IJCNN 2004: Time Series Prediction Competition

Time Series Prediction by Kalman Smoother with Cross-Validated Noise Density

Simo Särkkä, <*simo.sarkka@hut.fi*>

Aki Vehtari, <*aki.vehtari@hut.fi*>

Jouko Lampinen, <*jouko.lampinen@hut.fi*>

Helsinki University of Technology, Finland

Outline

- Bayesian Filtering and Smoothing
- Kalman Filtering and Smoothing
- Long Term Model
- Short Term Model
- Cross-Validation of Parameters
- Prediction Results

Bayesian Filtering and Smoothing

- Bayesian inference - Uncertainties are modeled as probability distributions
- Generic state space model

$$\mathbf{y}_k \sim p(\mathbf{y}_k | \mathbf{x}_k)$$

$$\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1})$$

$$\mathbf{x}_0 \sim p(\mathbf{x}_0)$$

- Where vector $\mathbf{x}_k$ is the state and vector $\mathbf{y}_k$ is the measurement

Bayesian Filtering and Smoothing

- The ultimate goal of a **filter** is to compute (filtered) posterior distribution of the current state, which is conditioned to measurements up to the current time step k:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k)$$

- The ultimate goal of a **smoother** is to compute (smoothed) posterior distributions of all the states, which are conditioned to all measurements up to a time step $T > k$:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

Kalman Filter and Smoother

- **Kalman filter** computes exactly the filtered posteriors of linear Gaussian models in form

$$p(\mathbf{y}_k|\mathbf{x}_k) = N(\mathbf{y}_k|\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)$$
$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N(\mathbf{x}_k|\mathbf{A}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1})$$
$$p(\mathbf{x}_0) = N(\mathbf{x}_0|\mathbf{m}_0, \mathbf{P}_0)$$

- Where vector $\mathbf{x}_k$ is the state and vector $\mathbf{y}_k$ is the measurement
- **Kalman smoother** computes the smoothed posteriors for the same linear Gaussian model

