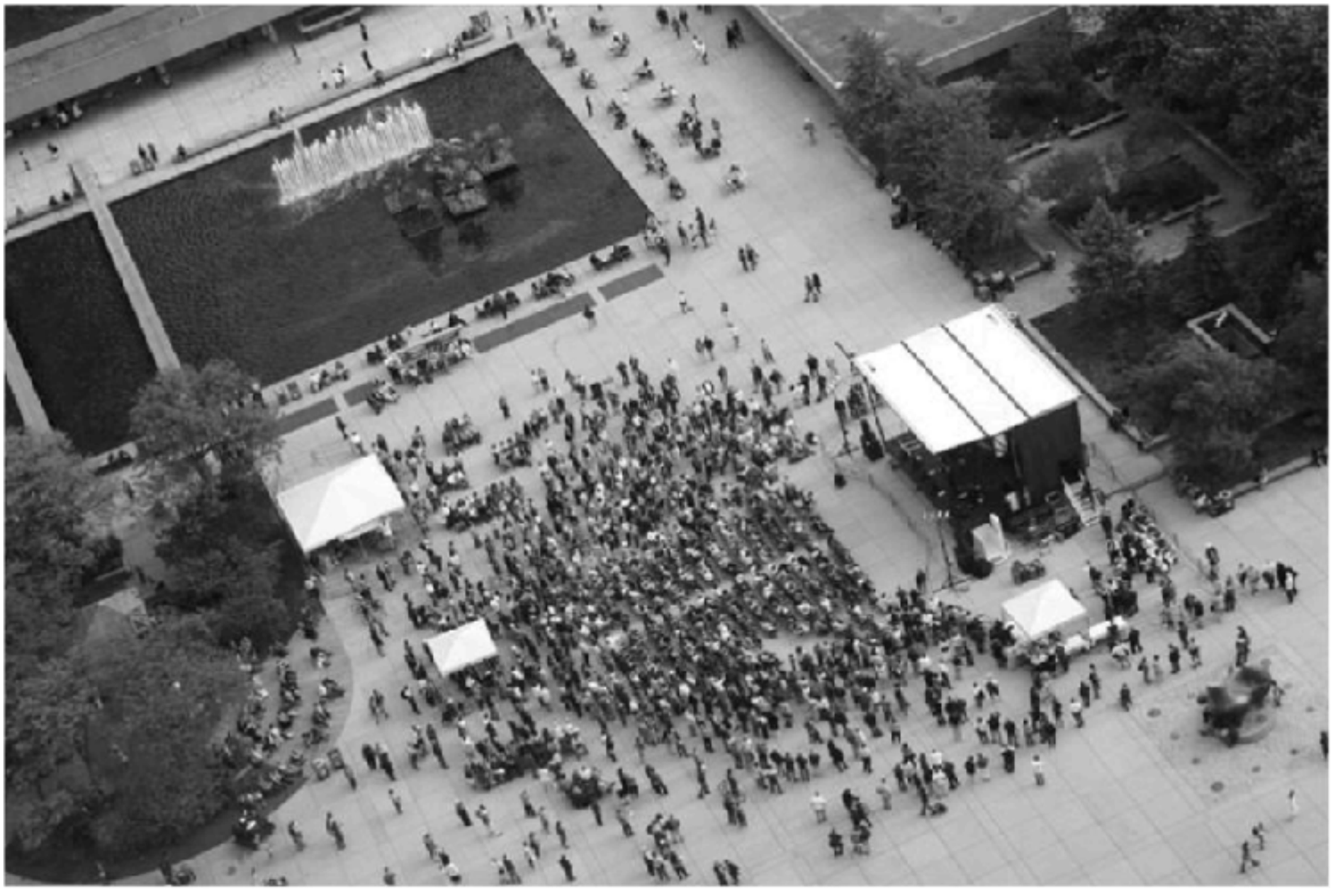
Kernel Ridge Regression and Gaussian Process Regression

Machine Learning Group, TU Berlin

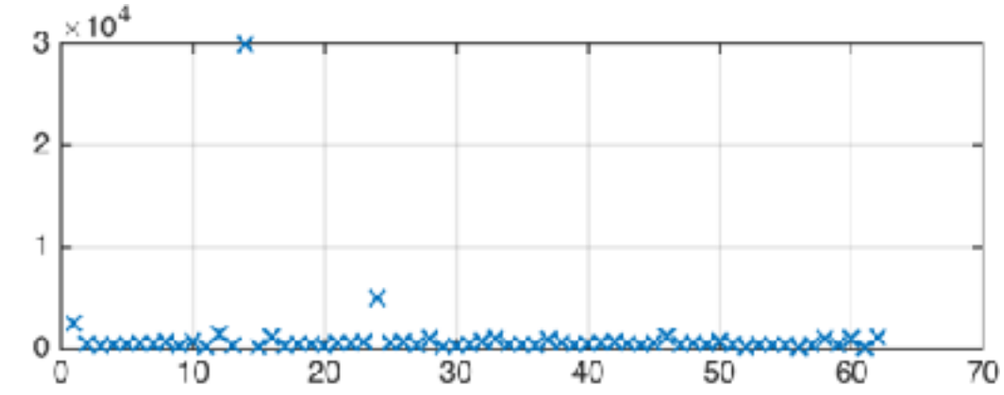
How many people?

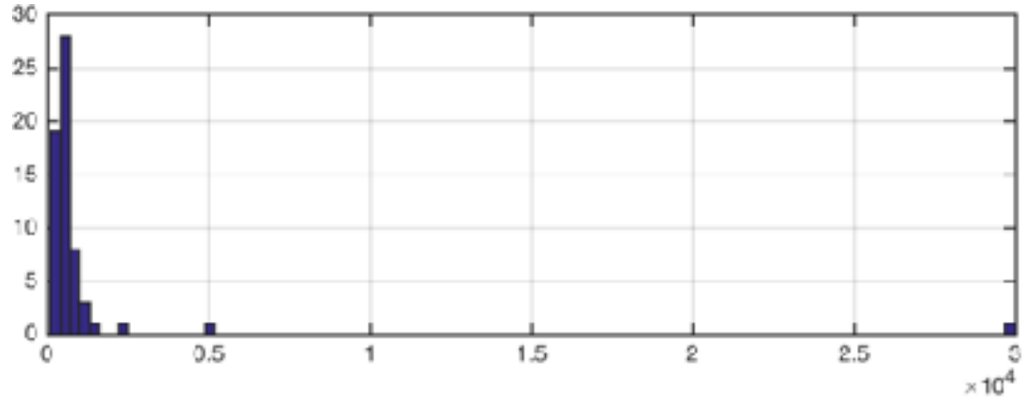


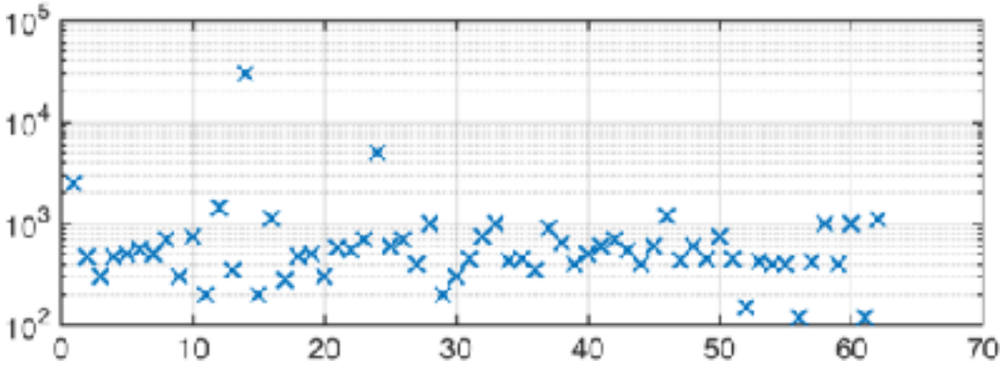
How many people?

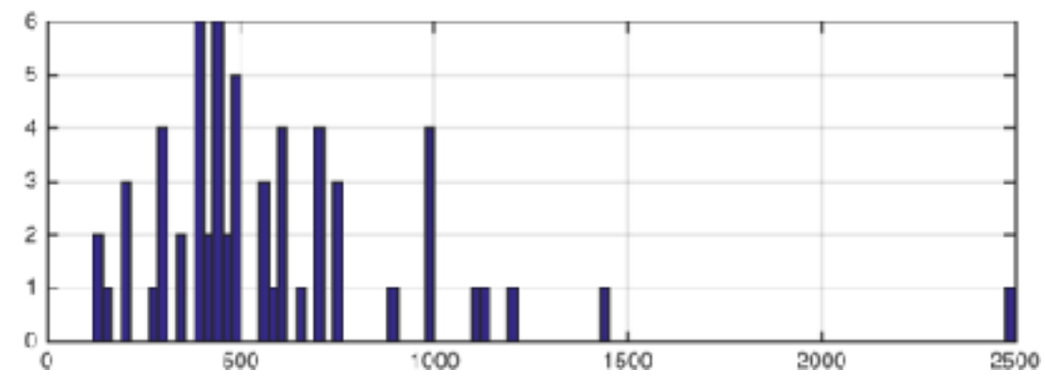
The data

- 62 votes
- Range: 120 — 30000









Averaging

- Arithmetic mean: $\hat{\mu}_{\mathrm{a}} = \frac{1}{N}\sum_{i=1}^{N} x_i = \arg\min_{\mu} \sum_{i=1}^{N} (\mu - x_i)^2$

Averaging

- Arithmetic mean: $\hat{\mu}_{\mathrm{a}} = \frac{1}{N}\sum_{i=1}^{N} x_i = \arg\min_{\mu} \sum_{i=1}^{N} (\mu - x_i)^2$

- Median: $\hat{\mu}_{\mathrm{m}} = x_{\lfloor \frac{N+1}{2} \rfloor} = \arg\min_{\mu} \sum_{i=1}^{N} |\mu - x_i|$, assuming sorted x_i

  → robust

Averaging

- Arithmetic mean: $\hat{\mu}_{\mathrm{a}} = \frac{1}{N}\sum_{i=1}^{N} x_i = \arg\min_{\mu} \sum_{i=1}^{N} (\mu - x_i)^2$

- Median: $\hat{\mu}_{\mathrm{m}} = x_{\lfloor \frac{N+1}{2} \rfloor} = \arg\min_{\mu} \sum_{i=1}^{N} |\mu - x_i|$, assuming sorted x_i

  → robust

- Generalized f-mean: $\hat{\mu}_{\mathrm{f}} = f^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right)$

$f(x)$	$= x$	:	arithmetic mean
$f(x)$	$= x^2$	:	root mean square
$f(x)$	$= \log(x)$	:	geometric mean
$f(x)$	$= 1/x$	:	harmonic mean

Teaching evaluation

If you want to fill out the form

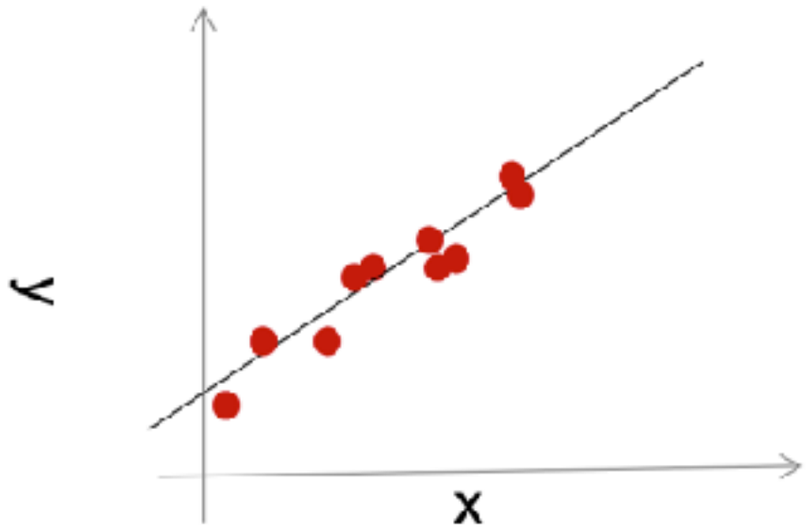
- Great!
- Please don’t fold the paper, scribble over the page delimiters/barcode.
- Average over all instructors incl. exercise session.
- Any questions, please ask.

Recap: OLS linear regression

Given a sample $\mathcal{D} = \{\mathcal{X}, \mathbf{y}\}$ of d-dim. inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top \in \mathbb{R}^{n \times d}$ and corresponding scalar outputs $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$, estimate a linear relationship between X and $\mathbf{y}$ assuming Gaussian distributed noise.

$$y_i = \mathbf{x}_i^\top \beta + \varepsilon_i$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$



Recap: OLS linear regression

Given a sample $\mathcal{D} = \{\mathcal{X}, \mathbf{y}\}$ of d-dim. inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top \in \mathbb{R}^{n \times d}$ and corresponding scalar outputs $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$, estimate a linear relationship between X and $\mathbf{y}$ assuming Gaussian distributed noise.

