Behandelte Themen

0. „Motivation“: Lernen in Statistik und Biologie

1. Überblick über statistische Datenmodellierungs-Verfahren

2. Das lineare Modell (Regression)

3. Perceptron und Multilagen-Perceptron (Funktionsapproximation)

4. Selbstorganisierende Merkmalskarten (Dichteschätzung)

5. Lernen von Datenmodellen

- Approximation: Bayes‘sches Schließen, MAP, ML
- Maximum Likelihood Schätzung und Fehlerminimierung
- Generalisierung und Regularisierung
- Optimierungsverfahren

6. Bayesianische Netze (Dichteschätzung und Funktionsapproximation)

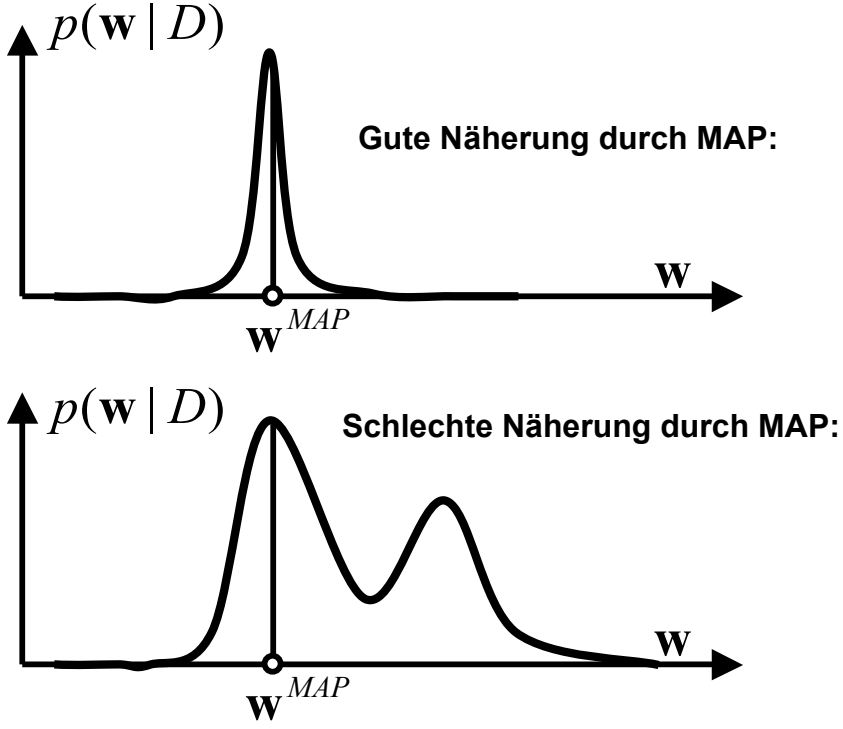
7. Kern-Trick und Support Vector Machine

Maximum a Posteriori (MAP) Schätzer:

- **Marginalisierung über Modelle oft schwierig**
- **Näherung: Verwende das Modell mit der größten a-posteriori-Wahrscheinlichkeit**

$$\mathbf{w}^{MAP} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid D)$$

- **Gute Näherung bei konzentriertem, symmetrischem posterior (=> viele Daten)**
- **Übergang zu frequentistischer Sicht (relative Häufigkeiten, ein wahres Modell)**



MAP Lernen: Maximiere posterior $p(D \mid \mathbf{w})\, p(\mathbf{w}) = \max$

Äquivalent : Minimiere -log posterior

$$-\ln p(D \mid \mathbf{w}) - \ln p(\mathbf{w}) + const = \min$$

⇧ ⇧

Später: Risikominimierung Regularisierung

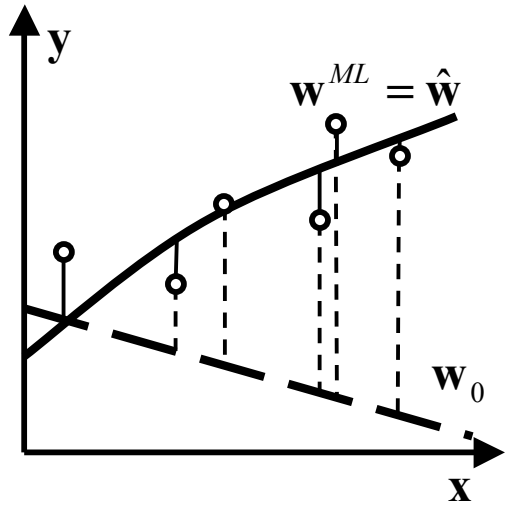
- **Nur ein Modell, aber das muss man finden! => „Optimierung“**

Maximum-Likelihood und Fehlerminimierung

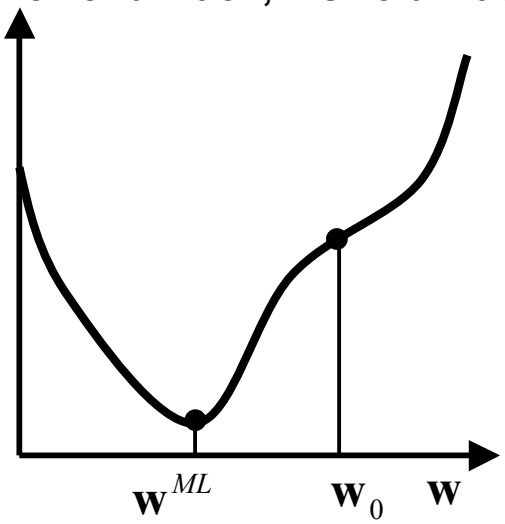
Betrachte überwachtes Lernproblem (Regression, Klassifikation)

- **<u>Likelihood:</u> Gegeben ein Rauschmodell $p_n(\mathbf{n})$, wie wahrscheinlich ist eine Abweichung des tatsächlichen vom vorhergesagten Output?**
- **<u>Negative log-Likelihood:</u> Wie unwahrscheinlich ist die Abweichung**
  Also: Wie groß ist der Fehler, den das Modell macht?

