typically some average of daily prices.

- $\Rightarrow$ Add a term $\alpha_5(S_0 - \hat{S}_0)^2$ to regularization prior $R = R(m_h, S_0)$, where S_0 is now random variable with measured mean (say) $\hat{S}_0$.

Uncertainty in S_0 cont.

- Furthermore, recall that upon changing variables to log moneyness $y = \log(K/S_0)$ (also $u(\tau, y) = C(\tau, S_0 \exp(y))/S_0$, $m(\tau, y) = \frac{1}{2}\sigma(\tau, K(y))^2$), obtain the nicer PDE

$$-\frac{\partial u}{\partial \tau} + m\left(\frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial y}\right) + b\frac{\partial u}{\partial y} \;=\; 0, \quad \tau > 0, y \in \Re.$$

  Now the uncertainty in S_0 has moved into the independent variable y!
- $\Rightarrow$ In addition to adding a term $\alpha_5(S_0 - \hat{S}_0)^2$ to $R = R(m_h, S_0)$, update also

$$\begin{aligned}\phi(m_h, S_0) \;&=\; \alpha_0 \sum_{i=1}^{l} \left((P(S_0)L_h(m_h)^{-1}q_h(S_0))_i - d_i\right)^2 \\ &+\; \alpha_4 \sum_{j=1}^{M_y} \left((1-\exp(y_j(S_0)))^+ - (1-\exp(y_j(\hat{S}_0)))^+\right)^2.\end{aligned}$$

(In practice set $\alpha_5 = 0$.)

Uncertainty in S_0 cont.

- Splitting method: Alternately freeze S_0 and m_h while solving for the other one.
- This converges fast because of the weak coupling, even when using the variable y.
- In preliminary experiments we see roughly a $1-2\%$ change in adjusted price.

Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners?
- Randomized algorithms for many data sets
- Conclusions

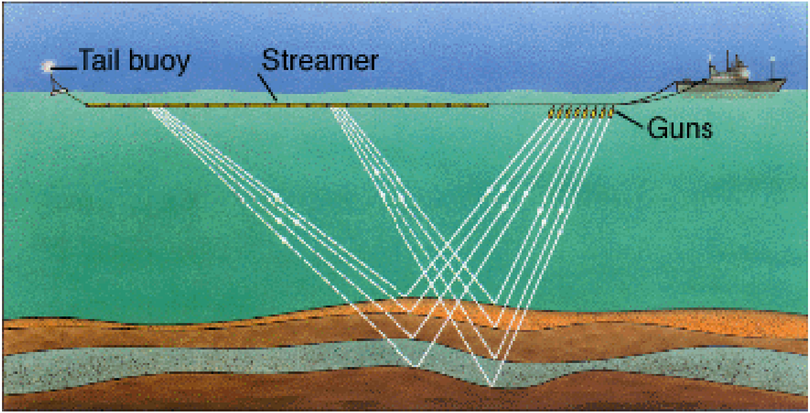
Examples

In some cases, it has been claimed that data completion/manipulation allows obtaining better results. Here are some examples:

- Matrix and tensor completion for seismic applications
  [da Silva & Herrmann, 2015; da Silva PhD thesis, 2017]
- Motion calibration and simulation of a soft body
  [Wang, Wu, Yin, A., Liu & Huang, 2015]

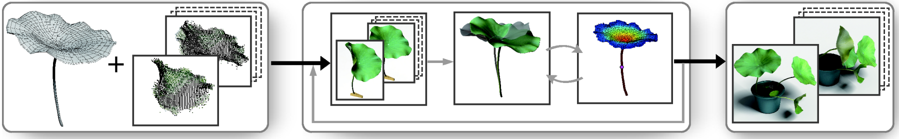
Example: full waveform inversion

Herrmann: use data completion of velocity field in order to solve this problem.



Motion capture and calibration (CS4)

- Physically-based deformation modelling in Computer Graphics
- Want to simulate and animate motion of a soft body, such as a plant under wind or water pressure, cloth, steak, face, etc.
- Can model by elastodynamics and porous media, but need to calibrate the model.
- Do that calibration by fitting example data obtained by sensor hardware: motion capturing and tracking.