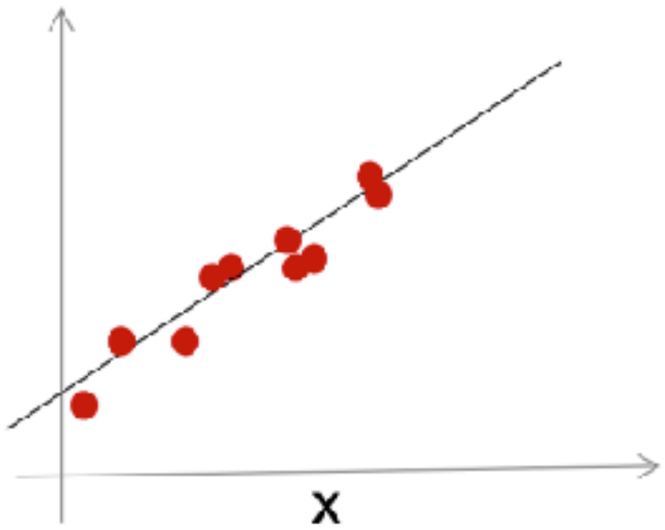
$$y_i = \mathbf{x}_i^\top \beta + \varepsilon_i$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$p(\mathcal{D}|\beta, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y_i - \mathbf{x}_i^\top \beta\right)^2}{2\sigma^2}\right)$$

$$l(\beta, \sigma^2) = \log\left(\frac{N}{\sqrt{2\pi}\sigma}\right) + \sum_{i=1}^{N}\left(-\frac{\left(y_i - \mathbf{x}_i^\top \beta\right)^2}{2\sigma^2}\right)$$

$$\{\hat{\beta}, \hat{\sigma}^2\} = \arg\min_{\{\beta, \sigma^2\}} l(\beta, \sigma^2)$$



Recap: OLS linear regression

Maximum-likelihood estimates:

$$\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}$$

$$\hat{\sigma}^2 = \frac{1}{n} ||\mathbf{y} - X(X^\top X)^{-1} X^\top \mathbf{y}||^2$$

Recap: OLS linear regression

Maximum-likelihood estimates:

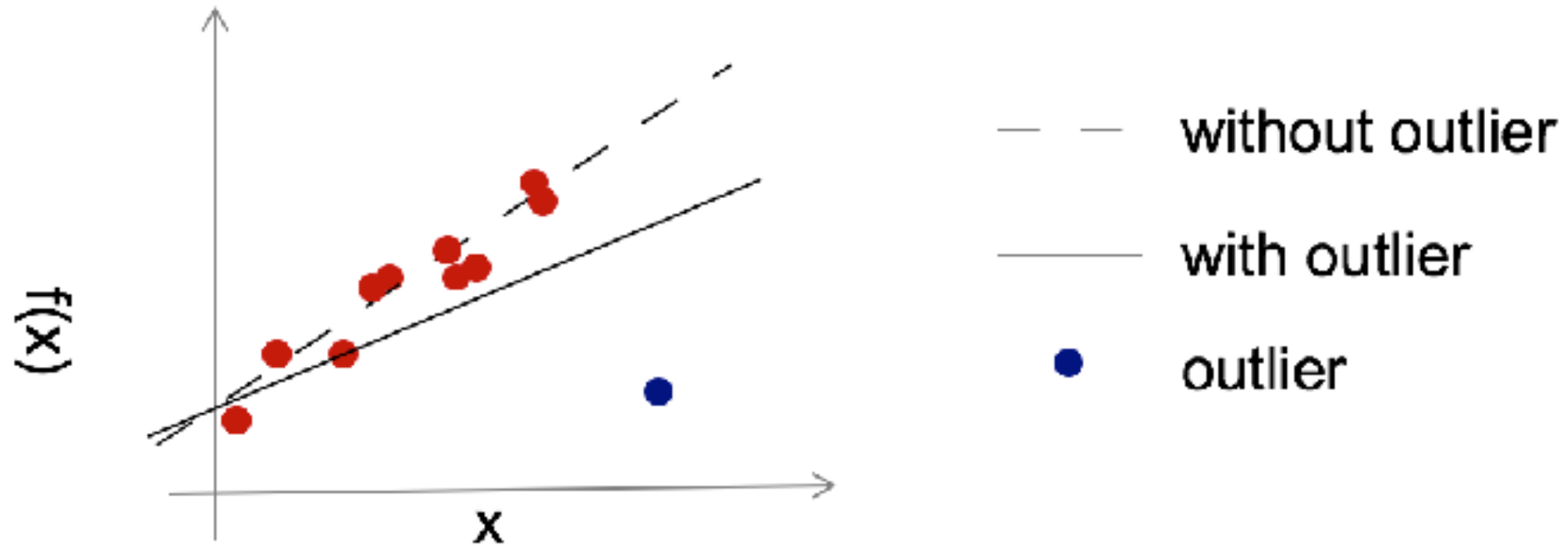
$$\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}$$

$$\hat{\sigma}^2 = \frac{1}{n} ||\mathbf{y} - X(X^\top X)^{-1} X^\top \mathbf{y}||^2$$

Four meaningful extensions:

1. Robust noise models
2. Regularization
3. Bayesian treatment
4. Kernelization/Nonlinearization

Robust noise models



OLS solution not robust: outliers dominate the quadratic loss function.

Robust loss functions

Solution: sub-quadratic loss functions

Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

- Huber-loss:

$$\begin{cases} \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta)^2 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \leq \delta, \\ \delta\,|y_i - \mathbf{x}_i^\top \beta| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

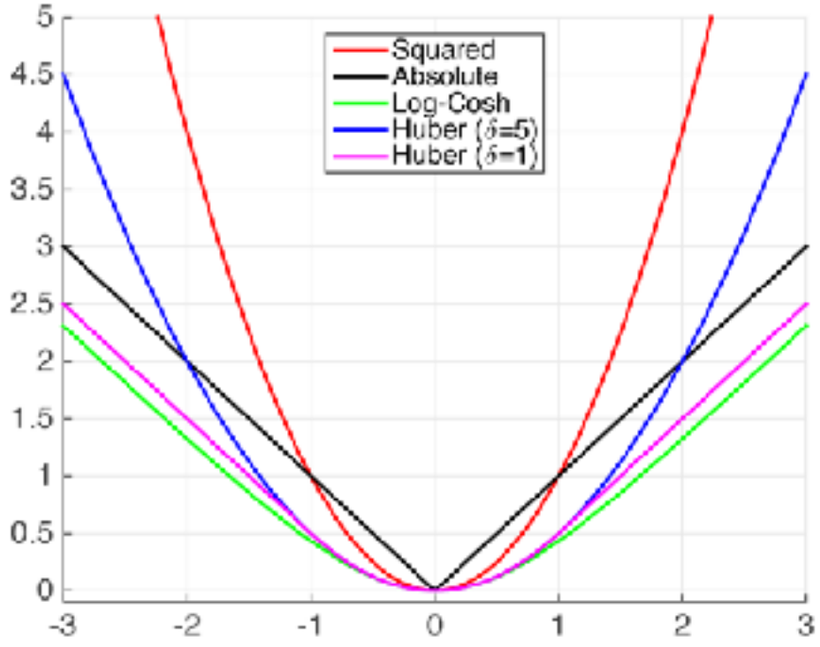
Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

- Huber-loss:

$$\begin{cases} \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta)^2 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \leq \delta, \\ \delta\,|y_i - \mathbf{x}_i^\top \beta| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Kilian Weinberger

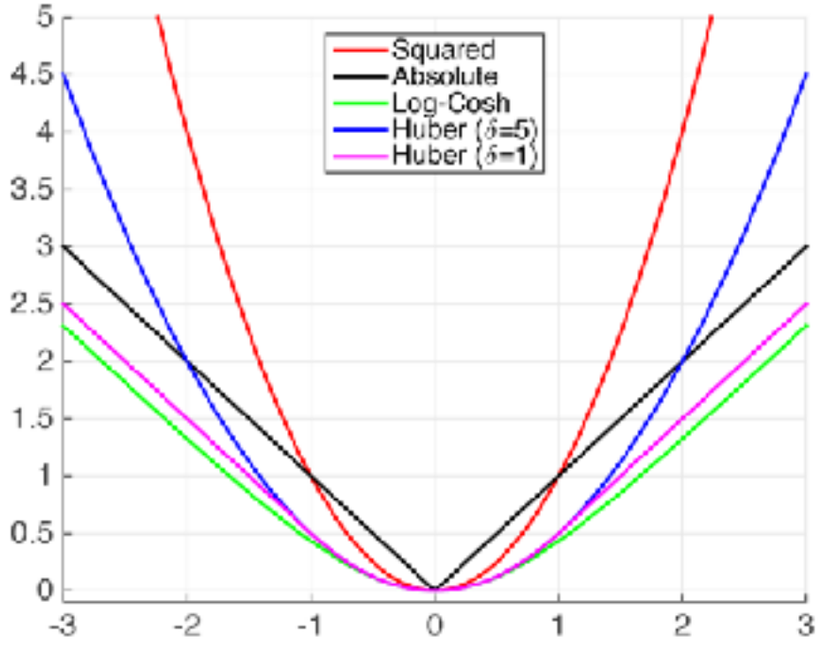
Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

- Huber-loss:

$$\begin{cases} \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta)^2 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \delta, \\ \delta\,|y_i - \mathbf{x}_i^\top \beta| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Kilian Weinberger

- ε-insensitive loss → Support Vector Regression

$$\begin{cases} 0 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \varepsilon, \\ |y_i - \mathbf{x}_i^\top \beta| - \varepsilon & \text{otherwise.} \end{cases}$$

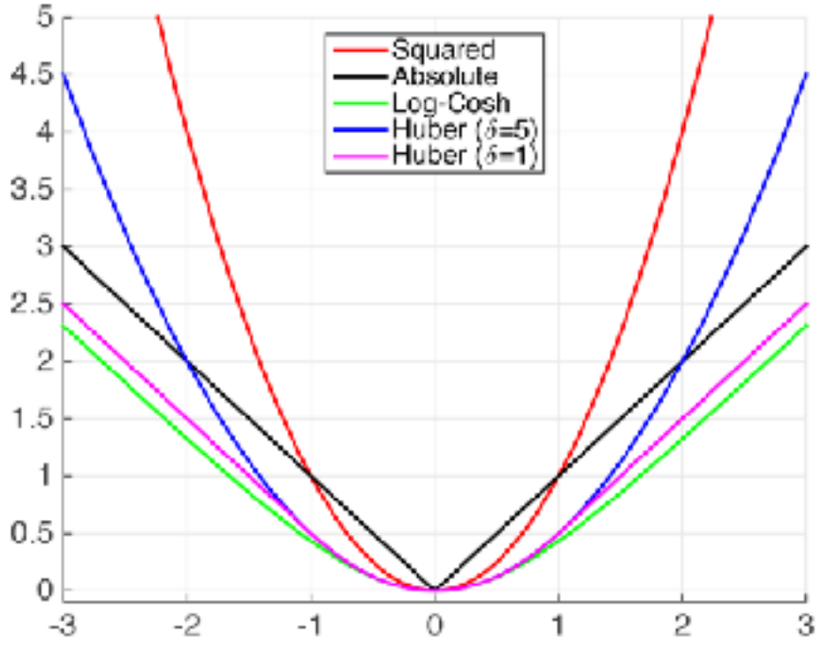
Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

- Huber-loss:

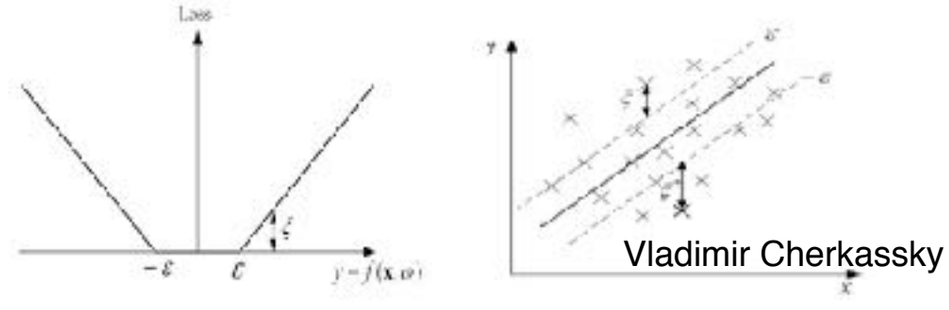
$$\begin{cases} \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta)^2 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \delta, \\ \delta\,|y_i - \mathbf{x}_i^\top \beta| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Kilian Weinberger

- ε-insensitive loss → Support Vector Regression

$$\begin{cases} 0 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \varepsilon, \\ |y_i - \mathbf{x}_i^\top \beta| - \varepsilon & \text{otherwise.} \end{cases}$$



Vladimir Cherkassky

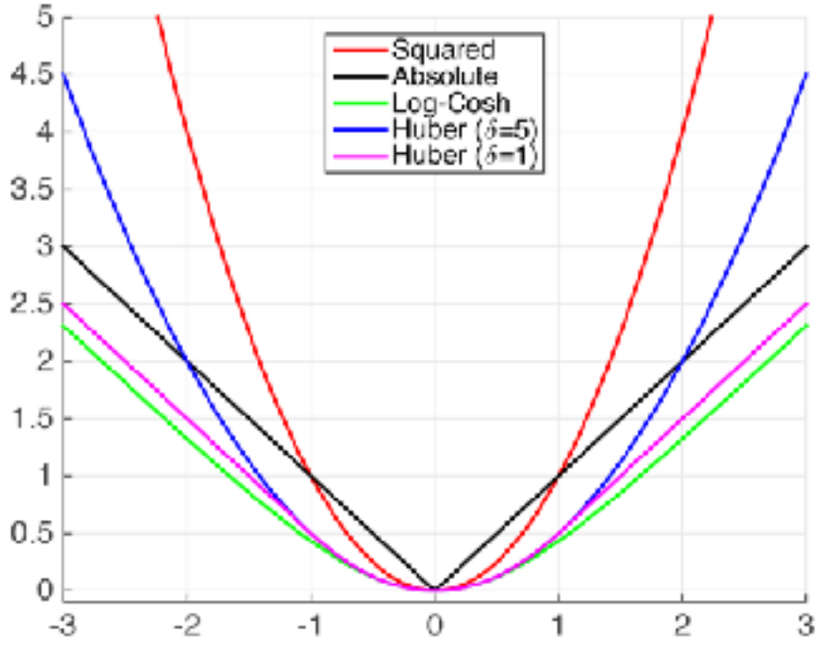
Robust loss functions

Solution: sub-quadratic loss functions

- Least-absolute deviation: $|y_i - \mathbf{x}_i^\top \beta|$

- Huber-loss:

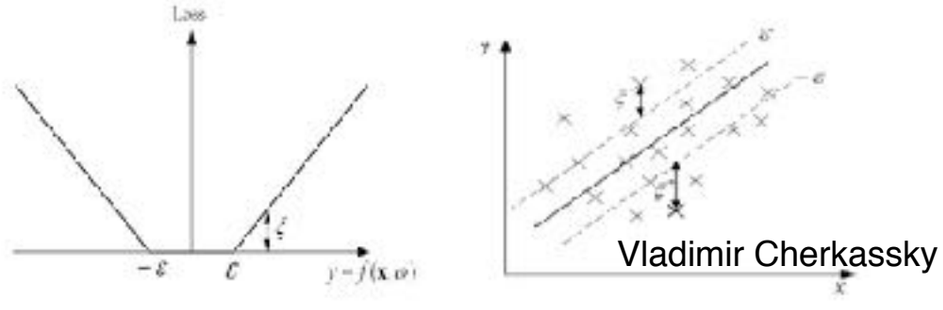
$$\begin{cases} \frac{1}{2}(y_i - \mathbf{x}_i^\top \beta)^2 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \delta, \\ \delta\,|y_i - \mathbf{x}_i^\top \beta| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Kilian Weinberger

- ε-insensitive loss → Support Vector Regression

$$\begin{cases} 0 & \text{for } |y_i - \mathbf{x}_i^\top \beta| \le \varepsilon, \\ |y_i - \mathbf{x}_i^\top \beta| - \varepsilon & \text{otherwise.} \end{cases}$$



Vladimir Cherkassky

→ Only points outside margin define solution → sparse in samples.