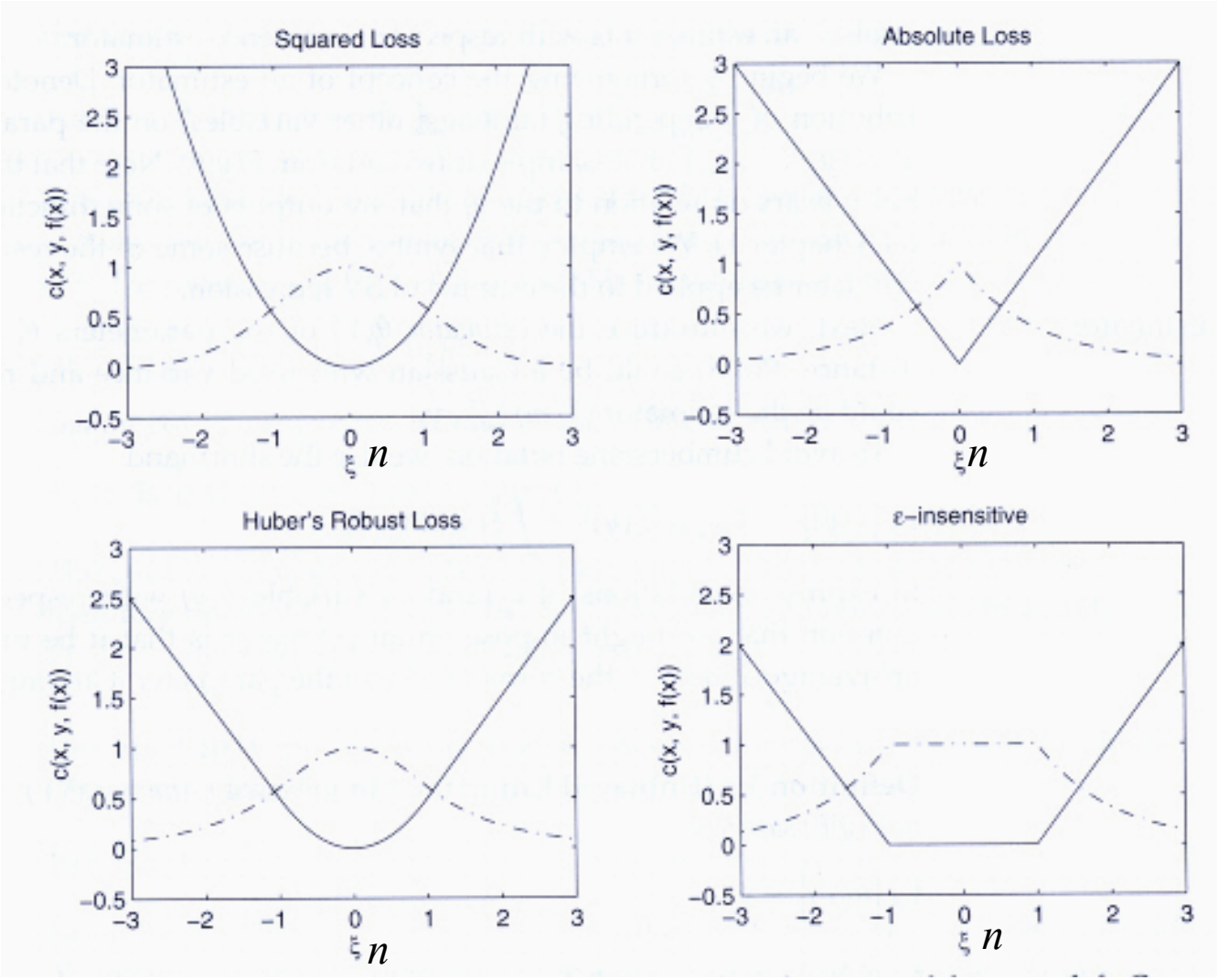
- **Teilweise synonym (beobachtet):**
  ++ negative log-Likelihood
  ++ neg. log. Rauschdichte
  -- „Risikofunktion“
  -- „Fehlerfunktion“
  -- „Verlustfunktion“
  -- „Energiefunktion“
- **Anpassen („Fitten“) eines Modells durch Risiko/Fehlerminimierung**

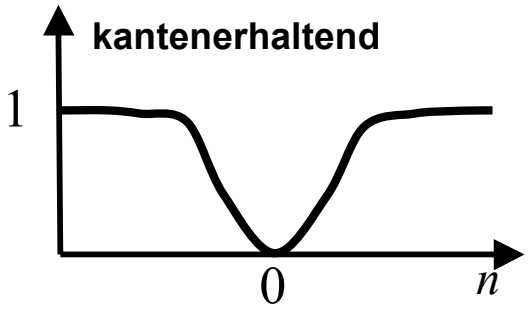


Fehlerfunktion, Risikofunktion



Fehlerfunktionen: Beispiele für Regression





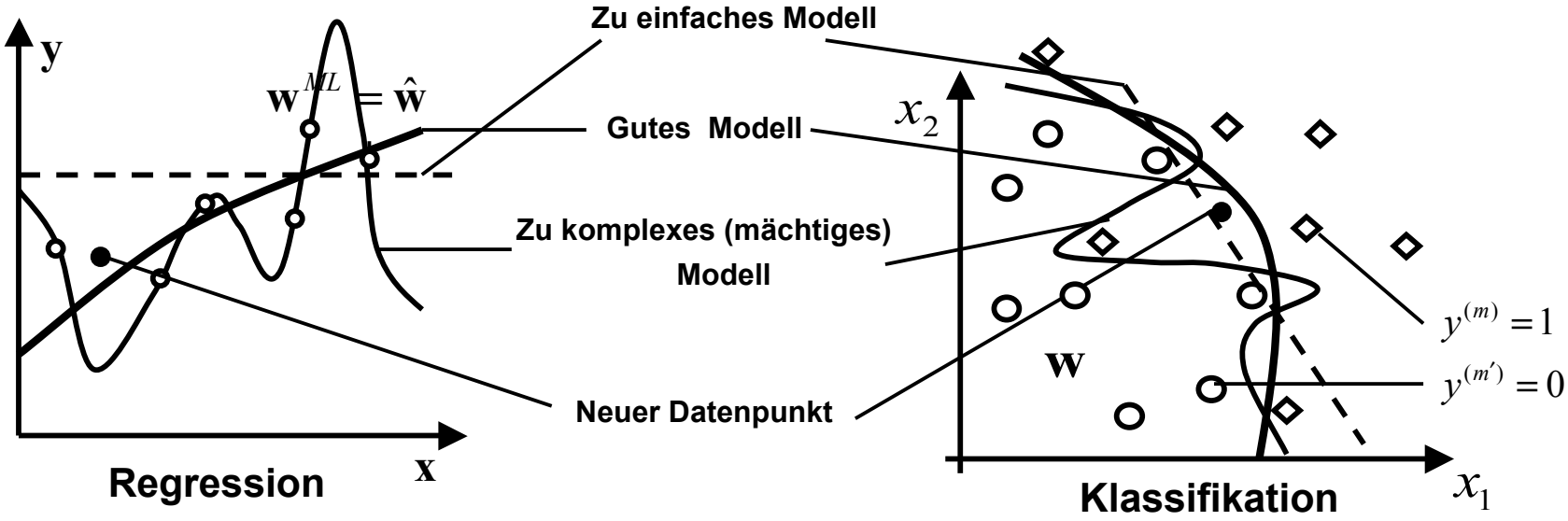
Lernen von Datenmodellen

Generalisierung und Regularisierung

- **Modell-Komplexität und Varianz-Bias Problem**
- **Generalisierung durch Regularisierung**
- **Kreuzvalidierung: Optimierung der Modellkomplexität**
- **Anwendungsbeispiel für Regularisierung: Funktionelle Kernspin-Bilder**

