Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse

1AgroParisTech, Paris
2University Paris-Sud 11, Orsay
3SSB Group, Paris

Change point detection methods and Applications
AgroParisTech, Paris, September, 11 2008

joint work with Sylvain Arlot
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y} \text{ independent,}\]

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \((t_i = i/n)\).

Noise \(\epsilon\): \(\mathbb{E}[\epsilon_i] = 0\) and \(\mathbb{E}[\epsilon_i^2] = 1\).

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y} \text{ independent},\]

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \((t_i = i/n)\).

Noise \(\epsilon\):
\[\mathbb{E}[\epsilon_i] = 0 \quad \text{and} \quad \mathbb{E}[\epsilon_i^2] = 1.\]

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y} \quad \text{independent},\]

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \((t_i = i/n)\).

Noise \(\epsilon\): \(\mathbb{E}[\epsilon_i] = 0\) and \(\mathbb{E}[\epsilon_i^2] = 1\).

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y} \text{ independent,}\]

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \((t_i = i/n)\).

Noise \(\epsilon\):

\[\mathbb{E}[\epsilon_i] = 0 \quad \text{and} \quad \mathbb{E}[\epsilon_i^2] = 1.\]

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y}\] independent,

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \((t_i = i/n)\).

Noise \(\epsilon\): \(\mathbb{E}[\epsilon_i] = 0\) and \(\mathbb{E}[\epsilon_i^2] = 1\).

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Statistical framework: regression on a fixed design

\[(t_1, Y_1), \ldots, (t_n, Y_n) \in [0, 1] \times \mathcal{Y} \quad \text{independent},\]

\[Y_i = s(t_i) + \sigma_i \epsilon_i \in \mathcal{Y} = [0, 1] \text{ or } \mathbb{R}\]

Instants \(t_i\): deterministic \(t_i = i/n\).

Noise \(\epsilon\): \(\mathbb{E}[\epsilon_i] = 0\) and \(\mathbb{E}[\epsilon_i^2] = 1\).

Noise level: \(\sigma_i\) (heteroscedastic)

Goal: estimate \(s\)
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions.
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions.
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions
Change-points detection framework

\[ Y_i = s(t_i) + \sigma_i \epsilon_i \]

- \( s \): piecewise constant with high jumps
- Heteroscedastic noise

Purpose and strategy:
- Estimate \( s \) to recover most of the significant jumps w.r.t. the noise level.
- We choose our estimator among piecewise constant functions
Estimation vs. Selection

- A change-point in a noisy region
- We do not systematically want to recover it
- Use the quadratic risk
- Detected change-points are meaningful
Estimation vs. Selection

- A change-point in a noisy region
- We do not systematically want to recover it
- Use the quadratic risk
- Detected change-points are meaningful
Estimation vs. Selection

- A change-point in a noisy region
- We do not systematically want to recover it
- Use the quadratic risk
- Detected change-points are meaningful
Estimation vs. Selection

- A change-point in a noisy region
- We do not systematically want to recover it
- Use the quadratic risk

- Detected change-points are meaningful
Estimation vs. Selection

- A change-point in a noisy region
- We do not systematically want to recover it
- Use the quadratic risk
- Detected change-points are meaningful
Loss function, least-squares risk and contrast

- **Loss function:**
  \[ \ell (s, u) = \|s - u\|_n^2 := \frac{1}{n} \sum_{i=1}^{n} (s(t_i) - u(t_i))^2 \]

- **Least-squares risk of an estimator \( \hat{s} \):**
  \[ R_n(\hat{s}) := \mathbb{E} \left[ \ell (s, \hat{s}) \right] \]

- **Empirical risk:**
  \[ P_n \gamma(u) := \frac{1}{n} \sum_{i=1}^{n} \gamma(u, (t_i, Y_i)) , \]

  with \[ \gamma(u, (x, y)) = (u(x) - y)^2 \]
Loss function, least-squares risk and contrast

- **Loss function:**

\[ \ell(s, u) = \|s - u\|^2_n := \frac{1}{n} \sum_{i=1}^{n} (s(t_i) - u(t_i))^2 \]

- **Least-squares risk of an estimator \( \hat{s} \):**

\[ R_n(\hat{s}) := \mathbb{E}[\ell(s, \hat{s})] \]

- **Empirical risk:**

\[ P_n\gamma(u) := \frac{1}{n} \sum_{i=1}^{n} \gamma(u, (t_i, Y_i)) \]

with \( \gamma(u, (x, y)) = (u(x) - y)^2 \)
Loss function, least-squares risk and contrast

- **Loss function:**

\[
\ell(s, u) = \|s - u\|^2_n := \frac{1}{n} \sum_{i=1}^{n} (s(t_i) - u(t_i))^2
\]

- **Least-squares risk of an estimator \( \hat{s} \):**

\[
R_n(\hat{s}) := \mathbb{E}[\ell(s, \hat{s})]
\]