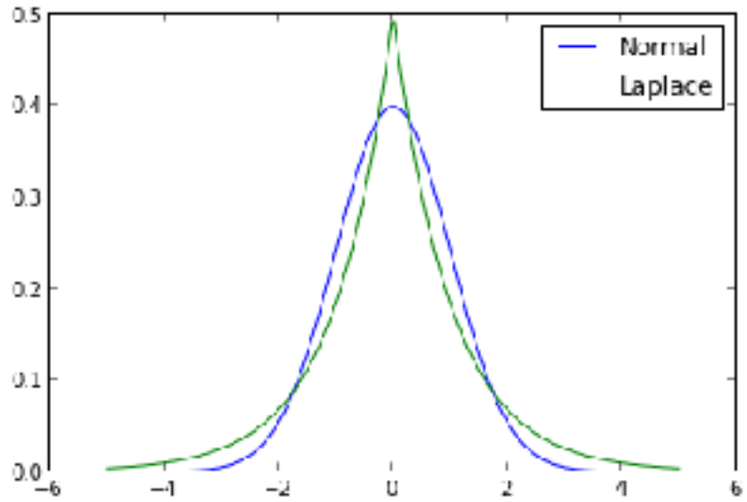
Robust loss functions

Notes:

- Even sublinear losses possible.
- No closed-form solution like for quadratic loss.
- Not even differentiable except for Huber-loss.

→ Iterative optimization using sub-gradients.

- Sub-quadratic loss functions correspond to ‘heavy-tailed’ noise distributions.



Austin Rochford

Regularization

Different problem: overfitting — occurs if $N \sim d$ or $N < d$.

- (always possible to fit a hyperplane through d-1 points in d dimensions)
- May occur for nonlinear regression even if N > d

Regularization

Different problem: overfitting — occurs if $N \sim d$ or $N < d$.

- (always possible to fit a hyperplane through d-1 points in d dimensions)
- May occur for nonlinear regression even if N > d

Solution: regularization $\hat{\beta} = \arg\min_{\beta} l_{\beta}(\varepsilon) + \lambda g(\beta)$

- Enforce model ‘simplicity’ by penalizing parameters.
- Introduce bias for the sake of reducing variance.
- Impose prior probability on parameters and obtain MAP estimate.

Regularization

Different problem: overfitting — occurs if $N \sim d$ or $N < d$.

- (always possible to fit a hyperplane through d-1 points in d dimensions)
- May occur for nonlinear regression even if N > d

Solution: regularization $\hat{\beta} = \arg\min_{\beta} l_{\beta}(\varepsilon) + \lambda g(\beta)$

- Enforce model ‘simplicity’ by penalizing parameters.
- Introduce bias for the sake of reducing variance.
- Impose prior probability on parameters and obtain MAP estimate.

Equivalent parameterizations:

$$\hat{\beta} = \arg\min_{\beta} l_{\beta}(\varepsilon) \text{ s.t. } g(\beta) \leq \kappa$$

$$\hat{\beta} = \arg\min_{\beta} g(\beta) \text{ s.t. } l_{\beta}(\varepsilon) \leq \varepsilon$$

Choices of regularizers

Ridge regression: shrink coefficients to zero (zero-mean Gaussian prior).

→ implicitly decrease degrees of freedom d.

$$l_\beta(\varepsilon) = \|\mathbf{y} - X\beta\|_2^2 \,, \quad g(\beta) = \|\beta\|_2^2$$

$$\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}$$

→ ℓ_2 -norm penalty has the same effect as adding uncorrelated noise.

Choices of regularizers

Ridge regression: shrink coefficients to zero (zero-mean Gaussian prior).

→ implicitly decrease degrees of freedom d.

$$l_\beta(\varepsilon) = \|\mathbf{y} - X\beta\|_2^2 \,, \quad g(\beta) = \|\beta\|_2^2$$

$$\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}$$

→ ℓ_2 -norm penalty has the same effect as adding uncorrelated noise.

Why not explicitly restrict dofs by looking for **sparse** β ?

ℓ_0-quasi-norm: $g(\beta) = \|\beta\|_0 = |\{i \,|\beta_i = 0\}|$

- Non-convex, hard to minimize.

Choices of regularizers

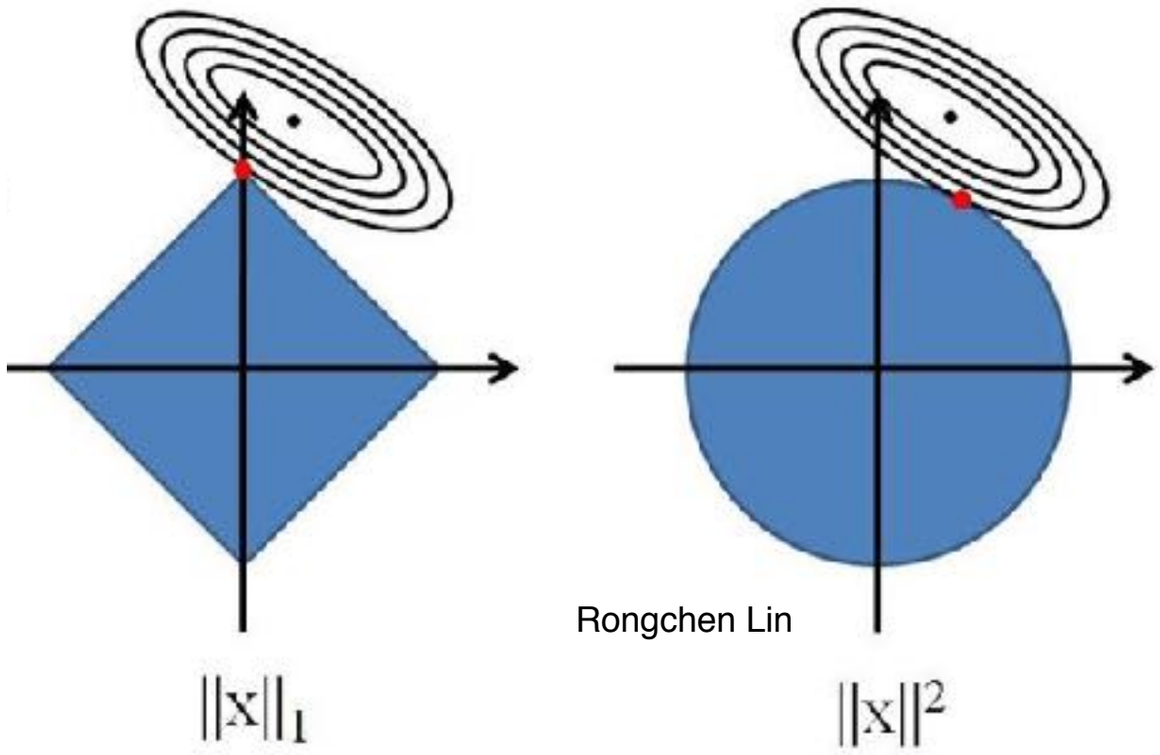
'Lasso' (Tibshirani, 1996)**:** $g(\beta) = \|\beta\|_1 = \sum_{i=1}^{d} |\beta_i|$

- Convex relaxation of the ℓ_0 quasi-norm.
- Also provides a sparse solution. **Why?**

Choices of regularizers

'Lasso' (Tibshirani, 1996)**:** $g(\beta) = \|\beta\|_1 = \sum_{i=1}^{d} |\beta_i|$

- Convex relaxation of the ℓ_0 quasi-norm.
- Also provides a sparse solution. **Why?**

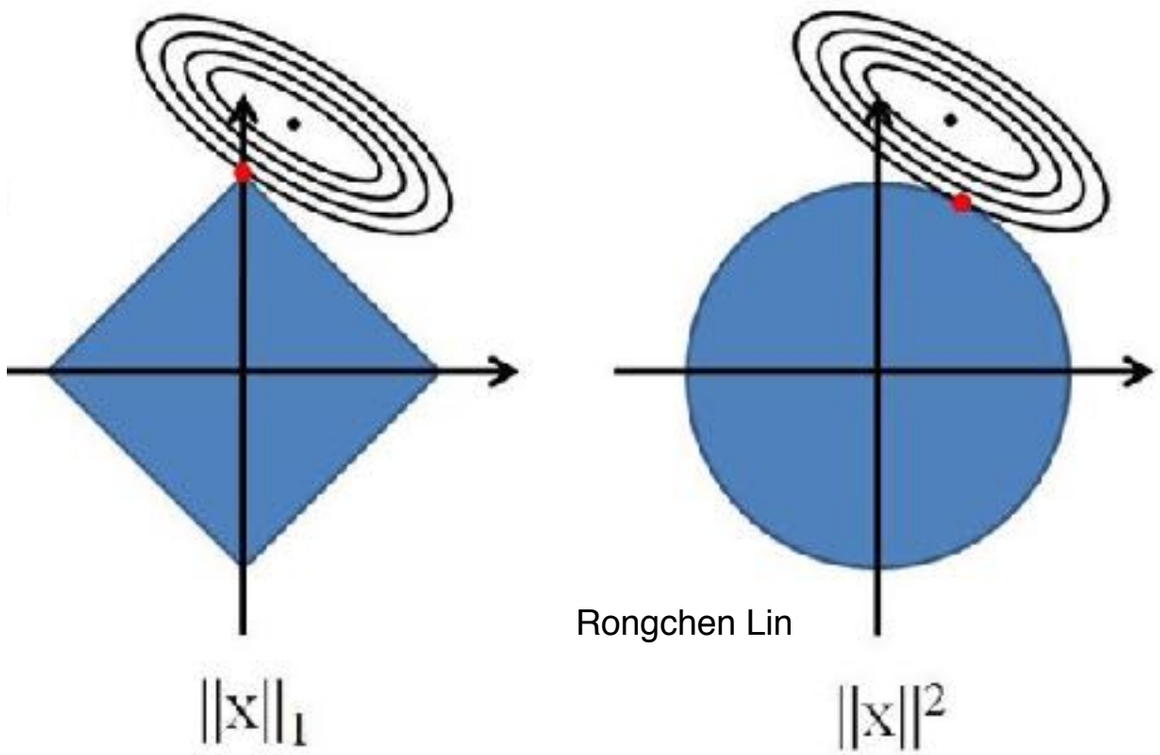


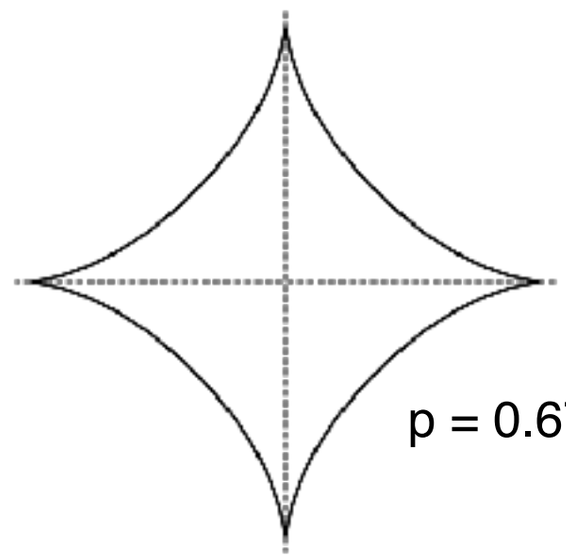
Choices of regularizers

'Lasso' (Tibshirani, 1996)**:** $g(\beta) = \|\beta\|_1 = \sum_{i=1}^{d} |\beta_i|$

- Convex relaxation of the ℓ_0 quasi-norm.
- Also provides a sparse solution. **Why?**

$$\|\beta\|_p = \left(\sum_{i=1}^{d} |\beta_i|^p \right)^{1/p}$$



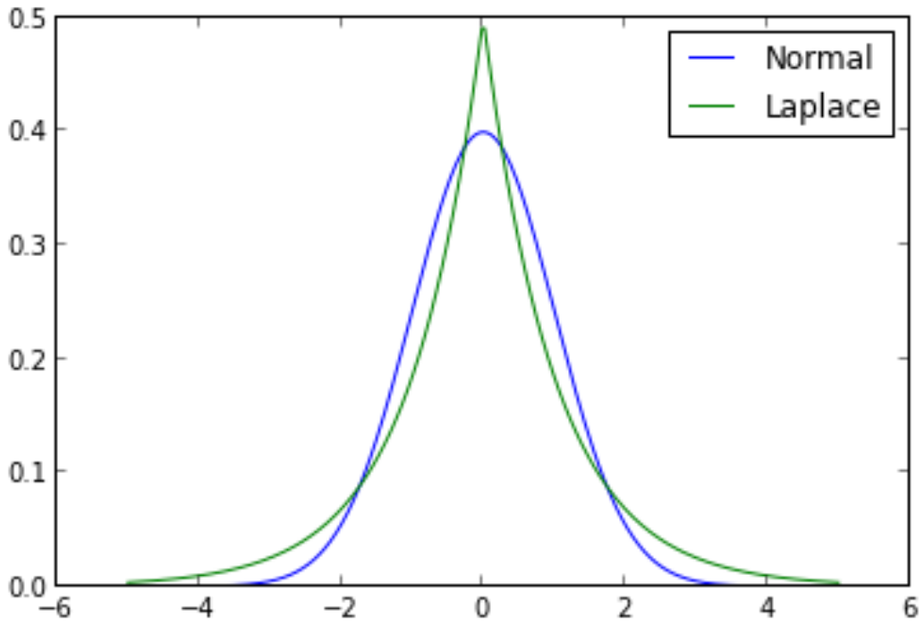


$$\|\beta\|_0 = \lim_{p \to 0} \|\beta\|_p$$

ℓ_1-norm sparsity cnt'd

Alternative explanation:

Laplace-distribution has more probability mass at zero than Gaussian.



Austin Rochford

(Here, we are not interested in its heavy tails.)

Choices of regularizers

Problem: if several features are correlated, Lasso selects only one.

→ Unstable solution, strongly affected by noise.

Choices of regularizers

Problem: if several features are correlated, Lasso selects only one.

→ Unstable solution, strongly affected by noise.

'Elastic net' (Zou, 2005): $g(\beta) = \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$

→ Selects either entire group of correlated variables or none of them.

Choices of regularizers

Problem: if several features are correlated, Lasso selects only one.

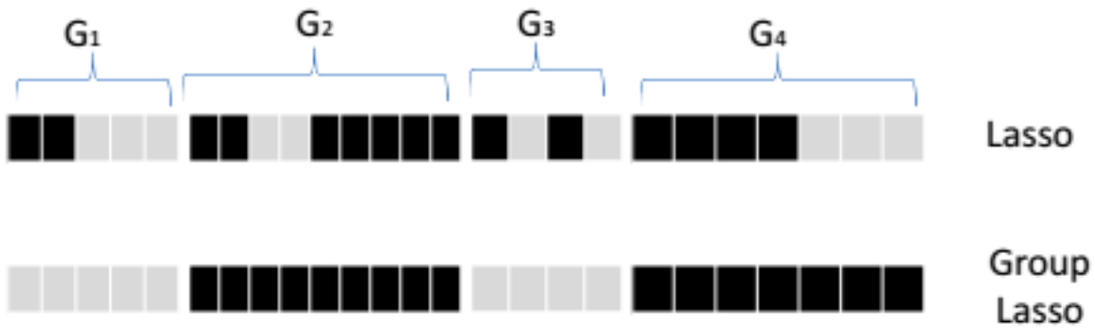
→ Unstable solution, strongly affected by noise.