Generalisierung und Regularisierung

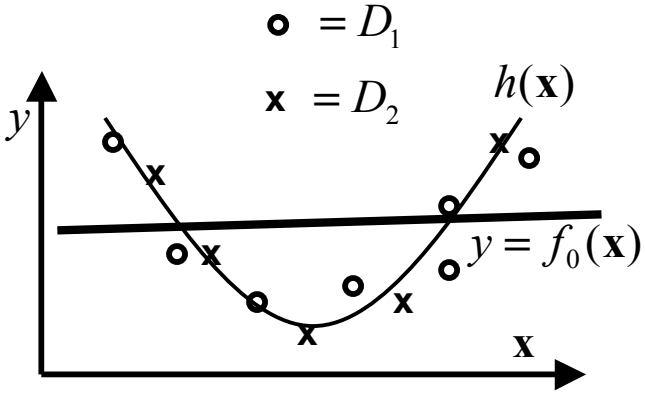
- **Betrachte überwachtes Lernproblem mit endlichem Datensatz** $(\mathbf{x}^{(m)}, \mathbf{y}^{(m)}),\ m = 1, \ldots, M$



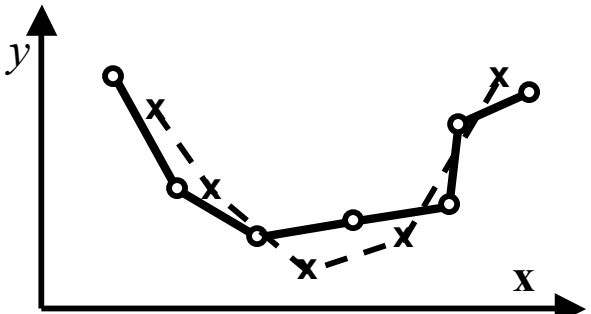
- **Beobachtungen:**
  - **-- Ein genügend komplexes Modell kann die Trainingsdaten beliebig genau erklären (kleiner Trainingsfehler)**
  - **-- Neue Datenpunkte werden dann aber schlechter erklärt (großer Testfehler, schlechte Generalisierungsfähigkeit, „Overfitting“)**
- **Finden der statistischen Struktur: Erkläre Daten mit möglichst einfachem Modell**

Etwas formaler: Das Bias-Varianz Problem

- **Ziel der Datenmodellierung:**
  -- Modellierung der statistischen Struktur h
  -- nicht Modellierung eines speziellen Datensatzes
- **Betrachte mittleres Verhalten über viele Datensätze:**
  $\mathbf{D} = \{D_1, D_2, \ldots\}$
- **Bias: Wie stark weicht das über alle Datensätze gemittelte Modell von dem wahren Modell ab?**
- **Varianz: Wie stark variiert das Modell (wie stark hängt es vom einzelnen Datensatz ab?)**
- **Sei $E_{\mathbf{D}}[.]$ der Erwartungswert über alle Datensätze**
  Für Regression und quadratischen Fehler:



Großer Bias, Varianz=0



Kleiner Bias, große Varianz

$$(f(\mathbf{x} \mid \mathbf{w}) - h(\mathbf{x}))^2 = (f(\mathbf{x} \mid \mathbf{w}) - E_{\mathbf{D}}[f(\mathbf{x} \mid \mathbf{w})] + E_{\mathbf{D}}[f(\mathbf{x} \mid \mathbf{w})] - h(\mathbf{x}))^2$$

$$E_{\mathbf{D}}[(f(\mathbf{x} \mid \mathbf{w}) - h(\mathbf{x}))^2] = (E_{\mathbf{D}}[f(\mathbf{x} \mid \mathbf{w})] - h(\mathbf{x}))^2 + E_{\mathbf{D}}[(f(\mathbf{x} \mid \mathbf{w}) - E_{\mathbf{D}}[f(\mathbf{x} \mid \mathbf{w})])^2]$$

Minimiere gemeinsam: **Bias** **Varianz**

Regularisierung: **Favorisierung von Modellen mit bestimmter (z.B. niedriger) Komplexität**

Bayes-Formalismus

-- negative log-Likelihood

-- negativer log-Prior

-- Max. a Posteriori

Schätztheorie

-- Fehlerfunktion L

-- Regularisierungsfunktion R

-- $F(\mathbf{w}) = L(\mathbf{w}) + \alpha R(\mathbf{w}) =! \min$

Beobachtung

-- Rauschen enthält alle (auch hohe) Frequenzen

-- Deterministische Zusammenhänge sind oft glatt (niedrige Frequenzen)

-- Sinnvolle Regularisierung: Bevorzuge „glatte“ Modelle

-- Glattere Modelle <= weniger Parameter, oder kleinere Gewichte

Beispiele für Regularisierungsfunktionen

- **Weight-Decay** $R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$
- **Kurven-basierte Regularisierung** $R(\mathbf{w}) = \sum_{m=1}^{M}\sum_{j=1}^{c}\sum_{i=1}^{d}\left(\left.\frac{\partial^2 f(\mathbf{x}\,|\,\mathbf{w})}{\partial x_i \partial x_j}\right|_{\mathbf{x}^{(m)}}\right)^2$

Regularisierung und Generalisierung

- **Regularisiertes Modell hat niedrige Varianz**
  => hängt nicht stark vom speziellen Datensatz ab
  => kann gut auf neue Datensätze übertragen werden
  => kann gut generalisieren.