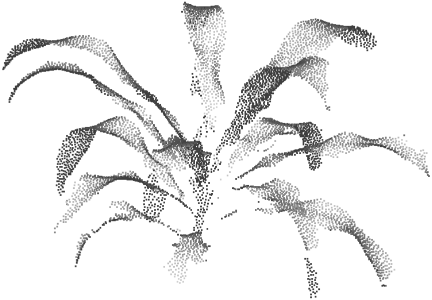


(a) Static shape (b) Point cloud sequence (c) Tracking (d) Reference shape optimization (e) Material optimization (f) Simulation

[Wang, Wu, Yin, A., Liu & Huang; siggraph '15]

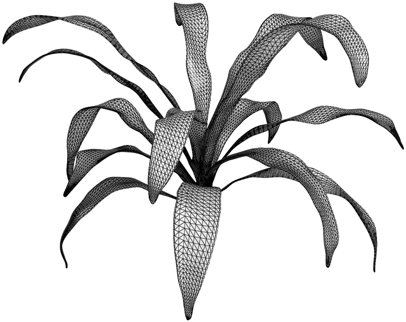
CAPTURING DATA

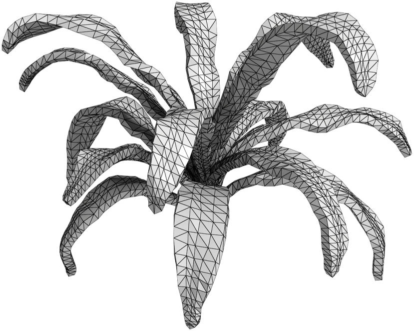
Left: three Kinect sensors are placed around the object;
Right: the deformation point cloud sequence is captured at 30*Hz*.



Capturing data cont.

Left: a (high resolution) surface mesh $\mathbb{S}$ with 15,368 vertices is used as a template to track captured point clouds;
Right: its (low resolution) corresponding volumetric mesh $\mathbb{T}$ with 9,594 nodes is used for spatial co-rotated linear FEM simulations.





Elastic deformation

- Denote reference shape by $\mathbf{X}$ and dynamic, deformed positions at a time instant t by $\mathbf{x} = \mathbf{x}(t)$.
- Element-wise stress-strain relationship using Hooke's law and Cauchy's linear strain tensor is

$$\sigma = \mathbf{E}\epsilon = \mathbf{E}\mathbf{B_e}(\mathbf{x_e} - \mathbf{X_e}),$$

where the 6×12 matrix $\mathbf{B_e} = \mathbf{B_e}(\mathbf{X_e})$ depends on $\mathbf{X_e}$ nonlinearly.

- For isotropic materials the 6×6 matrix $\mathbf{E}$ only depends on Young's modulus E and Poisson's ratio ν.
- Denoting the per-element rotation matrix obtained from polar decomposition by $\mathbf{R_e} = \mathbf{R_e}(\mathbf{x_e}(t), \mathbf{X_e})$, the element-wise elastic forces using the co-rotated linear approximation are

$$\begin{aligned} \mathbf{f_e}(E, \nu, \mathbf{X_e}, \mathbf{x_e}(t)) &= \mathbf{R_e}\mathbf{K_e}(\mathbf{R_e^T}\mathbf{x_e}(\mathbf{t}) - \mathbf{X_e}), \\ \mathbf{K_e} &= V_e\mathbf{B_e^T}\mathbf{E}\mathbf{B_e}, \end{aligned}$$

$\mathbf{K_e}$ is 12×12 element stiffness matrix and V_e is element volume.

Equations of motion

- Assemble force contributions from all FEM elements
- Summon Newton's 2nd law: at time t

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \hat{\mathbf{K}}\mathbf{x} = (\mathbf{R}\mathbf{K})\mathbf{X} + \mathbf{f}_{\mathrm{ext}}, \quad \hat{\mathbf{K}} = \mathbf{R}\mathbf{K}\mathbf{R}^{\mathbf{T}}.$$

- Stiffness matrix $\hat{\mathbf{K}}$ is sparse and is assembled from element contributions. Mass matrix $\mathbf{M}$ is lumped.
- Use Rayleigh damping: $\mathbf{D} = \alpha\mathbf{M} + \beta\hat{\mathbf{K}}$.
- Model calibration parameters in simplest case are

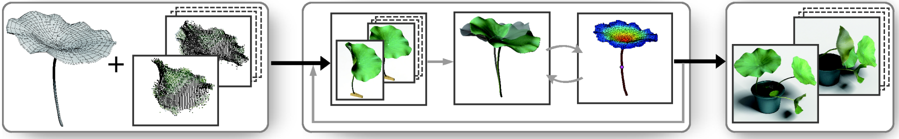
$$\mathbf{p} = (E, \nu, \alpha, \beta), \quad \text{so } \mathbf{m} = (\mathbf{p}, \mathbf{X}).$$

Often have more than one control point, for each of which there is a Young modulus.