'Elastic net' (Zou, 2005): $g(\beta) = \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$

→ Selects either entire group of correlated variables or none of them.

If meaningful grouping is known: Group Lasso (Yuan, 2006)

$$g(\beta) = \lambda \|\beta\|_{1,2} = \lambda \sum_{k=1}^{K} \|\beta_k\|_2^2 = \lambda \left\| \, \|\beta_1\|_2^2, \ldots, \|\beta_K\|_2^2 \, \right\|_1$$



Choices of regularizers

Tikhonov regularization: $g(\beta) = \|\Gamma\beta\|_2^2 \quad , \quad \Gamma \in \mathbb{R}^{L\times d}$

$$\hat{\beta} = (X^\top X + \Gamma^\top\Gamma)^{-1} X^\top \mathbf{y}$$

E.g., for features with a spatial structure

- Γ = Cosines/Wavelets → penalize energy at specific spatial scale
- Γ = discrete derivates → enforce general smoothness, penalize edges

Choices of regularizers

Tikhonov regularization: $g(\beta) = \|\Gamma\beta\|_2^2 \quad , \quad \Gamma \in \mathbb{R}^{L\times d}$

$$\hat{\beta} = (X^\top X + \Gamma^\top\Gamma)^{-1} X^\top \mathbf{y}$$

E.g., for features with a spatial structure

- Γ = Cosines/Wavelets → penalize energy at specific spatial scale
- Γ = discrete derivates → enforce general smoothness, penalize edges

Sparsity after transformation: $g(\beta) = \|\Gamma\beta\|_1 \quad , \quad \Gamma \in \mathbb{R}^{L\times d}$

- E.g. enforce small number of frequencies, edges, change points …

Example: total variation denoising

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \sum_i |\beta_i - \beta_{i+1}|$$

$$= \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \|\Gamma\beta\|_1 \quad , \quad \Gamma = \begin{pmatrix} 1 & -1 & 0 & \dots \\ 0 & 1 & -1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

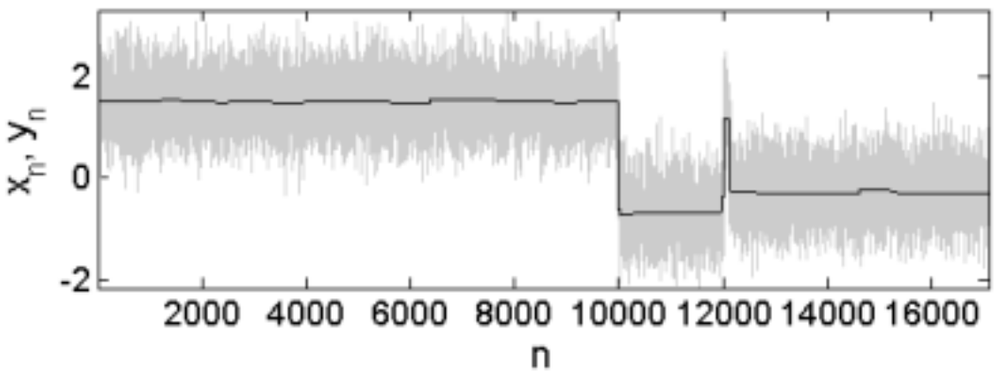
Enforces piecewise-constant solution.

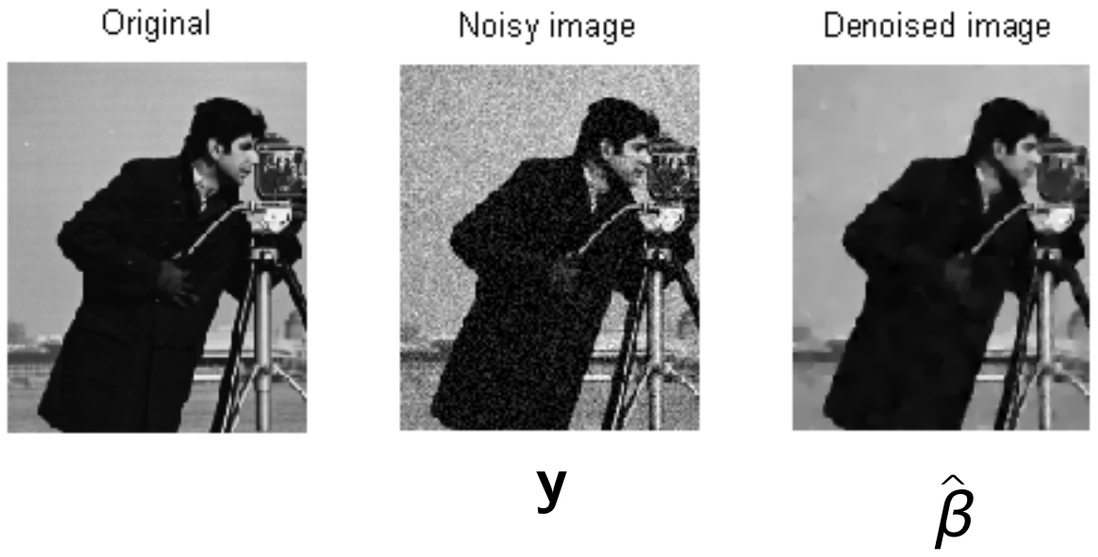
Example: total variation denoising

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \sum_i |\beta_i - \beta_{i+1}|$$

$$= \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \|\Gamma\beta\|_1 \quad , \quad \Gamma = \begin{pmatrix} 1 & -1 & 0 & \dots \\ 0 & 1 & -1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Enforces piecewise-constant solution.



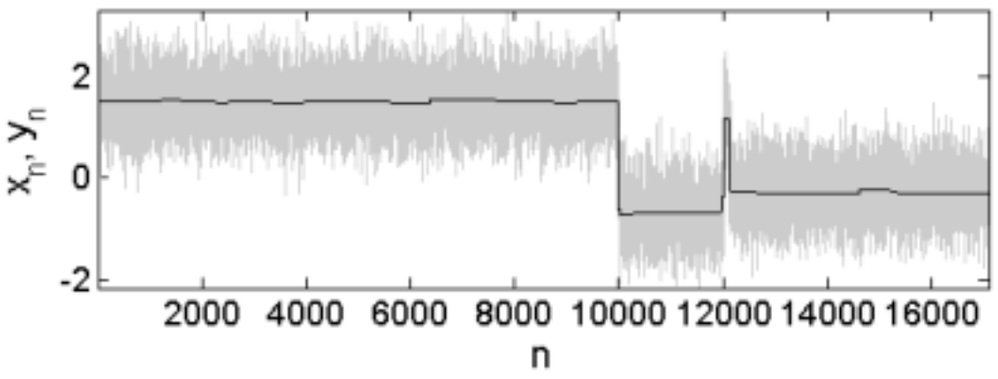


Example: total variation denoising

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \sum_i |\beta_i - \beta_{i+1}|$$

$$= \arg\min_{\beta} \|\mathbf{y} - \beta\|_2^2 + \lambda \|\Gamma\beta\|_1 \quad , \quad \Gamma = \begin{pmatrix} 1 & -1 & 0 & \dots \\ 0 & 1 & -1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Enforces piecewise-constant solution.



$\mathbf{y}$ $\hat{\beta}$

$$\Gamma = \begin{pmatrix} 1 & -2 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$: piecewise linear.

Example: EEG source reconstruction

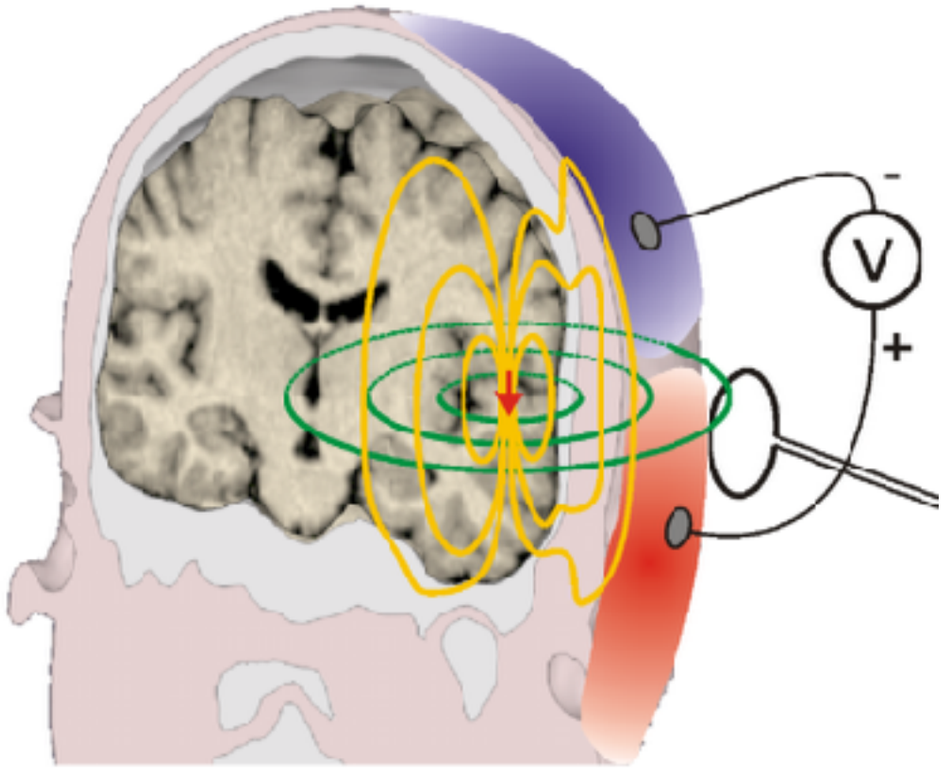
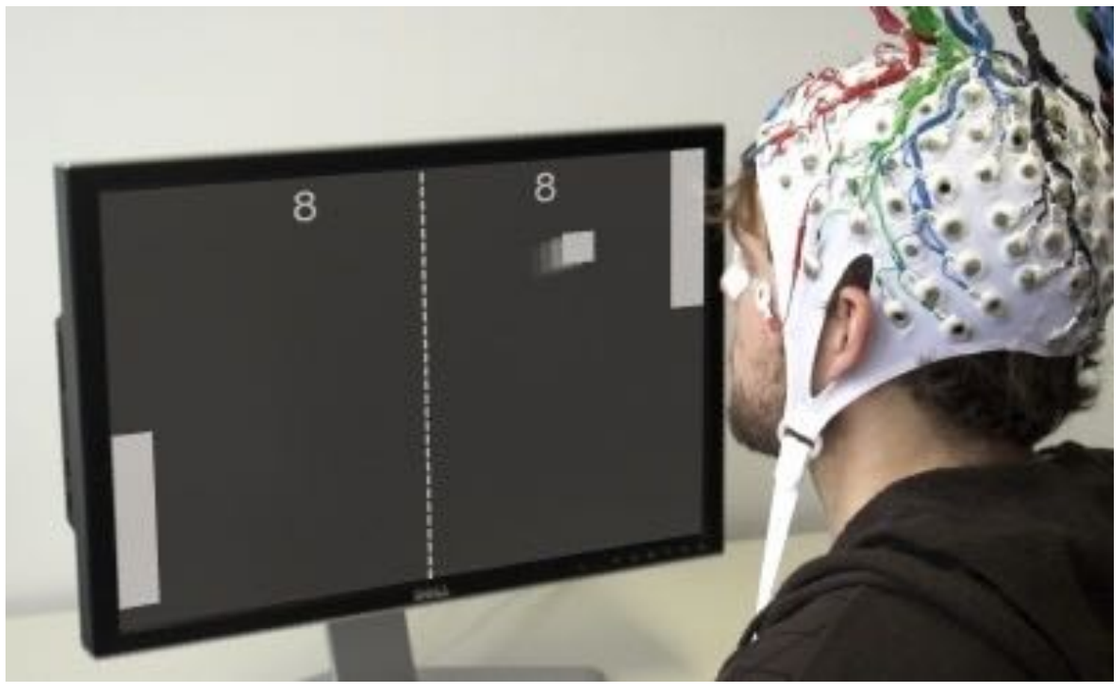