Optimierung von Regularisierungstermen

- **Problem: -- Regularisierungsterm ist Hypothese des Modellierers**
  -- kann nicht direkt aus Daten geschätzt werden
  -- Wie bestimmt man die freien Parameter (z.B. Gewicht α)?
- **Methode 1: Schätzung als Hyperparameter aus der Bayes-Evidenz**
- **Methode 2 (praktisch sehr wichtig): Kreuzvalidierung.**

Kreuzvalidierung

- **Beurteile das Modell anhand seiner Generalisierungsfähigkeit:**
- **Teile dazu die Daten in Trainings- und Validierungsdaten auf**
- **Lerne an Trainingsdaten; Bestimme Performance anhand der Validierungsdaten**
- **Bsp. für Algorithmus :**

1. Selektiere ein Regularisierungsgewicht α für sehr einfache Modelle

2. Teile dazu die Daten zufällig in Trainings- und Validierungsdaten auf: $D = D_{tr}, D_{te}$

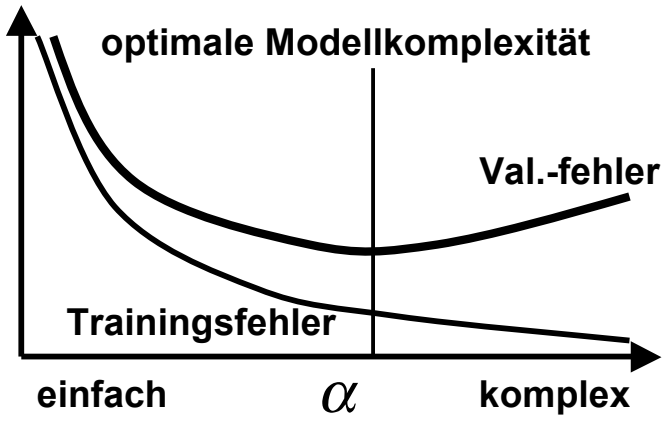
3. Optimiere $F(\mathbf{w})$ nur anhand der Trainingsdaten (siehe nächstes Kapitel)

4. Bestimme den Fehler anhand der Validierungsdaten

5. Wiederhole Schritte 2-4 oftmals mit neuen D_{tr}, D_{te}
=> mittlerer Validierungsfehler

6. Ändere α in Richtung komplexere Modelle,
gehe zu Schritt 2

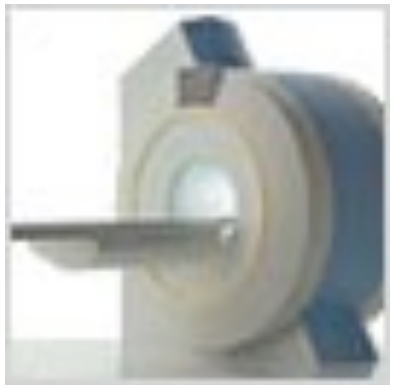
7. Stoppe, wenn der Validierungsfehler minimal ist



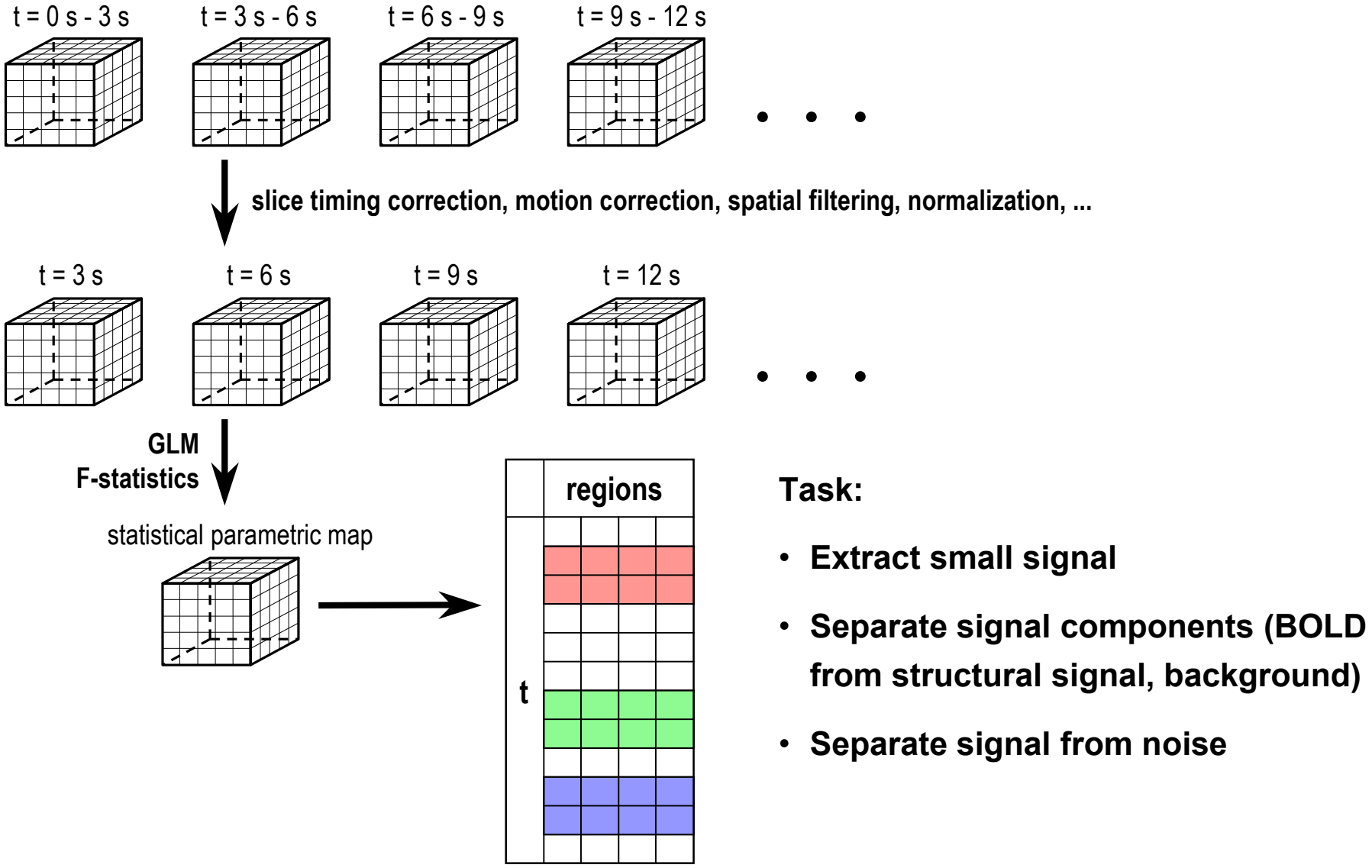
Anwendungsbeispiel für Regularisierung: Analyse von fMRI-Daten

Technik: funktionelle Kernspintomographie

- **„functional Magnetic Resonance Imaging“ (fMRI)**
- **Magnetmomente der Atome werden in einem starken Magnetfeld angeregt (Magnetresonanz)**
- **Sie relaxieren in den Grundzustand zurück**
- **Die Relaxation wird durch Gradientenfelder ortsabhängig gemacht (=> Imaging)**
- **Die Relaxation ist abhängig von der molekularen Umgebung**
  -- Wasserstoffkonzentration: strukturelle MRI
  -- Paramagnetische Substanzen (Hbr): funktionelle MRI



