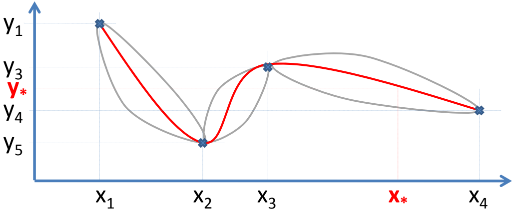
Gaussian Process Approach to Interpolation and Correlation Estimate

References:

Online information: **http://www.gaussianprocess.org/**

Nice easy book: **Gaussian Processes for Machine Learning**, by Rasmussen and Williams

Gaussian Processes - 1 metal



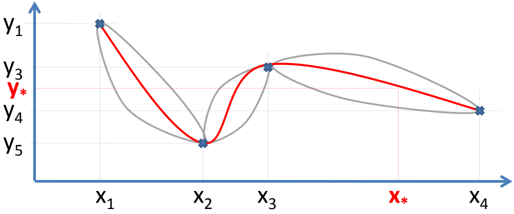
Prior assumption on both observed and interpolated points:

$$\left(\begin{array}{c} y_1 \\ \vdots \\ y_n \\ \hline y_* \end{array}\right) \stackrel{d}{=} \mathcal{N}\left(\mathbf{0}; \begin{pmatrix} K(x_i, x_j) & K(x_i, x_*) \\ K(x_*, x_i) & 1 \end{pmatrix}\right)$$

Many choices for **Kernel function**, e.g.

$$K(x_i, x_j) = \begin{cases} \exp\{-\frac{1}{h}|x_i - x_j|\}, & \text{Ornstein-Uhlenbeck} \\ \exp\{-\frac{1}{h}|x_i - x_j|^2\}, & \text{Radial basis functions} \end{cases}$$

Gaussian Processes - 1 metal



Posterior conditional on observations is normal

$$(y_*|(y_1,\ldots,y_n)') \stackrel{d}{=} \mathcal{N}\,(\,mean;\ var)$$
$$mean = K(x_*,x)K(x,x)^{-1}y$$
$$var = 1 - K(x_*,x)K(x,x)^{-1}K(x,x_*)$$

Gaussian Processes - 2 metals

- **Spatial** and metal-metal correlations can be **intertwined**
- We use **tensor product** Kernel function to disentangle

$$\begin{pmatrix} y_i^{(1)} \\ y_i^{(2)} \\ y_j^{(1)} \\ y_j^{(2)} \end{pmatrix} \stackrel{d}{=} \mathcal{N}\left(\mathbf{0}; \begin{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} & K(x_i, x_j)\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \\ K(x_i, x_j)\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} & \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \end{pmatrix}\right)$$

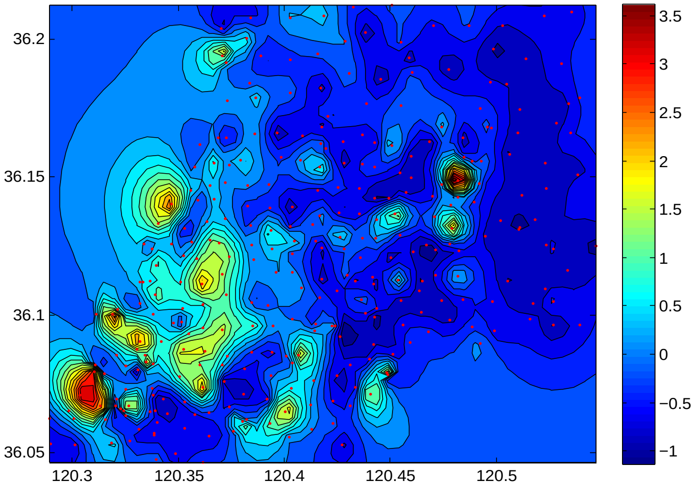
- We computed the **Posterior** of interpolation point given the observations – it's **still Normal** (of course)

Gaussian Processes - 2 metals

- Need to **learn** correlation (ρ) and bandwidth (h)
- Can do this via **Maximum Likelihood Estimation** (MLE)
- Alternatively, can compute the spatial correlation to determine h and then use MLE for ρ
- As $h \downarrow 0$, the MLE of correlation reduces to pair-wise metal-metal correlation