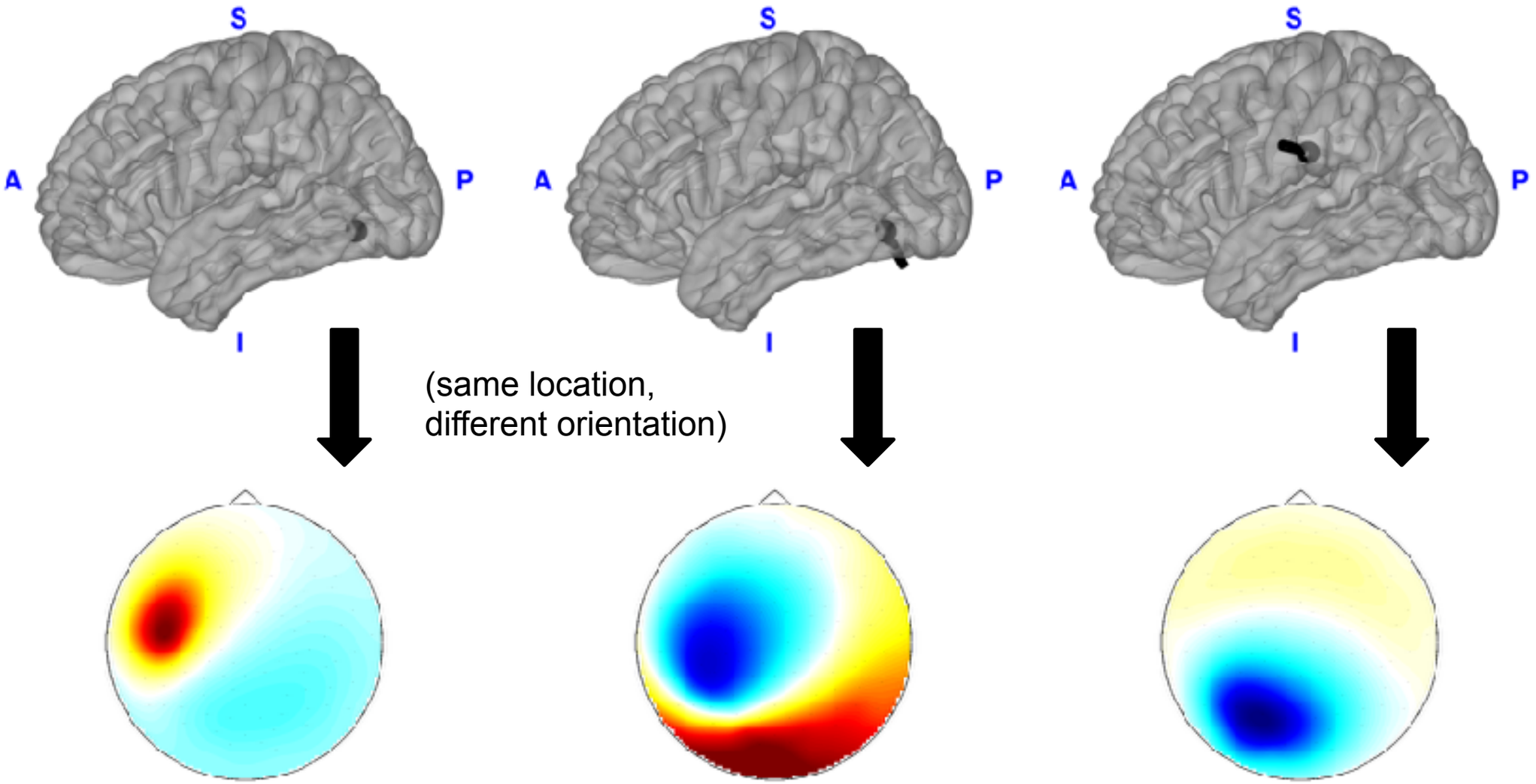
Figure by Lauri Parkkonen

~100 channels



Volume conduction

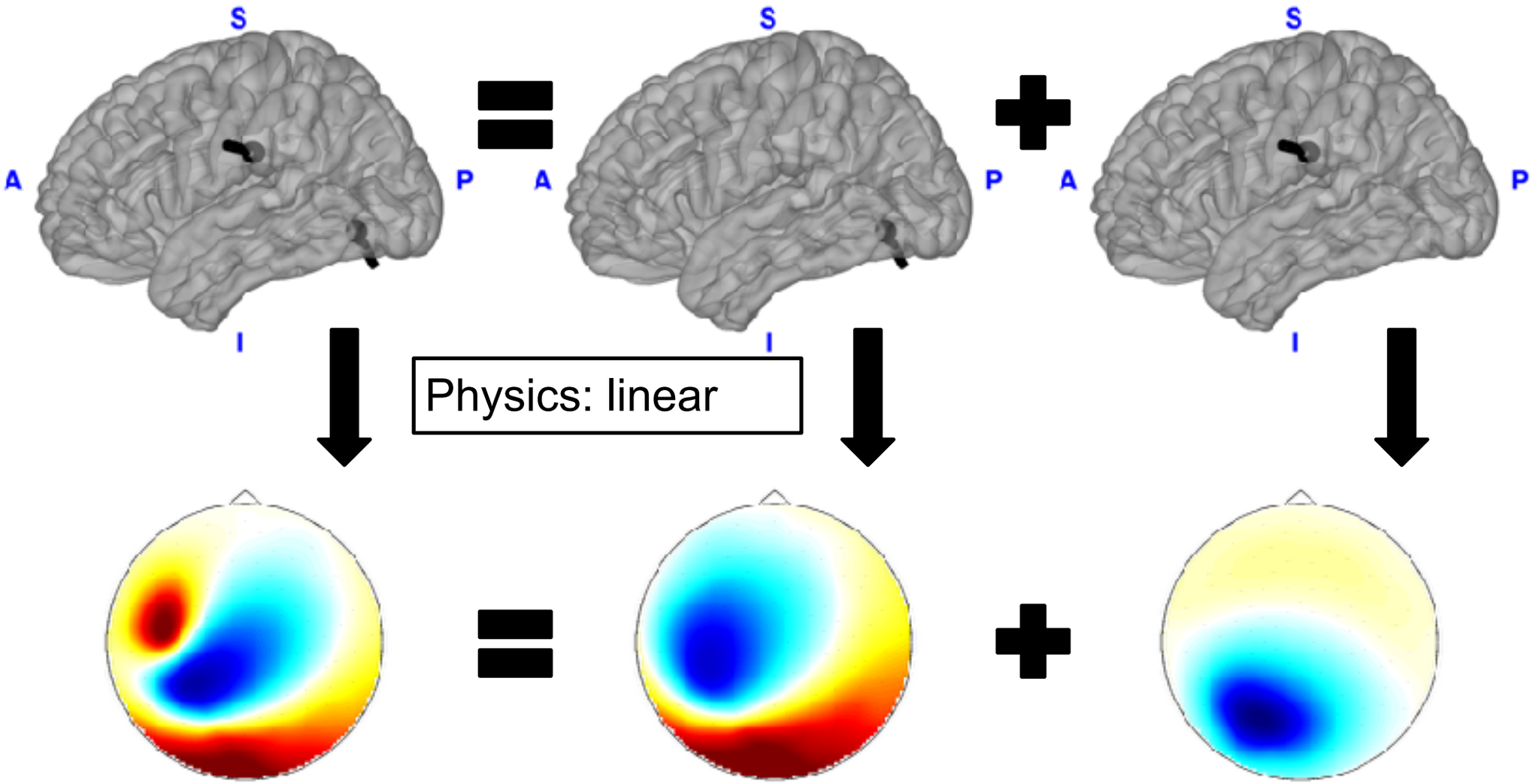
Brain source



EEG scalp potential

Volume conduction

Brain source



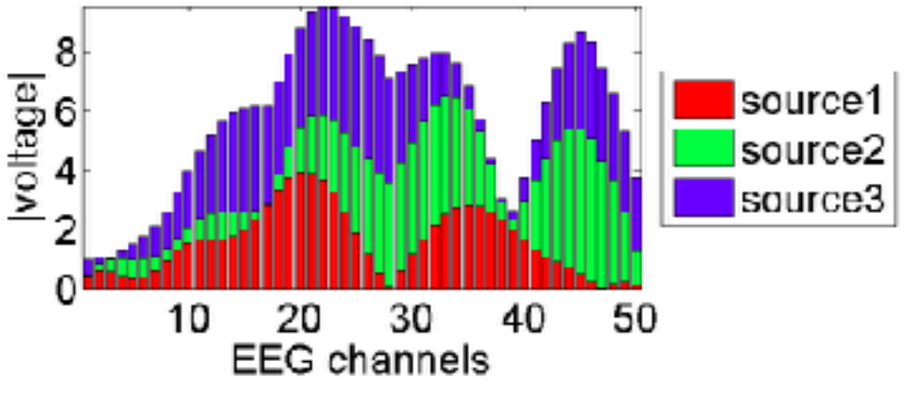
EEG scalp potential

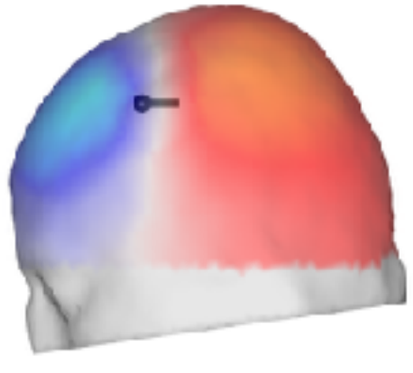
The linear forward model of EEG

EEG sensors measure linear superpositions of brain sources

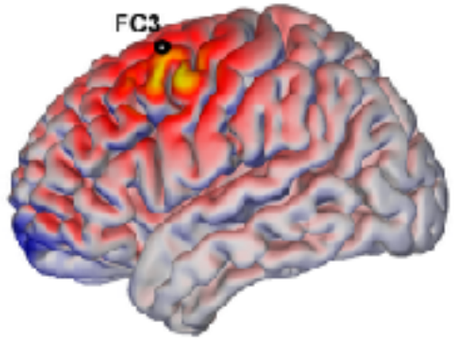
$$\mathbf{x}(t) = \mathbf{L}\mathbf{j}(t) + \boldsymbol{\epsilon}(t)$$

$\mathbf{x}(t)$	EEG scalp potential
$\mathbf{j}(t)$	Neuronal source currents
$\mathbf{L}$	Lead field (physical head model)
$\boldsymbol{\epsilon}(t)$	Noise and artifacts





column of **L**



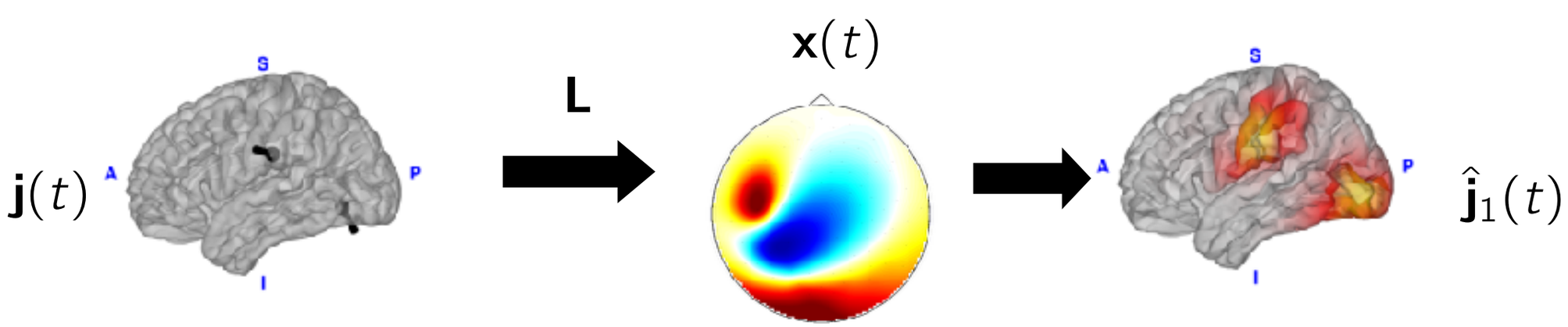
row of **L**

Inverse source reconstruction

$\mathbf{x}(t) = \mathbf{L}\mathbf{j}(t) + \boldsymbol{\epsilon}(t)$

Using known physical model $\mathbf{L}$, we can reconstruct sources $\mathbf{j}(t)$ from $\mathbf{x}(t)$.

→ Essentially a linear regression problem.



Advantages: anatomical localization/interpretation, SNR increase.

No unique solution, need subjective prior assumptions.

Spatial priors in inverse imaging

MAP approach: $\hat{\mathbf{j}}(t) = \arg\min_{\mathbf{j}} \|\mathbf{x}(t) - \mathbf{L}\mathbf{j}(t)\|_2^2 + \lambda g(\mathbf{j}(t))$

Smoothness

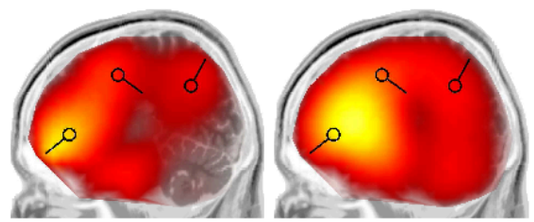
$$g(\mathbf{j}(t)) = \|\boldsymbol{\Gamma}\mathbf{j}(t)\|_2^2$$

Sparsity

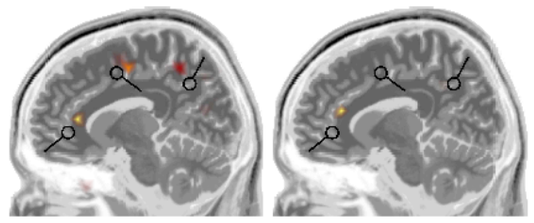
$$g(\mathbf{j}(t)) = \|\mathbf{j}(t)\|_1$$

Both

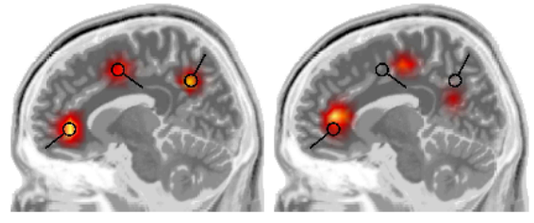
$$g(\mathbf{j}(t)) = \|\boldsymbol{\Gamma}\mathbf{j}(t)\|_2^2 + \alpha \|\mathbf{j}(t)\|_1$$



Smooth



Sparse

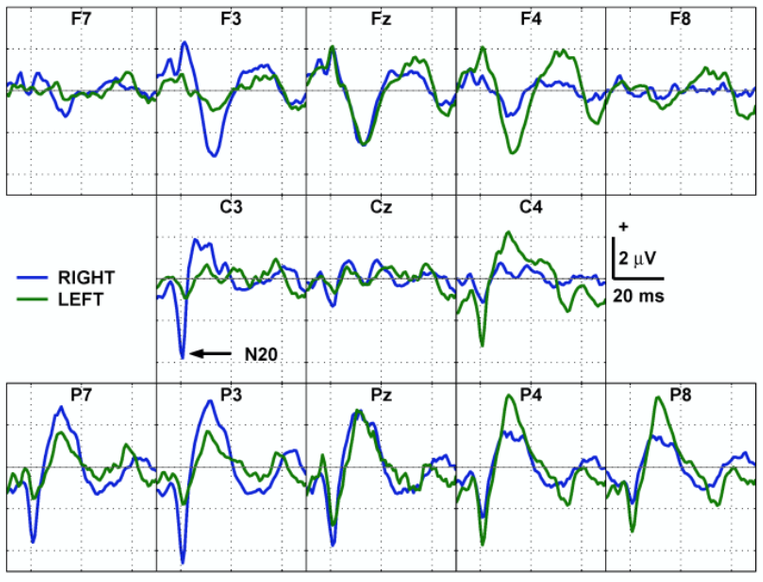


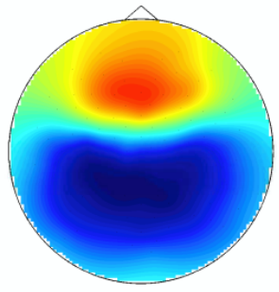
Both

Localization of hand areas in somatosensory cortex

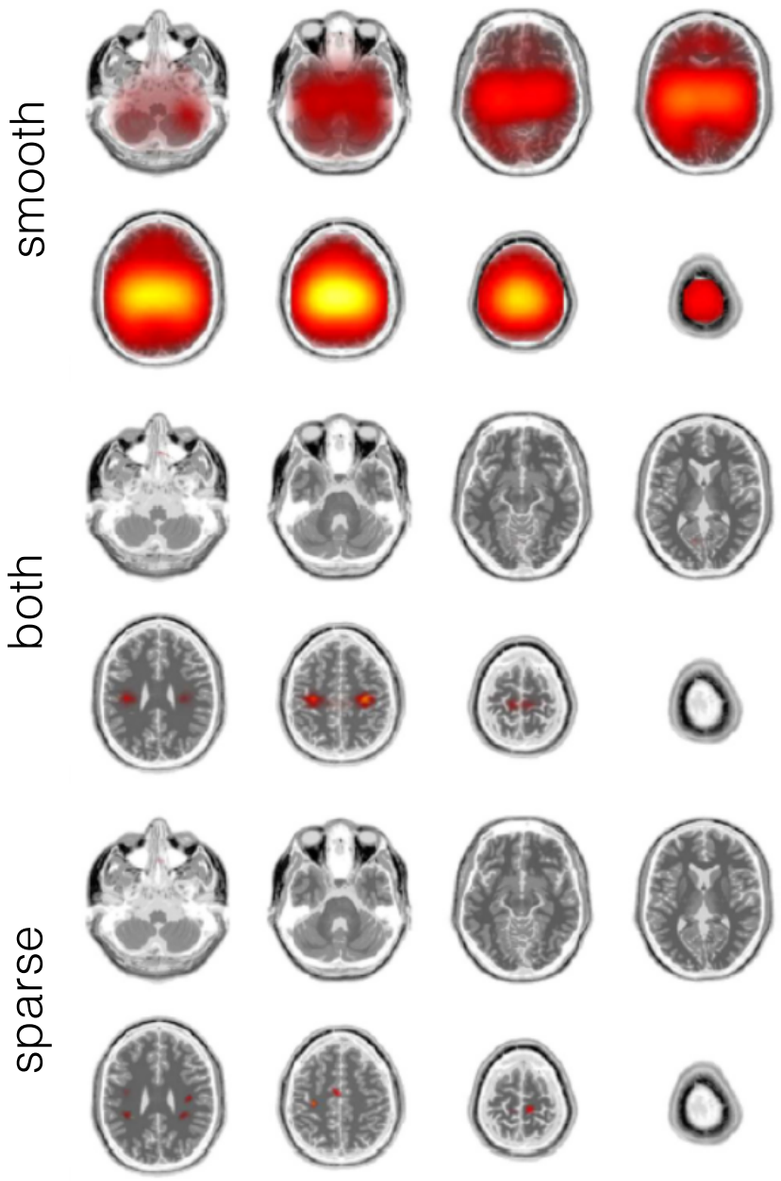
Electrical stimulation at both thumbs
(Median nerves)

