Cross-validation of Controlled Dynamic Models: Bayesian Approach

M. Kárný, P. Nedoma, V. Šmídl

Adaptive System Department, Institute of Information Theory and Automation, Prague.

16th IFAC Congress, July 6, 2005
Outline

1. Introduction
   - Motivation
   - Aim of the Work

2. Bayesian Approach
   - Formalizing the Validation Task
   - Validation by Data Splitting
   - Bayesian Learning

3. Validation of Dynamic Models
   - Validation Algorithm
   - Validation with Multiple Splits

4. Validation of Models from Exponential Family
   - Computational Efficient Validation
   - Validation using Stabilized Forgetting

5. Conclusion
Question arising in design of any model or controller:

Is our model good enough?

Possible approaches:

1. test it on the real problem (expensive),
2. simulation (requires reliable physical model),
3. expert knowledge (experience, craftsmanship),
4. formal criteria (limited range of models).

Each is suitable for different problems.
What is the best validation strategy for modelling of large high-dimensional data archives by mixtures of Gaussian and ARX models?
Motivation: Static Mixture

Validation:

- In this case (2D) we can use expert knowledge (visual inspection).
- What to do in higher-dimensions?
Motivation: Dynamic Mixture (2 ARX models)

Validation using prediction errors:

1) expert knowledge, 2) formal criteria, 3) What to do in higher-dimensions?
We seek a **formal** validation procedure that:

1. fits well into the Bayesian framework,
2. general enough to cover more complex models, (Including dynamic models with non-stationary parameters)
3. use as many standard results as possible, (we expect that learning procedure is available, at least approximate)
4. computationally feasible.
finds model, \( \mathcal{M}^* \), which is the best representation of the data (including model structure)
Formalizing the Validation Task

Learning

System 1

$\varnothing$

$M^*$

$\sqcap \hat{M}_1 \hat{M}_2$

System 2

$\varnothing$

$\hat{M}^*$

Validation

System 1

$\varnothing$

$M^*$

$M^*_V$

System 2

$\varnothing$

$\hat{M}^*$

$\sqcap \hat{M}_1 \hat{M}_2$

finds model, $\hat{M} \in M^*$, which is the best representation of the data (including model structure)

tests assumptions of $M^*$ in richer class $M^*_V$, using $\hat{M}$,
finds model, $\{0M \in M^*\}$, which is the best representation of the data (including model structure)

Validation is a ‘poor man’s’ attempt to relax assumptions of $M^*$ enforced by computational restrictions. The task is to choose $M^*_V$. 
Validation by Data Splitting

All available data are split into learning and validation parts.

<table>
<thead>
<tr>
<th>Learning data</th>
<th>Validation data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(1 \ldots \tau)$</td>
<td>$d(\tau + 1 \ldots, t)$</td>
</tr>
</tbody>
</table>

Learning data $d(1 \ldots \tau)$, are used to identify the model $M$, validation data $d(\tau + 1 \ldots, t)$, are used to test predictive abilities of $M$. 
Validation by Data Splitting

All available data are split into learning and validation parts.

<table>
<thead>
<tr>
<th>Learning data</th>
<th>Validation data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \tau )</td>
</tr>
<tr>
<td>( l_c )</td>
<td>( l_v )</td>
</tr>
</tbody>
</table>

learning data \( l_c = d(1 \ldots \tau) \), are used to identify the model \( l_M \),
validation data \( l_v = d(\tau + 1 \ldots, t) \), are used to test predictive
abilities of \( l_M \).

\( M^* \) is a class of single models,
\( M^*_V \) is a class of models \( M^* \) switching parameters at time \( \tau \).
Validation by Data Splitting

All available data are split into learning and validation parts.

<table>
<thead>
<tr>
<th>Learning data</th>
<th>Validation data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(\tau)</td>
</tr>
<tr>
<td>(\bar{t})</td>
<td></td>
</tr>
</tbody>
</table>

Learning data \(\mathcal{d} = d(1 \ldots \tau)\), are used to identify the model \(\mathcal{M}\), validation data \(\mathcal{d} = d(\tau + 1 \ldots , \bar{t})\), are used to test predictive abilities of \(\mathcal{M}\).

\(\mathcal{M}^*\) is a class of single models, 
\(\mathcal{M}_V^*\) is a class of models \(\mathcal{M}^*\) switching parameters at time \(\tau\).

We solve a **decision making problem** with two hypotheses:

- **\(H_0\)**: all data are represented by a single model: \(\mathcal{M}\),
- **\(H_1\)**: data sets \(\mathcal{d}\) and \(\mathcal{d}\) are generated by different models, \(\mathcal{M}\) and \(\mathcal{M}\), respectively.
Bayesian Learning for Dynamic Controlled Models

Proabilistic model:

\[
f(d(1\ldots \bar{t}), x(1\ldots \bar{t}) | x_0) = \prod_{t=p}^{\bar{t}} f(y_t | \psi_t, x_t) f(x_t | \psi_t, x_{t-1}) f(u_t | d(1\ldots \bar{t})) ,
\]

where: data \(d_t = [u_t, y_t]\), output \(y_t\), input \(u_t\), parameters (state) \(x_t\), regressor \(\psi_t = \psi(d(1\ldots \bar{t} - 1))\), regression length \(p\).

