3-D mass map reconstruction with sparsity regularization

Xiangchong Li, Naoki Yoshida, Masamune Oguri, Shiro Ikeda, Wentao Luo

August 14, 2020

Department of Physics, University of Tokyo

Background

Lensing

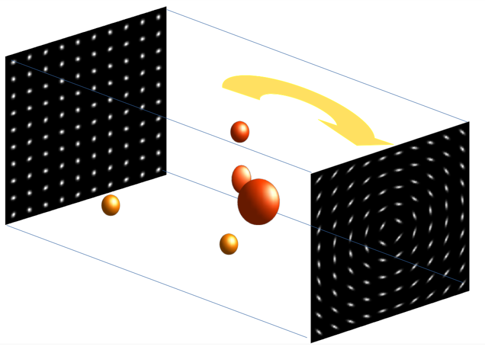
From $\delta(z_l)$ to $\gamma(z_s)$

$$\kappa(\vec{\theta}, z_s) = \boxed{\int_0^{z_s} dz_l K(z_l, z_s)}\, \delta(\vec{\theta}, z_l).$$

$$K(z_l, z_s) = \begin{cases} \frac{3H_0\Omega_M}{2c} \frac{\chi_l \chi_{sl}(1+z_l)}{\chi_s E(z_l)} & (z_s > z_l) \\ 0 & (else). \end{cases}$$

$$\gamma^t(\vec{\theta}, z_s) = \boxed{\int d^2\theta' D(\vec{\theta} - \vec{\theta'})}\, \kappa(\vec{\theta'}, z_s),$$

$$D(\vec{\theta}) = -\frac{1}{\pi}(\theta_1 - i\theta_2)^{-2}.$$



A non-local convolution in both line-of-sight direction and the transverse plane.

Model

Model space ϕ

$$\delta = \int d^3 r \phi(\vec{r}) x(\vec{r})$$

$$\delta = \mathbf{\Phi} x$$

- $\phi = \delta_D$
  - 2D [Kaiser & Squires 1995]
- $\phi = e^{i\vec{k}\cdot\vec{r}}$
  - Wiener filter (Ridge regularization) [Oguri et. al 2018]
- $\phi = \mathrm{starlets}$
  - lasso sparsity regularization (Glimpse3D) [Leonard et. al 2013]

χ^2 fitting

Lensing (DK)

$$\gamma = \mathbf{DK}\delta$$

Model (Φ)

$$\gamma = \mathbf{DK\Phi}x + \epsilon$$

Systematics (S)

$$\gamma = \mathbf{SDK\Phi}x + \epsilon$$

χ^2 fitting

$$\min\{\|\gamma - \mathbf{A}\mathrm{x}\|_2^2\}$$
$$x = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\gamma$$

Difficulties in the χ^2 fitting

ill-posed problem

$$\min\{\|\gamma - \mathbf{A}\mathrm{x}\|_2^2\}$$
$$x = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\gamma$$