- **Empirical risk:**

\[
P_n\gamma(u) := \frac{1}{n} \sum_{i=1}^{n} \gamma(u, (t_i, Y_i)),
\]

with \( \gamma(u, (x, y)) = (u(x) - y)^2 \)
Least-squares estimator

- \( (I_{\lambda})_{\lambda \in \Lambda_m} \): partition of [0, 1]
- \( S_m \): linear space of piecewise constant functions on \( (I_{\lambda})_{\lambda \in \Lambda_m} \)
- Empirical risk minimizer over \( S_m \) (= model):
  \[
  \hat{s}_m \in \arg \min_{u \in S_m} P_n \gamma(u, \cdot) = \arg \min_{u \in S_m} \frac{1}{n} \sum_{i=1}^{n} (u(t_i) - Y_i)^2 .
  \]

- Regressogram
  \[
  \hat{s}_m = \sum_{\lambda \in \Lambda_m} \hat{\beta}_{\lambda} 1_{I_{\lambda}} \quad \hat{\beta}_{\lambda} = \frac{1}{\text{Card} \{ t_i \in I_{\lambda} \}} \sum_{t_i \in I_{\lambda}} Y_i .
  \]
Least-squares estimator

- \((I_\lambda)_{\lambda \in \Lambda_m}\): partition of \([0, 1]\)
- \(S_m\): linear space of piecewise constant functions on \((I_\lambda)_{\lambda \in \Lambda_m}\)
- Empirical risk minimizer over \(S_m\) (= model):

\[
\hat{s}_m \in \arg \min_{u \in S_m} P_n \gamma(u, \cdot) = \arg \min_{u \in S_m} \frac{1}{n} \sum_{i=1}^{n} (u(t_i) - Y_i)^2.
\]

- Regressogram

\[
\hat{s}_m = \sum_{\lambda \in \Lambda_m} \hat{\beta}_\lambda 1_{I_\lambda} \quad \hat{\beta}_\lambda = \frac{1}{\text{Card} \{t_i \in I_\lambda\}} \sum_{t_i \in I_\lambda} Y_i.
\]
Least-squares estimator

- \((I_\lambda)_{\lambda \in \Lambda_m}\): partition of \([0, 1]\)
- \(S_m\): linear space of piecewise constant functions on \((I_\lambda)_{\lambda \in \Lambda_m}\)
- Empirical risk minimizer over \(S_m\) (= model):

\[
\hat{s}_m \in \arg \min_{u \in S_m} P_n \gamma(u, \cdot) = \arg \min_{u \in S_m} \frac{1}{n} \sum_{i=1}^{n} (u(t_i) - Y_i)^2 .
\]

- Regressogram

\[
\hat{s}_m = \sum_{\lambda \in \Lambda_m} \hat{\beta}_\lambda 1_{I_\lambda} \quad \hat{\beta}_\lambda = \frac{1}{\text{Card} \{ t_i \in I_\lambda \}} \sum_{t_i \in I_\lambda} Y_i .
\]
Least-squares estimator

- \((I_\lambda)_{\lambda \in \Lambda_m}\): partition of \([0, 1]\)
- \(S_m\): linear space of piecewise constant functions on \((I_\lambda)_{\lambda \in \Lambda_m}\)
- Empirical risk minimizer over \(S_m\) (\(=\) model):

\[
\hat{s}_m \in \arg \min_{u \in S_m} P_n \gamma(u, \cdot) = \arg \min_{u \in S_m} \frac{1}{n} \sum_{i=1}^{n} (u(t_i) - Y_i)^2.
\]

- Regressogram

\[
\hat{s}_m = \sum_{\lambda \in \Lambda_m} \hat{\beta}_\lambda 1_{I_\lambda} \quad \hat{\beta}_\lambda = \frac{1}{\text{Card}\{t_i \in I_\lambda\}} \sum_{t_i \in I_\lambda} Y_i.
\]
Data \((t_1, Y_1), \ldots, (t_n, Y_n)\)
Goal: reconstruct the signal
How many breakpoints?

$D = 1$

$D = 3$

$D = 9$

$D = 36$
How many breakpoints?

- $D = 1$
- $D = 3$
- $D = 9$
- $D = 36$
The oracle

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
Model selection

\[(S_m)_{m \in \mathcal{M}} \rightarrow (\hat{s}_m)_{m \in \mathcal{M}} \rightarrow \hat{s}_m \]

Goals:

1. Oracle inequality (in expectation, or with a large probability):

\[\ell(s, \hat{s}_m) \leq C \inf_{\mathcal{M}_n} \{\ell(s, \hat{s}_m) + R(m, n)\}\]

2. Adaptivity (provided \((S_m)_{m \in \mathcal{M}_n}\) is well chosen)
Model selection

\[(S_m)_{m \in \mathcal{M}} \longrightarrow (\hat{s}_m)_{m \in \mathcal{M}} \longrightarrow \hat{s}_m \]

Goals:

- **Oracle inequality (in expectation, or with a large probability):**
  \[\ell(s, \hat{s}_m) \leq C \inf_{\mathcal{M}_n} \{\ell(s, \hat{s}_m) + R(m, n)\}\]

- Adaptivity (provided \((S_m)_{m \in \mathcal{M}_n}\) is well chosen)
Model selection

\[
(S_m)_{m \in \mathcal{M}} \longrightarrow (\hat{s}_m)_{m \in \mathcal{M}} \longrightarrow \hat{s}_m
\]