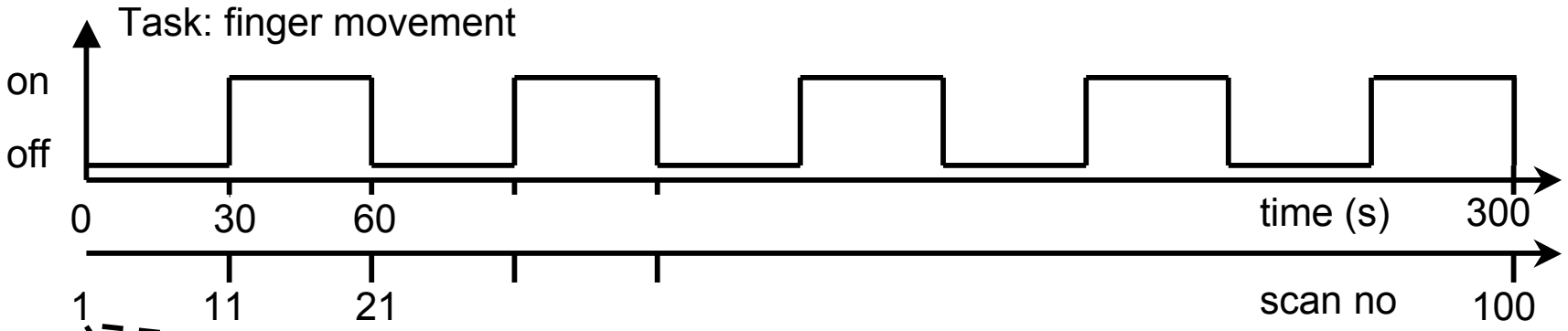
fMRI Data Format

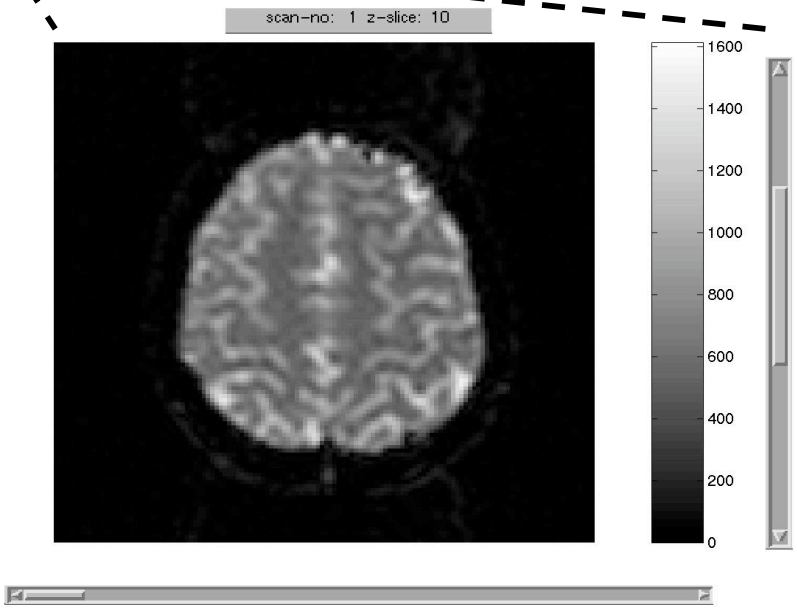


Task:

- **Extract small signal**
- **Separate signal components (BOLD from structural signal, background)**
- **Separate signal from noise**

Data Set mrea-g: Format





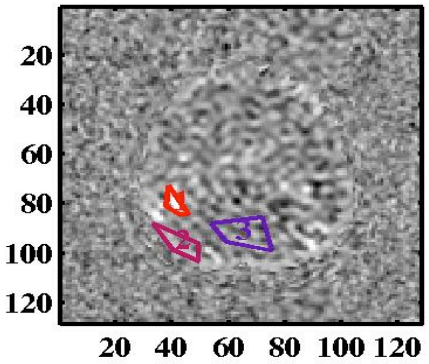
Raw data: scan 1, slice 10

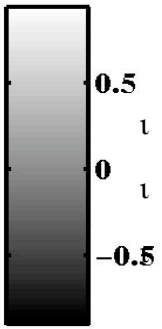
Raw data:

- EPI scans
- 100 scans,
- 128x128 slice size, 16 slices
- Scan time: 1.6 sec
- Scan interval: 3 sec

Data Set mrea-g: Visual Inspection of Time Series

Mean time series over three regions: Signal, drift, and background

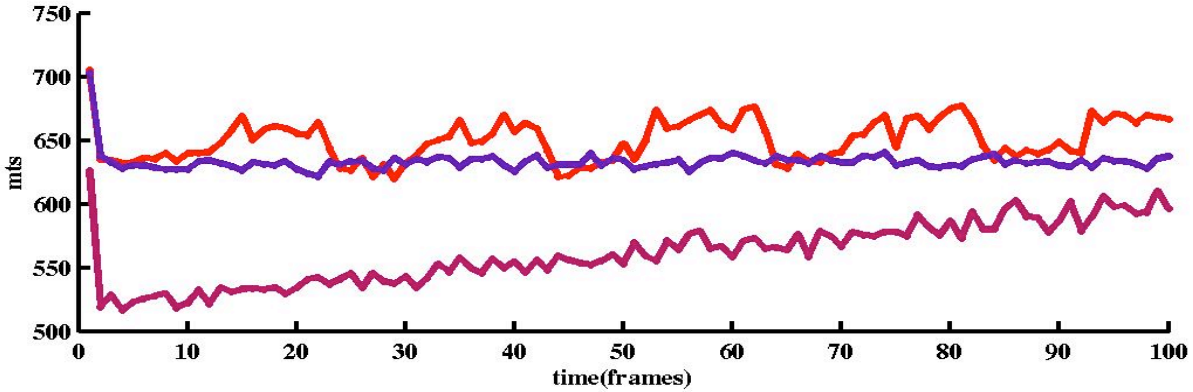




Task-related response

linear drift present

First frame has higher values (magnetic transient?)