→ N20 event-related potential in the EEG

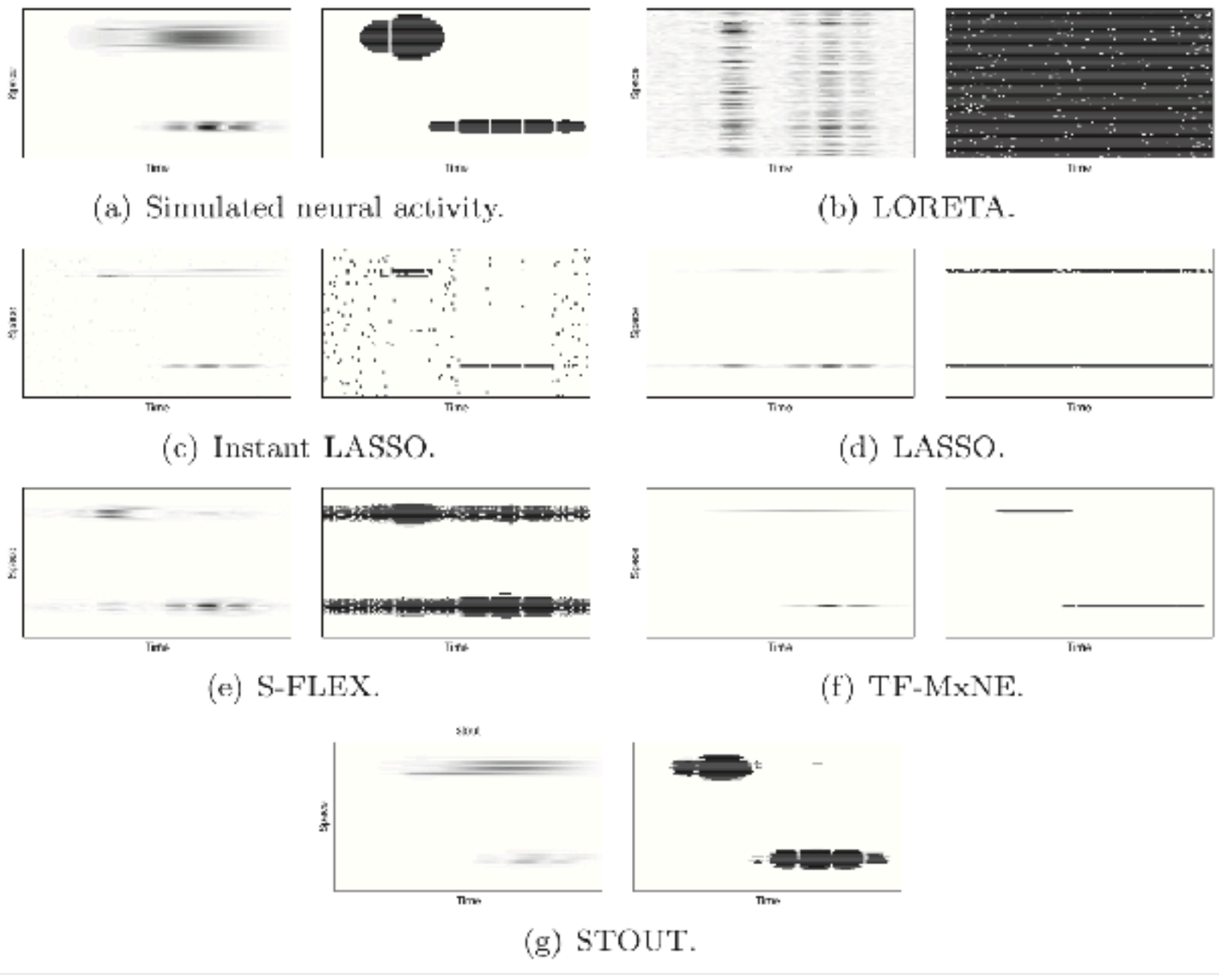




There should be two lateralized symmetric sources in the somatosensory cortices.



Reconstructing neural time series



(a) Simulated neural activity.

(b) LORETA.

(c) Instant LASSO.

(d) LASSO.

(e) S-FLEX.

(f) TF-MxNE.

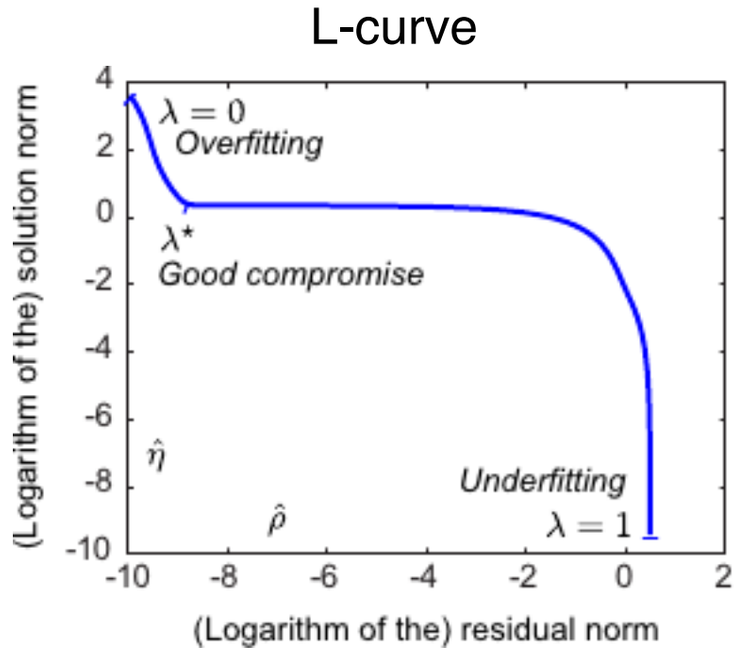
(g) STOUT.

Summary: choices of regularizers

- Choice of prior is a modeling decision just like choice of loss function, sometimes more important (especially if d > N).
- Arbitrary combinations of penalties possible and sometimes meaningful.
  - E.g. sparsity + sparse derivatives

Summary: choices of regularizers

- Choice of prior is a modeling decision just like choice of loss function, sometimes more important (especially if d > N).
- Arbitrary combinations of penalties possible and sometimes meaningful.
  - E.g. sparsity + sparse derivatives
- How to choose λ ?
  - Cross-validation
  - L-curve
  - Bayesian:
    - Compare marginal likelihood
    - ‘Integrate out’



Florent Brunet

Bayesian linear regression

Assume Gaussian likelihood and prior:

$$y_i = \mathbf{x}_i^\top \beta + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$
$$\beta \sim \mathcal{N}(0, \Sigma_\beta)$$

Bayesian linear regression

Assume Gaussian likelihood and prior:

$$
\begin{aligned}
y_i &= \mathbf{x}_i^\top \beta + \varepsilon_i \\
\varepsilon_i &\sim \mathcal{N}(0, \sigma_\varepsilon^2) \\
\beta &\sim \mathcal{N}(0, \Sigma_\beta)
\end{aligned}
$$

Bayes' rule:

$$
\begin{aligned}
p(\beta|X, \mathbf{y}) &= \frac{p(\mathbf{y}|X, \beta) \times p(\beta)}{p(\mathbf{y}|X)} \\
&= \frac{1}{Z} \prod_{i=1}^{N} \frac{1}{\sqrt{2\sigma_\varepsilon^2 \pi}} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \beta)^2}{2\sigma_\varepsilon^2}\right) \times \frac{1}{(2\pi)^{k/2}|\Sigma_\beta|^{1/2}} \exp\left(-\frac{1}{2}\beta^\top \Sigma_\beta^{-1} \beta\right)
\end{aligned}
$$

Bayesian linear regression

Posterior is also Gaussian: (similar derivations as in lecture 3)

$$p(\beta|X,\mathbf{y}) \sim \mathcal{N}\left((X^\top X + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} X^\top \mathbf{y}, (\sigma_\varepsilon^{-2} X^\top X + \Sigma_\beta^{-1})^{-1}\right)$$

Bayesian linear regression

Posterior is also Gaussian: (similar derivations as in lecture 3)

$$p(\beta|X, \mathbf{y}) \sim \mathcal{N}\left((X^\top X + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} X^\top \mathbf{y}, (\sigma_\varepsilon^{-2} X^\top X + \Sigma_\beta^{-1})^{-1}\right)$$

For Gaussians: mode = mean

→ MAP estimate: $\hat{\beta}_{\text{MAP}} = (X^\top X + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} X^\top \mathbf{y}$

- Equivalent to ridge regression/Tikhonov regularization estimate for appropriate choices of $\lambda, \Gamma, \sigma_\varepsilon^2, \Sigma_\beta$
- OLS is recovered for $|\Sigma_\beta| \to \infty$ (when prior becomes non-informative)

Bayesian linear regression

Predictive distribution is also Gaussian:

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \beta) p(\beta|X, \mathbf{y}) \mathrm{d}\beta$$

$$\sim \mathcal{N}\left(\mathbf{x}_*^\top (X^\top X + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} X^\top \mathbf{y}, \mathbf{x}_*^\top (\sigma_\varepsilon^{-2} X^\top X + \Sigma_\beta^{-1})^{-1} \mathbf{x}_*\right)$$

$$\sim \mathcal{N}\left(\mathbf{x}_*^\top \hat{\beta}_{\mathrm{MAP}}, \mathbf{x}_*^\top \Sigma_{\mathrm{post}}^{-1} \mathbf{x}_*\right)$$

Bayesian linear regression

Predictive distribution is also Gaussian:

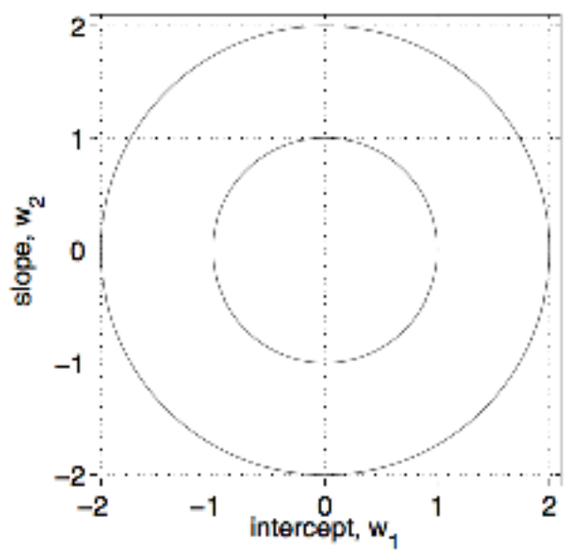
$$
\begin{aligned}
p(y_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(y_*|\mathbf{x}_*, \beta) p(\beta|X, \mathbf{y}) \mathrm{d}\beta \\
&\sim \mathcal{N}\left(\mathbf{x}_*^\top (X^\top X + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} X^\top \mathbf{y}, \mathbf{x}_*^\top (\sigma_\varepsilon^{-2} X^\top X + \Sigma_\beta^{-1})^{-1} \mathbf{x}_*\right) \\
&\sim \mathcal{N}\left(\mathbf{x}_*^\top \hat{\beta}_{\text{MAP}}, \mathbf{x}_*^\top \Sigma_{\text{post}}^{-1} \mathbf{x}_*\right)
\end{aligned}
$$

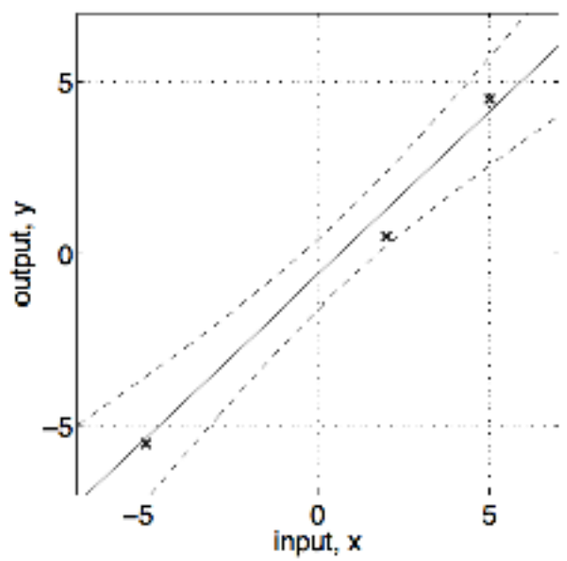
Notes:

- The mean often serves as predicted value; the variance as an estimate of confidence.
- As everything is Gaussian and linear, posterior and pred. distribution can be derived easily (as we have done in exercise 2 for OLS).