1. Shape noise ($\sigma_\gamma \sim 0.3$, $\gamma \sim 0.03$)
2. Masks (have to do zero padding)

χ^2 fitting with regularization

regularizations

$$\|\gamma - \mathbf{A}x\|_2^2 + C(x)$$

1. Ridge regularization (Wiener filter) [Oguri et. al 2018]

$$C(x) \sim \|x\|_2^2$$

2. Lasso regularization (Glimpse3D) [Leonard et. al 2013]

$$C(x) \sim \|x\|_1^1$$

3. Total Square Variance (TSV) regularization

$$C(x) \sim \sum_i (x_{i+1} - x_i)^2$$

Bayesian Interpretion of regularization

Prior

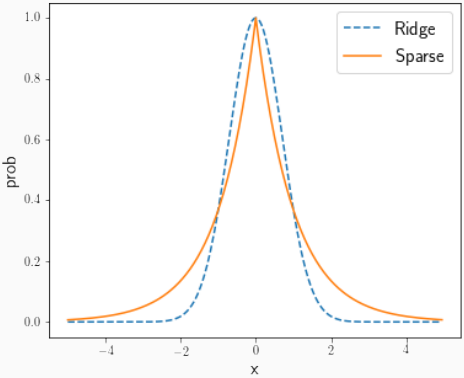
$$P(x|\gamma) \propto P(x)P(\gamma|x)$$

Ridge regularization

$$P(x) \propto exp(-\|x\|_2^2)$$

Sparse regularization

$$P(x) \propto exp(-\|x\|_1^1)$$



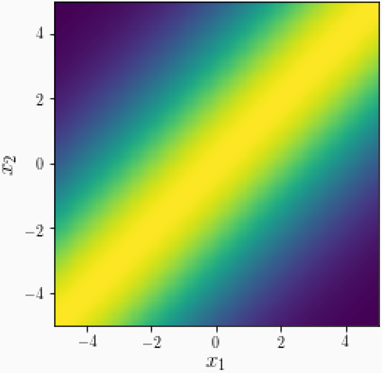
Bayesian Interpretion of regularizations

Prior

$$P(x|\gamma) \propto P(x)P(\gamma|x)$$

TSV

$$P(x) \propto exp(-\sum_i (x_i - x_{i+1})^2)$$



Method

Model Dictionary

We use surface density profile of [Takada & Jain (2003)] as the transverse profile of our basis vector.

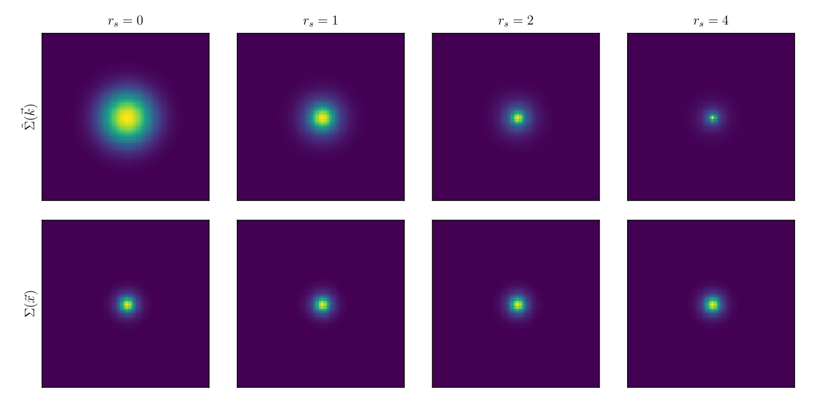
$$F(r) = \begin{cases} -\frac{\sqrt{c^2-r^2}}{(1-r^2)(1+c)} + \frac{\mathrm{arccosh}\left(\frac{\mathrm{r}^2+\mathrm{c}}{\mathrm{r}(1+\mathrm{c})}\right)}{(1-r^2)^{3/2}} & (r < 1), \\ \frac{\sqrt{c^2-1}}{3(1+c)}\left(1 + \frac{1}{c+1}\right) & (r = 1), \\ -\frac{\sqrt{c^2-r^2}}{(1-r^2)(1+c)} + \frac{\mathrm{arccos}\left(\frac{\mathrm{r}^2+\mathrm{c}}{\mathrm{r}(1+\mathrm{c})}\right)}{(r^2-1)^{3/2}} & (1 < r \le c), \\ 0 & (r > c). \end{cases}$$

In the line-of-sight direction, we use Dirac delta function

$$\phi(\vec{r}) = F(|\vec{\theta}|/r_s)\delta_D(z)$$

Model Dictionary

Smoothed basis vectors



pixel size $= 1$ arcmin, Gaussian smoothing scale $= 1.5$ arcmin, redshift bin $\Delta z = 0.05$, 4 dictionary frames. The indexing $x[s, i, j, k]$ or x_α

Regularized ML Fitting

The Loss function

$$L(x) = \frac{1}{2}\|\gamma - \mathbf{A}x\|_2^2 +$$

$$\frac{1}{2}\tau \sum_{i,j}\left[(x[s,i+1,j,k] - x[s,i,j,k])^2 + x[s,i,j+1,k] - x[s,i,j,k])^2\right]$$

$$+ \lambda\sigma(x)\|x\|_1.$$

$$\begin{aligned} L(x) &= \frac{1}{2}(\gamma_\alpha - A^*_{\alpha\beta}x_\beta)(\gamma_\alpha - A_{\alpha\mu}x_\mu) \\ &+ \frac{\tau}{2}[(D^1_{\alpha\beta}x_\beta)(D^1_{\alpha\mu}x_\mu) + (D^2_{\alpha\beta}x_\beta)(D^2_{\alpha\mu}x_\mu)] \\ &+ \lambda\sigma_\beta\left\|x_\beta\right\|_1, \end{aligned} \tag{1}$$

where $\mathbf{D^{1,2}}$ refers to finite difference operator. Here we adopt Einstein notation on α, β, μ, ν.