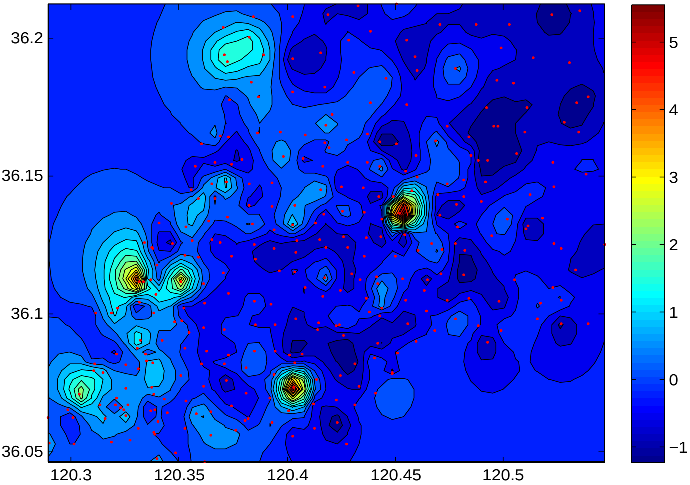
Gaussian Processes – results

Cd – $h = 0.01$, estimated $\rho = 0.14$ with *As*



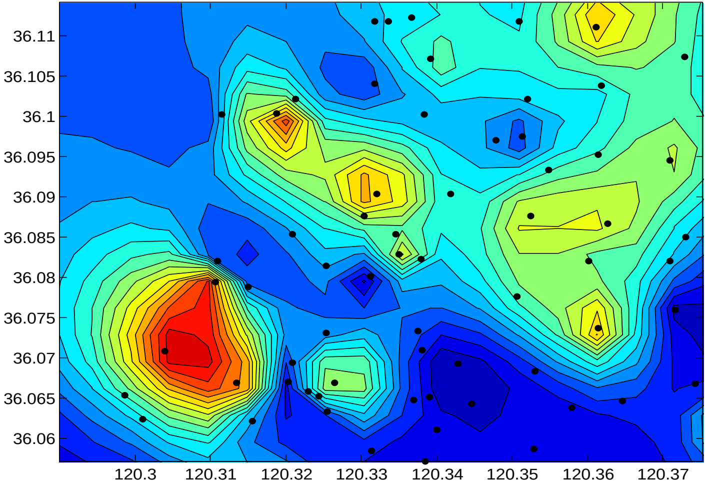
Gaussian Processes – results

As – $h = 0.01$, **estimated** $\rho = 0.14$ **with** *Cd*



Gaussian Processes – results

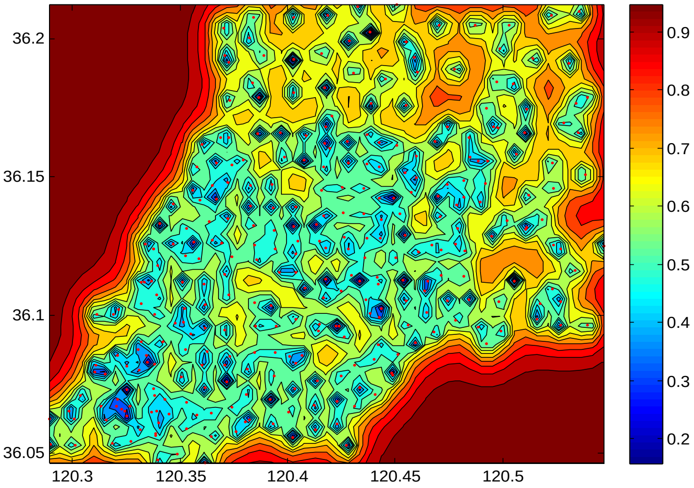
Cd – $h = 0.01$, estimated $\rho = 0.14$ with As – zoomed in



Notice that **extremes not always at data points**

Gaussian Processes – results

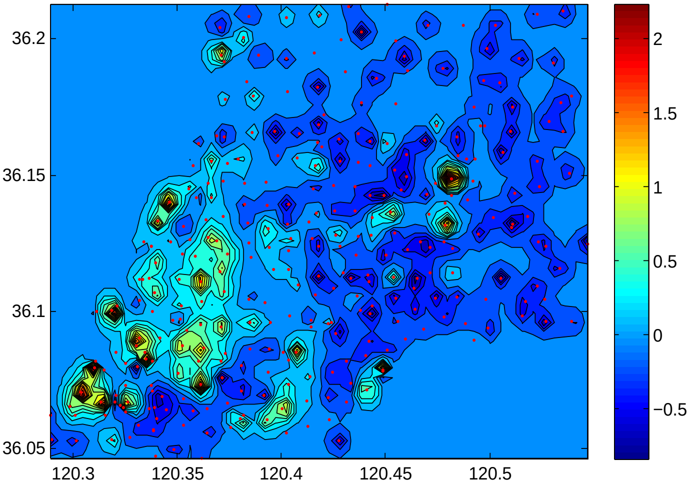
Cd – $h = 0.01$, estimated $\rho = 0.14$ with As – posterior std.dev



Notice that **variance drops to zero at data points**

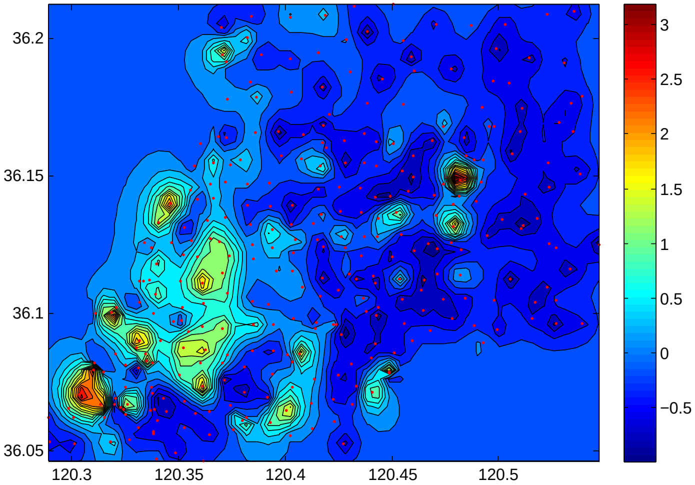
Gaussian Processes – varying bandwidth

Cd – $h = 0.0025$**, estimated** $\rho = 0.21$ **with** As



Gaussian Processes – varying bandwidth

Cd – $h = 0.005$**, estimated** $\rho = 0.17$ **with** As



Gaussian Processes – varying bandwidth

Cd – $h = 0.01$, **estimated** $\rho = 0.14$ **with** As