State-estimation: \(f(x_t | d(1\ldots \bar{t}))\), prior \(f(x_0)\),
Output-prediction: \(f(y_t | u_t, d(1\ldots \bar{t} - 1))\),
Model likelihood:

\[
\mathcal{L}(d(1\ldots \bar{t}), \mathcal{M}) = \prod_{t=1}^{\bar{t}} f(y_t | u_t, d(1\ldots \bar{t} - 1)) ,
\]

Model selection:

\[
\hat{\mathcal{M}} = \arg \min_{\mathcal{M} \in \mathcal{M}^*} \mathcal{L}(d(1\ldots \bar{t}), \mathcal{M}) .
\]
Validation Algorithm

1. Choose model class $M^*$, prior $f(x_0)$, and splitting moment $\tau$.

2. Use $\lfloor ld$ to learn model $\lfloor l M$. Choose $f(x_\tau)$ (possibly using $\lfloor l M$).

3. Continue with learning on $\lfloor v d$: 1) learn $\lfloor o M$, starting from $\lfloor l M$, 2) learn $\lfloor v M$, starting from $f(x_\tau)$.

4. Evaluate likelihoods $L(\cdot)$ of all models and probability $f(H_0|d) = (1 + L(\lfloor l d, \lfloor l M) L(\lfloor v d, \lfloor v M) L(\lfloor o M))^{-1}$.

5. Model class $M^*$ is valid if $f(H_0|d)$ is close to 1.
Choose model class $\mathcal{M}^*$, prior $f(x_0)$, and splitting moment $\tau$. 
Validation Algorithm

Choose model class $\mathcal{M}^*$, prior $f(x_0)$, and splitting moment $\tau$.

Use $\dot{\mathcal{M}}$ to learn model $\dot{\mathcal{M}}$. Choose $f(x_{\tau})$ (possibly using $\dot{\mathcal{M}}$).
Validation Algorithm

1. Choose model class $\mathcal{M}^*$, prior $f(x_0)$, and splitting moment $\tau$.
2. Use $\mathcal{U}d$ to learn model $\mathcal{U}M$. Choose $f(x_\tau)$ (possibly using $\mathcal{U}M$).
3. Continue with learning on $\mathcal{V}d$:
   1. learn $\mathcal{O}M$, starting from $\mathcal{U}M$,
   2. learn $\mathcal{V}M$, starting from $f(x_\tau)$.
Choose model class $\mathcal{M}^*$, prior $f(x_0)$, and splitting moment $\tau$.

Use $\downarrow d$ to learn model $\downarrow \mathcal{M}$. Choose $f(x_\tau)$ (possibly using $\downarrow \mathcal{M}$).

Continue with learning on $\downarrow \nu d$:

1. learn $\downarrow \omega \mathcal{M}$, starting from $\downarrow \mathcal{M}$,
2. learn $\downarrow \nu \mathcal{M}$, starting from $f(x_\tau)$.

Evaluate likelihoods $\mathcal{L}(\cdot)$ of all models and probability

$$f(H_0|d) = \left(1 + \frac{\mathcal{L}(\downarrow d, \downarrow \mathcal{M})\mathcal{L}(\downarrow \nu d, \downarrow \nu \mathcal{M})}{\mathcal{L}(d, \downarrow \omega \mathcal{M})}\right)^{-1},$$
Validation Algorithm

1. Choose model class $\mathcal{M}^*$, prior $f(x_0)$, and splitting moment $\tau$.
2. Use $\mathcal{L}d$ to learn model $\mathcal{L}\mathcal{M}$. Choose $f(x_\tau)$ (possibly using $\mathcal{L}\mathcal{M}$).
3. Continue with learning on $\mathcal{L}v_d$:
   1. learn $\mathcal{L}o\mathcal{M}$, starting from $\mathcal{L}\mathcal{M}$,
   2. learn $\mathcal{L}v\mathcal{M}$, starting from $f(x_\tau)$.
4. Evaluate likelihoods $\mathcal{L}(\cdot)$ of all models and probability
   \[
   f(H_0|d) = \left(1 + \frac{\mathcal{L}(\mathcal{L}d, \mathcal{L}\mathcal{M})\mathcal{L}(\mathcal{L}v_d, \mathcal{L}v\mathcal{M})}{\mathcal{L}(d, \mathcal{L}o\mathcal{M})}\right)^{-1},
   \]
5. Model class $\mathcal{M}^*$ is valid if $f(H_0|d)$ is close to 1.
Experiment: ARX model

process: ARX(2)
model: ARX(2)

process: ARX(4)
model: ARX(2)
Validation with Multiple Splitting Moments

We replace the known splitting moment $\tau$ by a grid $\tau^* = [\tau_1, \ldots, \tau_n]$.