Solving the ML problem

The quadratic term

$$G(x) = \frac{1}{2}(\gamma_\alpha - A^*_{\alpha\beta}x_\beta)(\gamma_\alpha - A_{\alpha\mu}x_\mu) \\ + \frac{\tau}{2}[(D^1_{\alpha\beta}x_\beta)(D^1_{\alpha\mu}x_\mu) + (D^2_{\alpha\beta}x_\beta)(D^2_{\alpha\mu}x_\mu)],$$

Coordinate descent Algorithm

Begin with $x^{(0)} = 0$, for one coordinate $t[s, i, j, k]$

$$x_t^{(n+1)} = \mathrm{ST}_{\lambda\sigma_t}(x_t^{(n)} - \frac{\partial_t G(x^{(n)})}{A_{\alpha t}A_{\alpha t} + 4\tau}),$$

where ST is the soft thresholding function defined as

$$\mathrm{ST}_\lambda(x) = \mathrm{sign}(x)\max(|x| - \lambda, 0).$$

We set $\lambda = 4$, $\tau = \frac{1}{N}\sum_t^N(A_{\alpha t}A_{\alpha t}) \times 20$, $\sigma(x)$ the noise expected on projector x.

Solving the ML problem

Coordinate descent Algorithm

$$x_t^{(n+1)} = \mathrm{ST}_{\lambda\sigma_t}\left(x_t^{(n)} - \frac{\partial_t G(x^{(n)})}{A_{\alpha t}A_{\alpha t} + 4\tau}\right),$$

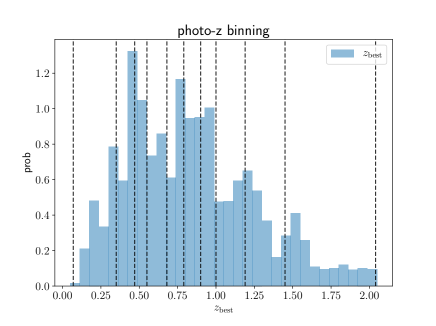
$$\mathrm{ST}_\lambda(x) = \mathrm{sign}(x)\max(|x| - \lambda, 0).$$

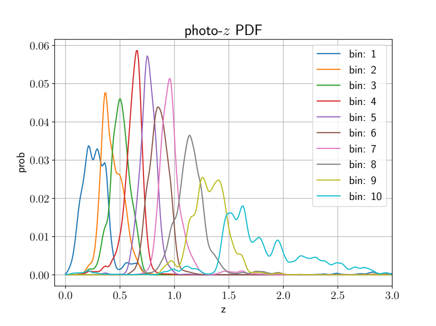
- Begin with $x^{(0)} = 0$,
- for n=1... N_{iter}
  1. select one coordinate $t[s, i, j, k]$
  2. $\bar{x}_t^{(n)} = x_t^{(n)} - \frac{\partial_t G(x^{(n)})}{A_{\alpha t}A_{\alpha t}+4\tau}$
  3. - if $\bar{x}_t^{(n)} > \lambda\sigma_t$: $x_t^{(n+1)} = 0$.
     - else: $x_t^{(n+1)} = \bar{x}_t^{(n)} - \lambda\sigma_t$