Long Term Model

- The selected continuous dynamic linear model for long term prediction is

$$\ddot{x}(t) = w(t)$$

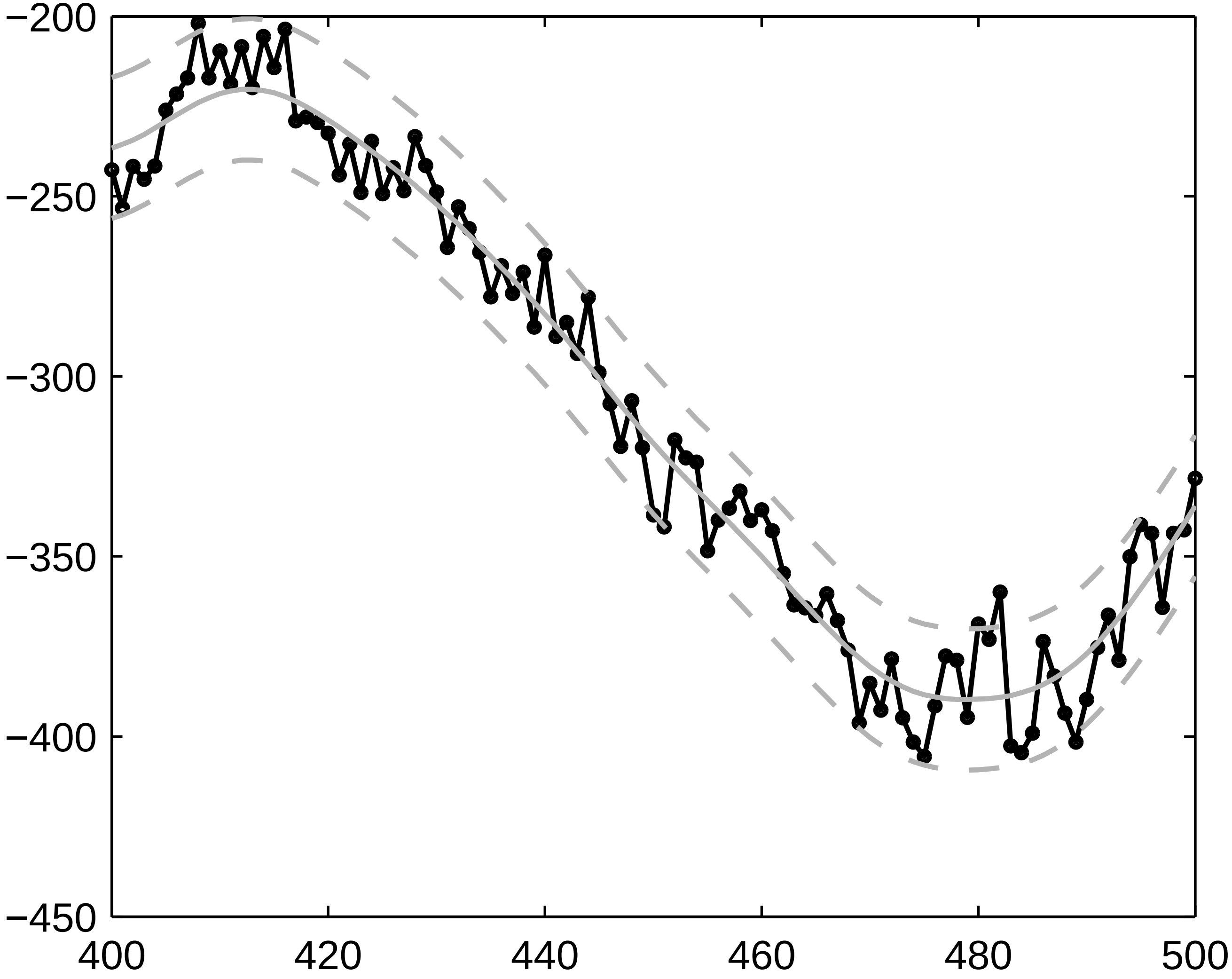
- Equivalent discrete representation

$$\begin{pmatrix} x_k \\ \dot{x}_k \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{pmatrix} + \begin{pmatrix} q^x_{1,k-1} \\ q^x_{2,k-1} \end{pmatrix}$$