Motion tracking and inverse problem

- Motion tracking: physically-based probabilistic tracking: for given $\mathbf{p}$ and $\mathbf{X}$, find captured trajectory $\mathbf{x} = \hat{\mathbf{x}}_t$.
- This tracking problem involves an inference (EM) algorithm.
- Inverse problem: deformation parameter estimation:

$$\min_{\mathbf{p},\mathbf{X}} \sum_t \|\mathbf{x}_t - \hat{\mathbf{x}}_\mathbf{t}\|^2.$$



(a) Static shape (b) Point cloud sequence (c) Tracking (d) Reference shape optimization (e) Material optimization (f) Simulation

Data manipulation to the rescue

- Employ a splitting method between **p** and **X**. Fortunately this works very well because of weak coupling between these unknown groups.
- Use Nelder-Mead for **p** as there are many local minima. However, this is good only for a few unknowns.
- Many more nontrivial details are described in the paper.
- Nice looking results are obtained! see the videos and a separate talk.
- However, the calibration (parameter estimation) part of the process uses manufactured data.
- **Attempts to estimate the parameters directly from the point cloud data have not worked out well**

Data manipulation to the rescue

- Employ a splitting method between **p** and **X**. Fortunately this works very well because of weak coupling between these unknown groups.
- Use Nelder-Mead for **p** as there are many local minima. However, this is good only for a few unknowns.
- Many more nontrivial details are described in the paper.
- Nice looking results are obtained! see the videos and a separate talk.
- However, the calibration (parameter estimation) part of the process uses manufactured data.
- **Attempts to estimate the parameters directly from the point cloud data have not worked out well**

Outline

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners
- Randomized algorithms for many data sets
- Conclusions

Inverse problem with s data sets (CS5)

After discretization and for our problems of interest:

$$\begin{aligned} \mathbf{d}_i &= \mathbf{f}_i(\mathbf{m}) + \boldsymbol{\eta}_i, \quad i = 1, 2, \ldots, s \\ \mathbf{f}_i(\mathbf{m}) &= P\mathbf{u}_i = PG(\mathbf{m})\mathbf{q}_i \end{aligned}$$

Calculating "$G(\mathbf{m})\mathbf{q}_i$" for each i is costly!

- $\mathbf{d}_i \in \mathbb{R}^l$ is the measurement obtained in the i^{th} experiment
- $\mathbf{f}_i$ is the known forward operator for the i^{th} experiment
- $\mathbf{m} \in \mathbb{R}^{l_m}$ is the sought-after model
- $\boldsymbol{\eta}_i$ is the noise incurred in the i^{th} experiment
- s is the total number of experiments
- $\mathbf{u}_i \in \mathbb{R}^{l_u}$ is the ith field
- $\mathbf{q}_i \in \mathbb{R}^{l_u}$ is the ith source
- G^{-1} is a square matrix discretizing the PDE with the BC
- $P = P_i$ is the projection matrix for the i^{th} experiment

Application: DC resistivity

- PDE with multiple sources

$$\nabla \cdot (\mu(\mathbf{x})\nabla u_i) = q_i, \quad i = 1, \ldots, s,$$
$$\frac{\partial u_i}{\partial \nu}|_{\partial\Omega} = 0.$$

- Conductivity $\mu(\mathbf{x})$ is expressed as a point-wise function of $m(\mathbf{x})$ (e.g., use tanh to incorporate known bounds on μ).
- The operator $G(\mathbf{m})$ is the inverse of the above PDE discretized on a staggered grid.
- Use different selections of sources q_i, yielding corresponding fields u_i.
- Data is measured only on part of the domain's boundary.
- Use any prior we may have for this very difficult problem!