1D example

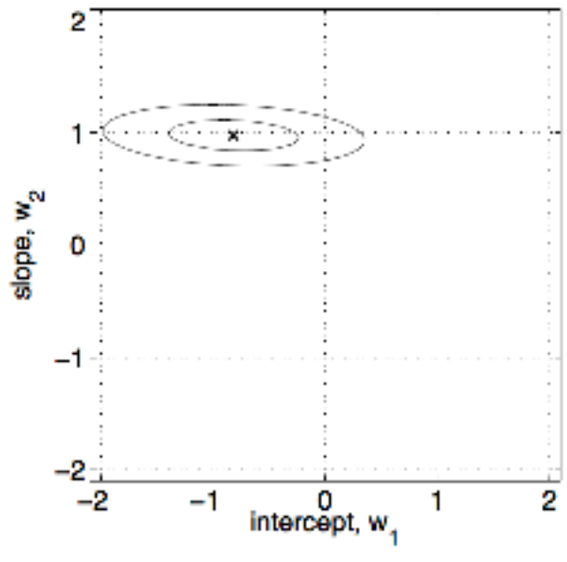
Prior

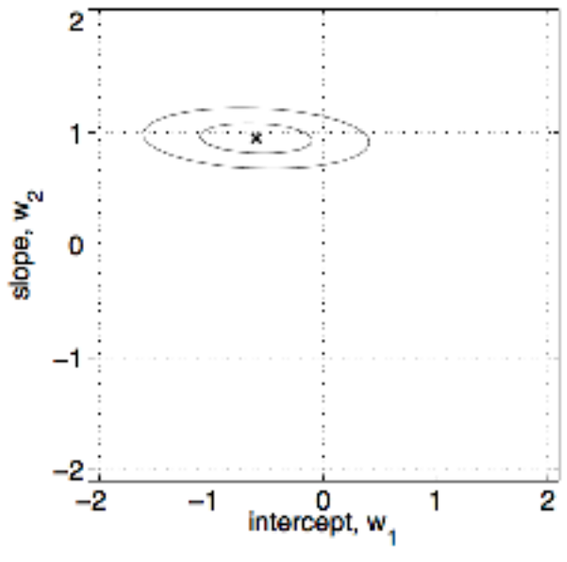




Data + predictive distribution

Likelihood



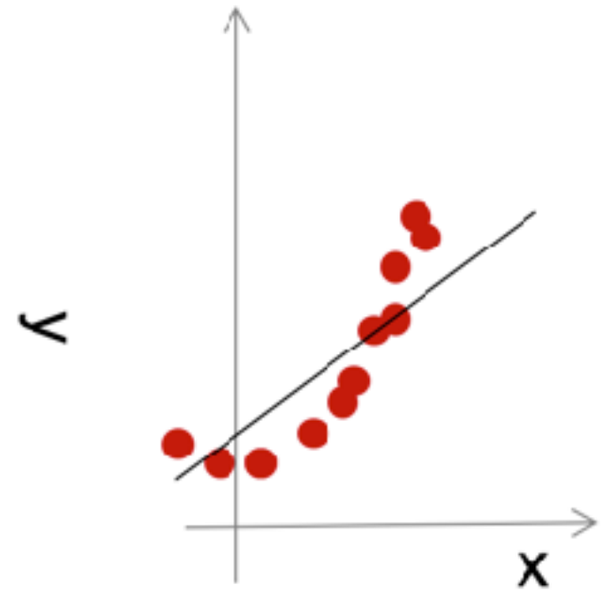


Posterior

Rasmussen and Williams

Non-linear regression

Problem: if the relationship between input and output **not linear**, the linear model gives poor predictions.



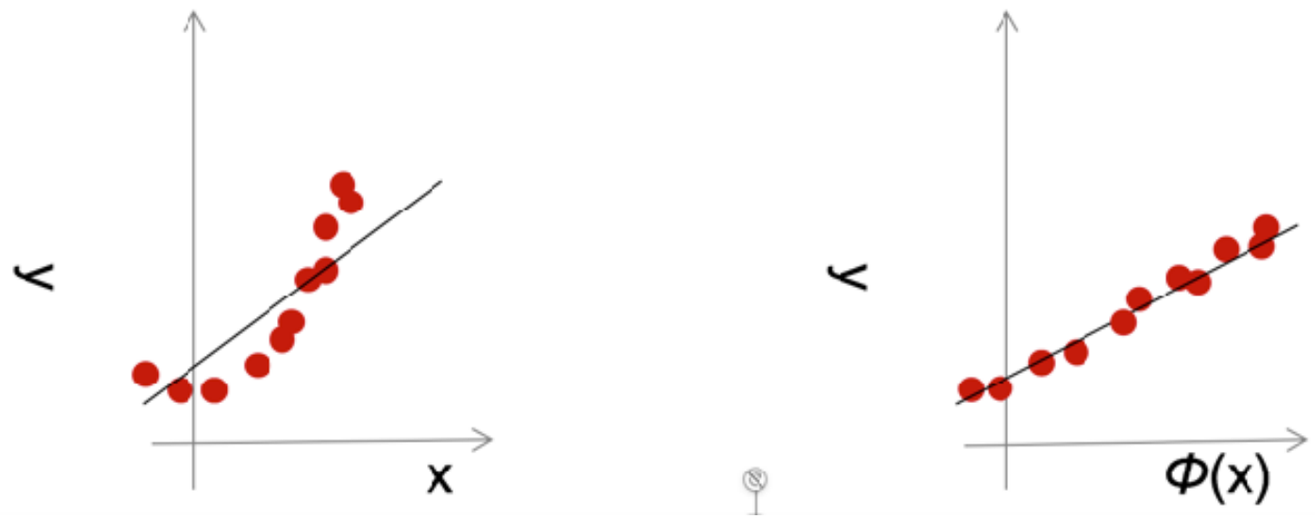
→ **Need to consider general non-linear model** $y_i = f(\mathbf{x}_i) + \varepsilon_i$ **.**

(Previously $f(\mathbf{x}_i) = \mathbf{x}_i^\top \beta$)

Non-linear embedding

Simple way to introduce non-linearity: non-linear embedding $\beta \rightarrow \phi(\beta)$

Project data via non-linear function Φ into a feature space where the relationship between input and output is approximately linear function.



E.g. polynomial expansion $\phi([x_1, x_2, 1]^\top) = [x_1, x_2, x_1^2, x_2^2, x_1 x_2, 1]^\top$

Note: last dimension determines intercept.

Non-linear regression

Resulting model is non-linear in the data but linear in the parameters:

$$y_i = \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\beta \sim \mathcal{N}(0, \Sigma_\beta) \quad , \quad \beta \in \mathbb{R}^{d_\phi}$$

$$\rightarrow$$

Non-linear regression

Resulting model is non-linear in the data but linear in the parameters:

$$y_i = \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$
$$\beta \sim \mathcal{N}(0, \Sigma_\beta) \quad , \quad \beta \in \mathbb{R}^{d_\phi}$$

→ Can be estimated using any of the methods discussed above by exchanging $\mathbf{x}$ with $\phi(\mathbf{x})$.

Non-linear regression

Resulting model is non-linear in the data but linear in the parameters:

$$y_i = \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$
$$\beta \sim \mathcal{N}(0, \Sigma_\beta) \quad , \quad \beta \in \mathbb{R}^{d_\phi}$$

→ Can be estimated using any of the methods discussed above by exchanging $\mathbf{x}$ with $\phi(\mathbf{x})$.

- Ridge Regression: $\hat{\beta} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top \mathbf{y}$

Non-linear regression

Resulting model is non-linear in the data but linear in the parameters:

$$y_i = \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i$$
$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$
$$\beta \sim \mathcal{N}(0, \Sigma_\beta) \quad , \quad \beta \in \mathbb{R}^{d_\phi}$$

→ Can be estimated using any of the methods discussed above by exchanging $\mathbf{x}$ with $\phi(\mathbf{x})$.

- Ridge Regression: $\hat{\beta} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top \mathbf{y}$
- Bayesian predictive distribution:

$$p(y_* | \mathbf{x}_*, X, \mathbf{y}) \sim \mathcal{N}\left(\phi_*^\top (\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} \Phi^\top \mathbf{y}, \phi_*^\top (\sigma_\varepsilon^{-2} \Phi^\top \Phi + \Sigma_\beta^{-1})^{-1} \phi_*\right)$$

where $\Phi = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N))^\top \in \mathbb{R}^{N \times d_\phi}$, $\phi_* = \phi(\mathbf{x}_*) \in \mathbb{R}^{d_\phi}$.

Kernel regression

Sometimes, it may be easier to find a kernel (similarity measure)

$k(\mathbf{x}_i, \mathbf{x}_j)$ than a non-linear embedding $\phi(\mathbf{x})$.

Kernel regression

Sometimes, it may be easier to find a kernel (similarity measure)

$k(\mathbf{x}_i, \mathbf{x}_j)$ than a non-linear embedding $\phi(\mathbf{x})$.

Perhaps not surprisingly, we can kernelize the non-linear model.

Kernel regression

Sometimes, it may be easier to find a kernel (similarity measure)

$k(\mathbf{x}_i, \mathbf{x}_j)$ than a non-linear embedding $\phi(\mathbf{x})$.

Perhaps not surprisingly, we can kernelize the non-linear model.

Predictive distribution:

$$
\begin{aligned}
p(y_*|\mathbf{x}_*, X, \mathbf{y}) &\sim \mathcal{N}\left(\phi_*^\top(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})^{-1}\Phi^\top\mathbf{y},\, \phi_*^\top(\sigma_\varepsilon^{-2}\Phi^\top\Phi + \Sigma_\beta^{-1})^{-1}\phi_*\right) \\
&\sim \mathcal{N}\big(\phi_*^\top\Sigma_\beta\Phi^\top(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}, \\
&\qquad \phi_*^\top\Sigma_\beta\phi_* - \phi_*^\top\Sigma_\beta\Phi^\top(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}\Phi\Sigma_\beta\phi_*\big) \\
&\sim \mathcal{N}\left(\mathbf{k}_*^\top(K + \sigma_\varepsilon^2 I)^{-1}\mathbf{y},\, k_{*,*} - \mathbf{k}_*^\top(K + \sigma_\varepsilon^2 I)^{-1}\mathbf{k}_*\right) \ ,
\end{aligned}
$$

where $\mathbf{k}_*^\top = \phi_*^\top\Sigma_\beta\Phi^\top \in \mathbb{R}^N$, $K = \Phi\Sigma_\beta\Phi^\top \in \mathbb{R}^{N\times N}$, $k_{*,*} = \phi_*^\top\Sigma_\beta\phi_* \in \mathbb{R}$.

Proof (mean)

$$\boldsymbol{\phi}_*^\top(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})^{-1}\Phi^\top\mathbf{y}$$

$$= \boldsymbol{\phi}_*^\top(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})^{-1}\Phi^\top\overbrace{(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}}^{I}\mathbf{y}$$

$$= \boldsymbol{\phi}_*^\top(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})^{-1}(\Phi^\top\Phi\Sigma_\beta\Phi^\top + \Phi^\top\sigma_\varepsilon^2 I)(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}$$

$$= \boldsymbol{\phi}_*^\top\underbrace{(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})^{-1}(\Phi^\top\Phi + \sigma_\varepsilon^2\Sigma_\beta^{-1})}_{I}\Sigma_\beta\Phi^\top(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}$$

$$= \boldsymbol{\phi}_*^\top\Sigma_\beta\Phi^\top(\Phi\Sigma_\beta\Phi^\top + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}$$

$$= \mathbf{k}_*^\top(K + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}$$

Proof (mean)

$$
\begin{aligned}
&\boldsymbol{\phi}_*^\top (\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} \Phi^\top \mathbf{y} \\
&= \boldsymbol{\phi}_*^\top (\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} \Phi^\top \overbrace{(\Phi \Sigma_\beta \Phi^\top + \sigma_\varepsilon^2 I)(\Phi \Sigma_\beta \Phi^\top + \sigma_\varepsilon^2 I)^{-1}}^{I} \mathbf{y} \\
&= \boldsymbol{\phi}_*^\top (\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} (\Phi^\top \Phi \Sigma_\beta \Phi^\top + \Phi^\top \sigma_\varepsilon^2 I)(\Phi \Sigma_\beta \Phi^\top + \sigma_\varepsilon^2 I)^{-1} \\
&= \boldsymbol{\phi}_*^\top \underbrace{(\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})^{-1} (\Phi^\top \Phi + \sigma_\varepsilon^2 \Sigma_\beta^{-1})}_{I} \Sigma_\beta \Phi^\top (\Phi \Sigma_\beta \Phi^\top + \sigma_\varepsilon^2 I)^{-1} \mathbf{y} \\
&= \boldsymbol{\phi}_*^\top \Sigma_\beta \Phi^\top (\Phi \Sigma_\beta \Phi^\top + \sigma_\varepsilon^2 I)^{-1} \mathbf{y} \\
&= \mathbf{k}_*^\top (K + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}
\end{aligned}
$$

Proof for variance: see Rasmussen and Williams, 2006.

Kernel regression

Notes:

- $K_{i,j} = \phi(\mathbf{x}_i)^\top \Sigma_\beta \phi(\mathbf{x}_j) = \tilde{\phi}(\mathbf{x}_i)^\top \tilde{\phi}(\mathbf{x}_j)$, where $\tilde{\phi}(\mathbf{x}) = \Sigma_\beta^{1/2} \phi(\mathbf{x})$ defines a valid kernel.

Kernel regression

Notes:

- $K_{i,j} = \phi(\mathbf{x}_i)^\top \Sigma_\beta \phi(\mathbf{x}_j) = \tilde{\phi}(\mathbf{x}_i)^\top \tilde{\phi}(\mathbf{x}_j)$, where $\tilde{\phi}(\mathbf{x}) = \Sigma_\beta^{1/2} \phi(\mathbf{x})$ defines a valid kernel.
- The MAP estimate $\hat{y}_* = \mathbf{k}_*^\top (K + \sigma_\varepsilon^2 I)^{-1} \mathbf{y} = \sum_{i=1}^{N} a_i k(\mathbf{x}_*, \mathbf{x}_i)$ is called the **Kernel ridge regression** estimate.
- Solution lies in the subspace of $\mathbb{R}^{d_\phi}$ spanned by the N data points.

Kernel regression

Notes:

- $K_{i,j} = \phi(\mathbf{x}_i)^\top \Sigma_\beta \phi(\mathbf{x}_j) = \tilde{\phi}(\mathbf{x}_i)^\top \tilde{\phi}(\mathbf{x}_j)$, where $\tilde{\phi}(\mathbf{x}) = \Sigma_\beta^{1/2} \phi(\mathbf{x})$ defines a valid kernel.
- The MAP estimate $\hat{y}_* = \mathbf{k}_*^\top (K + \sigma_\varepsilon^2 I)^{-1} \mathbf{y} = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_*, \mathbf{x}_i)$ is called the **Kernel ridge regression** estimate.
- Solution lies in the subspace of $\mathbb{R}^{d_\phi}$ spanned by the N data points.
- In the **kernel (=dual)** formulation, K is of size $N \times N$ (not $d_\phi \times d_\phi$).

→ Dual formulation is beneficial if $d_\phi > N$,

even in the linear case $\phi(\mathbf{x}) = \mathbf{x}$!

Gaussian Processes

Alternative view for kernel regression.

Stochastic process: generalization of a multivariate random variable to infinite dimensions = functions.

Gaussian Processes

Alternative view for kernel regression.

Stochastic process: generalization of a multivariate random variable to infinite dimensions = functions.

Gaussian process (GP): a random function, for which any finite subset of values has a multivariate Gaussian distribution.

A GP f(**x**) is fully specified by

- A mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$.
- A covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))\,(f(\mathbf{x}') - m(\mathbf{x}'))\right]$.

Notation: $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$.