Goals:

- **Oracle inequality** (in expectation, or with a large probability):

\[
\ell(s, \hat{s}_m) \leq C \inf_{\mathcal{M}_n} \{\ell(s, \hat{s}_m) + R(m, n)\}
\]

- **Adaptivity** (provided \((S_m)_{m \in \mathcal{M}_n}\) is well chosen)
Collection of models

- For any $m \in \mathcal{M}_n$, $(I_\lambda)_{\lambda \in \Lambda_m}$ denotes a partition of $[0, 1]$ such that
  \[ I_\lambda = [t_{i_k}, t_{i_{k+1}}) \]

- $S_m$: linear space of piecewise constant functions on $(I_\lambda)_{\lambda \in \Lambda_m}$ with $\dim(S_m) = D_m$

\[ \forall 1 \leq D \leq n - 1, \quad \text{Card} \{ m \in \mathcal{M}_n \mid D_m = D \} = \binom{n-1}{D-1} \]
For any $m \in \mathcal{M}_n$, $(I_\lambda)_{\lambda \in \Lambda_m}$ denotes a partition of $[0, 1]$ such that

$$I_\lambda = [t_{i_k}, t_{i_k+1})$$

$S_m$: linear space of piecewise constant functions on $(I_\lambda)_{\lambda \in \Lambda_m}$ with $\dim(S_m) = D_m$

$$\forall 1 \leq D \leq n - 1, \quad \text{Card} \{ m \in \mathcal{M}_n \mid D_m = D \} = \binom{n - 1}{D - 1}$$
Collection of models

- For any $m \in \mathcal{M}_n$, $(I_\lambda)_{\lambda \in \Lambda_m}$ denotes a partition of $[0, 1]$ such that
  
  $$I_\lambda = [t_{i_k}, t_{i_{k+1}}]$$

- $S_m$: linear space of piecewise constant functions on $(I_\lambda)_{\lambda \in \Lambda_m}$ with $\dim(S_m) = D_m$

$$\forall 1 \leq D \leq n - 1, \quad \text{Card}\{m \in \mathcal{M}_n \mid D_m = D\} = \binom{n - 1}{D - 1}$$
The collection complexity idea

Usual approach:

- Bias-variance tradeoff
- Mallows’ $C_p$:

$$C_p(m) = P_n \gamma(\hat{s}_m) + 2\sigma^2 \frac{D_m}{n}$$

This approach is useless:

- Mallows’ $C_p$ overfits (Figure)
- The collection complexity is involved in this phenomenon

$C_p$ criterion

Oracle dimension: 5
The collection complexity idea

Usual approach:

- Bias-variance tradeoff
- Mallows’ $C_p$:

$$C_p(m) = P_n\gamma(\hat{s}_m) + 2\sigma^2 \frac{D_m}{n}$$

This approach is useless:

- Mallows’ $C_p$ overfits (Figure)
- The collection complexity is involved in this phenomenon
The collection complexity idea

Usual approach:

- Bias-variance tradeoff
- Mallows’ $C_p$:

$$C_p(m) = P_n \gamma(\hat{s}_m) + 2\sigma^2 \frac{D_m}{n}$$

This approach is useless:

- Mallows’ $C_p$ overfits (Figure)
- The collection complexity is involved in this phenomenon
The collection complexity idea

Usual approach:
- Bias-variance tradeoff
- Mallows’ $C_p$:

$$C_p(m) = P_n \gamma(\hat{s}_m) + 2\sigma^2 \frac{D_m}{n}$$

This approach is useless:
- Mallows’ $C_p$ overfits (Figure)
- The collection complexity is involved in this phenomenon
The collection complexity idea

Usual approach:

- Bias-variance tradeoff
- Mallows’ $C_p$:

$$C_p(m) = P_n \gamma(\hat{s}_m) + 2\sigma^2 \frac{D_m}{n}$$

This approach is useless:

- Mallows’ $C_p$ overfits (Figure)
- The collection complexity is involved in this phenomenon

\[\text{C}_p\text{ criterion}\]

Oracle dimension: 5
Model collection complexity and overfitting

Regular partitions

At least 10 points

At least 15 points

At least 5 points
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[ \forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_{n \gamma}(u) \]
\[ \hat{m} = \arg \min_{m \in M} \{ P_{n \gamma}(\hat{s}_m) + \text{pen}(D_m) \} \]

Algorithm 1': (=Algorithm 1)
\[ \forall D, \quad \hat{s}_{\tilde{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_{n \gamma}(u) \]
\[ \hat{D} = \arg \min_{D \in \mathcal{D}} \{ P_{n \gamma}(\hat{s}_{\tilde{m}(D)}) + \text{pen}(D_m) \} \]

where
\[ \tilde{S}_D = \bigcup_{m \mid D_m = D} S_m \]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[ \forall m, \quad \hat{s}_m = \text{arg min}_{u \in S_m} P_n \gamma(u) \]
\[ \hat{m} = \text{arg min}_{m \in \mathcal{M}} \{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \} \]