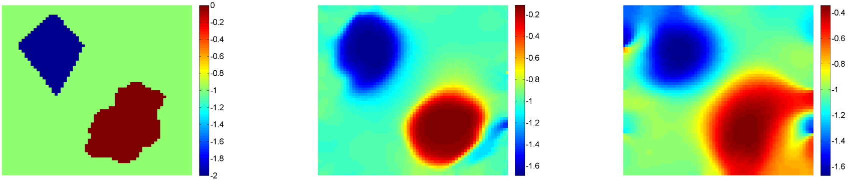
DC resistivity experiment setup

- Domain Ω is the unit square. Sources are of the form

$$q_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_1^i) - \delta(\mathbf{x} - \mathbf{x}_2^i)$$

  with $\mathbf{x}_1$ positive unit point source on west boundary, $\mathbf{x}_2$ negative unit point source on east boundary. Vary p boundary wall locations to get $s = p^2$ data sets.
- Receivers are all grid points on north and south walls. No sources or receivers at corners.
- Uniform 64×64 mesh
- For bounds set $\mu_{\max} = 1.2 \max \mu(\mathbf{x})$, $\mu_{\min} = 1.2^{-1} \min \mu(\mathbf{x})$
- PCG inner iteration limit $r = 20$; cgtol $= 1.e\text{-}3$.

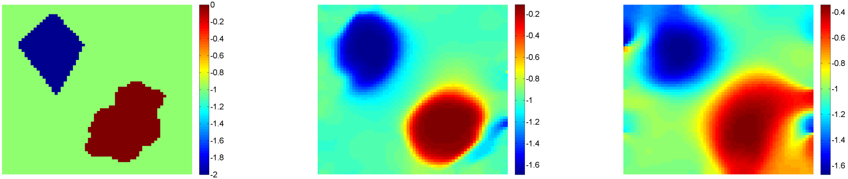
Example: $\mu_I = .1$, $\mu_{II} = 1$, $\mu_{III} = .01$, noise 2%



(a) True model (b) s=3,969 (c) s=49

- Thus, we want s larger for better reconstruction quality.
- But the cost of solving the problem grows very fast! (at least linearly with s). Need to find more efficient approximations for evaluating misfit function $\phi(\mathbf{m})$.

Example: $\mu_I = .1,\ \mu_{II} = 1,\ \mu_{III} = .01$, noise 2%



(d) True model (e) s=3,969 (f) s=49

- Thus, we want s larger for better reconstruction quality.
- But the cost of solving the problem grows very fast! (at least linearly with s). Need to find more efficient approximations for evaluating misfit function $\phi(\mathbf{m})$.

Monte Carlo to approximate the misfit trace

- Let $B(\mathbf{m}) = F(\mathbf{m}) - D \in \mathbb{R}^{l \times s}$. In kth iteration, $\mathbf{m} = \mathbf{m}_k$.
- Then $A = B^T B$ is implicit symmetric positive semi-definite (SPSD); effectively, can only carry out matrix-vector products $A * \mathbf{v}$ with this $s \times s$ matrix.

$$\phi(\mathbf{m}) = \|B(\mathbf{m})\|_F^2 = tr(B^T B) = \mathbb{E}(\mathbf{w}^T A \mathbf{w}).$$

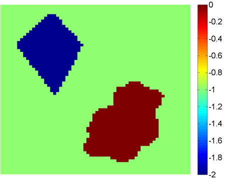
- Approximating expectation $\Leftrightarrow$ Approximating the trace $\phi(\mathbf{m}) = tr(A)$
  - Monte-Carlo approximation

$$tr(A) \approx \frac{1}{s_k} \sum_{j=1}^{s_k} \mathbf{w}_j^T A \mathbf{w}_j = \frac{1}{s_k} \sum_{j=1}^{s_k} \|B \mathbf{w}_j\|_2^2 .$$

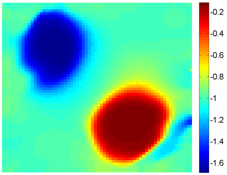
  - Note we can obtain exact trace using $s_k = s$ samples with $\mathbf{w}_j$ a scaled jth column of identity; but we want $s_k \ll s$.