- Measurement model is

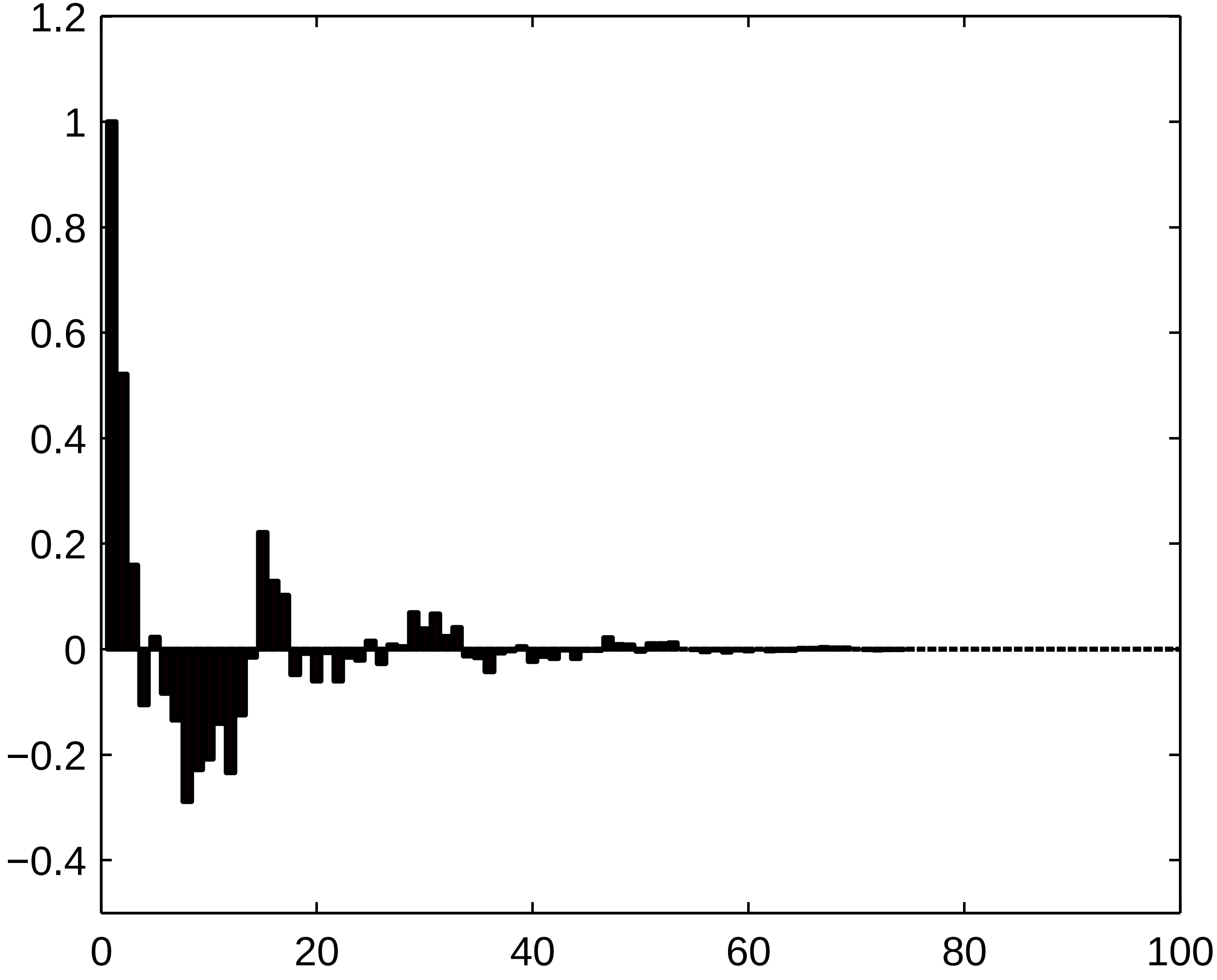
$$y_k = x_k + r^x_k, \quad r^x_k \sim N(0, \sigma^2_x)$$