Gaussian Processes

The Gaussian prior over regression weights β in the Bayesian (non-)linear model can be seen as a GP prior over the functions $f(\mathbf{x}) = \phi(\mathbf{x})^\top \beta$.

$$y_i = \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i$$
$$\beta \sim \mathcal{N}(0, \Sigma_\beta)$$

$\rightarrow$

Gaussian Processes

The Gaussian prior over regression weights β in the Bayesian (non-)linear model can be seen as a GP prior over the functions $f(\mathbf{x}) = \phi(\mathbf{x})^\top \beta$.

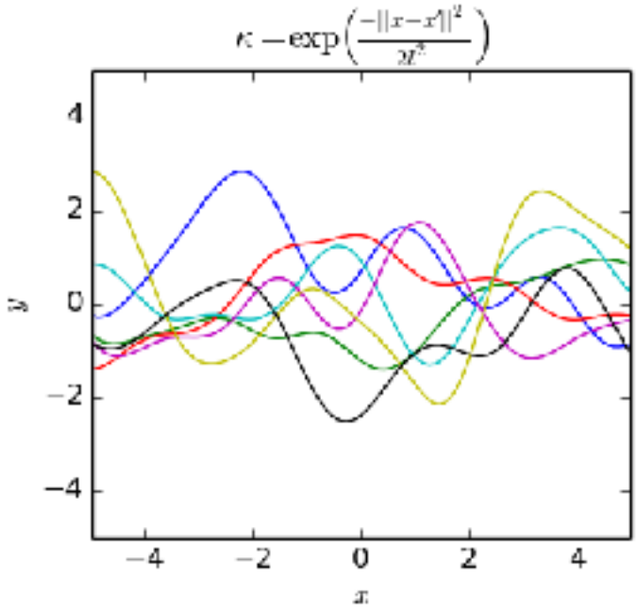
$$
\begin{aligned}
y_i &= \phi(\mathbf{x}_i)^\top \beta + \varepsilon_i \\
\beta &\sim \mathcal{N}(0, \Sigma_\beta)
\end{aligned}
$$

$$
\begin{aligned}
\rightarrow \qquad f(\mathbf{x}) &\sim \mathcal{GP}\left(\phi(\mathbf{x})^\top \mathbb{E}[\beta], \phi(\mathbf{x})^\top \mathbb{E}[\beta\beta^\top]\phi(\mathbf{x}'))\right) \\
&\sim \mathcal{GP}\left(0, \phi(\mathbf{x})^\top \Sigma_\beta \phi(\mathbf{x}')\right) \\
&\sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right)
\end{aligned}
$$

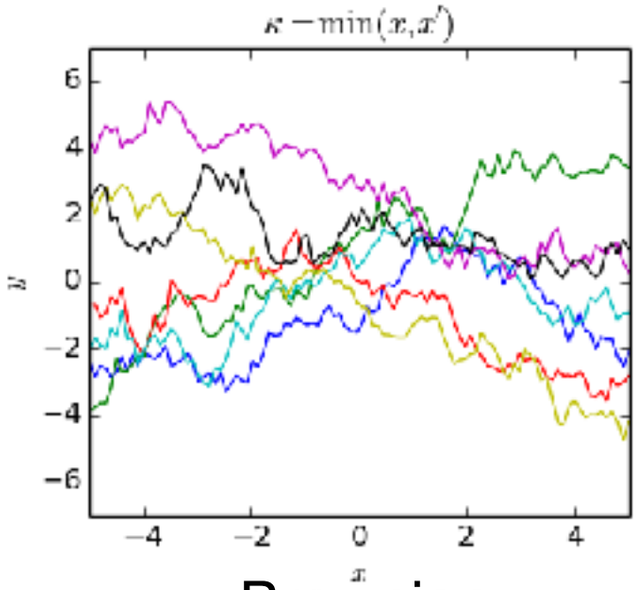
Per Mercer's definition, every kernel defines a valid covariance function.

GP covariance functions

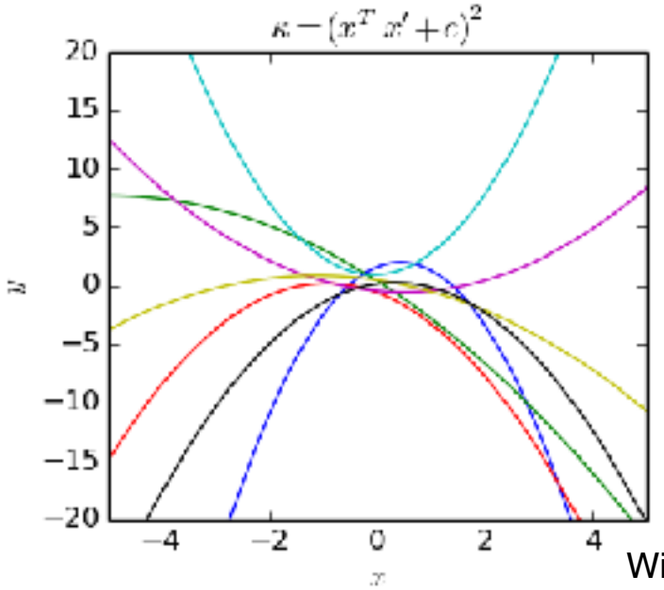
Assume zero-mean GPs. Then the covariance function alone defines what functions have high prior probability.



RBF



Brownian motion



Quadratic

Wikipedia

Most kernels have a parameter for adjusting the characteristic length scale of the variation (e.g., l in RBF).

Kernels

Specialized kernels have been invented to compare

- Text strings
- E-mail/network traffic metadata
- DNA segments
- Molecule descriptors
- Trees (e.g. language)
- Images
- Distributions

→ No necessity to embed discrete structures into $\mathbb{R}^d$.

Kernels

Specialized kernels have been invented to compare

- Text strings
- E-mail/network traffic metadata
- DNA segments
- Molecule descriptors
- Trees (e.g. language)
- Images
- Distributions

→ No necessity to embed discrete structures into $\mathbb{R}^d$.

→ ML II lecture

GP prediction

Joint distribution of observed and unobserved output variables is Gaussian.

$$p\left(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix}\middle| X, \mathbf{x}_*\right) \sim \mathcal{N}\left(0, \begin{bmatrix}K+\sigma_\varepsilon^2 I & \mathbf{k}_*\\ \mathbf{k}_*^\top & k_{*,*}\end{bmatrix}\right)$$

Predictive distribution is obtained by conditioning on the observed outputs:

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \frac{p\left(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix}\middle| X, \mathbf{x}_*\right)}{p(\mathbf{y}|X)} \text{ , where } p(\mathbf{y}|X) \sim \mathcal{N}(0, K) \text{ .}$$

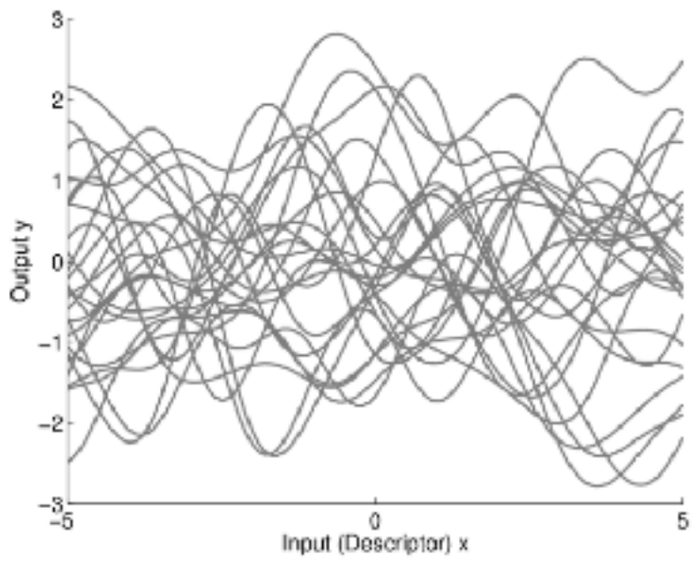
Leads to the same predictive distribution as Bayesian kernel regression:

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) \sim \mathcal{N}\left(\mathbf{k}_*^\top (K+\sigma_\varepsilon^2 I)^{-1}\mathbf{y}, k_{*,*} - \mathbf{k}_*^\top (K+\sigma_\varepsilon^2 I)^{-1}\mathbf{k}_*\right) \textbf{ .}$$

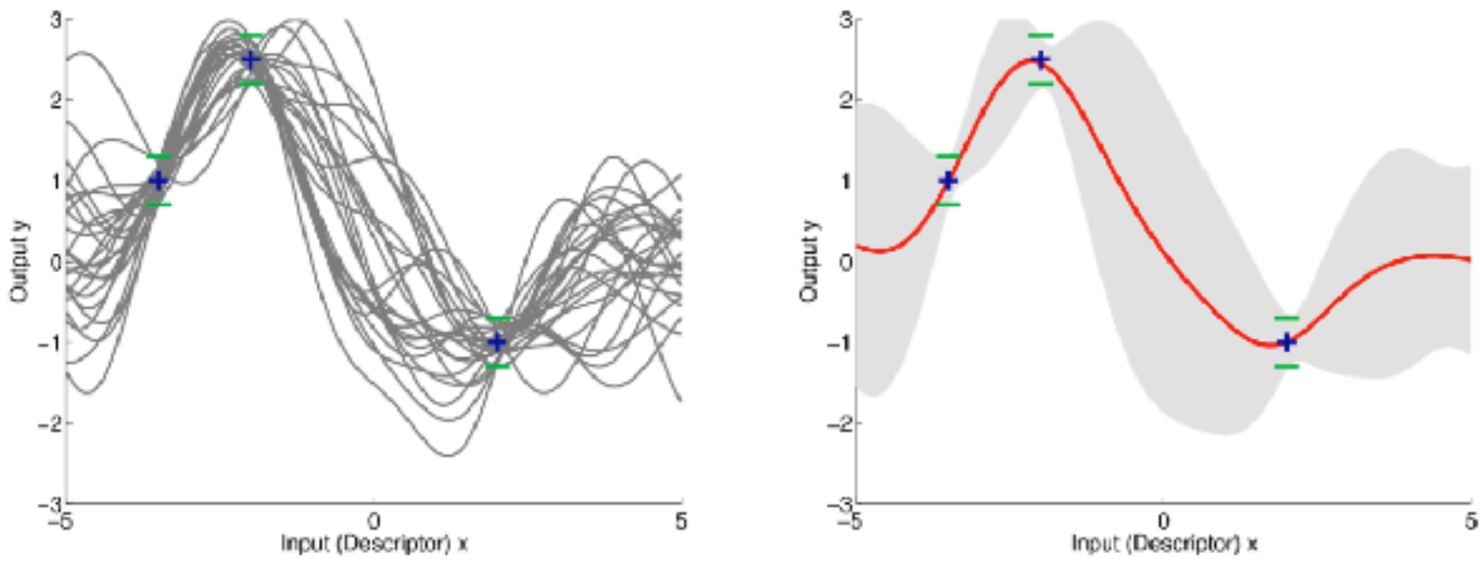
GP prediction

Interpretation:

- Posterior (implicit) discards those functions from the prior that are inconsistent with the observed data.
- Predictive distribution is posterior evaluated at test input.



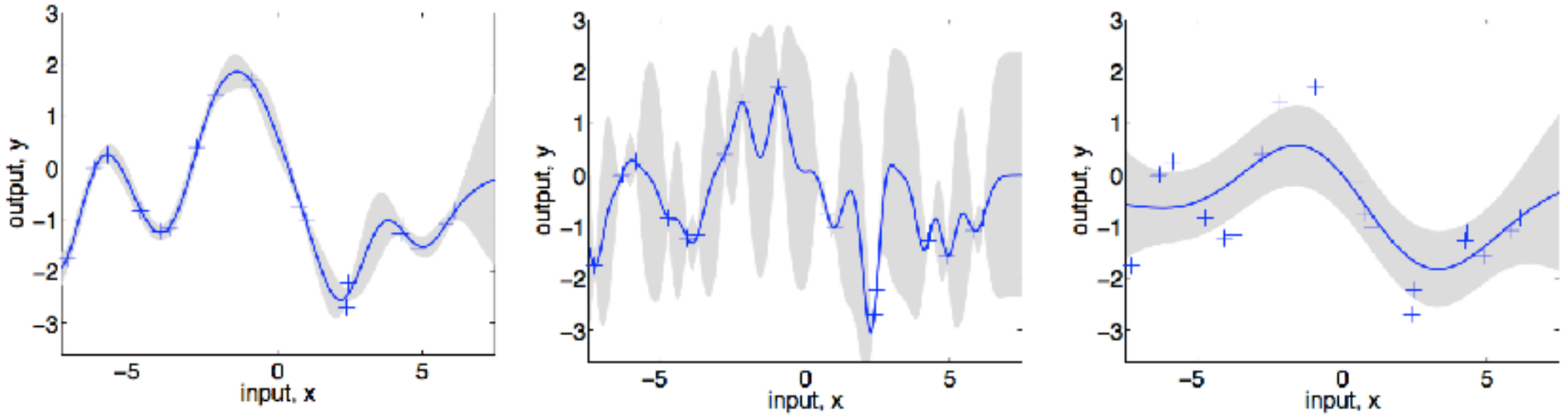
prior



posterior/predictive distribution

Length scales

Example: RBF kernels with different length scales.



(a), $\ell = 1$ (b), $\ell = 0.3$ (c), $\ell = 3$

Rasmussen and Williams

Figure 2.5: (a) Data is generated from a GP with hyperparameters $(\ell, \sigma_f, \sigma_n) = (1, 1, 0.1)$, as shown by the + symbols. Using Gaussian process prediction with these hyperparameters we obtain a 95% confidence region for the underlying function f (shown in grey). Panels (b) and (c) again show the 95% confidence region, but this time for hyperparameter values $(0.3, 1.08, 0.00005)$ and $(3.0, 1.16, 0.89)$ respectively.

Summary

Design choices in regression

- Noise model / loss function $p(\varepsilon)/l_\beta(\varepsilon)$.
  - Gaussian/quadratic simplifies calculations, but others more robust.
- Prior/regularizer $p(\beta)/g(\beta)$.
  - Again Gaussian/Tikhonov simple, but cannot achieve sparsity.
- MAP vs. predictive distribution.
  - Analytic predictive distribution again only in Gaussian case.
- Linear or non-linear.
  - Choice of non-linear embedding ϕ or choice of kernel k.
- Feature space or sample space (primal or dual) formulation.

Summary

- Choices should be made based on data properties
  - Outliers?
  - Non-linearities expected?
  - How many dimensions?
  - How many training/test points?

Summary

- Choices should be made based on data properties
  - Outliers?
  - Non-linearities expected?
  - How many dimensions?
  - How many training/test points?

Literature

Gaussian Processes for Machine Learning. Carl Edward Rasmussen and Christopher K. I. Williams, MIT Press, 2006.

http://www.gaussianprocess.org/gpml