Using stabilized Gauss-Newton with total variation (TV) added

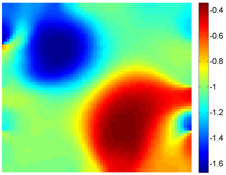
Method	Vanilla (3,969)	Gaussian (3,969)	Vanilla (49)
Work	476,280	4,618	5,978



(g) True model



(h) Gaussian, s=3,969

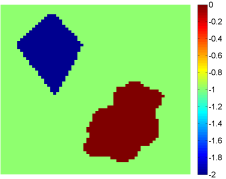


(i) Vanilla, s=49

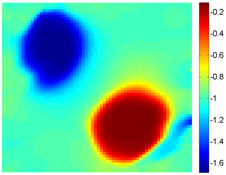
BUT what if $P = P_i$ varies with i, i.e., data for different experiments is not given at same locations?
Can no longer write $\sum_{i=1}^{s} w_i P G(\mathbf{m}) \mathbf{q}_i = \sum_{i=1}^{s} P G(\mathbf{m})(w_i \mathbf{q}_i)$, and the magic of the randomized algorithm is gone.

Using stabilized Gauss-Newton with total variation (TV) added

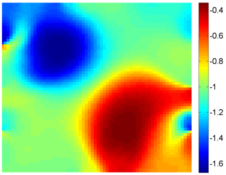
Method	Vanilla (3,969)	Gaussian (3,969)	Vanilla (49)
Work	476,280	4,618	5,978



(j) True model



(k) Gaussian, s=3,969



(l) Vanilla, s=49

BUT what if $P = P_i$ varies with i, i.e., data for different experiments is not given at same locations?
Can no longer write $\sum_{i=1}^{s} w_i P G(\mathbf{m}) \mathbf{q}_i = \sum_{i=1}^{s} P G(\mathbf{m})(w_i \mathbf{q}_i)$, and the magic of the randomized algorithm is gone.

Data approximation methods

Let receivers of ith data set be in $\Gamma_i \subset \partial\Omega, \; i = 1, 2, \ldots, s$. Want to extend data to the union $\Gamma = \bigcup \Gamma_i \subseteq \partial\Omega$.

1. DCT, wavelets, curvelets for each i. Advantage: leverage the recent advances in compressive sensing and sparse ℓ_1 methods.
2. Piecewise linear interpolation for each v_i. Advantage: very simple.
3. L2G: data completion function $v_i \in H^1(\Gamma)$ solves discretization of

$$\min_v \|v - \mathbf{d}_i\|^2_{L_2(\Gamma_i)} + \beta \, \|\nabla_S v\|^2_{L_2(\Gamma)} .$$

4. Data completion function $v_i \in H^2(\Gamma)$ solves discretization of

$$\min_v \|v - \mathbf{d}_i\|^2_{L_2(\Gamma_i)} + \beta \, \|\Delta_S v\|^2_{L_2(\Gamma)} .$$

Which method to use?

Choosing data approximation method

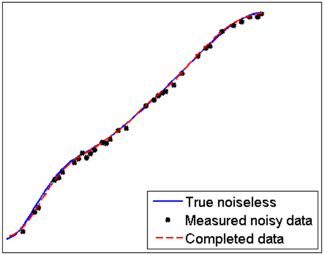
- Use Mathematics, not Politics, to select method: this approach can work in practice.
- Concentrate on EIT / DC resistivity with piecewise smooth conductivity $\mu(\mathbf{x})$.

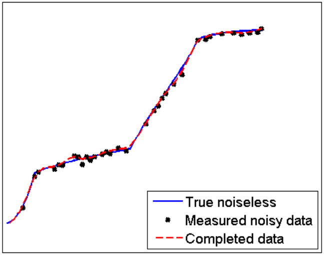
1. If μ's discontinuities are all away from boundary, then $u \in H^2(\Gamma)$. So, use regularization with Laplacian (**Option 4**).
2. If μ has discontinuities which extend to boundary, then $u \in H^1(\Gamma)$. So, use L2G (**Option 3**).

See [Roosta, Doel & A., 2014] for theorems justifying the above.