Long Term Model



Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Long Term Model



Short Term Model

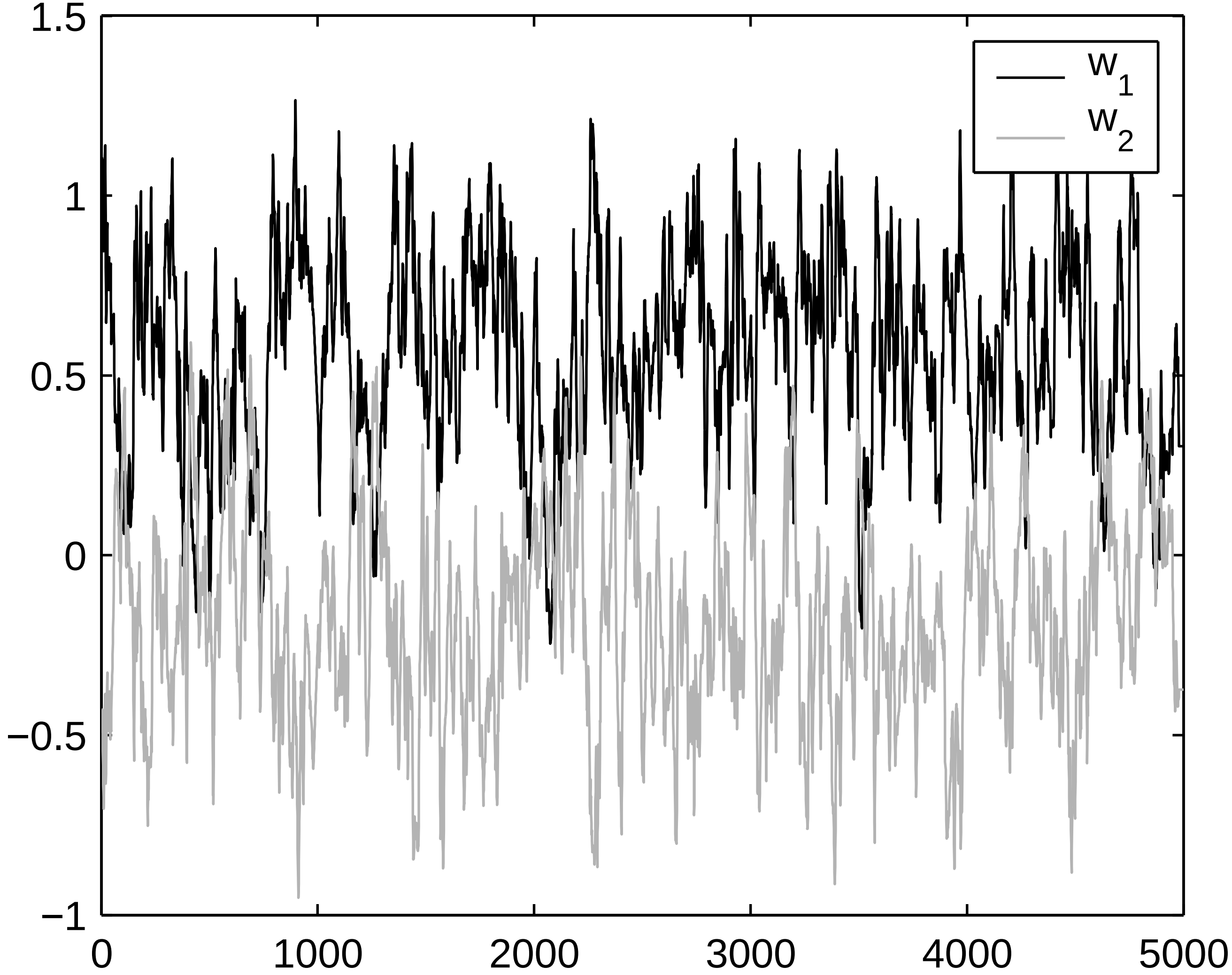
- Time Varying AR-model for residual:

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{v}_k^{\mathrm{ar}}$$
$$e_k = \sum_i w_{i,k} e_{k-i} + r_k^{\mathrm{ar}}.$$

- The weight vector $\mathbf{w}_k$ is to be estimated from the known part of residual time series with Kalman filter and smoother

- The weight vector is predicted over the missing parts - measurements are modeled as missing

Short Term Model



Short Term Model

- Estimated short term residual is obtained from

$$d_k = \sum_i w_{i,k} d_{k-i} + v_k^{\mathrm{p}}$$

$$e_k = d_k + r_k^p, \quad r_k^p \sim N(0, \sigma_p^2),$$

- Given the weight sequence, can be solved by Kalman filter and smoother

- The final result is the sum of long and short term estimates

$$\hat{y}_k = \hat{x}_k + \hat{d}_k$$

Prediction Method in Practice

1. **Long term prediction:** Run *Kalman filter* over the data sequence and store the estimated means and covariances. Run *Kalman smoother* over the Kalman filter estimation result