General Linear Model: Design Matrix

Principle: $\mathbf{X} = \mathbf{Hw} + \mathbf{n}$

Mrea-g data set

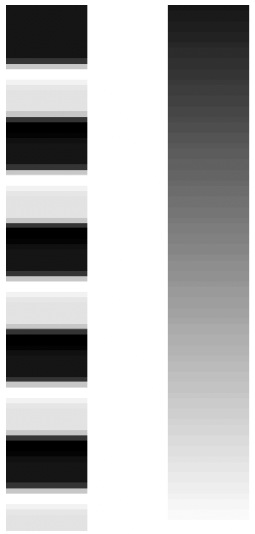
$\mathbf{X}$ = matrix of time series

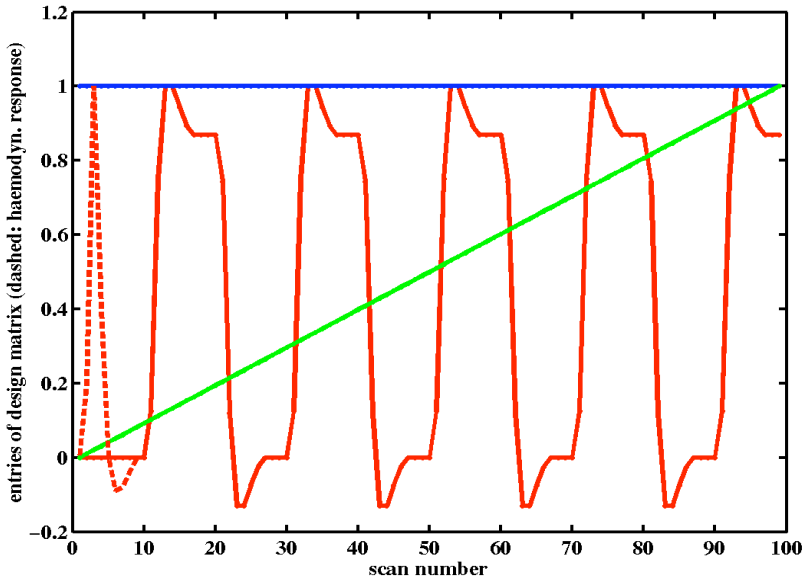
$\mathbf{H}$ = Design matrix

$\mathbf{w}$= parameter (estimate used for t-test)

$\mathbf{n}$ = noise (estimate used for t-test)

Entries of design matrix used:

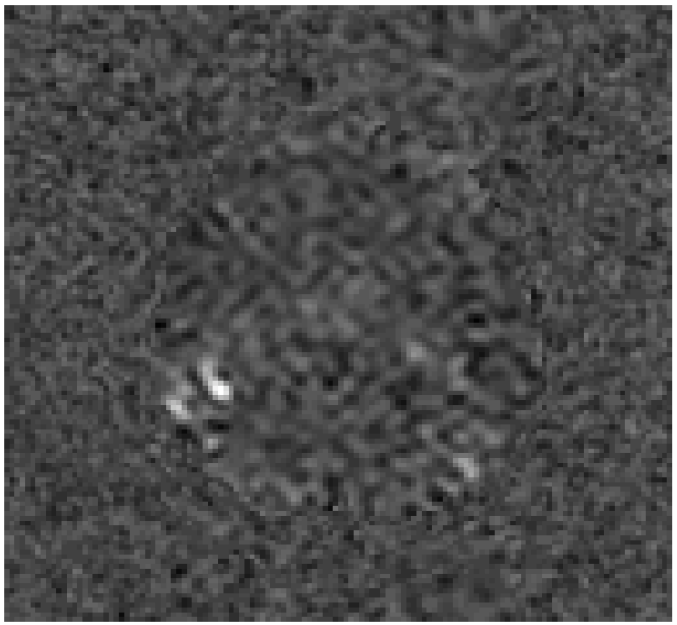




GLM: Statistical Parametric Maps (SPM)

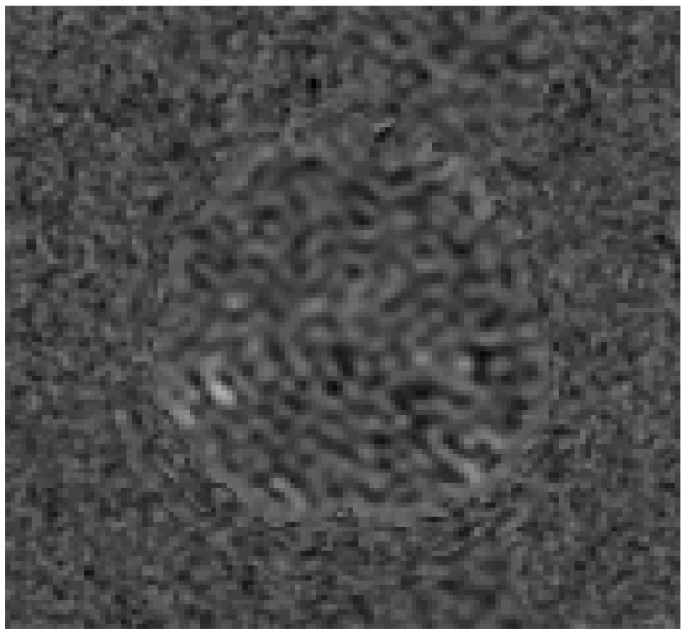
Mrea-g data set

3 component GLM on raw data



$w_1(x, y)$

Raw t-test on raw data



$t(x, y)$

- Higher absolute t-value in GLM (due to convolution with haemodynamic response
- t-values along head border are not suppressed (beause drift is modeled)

Regularized GLM

Introduce regularizing terms (Bayes: Priors):