Choosing data approximation method

Experiments with 50% data completion and 5% noise:



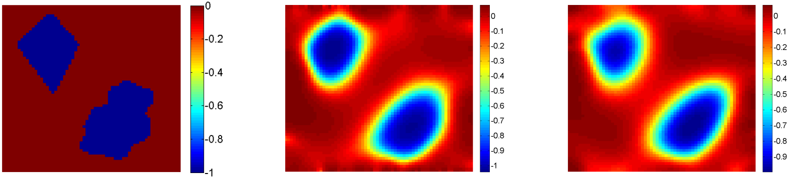


Left: Laplacian for $u \in H^2(\Gamma)$. Right: L2G for $u \in H^1(\Gamma)$

$\mu_I = .1,\ \mu_{II} = 1,$ NOISE 3%, $s = 961,$ COMP. 20%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

Method	Random Subset	Data Completion
Work	3,367	1,597

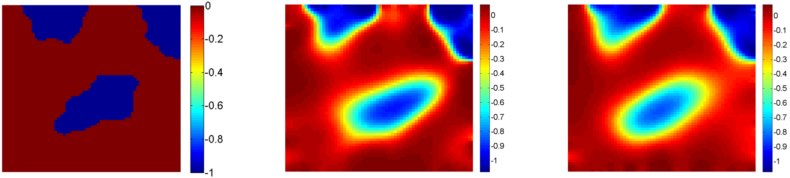


Left: true. Center: RS. Right: DC

$\mu_I = .1,\ \mu_{II} = 1$, NOISE 3%, $s = 961$, COMP. 20%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

Method	Random Subset	Data Completion
Work	6,302	2,769

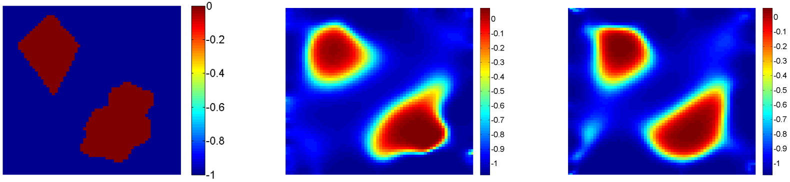


Left: true. Center: RS. Right: DC

$\mu_I = 1,\ \mu_{II} = .1,$ NOISE $5\%,\ s = 961,$ COMP. 50%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

Method	Random Subset	Data Completion
Work	5,139	2,320

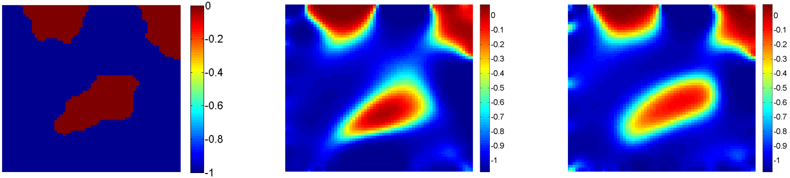


Left: true. Center: RS. Right: DC

$\mu_I = 1,\ \mu_{II} = .1,$ NOISE $5\%,\ s = 961,$ COMP. 50%

Compare using Gaussian distribution with data completion vs random subset which does not require data completion.

Method	Random Subset	Data Completion
Work	5,025	1,818



Left: true. Center: RS. Right: DC

Summary and moreover

- In the accompanying paper there are also 3D results, similar to the ones for the case with the same receivers.
- Both variants with and without level set method were tried.
- The Data Completion method was always faster than Random Subset by a factor of at least 2 and up to 4.
- Data completion of up to $\approx 50\%$ works fine. But reconstructions deteriorate upon completing scarcer data!

OUTLINE

- Motivation
- Completing scarce (sparse) data
- Uncertainty in data locations
- Forced to cut corners?
- Randomized algorithms for many data sets
- Conclusions

Conclusions

- Data completion and other statistically unholy manipulations such as ignoring location uncertainty are **not an ideal undertaking** from a theoretical point of view.
- But in practical situations it is often **quietly done** by mathematicians, computer scientists and engineers alike.
- We have seen instances where (more massive) such practices **should be avoided**.
- We have seen instances where such practices **can be tolerated**, typically when other uncertainties dominate.
- We have seen instances where such practices seem **essential for obtaining plausible results**, and where better algorithms are further sought.
- The larger the proportion of missing data, the harder it is to produce an adequate completed set.

Bayes made me do it