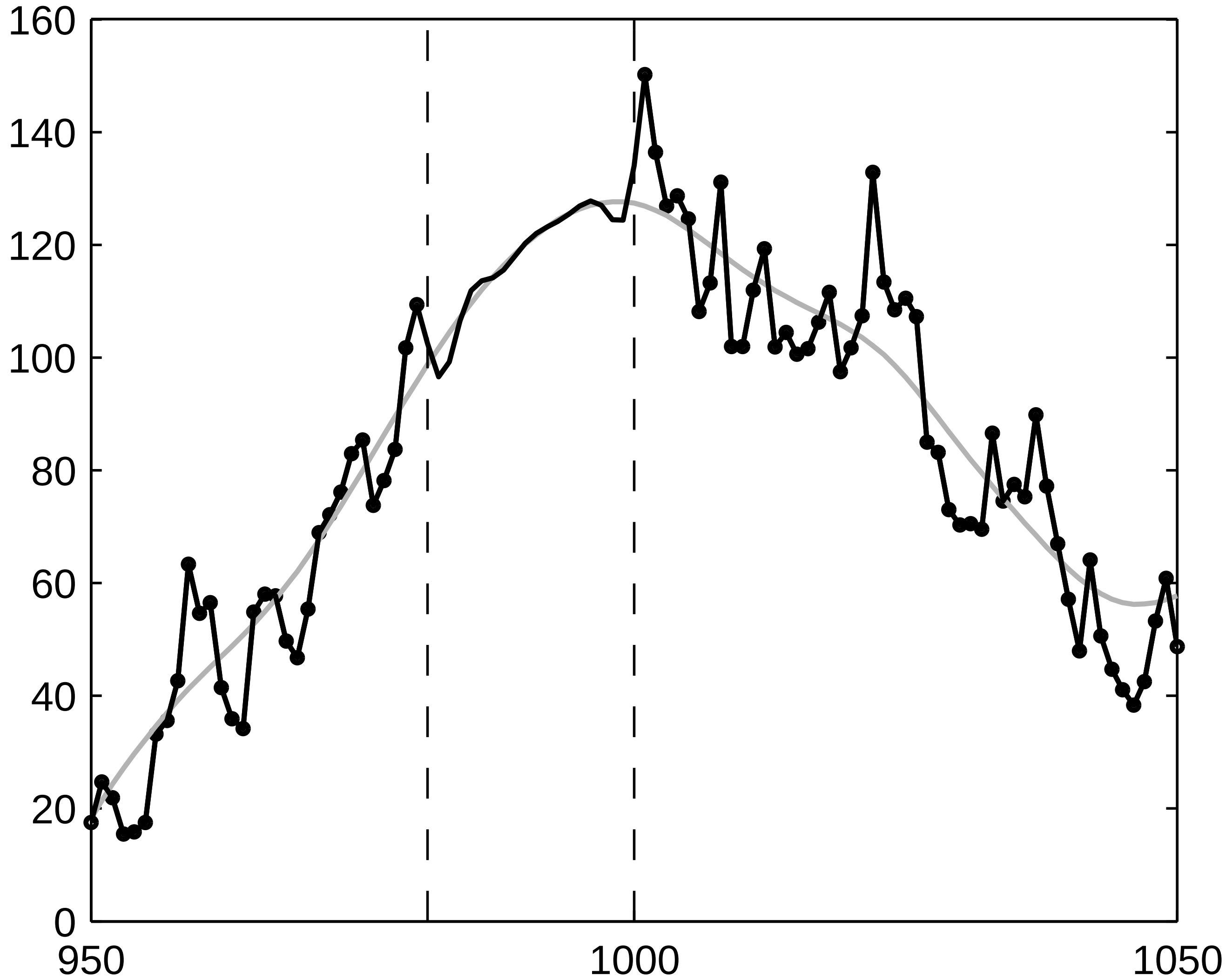
2. **AR-weight estimation:** Run *Kalman filter* and then *Kalman smoother* over the residual sequence to estimate the weights

3. **Short term residual estimation:** Run *Kalman filter* and *Kalman smoother* over the residual sequence to get the estimated short term residual signal