Learning $\longrightarrow$ Validation

\[
\begin{array}{ccccccc}
1 & \tau_1 & \tau_2 & \ldots & \tau_n & \bar{t} \\
\end{array}
\]

The number of hypothesis in the decision making problem is much higher.
Validation with Multiple Splitting Moments

We replace the known splitting moment $\tau$ by a grid $\tau^* = [\tau_1, \ldots, \tau_n]$.

Learning          Validation

\[ \begin{array}{ccccccc}
1 & \tau_1 & \tau_2 & \ldots & \tau_n & \bar{t} \\
\end{array} \]

The number of hypothesis in the decision making problem is much higher.

Can be set up as decision making problem, using loss-functions.

Results:

Maximum-likelihood: choose $H_{\hat{i}}$ with

\[
\hat{i} = \arg \max_{i=0,1} \left( \max_{\tau \in \tau^*} f(H_i|d) \right).
\]

Marginal-likelihood: treating $\tau$ as random variable:

\[
\mathcal{L}(d(1\ldots\bar{t}), \mathcal{H}) = \sum_{\tau} \mathcal{L}(d(1\ldots\bar{t}), \mathcal{H}|\tau) f(\tau).
\]
Validation in Exponential Family

Estimation of models from exponential family is computationally attractive, since there exist **sufficient statistics** (data compression).

\[
\begin{array}{cccc}
V_1 & V_2 & \cdots & V_n \\
1 & \tau_1 & \tau_2 & \cdots & \tau_n & \bar{t}
\end{array}
\]

Computational advantages:

1. cheap evaluation of marginal likelihoods,
   \[ \mathcal{L}(\hat{l}d, \mathcal{M}) = \mathcal{L}(\hat{l}V, \mathcal{M}). \]
2. sufficient statistics are additive:
   - Learning statistics: \[ \hat{l}V(\tau = 2) = V_0 + V_1 + V_2. \]
   - Validation statistics: \[ \hat{v}V(\tau = 2) = \hat{l}V_0 + V_3 + V_4 + \ldots + V_{n+1}. \]
   - Overall statistics: \[ \hat{o}V = V_0 + V_1 + \ldots + V_{n+1}. \]

Validation is only minor computational overhead over learning.
Validation using Stabilized Forgetting

In exponential family, **stabilized forgetting** is defined via hypothesis:

\[ H_0: \text{model parameters are determined by (alternative) statistics } \overline{V}, \]

\[ H_1: \text{model parameters are determined by exponential window on recent data } d(t-k, \ldots, t), V_{\text{data}}. \]

By setting \( f(H_0) = \phi \), the posterior distribution of parameters is given by

\[
\overline{\text{lo}}V = (1 - \phi) V_{\text{data}} + \phi \overline{V}.
\]
Validation using Stabilized Forgetting

In exponential family, **stabilized forgetting** is defined via hypothesis:

\[ H_0: \text{ model parameters are determined by (alternative) statistics } \overline{V}, \]

\[ H_1: \text{ model parameters are determined by exponential window on recent data } d(t - k, \ldots, t), V_{\text{data}}. \]

By setting \( f(H_0) = \phi \), the posterior distribution of parameters is given by

\[ \overline{\text{lo}}V = (1 - \phi) V_{\text{data}} + \phi \overline{V}. \]

Validation is **inverse** problem to forgetting:

- (alternative) statistics \( \overline{V} \) are chosen as those of the validated model \( \overline{\text{lo}}V \).
- parallel run of two learning algorithms with stabilized forgetting:
  \( \phi \to 1 \), appropriate for **valid** model, \( \overline{\text{lo}}V \),
  \( \phi \to 0 \), appropriate for **invalid** model, \( \overline{\text{lo}}V \).
- test model likelihood of each variant
Application to Mixture Models

Learning:

- projection into exponential family (using conditional independence assumption).
- approximate on-line inference of component parameters (on-line EM, QB, VB, PB),
- sensitive to the choice of prior.
Application to Mixture Models

Learning:

- projection into exponential family (using conditional independence assumption).
- approximate on-line inference of component parameters (on-line EM, QB, VB, PB),
- sensitive to the choice of prior.

Validation:

<table>
<thead>
<tr>
<th>structure</th>
<th>‘correct’</th>
<th>Exponential</th>
<th>forgetting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x_0)$</td>
<td>estimated</td>
<td>fixed</td>
<td>fixed</td>
</tr>
<tr>
<td>$f(x_T)$</td>
<td>non-informative</td>
<td>informative</td>
<td>non-informative</td>
</tr>
<tr>
<td>cost</td>
<td>non-informative</td>
<td>using $MU$</td>
<td>non-informative</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>low</td>
<td>medium</td>
</tr>
</tbody>
</table>
Model validation was formalized as a special case of model selection and hypothesis testing,

We have presented a validation method which tests the assumption of time-invariance of the model,

Advantages:
- applicable for all models for which (i) learning procedure, and (ii) evaluation of data likelihood are available,
- computationally efficient algorithm for exponential family,

Disadvantage:
- fails to capture some cases of mis-modelling.