Algorithm 1': (=Algorithm 1)
\[ \forall D, \quad \hat{s}_{m(D)} = \text{arg min}_{u \in \tilde{S}_D} P_n \gamma(u) \]
\[ \hat{D} = \text{arg min}_{D \in \mathcal{D}} \{ P_n \gamma(\hat{s}_{m(D)}) + \text{pen}(D_m) \} \]

where
\[ \tilde{S}_D = \bigcup_{m|D_m=D} S_m \]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:

\[
\forall m, \quad \hat{s}_m = \arg\min_{u \in S_m} P_n \gamma(u)
\]

\[
\hat{m} = \arg\min_{m \in M} \{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \}
\]

Algorithm 1': (Algorithm 1)

\[
\forall D, \quad \hat{s}_{\hat{m}(D)} = \arg\min_{u \in \tilde{S}_D} P_n \gamma(u)
\]

\[
\hat{D} = \arg\min_{D \in \mathcal{D}} \{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \}
\]

where

\[
\tilde{S}_D = \bigcup_{m | D_m = D} S_m
\]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:

\[ \forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_n \gamma(u) \]

\[ \hat{m} = \arg \min_{m \in M} \left\{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \right\} \]

Algorithm 1': (=Algorithm 1)

\[ \forall D, \quad \hat{s}_{\hat{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_n \gamma(u) \]

\[ \hat{D} = \arg \min_{D \in D} \left\{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \right\} \]

where

\[ \tilde{S}_D = \bigcup_{m | D_m = D} S_m \]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[
\forall m, \quad \hat{s}_m = \arg\min_{u \in S_m} P_n \gamma(u)
\]

\[
\hat{m} = \arg\min_{m \in M} \{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \}
\]

Algorithm 1’: (Algorithm 1)
\[
\forall D, \quad \hat{s}_{\hat{m}(D)} = \arg\min_{u \in \tilde{S}_D} P_n \gamma(u)
\]

\[
\hat{D} = \arg\min_{D \in \mathcal{D}} \{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \}
\]

where
\[
\tilde{S}_D = \bigcup_{m | D_m = D} S_m
\]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[ \forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_n \gamma(u) \]
\[ \hat{m} = \arg \min_{m \in \mathcal{M}} \{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \} \]

Algorithm 1': (Algorithm 1)
\[ \forall D, \quad \hat{s}_{\hat{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_n \gamma(u) \]
\[ \hat{D} = \arg \min_{D \in D} \{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \} \]

where
\[ \tilde{S}_D = \bigcup_{m|D_m=D} S_m \]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[ \forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_n \gamma(u) \]
\[ \hat{m} = \arg \min_{m \in M} \left\{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \right\} \]

Algorithm 1': (=Algorithm 1)
\[ \forall D, \quad \hat{s}_{\hat{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_n \gamma(u) \]
\[ \hat{D} = \arg \min_{D \in D} \left\{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \right\} \]

where
\[ \tilde{S}_D = \bigcup_{m | D_m = D} S_m \]
Algorithm 1:

\[ \forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_n \gamma(u) \]

\[ \hat{m} = \arg \min_{m \in \mathcal{M}} \left\{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \right\} \]

Algorithm 1' (Algorithm 1):

\[ \forall D, \quad \hat{s}_{\hat{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_n \gamma(u) \]

\[ \hat{D} = \arg \min_{D \in D} \left\{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \right\} \]

where

\[ \tilde{S}_D = \bigcup_{m \mid D_m = D} S_m \]
Birgé and Massart (2001): Homoscedastic

Algorithm 1:
\[
\forall m, \quad \hat{s}_m = \arg \min_{u \in S_m} P_n \gamma(u)
\]
\[
\hat{m} = \arg \min_{m \in \mathcal{M}} \{ P_n \gamma(\hat{s}_m) + \text{pen}(D_m) \}
\]

Algorithm 1': (=Algorithm 1)
\[
\forall D, \quad \hat{s}_{\hat{m}(D)} = \arg \min_{u \in \tilde{S}_D} P_n \gamma(u)
\]
\[
\hat{D} = \arg \min_{D \in \mathcal{D}} \{ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D_m) \}
\]

where
\[
\tilde{S}_D = \bigcup_{m | D_m = D} S_m
\]
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
  - $\tilde{S}_D = \bigcup_{m, |D_m = D} S_m$
  - $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:
  $$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m \mid D_m = D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?

Segmentation in the mean of heteroscedastic data via resampling
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m \mid D_m = D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m|D_m=D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure

- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m: D_m=D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure

- Curves may be superimposed
- Complexity of $\tilde{S}_D$:
  \[
  \text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)
  \]

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m \mid D_m = D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure

- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m : D_m = D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m | D_m = D} S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Complexity measure in the homoscedastic setup

- Complexity of $S_m$: of order $D_m/n$
- $\tilde{S}_D = \bigcup_{m} \left| D_m = D \right| S_m$
- $\tilde{S}_D$: more complex than any $S_m$

Effective complexity measure
- Curves may be superimposed
- Complexity of $\tilde{S}_D$:

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

What about the heteroscedastic setting?
Heteroscedastic data

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:
$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:
$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:

$\frac{D_m}{n} + C_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:

Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:

Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:

$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:

We will use resampling to measure the complexity

Segmentation in the mean of heteroscedastic data via resampling
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:
$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:

Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:

Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:

$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:

We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:
$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:
Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:
Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:
$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:
We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:

Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:

Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:

$$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$$

is a reliable complexity measure

Heteroscedastic:

We will use resampling to measure the complexity
Strategy: Resampling with rich collections

With NOT TOO RICH collections of models

Homoscedastic:

Penalties like $c_1 \frac{D_m}{n}$ are reliable complexity measures

Heteroscedastic:

Resampling outperforms upon penalties $c_1 \frac{D_m}{n}$ (Arlot (08))

With RICH collections of models

Homoscedastic:

$c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right)$ is a reliable complexity measure

Heteroscedastic:

We will use resampling to measure the complexity
Cross-validation principle

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
Cross-validation principle
The training set: Computation of the estimator
The training set: Computation of the estimator
The test set: Quality assessment
The test set: Quality assessment
V-Fold cross-validation (VFCV)

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:
  \[
  (X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n)
  \]
  
  Training set \hspace{1cm} Test set $B_j$

  \[
  \hat{s}_m^{(-j)} = \arg \min_{u \in S_m} \left\{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right\}
  \]

  \[
  P_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(X_i, Y_i) \rightarrow P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right)
  \]

- $VFCV \rightarrow \hat{m} \in \arg \min_{\mathcal{M}_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right) \right\}$

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
V-Fold cross-validation (VFCV)

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:

$$(X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n)$$

Training set $\hat{s}_m^{(-j)} = \arg \min_{u \in S_m} \left\{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right\}$

$$P_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(X_i, Y_i) \quad \rightarrow \quad P_n^{(j)} \gamma(\hat{s}_m^{(-j)})$$

$\cdot$ VFCV $\rightarrow \hat{m} \in \arg \min_{\mathcal{M}_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma(\hat{s}_m^{(-j)}) \right\}$
V-Fold cross-validation (VFCV)

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:

$$(X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n)$$

<table>
<thead>
<tr>
<th>Training set</th>
<th>Test set $B_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_m^{(-j)} = \arg \min_{u \in S_m} \left{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right}$</td>
<td></td>
</tr>
<tr>
<td>$p_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(x_i, y_i) \rightarrow P_n^{(j)} \gamma \left( s_m^{(-j)} \right)$</td>
<td></td>
</tr>
</tbody>
</table>

- VFCV $\rightarrow \hat{m} \in \arg \min_{M_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( s_m^{(-j)} \right) \right\}$
**V-Fold cross-validation (VFCV)**

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:

  $$
  (X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n)
  $$

  Training set  

  Test set $B_j$

$$
\hat{s}_m^{(-j)} = \arg \min_{u \in S_m} \left\{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right\}
$$

$$
P_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(X_i, Y_i) \quad \rightarrow \quad P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right)
$$

- VFCV $\rightarrow \hat{m} \in \arg \min_{\mathcal{M}_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right) \right\}$
V-Fold cross-validation (VFCV)

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:

$$\left\{ (X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n) \right\}$$

Training set \hspace{1cm} Test set $B_j$

$$\hat{s}_m(-j) = \arg \min_{u \in S_m} \left\{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right\}$$

$$P_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(X_i, Y_i) \quad \rightarrow \quad P_n^{(j)} \gamma \left( \hat{s}_m(-j) \right)$$

\[ \text{VFCV} \rightarrow \hat{m} \in \arg \min_{M_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_m(-j) \right) \right\} \]
V-Fold cross-validation (VFCV)

- Randomly split the data into $V$ disjoint subsets ($B_j$) of size $p \approx n/V$
- For each $1 \leq j \leq V$:

\[
\begin{align*}
(X_1, Y_1), \ldots, (X_{n-p}, Y_{n-p}), &\quad (X_{n-p+1}, Y_{n-p+1}), \ldots, (X_n, Y_n)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Training set</th>
<th>Test set $B_j$</th>
</tr>
</thead>
</table>

\[
\hat{s}_m^{(-j)} = \arg \min_{u \in S_m} \left\{ \frac{1}{n-p} \sum_{i=1}^{n-p} \gamma(u, (X_i, Y_i)) \right\}
\]

\[
P_n^{(j)} = \frac{1}{p} \sum_{i=n-p+1}^{n} \delta(X_i, Y_i) \quad \rightarrow \quad P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right)
\]

- VFCV $\rightarrow \hat{m} \in \arg \min_{\mathcal{M}_n} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_m^{(-j)} \right) \right\}$
BM, VFCV: choosing the number of breakpoints

First step:

∀D, \( \hat{m}(D) = \text{Argmin}_{m \mid D_m = D} P_n \gamma(\hat{s}_m) \)

Second Step:

- **BM (Homoscedastic):**
  \[
P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right]
  \]

- **VFCV (Homoscedastic and Heteroscedastic):**
  \[
  \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma(\hat{s}_{\hat{m}^{(-j)}(D)}) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right]
  \]
BM, VFCV: choosing the number of breakpoints