$$P(w(x,y)) \equiv P(\mathbf{w}) = P_A(\mathbf{w}) P_{MRF}(\mathbf{w})$$

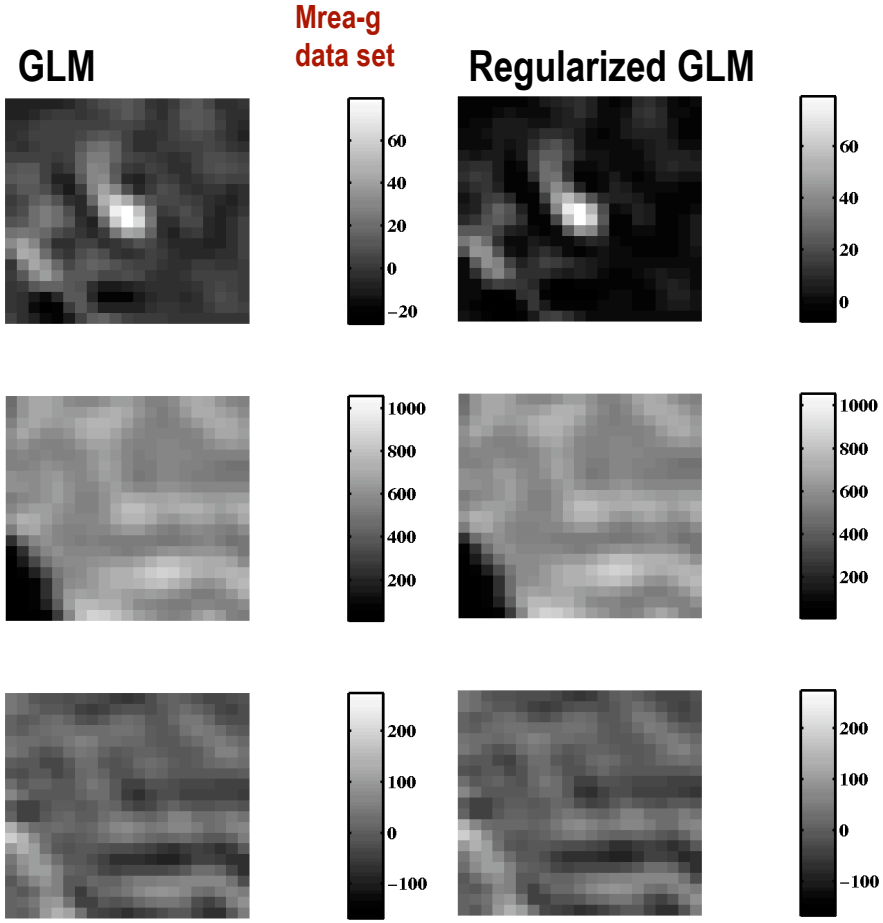
- **For positive sign of BOLD signal**

$$-\ln P_A = \sum_{x,y} \ln\left(1 + \exp(-w(x,y)/\sigma_A)\right)$$

- **For smoothness in space (Markov Random Field Prior)**

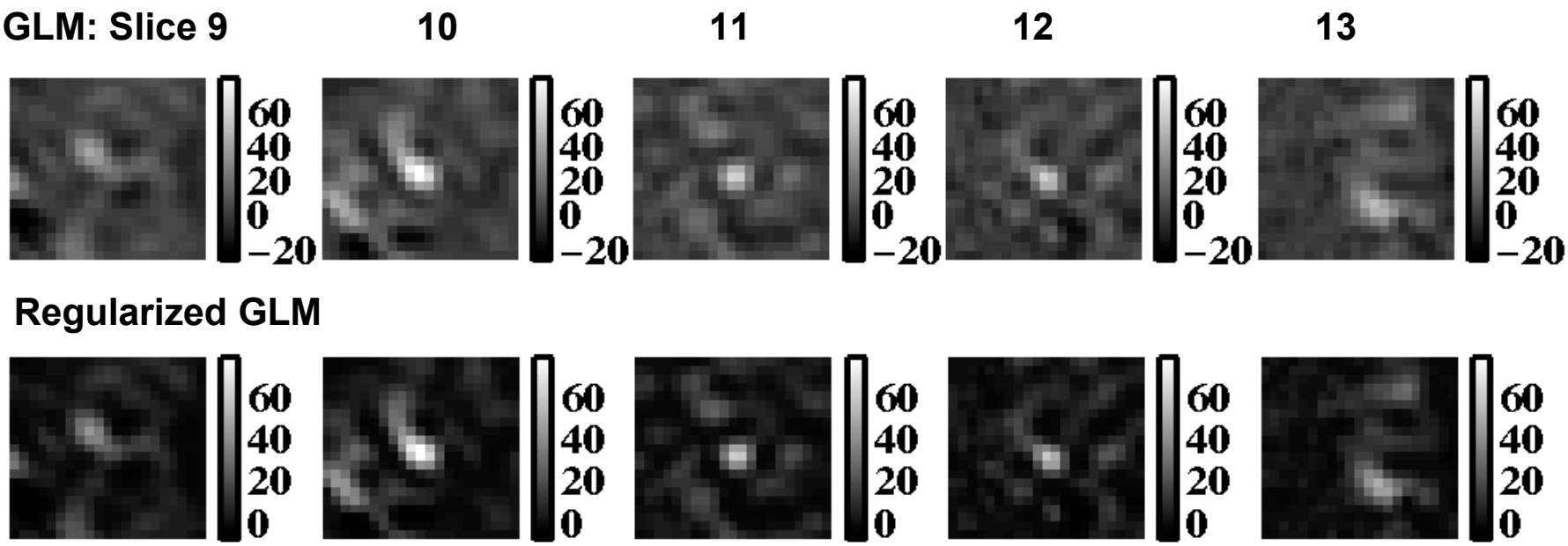
$$-\ln P_{MRF} =$$

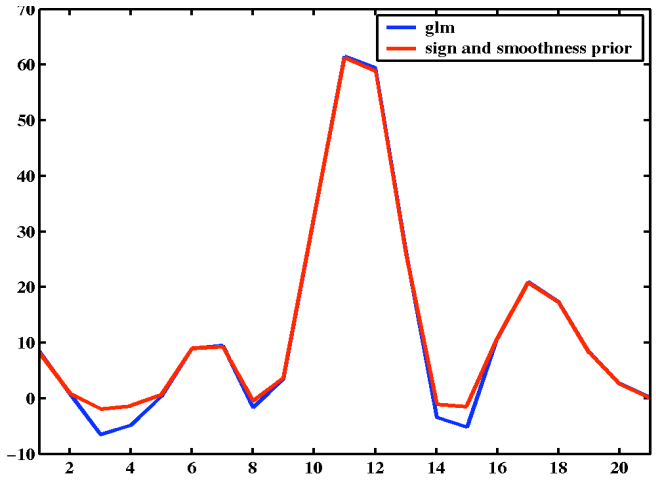
$$\sum_{x,y} \frac{(w(x+1,y) - w(x,y))^2 + (w(x,y+1) - w(x,y))^2}{2\sigma_{MRF}^2}$$



Regularized GLM: Stimulus-Related Signal

Mrea-g data set





Top: Five consecutive slices (small dataset)

Left: 1D-profile through active region (sl. 11, line 12)

Regularized GLM (Edge preserving MRF)