Cross-Validation of Noise Variances

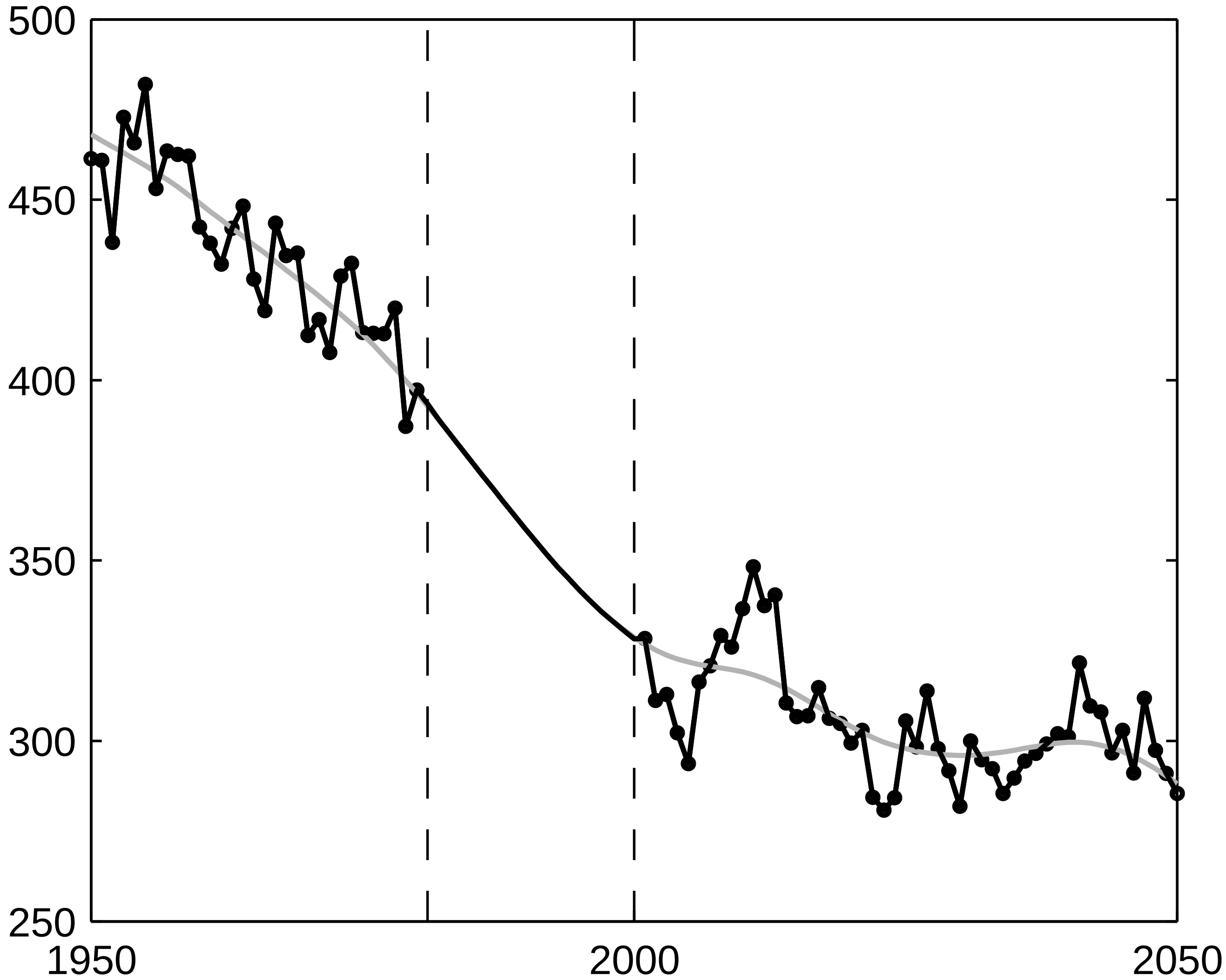
- Kalman filter and smoother have the noise parameters that need to be selected some way
- Measurement noises can be selected “by guessing”, because they depend on process noises
- Process noises were selected by cross-validation using the cost criterion of the competition
- The final prediction process noise was selected to have a suitably low value