- First step:

\[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m|D_m=D} P_n \gamma(\hat{s}_m) \]

- Second Step:
  1. BM (Homoscedastic):

\[
P_n \gamma(\hat{s}_m(D)) + \text{pen}(D) \approx \mathbb{E} \left[ \| s - \hat{s}_m(D) \|^2 \right]
\]

  2. VFCV (Homoscedastic and Heteroscedastic):

\[
\frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_{m(-j)}(D) \right) \approx \mathbb{E} \left[ \| s - \hat{s}_m(D) \|^2 \right]
\]

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
BM, VFCV: choosing the number of breakpoints

- First step:

\[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m|D_m = D} P_n \gamma(\hat{s}_m) \]

- Second Step:
  1. BM (Homoscedastic):

\[ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right] \]

  2. VFCV (Homoscedastic and Heteroscedastic):

\[ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_{\hat{m}^{(j)}(D)} \right) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right] \]
BM, VFCV: choosing the number of breakpoints

- First step:
  \[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m \mid D_m = D} P_n \gamma(\hat{s}_m) \]

- Second Step:
  1. BM (Homoscedastic):
     \[ P_n \gamma(\hat{s}_{\hat{m}(D)}) + \text{pen}(D) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right] \]
  2. VFCV (Homoscedastic and Heteroscedastic):
     \[ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma(\hat{s}_{\hat{m}(-j)(D)}) \approx \mathbb{E} \left[ \| s - \hat{s}_{\hat{m}(D)} \|^2 \right] \]
BM, VFCV: performance (number of breakpoints)

### Homoscedastic

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Ideal</th>
<th>VFCV</th>
<th>BM</th>
<th>$C_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.19 $\pm$ 0.28</td>
<td>3.35 $\pm$ 0.35</td>
<td>2.96 $\pm$ 0.36</td>
<td>62.1 $\pm$ 7.1</td>
</tr>
<tr>
<td>3</td>
<td>4.09 $\pm$ 0.24</td>
<td>5.12 $\pm$ 0.3</td>
<td>4.84 $\pm$ 0.32</td>
<td>19.3 $\pm$ 1.2</td>
</tr>
<tr>
<td>5</td>
<td>4.04 $\pm$ 0.57</td>
<td>7.33 $\pm$ 0.9</td>
<td>38.5 $\pm$ 2.8</td>
<td>131 $\pm$ 14</td>
</tr>
<tr>
<td>6</td>
<td>2.13 $\pm$ 0.18</td>
<td>3.47 $\pm$ 0.28</td>
<td>5.1  $\pm$ 0.39</td>
<td>94.7 $\pm$ 7.7</td>
</tr>
</tbody>
</table>

### Heteroscedastic

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Ideal</th>
<th>VFCV</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.19 $\pm$ 0.28</td>
<td>3.35 $\pm$ 0.35</td>
<td>2.96 $\pm$ 0.36</td>
</tr>
<tr>
<td>3</td>
<td>4.09 $\pm$ 0.24</td>
<td>5.12 $\pm$ 0.3</td>
<td>4.84 $\pm$ 0.32</td>
</tr>
<tr>
<td>5</td>
<td>4.04 $\pm$ 0.57</td>
<td>7.33 $\pm$ 0.9</td>
<td>38.5 $\pm$ 2.8</td>
</tr>
<tr>
<td>6</td>
<td>2.13 $\pm$ 0.18</td>
<td>3.47 $\pm$ 0.28</td>
<td>5.1  $\pm$ 0.39</td>
</tr>
</tbody>
</table>

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
Summary of main ideas

- Resampling turns out to be reliable with rich collections
  - Homoscedastic setup: Resampling performs almost as well as BM
  - Heteroscedastic setup: Resampling strongly outperforms upon BM
Summary of main ideas

- Resampling turns out to be reliable with rich collections.

- Homoscedastic setup: Resampling performs almost as well as BM.

- Heteroscedastic setup: Resampling strongly outperforms upon BM.
Summary of main ideas

- Resampling turns out to be reliable with rich collections

- Homoscedastic setup: Resampling performs almost as well as BM

- Heteroscedastic setup: Resampling strongly outperforms upon BM
Overfitting of ERM

ERM only takes into account the fit to the data

Overfitting may occur when

\[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m | D_m = D} P_n \gamma(u) \]

Overfitting is all the more strong as \( \tilde{S}_D \) is large

In our setting,

\[ \text{Card } \{ m | D_m = D \} = \binom{n-1}{D-1} \]

Idea:

ERM may be replaced by resampling to provide \( \{ \hat{m}(D) \}_D \)
Overfitting of ERM

- **ERM only takes into account the fit to the data**
- **Overfitting may occur when**
  \[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m|D_m=D} P_n \gamma(u) \]

- Overfitting is all the more strong as \( \tilde{S}_D \) is large
- In our setting,
  \[ \text{Card} \{ m | D_m = D \} = \binom{n-1}{D-1} \]

**Idea:**

ERM may be replaced by resampling to provide \( \{ \hat{m}(D) \}_D \)
Overfitting of ERM

- ERM only takes into account the fit to the data
- Overfitting may occur when

\[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m \mid D_m = D} P_n \gamma(u) \]