Introduce regularizing terms (Bayes: Priors):

$$P(w(x,y)) \equiv P(\mathbf{w}) = P_A(\mathbf{w}) P_{EPS}(\mathbf{w})$$

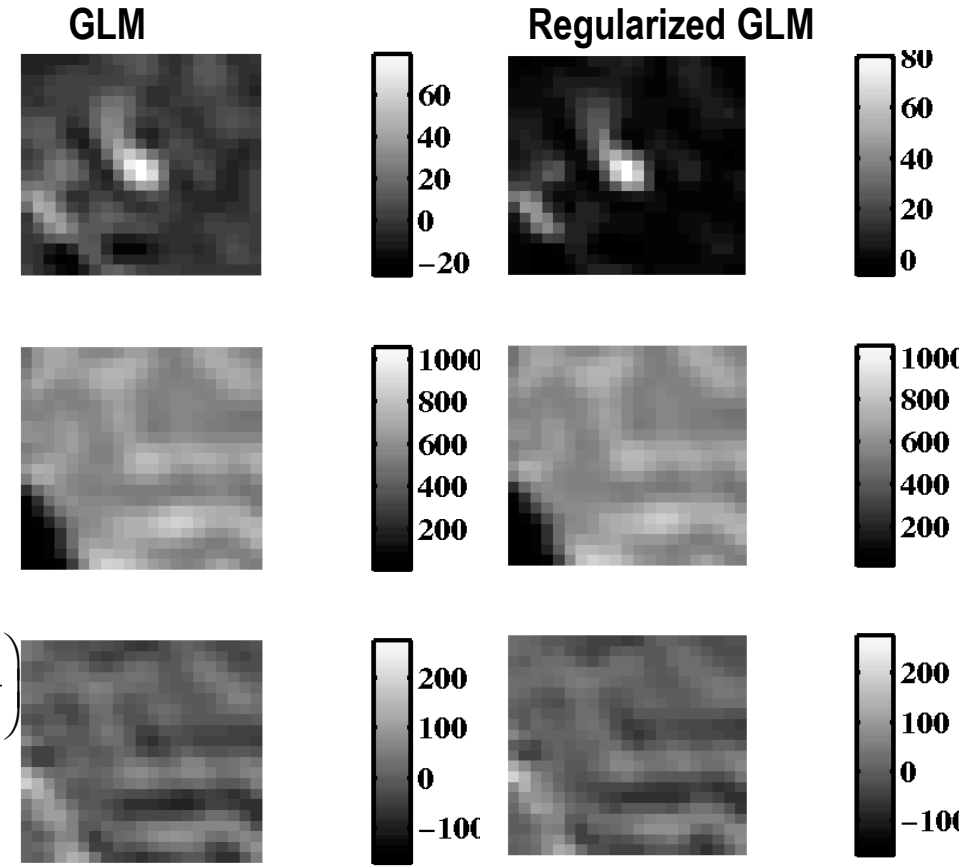
- **For positive sign of BOLD signal**

$$-\ln P_A = \sum_{x,y} \ln\left(1 + \exp(-w(x,y)/\sigma_A)\right)$$

- **For edge-preserving smoother**

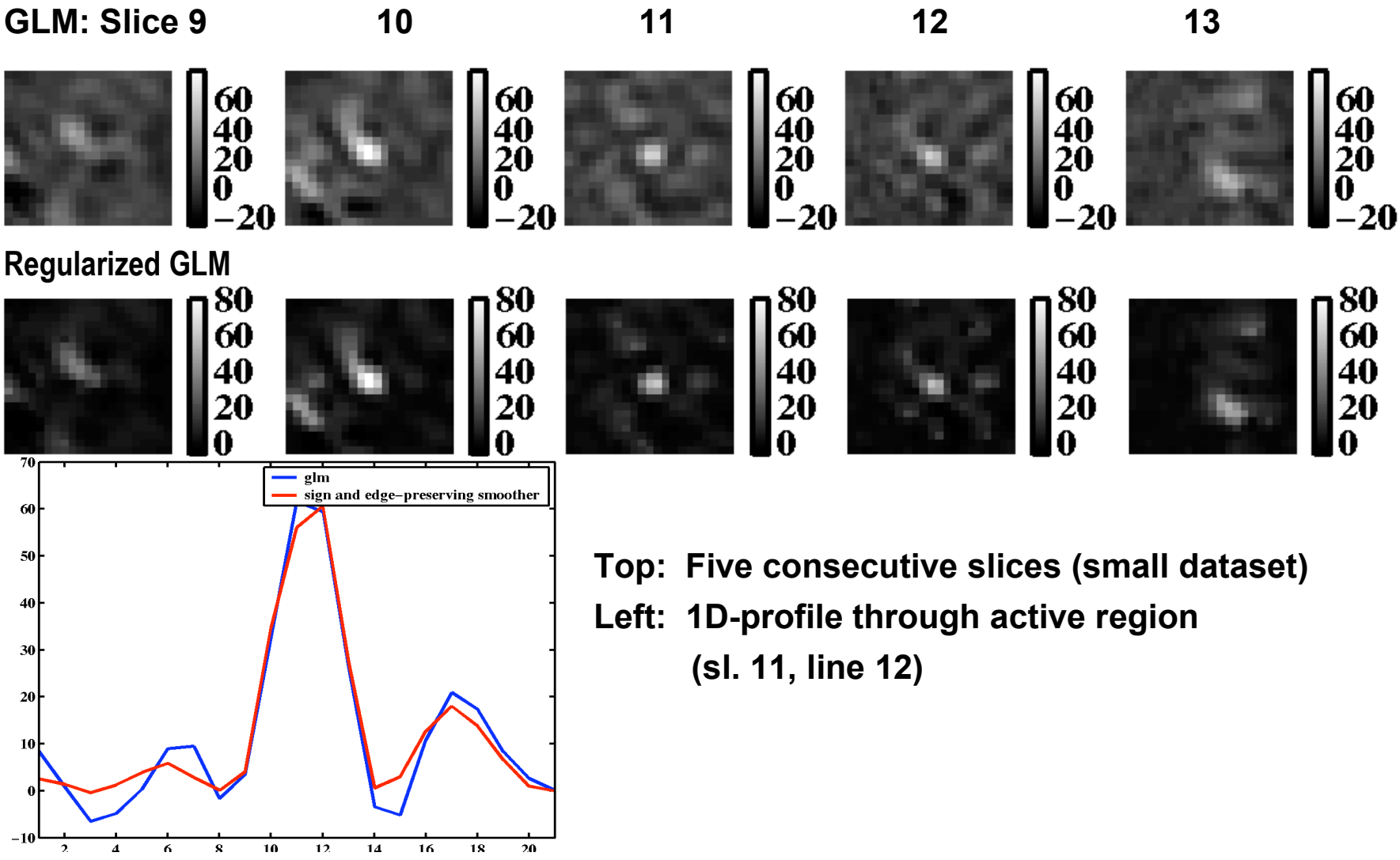
$$-\ln P_{EPS} =$$

$$\sum_{x,y} 1 - \exp\left(\frac{(w(x+1,y) - w(x,y))^2 + (w(x,y+1) - w(x,y))^2}{2\sigma_{MRF}^2}\right)$$



Regularized GLM (EP-MRF): Stimulus-Related Signal

Mrea-g data set



Top: Five consecutive slices (small dataset)

Left: 1D-profile through active region (sl. 11, line 12)