Test on HSC-like Simulation

Simulation

Binning in line-of-sight direction

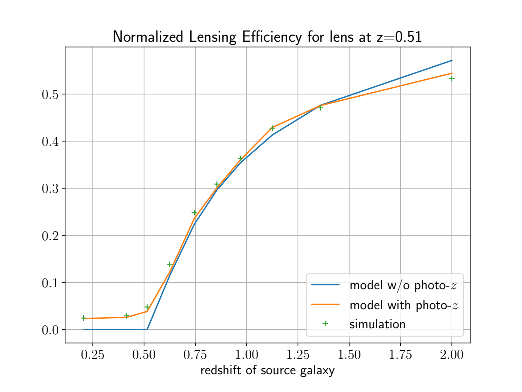




Simulation

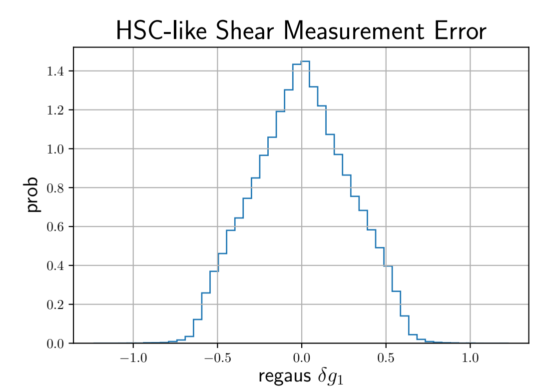
Photo-z error

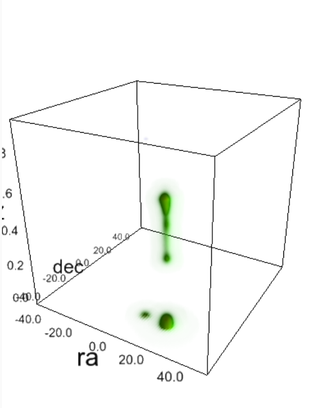
$$n_{zp}(z) = \left\langle P(z|z_p) \right\rangle$$



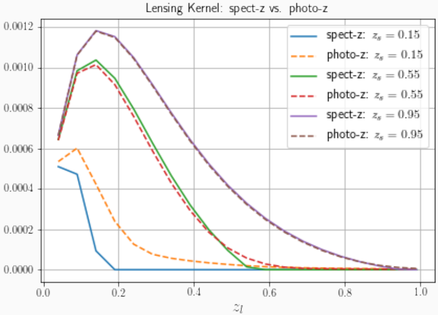
Simulation

Noise





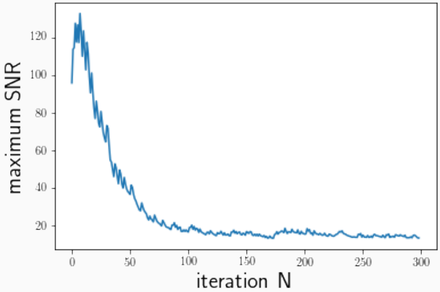
dec
ra
0.4
0.2
-40.0
-20.0
0.0
20.0
40.0



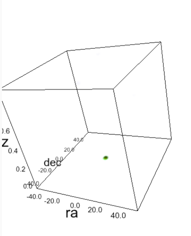
Lensing Kernel: spect-z vs. photo-z
spect-z: $z_s = 0.15$
photo-z: $z_s = 0.15$
spect-z: $z_s = 0.55$
photo-z: $z_s = 0.55$
spect-z: $z_s = 0.95$
photo-z: $z_s = 0.95$
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
0.0
0.2
0.4
0.6
0.8
1.0
z_l

Glimpse3D (Leonard et. al 2013)

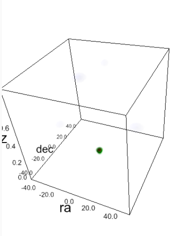
- Begin with $x^{(0)} = 0$, $s^{(0)} = 0$
- for n=1...N_{iter}
  1. choose the coordinate $t^{(n)}$ s.t. $\max\{\frac{\partial_t G(x^{(n)})}{A_{\alpha t} A_{\alpha t} + 4\tau} / (\sigma_t)\}$
  2. - if $\mathrm{SNR_{max}} > \mathrm{s^{(n-1)}}$: $s^{(n)} = (\mathrm{SNR_{max}} + \mathrm{SNR_{max2}})/2$
     - ...



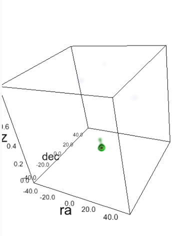
Outcomes2



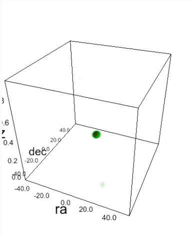
$z = 0.01, logM = 14.5$



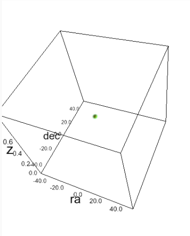
$z = 0.01, logM = 15.2$



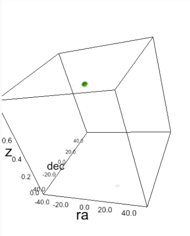
$z = 0.13, logM = 15.5$



$z = 0.26, logM = 14.7$



$z = 0.51, logM = 15.2$



$z = 0.88, logM = 15.2$

Thanks!