- Overfitting is all the more strong as \( \tilde{S}_D \) is large
- In our setting,

\[ \text{Card} \left( \{ m \mid D_m = D \} \right) = \binom{n - 1}{D - 1} \]

Idea:

ERM may be replaced by resampling to provide \( \{ \hat{m}(D) \}_D \)
Overfitting of ERM

- ERM only takes into account the fit to the data
- Overfitting may occur when
  \[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m | D_m = D} \mathbb{P}_n \gamma(u) \]
- Overfitting is all the more strong as \( \tilde{S}_D \) is large
- In our setting,
  \[ \text{Card} (\{m \mid D_m = D\}) = \binom{n-1}{D-1} \]

Idea:

ERM may be replaced by resampling to provide \( \{\hat{m}(D)\}_D \)
Overfitting of ERM

- ERM only takes into account the fit to the data
- Overfitting may occur when
  \[ \forall D, \hat{m}(D) = \text{Argmin}_{m \mid D_m = D} P_n \gamma(u) \]
- Overfitting is all the more strong as \( \tilde{S}_D \) is large
- In our setting,
  \[ \text{Card} (\{m \mid D_m = D\}) = \binom{n - 1}{D - 1} \]

Idea:

ERM may be replaced by resampling to provide \( \{\hat{m}(D)\}_D \)
Overfitting of ERM

ERM only takes into account the fit to the data
Overfitting may occur when

\[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m \mid D_m = D} P_n \gamma(u) \]

Overfitting is all the more strong as \( \tilde{S}_D \) is large
In our setting,

\[ \text{Card} \left( \{ m \mid D_m = D \} \right) = \binom{n-1}{D-1} \]

Idea:

ERM may be replaced by resampling to provide \( \{ \hat{m}(D) \}_D \)
Overfitting of ERM

- ERM only takes into account the fit to the data
- Overfitting may occur when
  \[ \forall D, \quad \hat{m}(D) = \text{Argmin}_{m\mid D_m = D} P_n\gamma(u) \]

- Overfitting is all the more strong as \( \tilde{S}_D \) is large
- In our setting,
  \[ \text{Card} (\{m \mid D_m = D\}) = \binom{n-1}{D-1} \]

**Idea:**

ERM may be replaced by resampling to provide \( \{\hat{m}(D)\}_D \)
Algorithms

Goal:

See whether resampling outperforms upon ERM

Alternatives:

- ERM:

  \[ \forall D, \quad \hat{m}(D) = \arg \min_{m|D_m=D} P_n \gamma(\hat{s}_m) \]

- Leave-one-out (LOO):

  \[ \forall D, \quad \hat{m}(D) = \arg \min_{m|D_m=D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma(\hat{s}_m^{(-i)}) \]
Goal:

See whether resampling outperforms upon ERM

Alternatives:

- **ERM:**
  \[
  \forall D, \quad \hat{m}(D) = \arg\min_{m|D_m=D} P_n \gamma(\hat{s}_m)
  \]

- **Leave-one-out (LOO):**
  \[
  \forall D, \quad \hat{m}(D) = \arg\min_{m|D_m=D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma(\hat{s}_m^{(-i)})
  \]
Algorithms

Goal:

See whether resampling outperforms upon ERM

Alternatives:

ERM:

$$\forall D, \hat{m}(D) = \arg \min_{m|D_m = D} p_n \gamma(\hat{s}_m)$$

Leave-one-out (LOO):

$$\forall D, \hat{m}(D) = \arg \min_{m|D_m = D} \frac{1}{n} \sum_{i=1}^{n} p_n(i) \gamma(\hat{s}_{m}^{(i)})$$
Goal:

See whether resampling outperforms upon ERM

Alternatives:

- **ERM:**

\[ \forall D, \quad \hat{m}(D) = \arg \min_{m \mid D_m = D} P_n \gamma(\hat{s}_m) \]

- Leave-one-out (LOO):

\[ \forall D, \quad \hat{m}(D) = \arg \min_{m \mid D_m = D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma(\hat{s}_m^{(-i)}) \]
Algorithms

Goal:

See whether resampling outperforms upon ERM

Alternatives:

- ERM:
  \[
  \forall D, \quad \hat{m}(D) = \arg \min_{m|D_m=D} P_n \gamma(\hat{s}_m)
  \]

- Leave-one-out (LOO):
  \[
  \forall D, \quad \hat{m}(D) = \arg \min_{m|D_m=D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma(\hat{s}_m^{(-i)})
  \]
Quality of the segmentations: Homoscedastic
Quality of the segmentations: Heteroscedastic

**Segmentation in the mean of heteroscedastic data via resampling**

Alain Celisse
Overfitting with ERM: dimension 6

Segmentation in the mean of heteroscedastic data via resampling

Alain Celisse
Overfitting with ERM: dimension 7
Overfitting with ERM: dimension 8
Overfitting with ERM: dimension 9
Overfitting with ERM: dimension 10

Segmentation in the mean of heteroscedastic data via resampling
Global algorithm description