Final results: 981–1000



Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Final results: 1981–2000

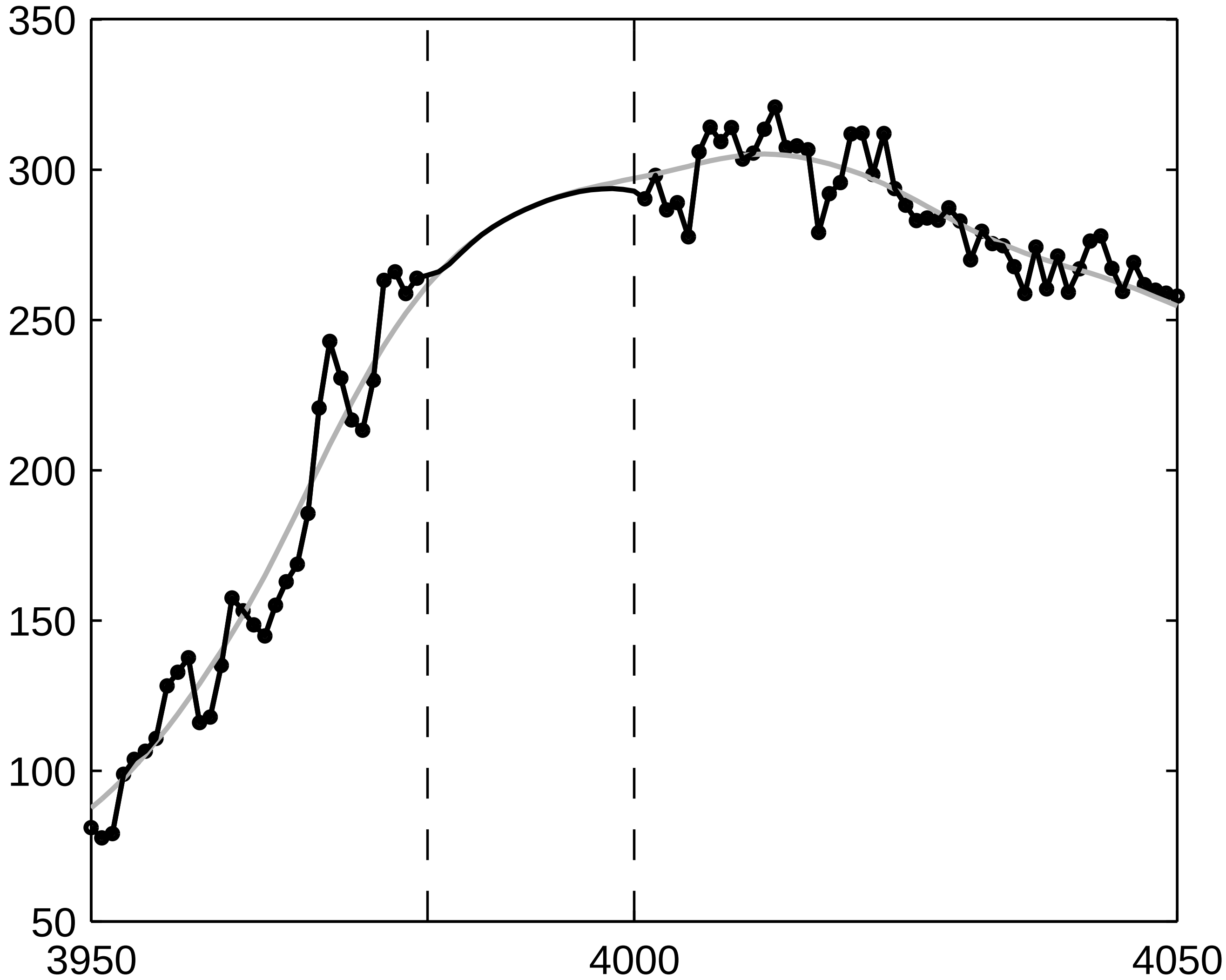


Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Final results: 2981–3000

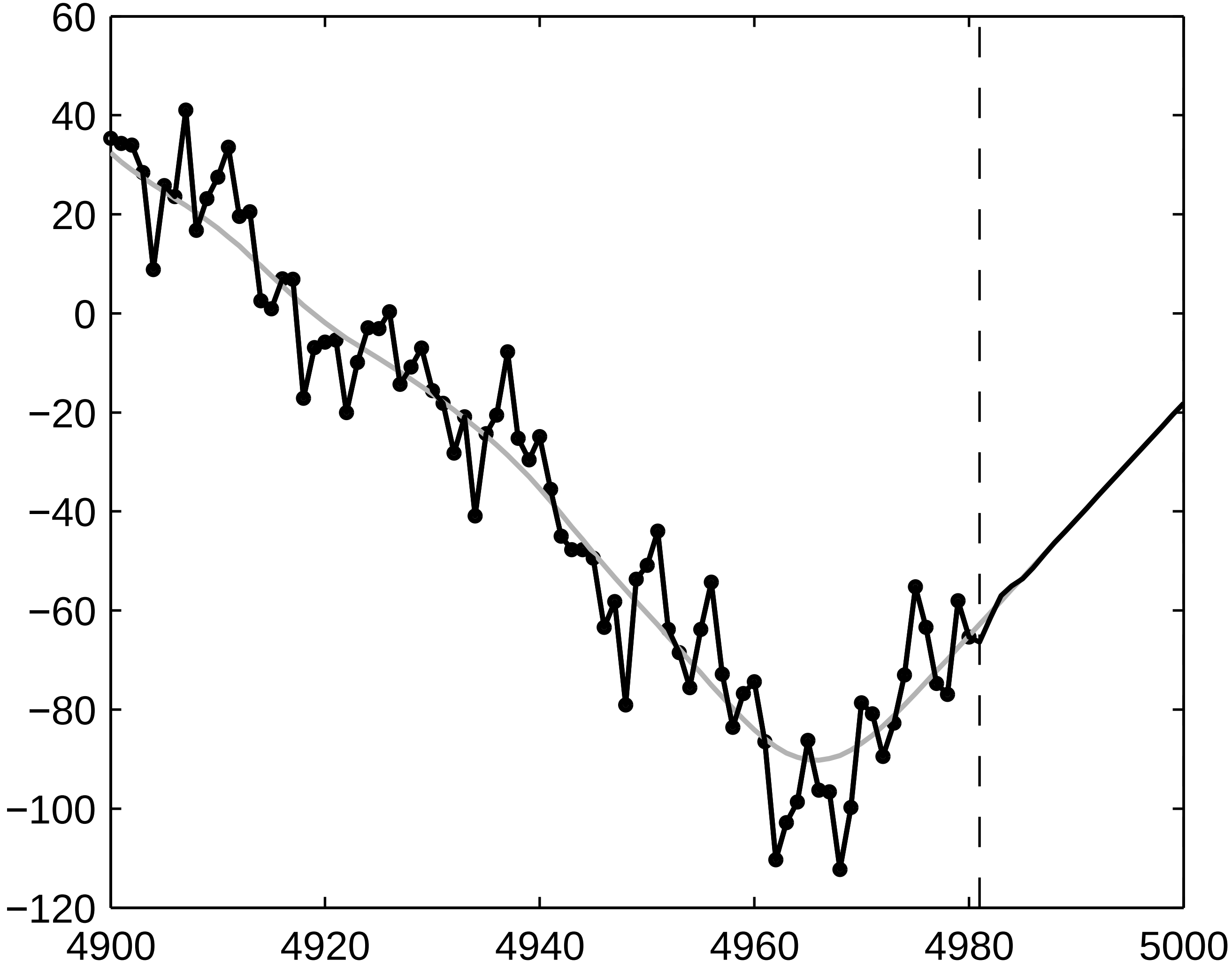
Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Final results: 3981–4000



Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Final results: 4981–5000



Gray line is the long term prediction, black line is the final prediction and dots are the known data points.

Summary of Method

- Long term prediction is based on Gaussian linear state space model
- Short term prediction was performed by a time varying AR-process model
- The noise parameters of Kalman filters and smoothers were selected by cross-validation
- The final prediction was done using the estimated noise parameters as if they were known in advance

Why Does it Work?

- The long term stochastic differential equation model

$$\ddot{x}(t) = w(t)$$

  is equivalent to Tikhonov regularization

- The class of functions is that of Gaussian processes, which are often used in Bayesian machine learning

- The short term model is also a Gaussian process model

- Cross-validation explicitly maximized the generalization ability for the cost function used in the competition