1. LOO at the first step to choose the best segmentation for each dimension:

\[ \forall D, \quad \hat{m}(D) = \arg \min_{m|D_m=D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma \left( \hat{s}_m^{(-i)} \right) \]

2. Use the VFCV to choose the number of breakpoints

\[ \hat{D} = \arg \min_D \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_{\hat{m}^{(-j)}(D)} \right) \right\} \]
Global algorithm description

1. LOO at the first step to choose the best segmentation for each dimension:

$$\forall D, \quad \hat{m}(D) = \arg\min_{m|D_m=D} \frac{1}{n} \sum_{i=1}^{n} P_n^{(i)} \gamma \left( \hat{s}_m^{(-i)} \right)$$

2. Use the VFCV to choose the number of breakpoints

$$\hat{D} = \arg\min_{D} \left\{ \frac{1}{V} \sum_{j=1}^{V} P_n^{(j)} \gamma \left( \hat{s}_{\hat{m}^{(-j)}(D)} \right) \right\}$$
The Bt474 Cell lines

- These are epithelial cells
- Obtained from human breast cancer tumors
- A test genome is compared to a reference male genome
- We only consider chromosomes 1 and 9
The Bt474 Cell lines

- These are epithelial cells
- Obtained from human breast cancer tumors
- A test genome is compared to a reference male genome
- We only consider chromosomes 1 and 9
The Bt474 Cell lines

- These are epithelial cells
- Obtained from human breast cancer tumors
- A test genome is compared to a reference male genome
- We only consider chromosomes 1 and 9
The Bt474 Cell lines

- These are epithelial cells
- Obtained from human breast cancer tumors
- A test genome is compared to a reference male genome
- We only consider chromosomes 1 and 9
Results: Chromosome 9

LOO+VFCV

Homoscedastic model (Picard et al. (05))

Heteroscedastic model (Picard et al. (05))
Results: Chromosome 1

Homoscedastic model (Picard et al. (05))

Heteroscedastic model (Picard et al. (05))
Conclusion

- We have designed a resampling-based procedure
- It may be effectively applied to the change-points detection problem
- It behaves almost as well as BM in a homoscedastic framework
- It works well in a fully heteroscedastic setup

- This methodology may be extended to other resampling schemes
- This algorithm relies on the dimension as a criterion to gather models: We may think about alternative criteria.
We have designed a resampling-based procedure.

- It may be effectively applied to the change-points detection problem.
  - It behaves almost as well as BM in a homoscedastic framework.
  - It works well in a fully heteroscedastic setup.

- This methodology may be extended to other resampling schemes.
- This algorithm relies on the dimension as a criterion to gather models: We may think about alternative criteria.
We have designed a resampling-based procedure

- It may be effectively applied to the change-points detection problem
- It behaves almost as well as BM in a homoscedastic framework
- It works well in a fully heteroscedastic setup

- This methodology may be extended to other resampling schemes
- This algorithm relies on the dimension as a criterion to gather models: We may think about alternative criteria.
Conclusion

- We have designed a resampling-based procedure
- It may be effectively applied to the change-points detection problem
- It behaves almost as well as BM in a homoscedastic framework
- It works well in a fully heteroscedastic setup

- This methodology may be extended to other resampling schemes
- This algorithm relies on the dimension as a criterion to gather models: We may think about alternative criteria.
We have designed a resampling-based procedure
It may be effectively applied to the change-points detection problem
It behaves almost as well as BM in a homoscedastic framework
It works well in a fully heteroscedastic setup

This methodology may be extended to other resampling schemes
This algorithm relies on the dimension as a criterion to gather models: We may think about alternative criteria.
Conclusion

- We have designed a resampling-based procedure
- It may be effectively applied to the change-points detection problem
- It behaves almost as well as BM in a homoscedastic framework
- It works well in a fully heteroscedastic setup

- This methodology may be extended to other resampling schemes
- **This algorithm relies on the dimension as a criterion to gather models**: We may think about alternative criteria.
Thank you!
<table>
<thead>
<tr>
<th>$s$</th>
<th>$\sigma_{c,v}$</th>
<th>ERM+VFCV</th>
<th>LOO+VFCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>1.88 ± 0.088</td>
<td>1.9 ± 0.091</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.65 ± 0.11</td>
<td>3.77 ± 0.12</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1.75 ± 0.007</td>
<td>1.75 ± 0.007</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.99 ± 0.015</td>
<td>1.99 ± 0.015</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1 ± 5e-017</td>
<td>1 ± 5e-017</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>3.03 ± 0.2</td>
<td>2.8 ± 0.19</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>3.65 ± 0.056</td>
<td>3.46 ± 0.053</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>4.65 ± 0.09</td>
<td>4.37 ± 0.087</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.38 ± 0.23</td>
<td>3.68 ± 0.22</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>6.91 ± 0.34</td>
<td>6.22 ± 0.31</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3.81 ± 0.17</td>
<td>3.76 ± 0.17</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.77 ± 0.012</td>
<td>1.77 ± 0.012</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>2.86 ± 0.05</td>
<td>2.67 ± 0.045</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4.64 ± 0.092</td>
<td>4.33 ± 0.093</td>
</tr>
</tbody>
</table>