# Variable selection and parameter tuning in high-dimensional prediction

Christoph Bernau and Anne-Laure Boulesteix

Institut für Medizinische Informationsverarbeitung, Biometrie und Epidemiologie
Ludwig-Maximilians-Universität München

COMPSTAT 2010, 23. August 2010

# Prediction based on high-dimensional data

**X**: a $n \times p$ matrix containing $n$ observations of $p$ variables, possibly with $n \ll p$.
**Examples:** microarray data, chemometric data, proteomic data, metabolomic data

| | $X_1$ | ... | ... | $X_p$ |
|---|---|---|---|---|
| *Pat* 1 | . | . | . | . |
| ... | . | . | . | . |
| ... | . | . | . | . |
| *Pat n* | . | . | . | . |

**Y**: a response variable to be predicted.

**Examples:** responder/non-responder, diseased/healthy

# Variable selection

- Many variables are irrelevant for the prediction problem.
- **Variable selection** is often useful as a **preliminary step** to model selection.
- **Example:**
  1. Rank the variables according the absolute value of the t-statistic.
  2. Select the $p^* = 100$ top-ranking variables and use them for model selection.

Boulesteix et al, 2008. Evaluating microarray-based classifiers. Cancer Informatics 6:77–97.

# Variable selection and cross-validation

- In small sample settings, prediction error rates are often estimated through cross-validation (CV) or related approaches (repeated subsampling, bootstrap).
- It is then essential to consider variable selection as a part of model selection and perform it **for each CV iteration successively**.
- Otherwise the error rate may be considerably underestimated (Ambroise and McLahan 2002).

A.-L. Boulesteix, 2007. WilcoxCV: an R package for fast variable in cross-validation. Bioinformatics 23:1702–1704.

# Parameter tuning

- Many classification methods involve a **parameter** that has to be tuned.
- **Examples:**
  - the number $k$ of nearest neighbors in the *kNN* algorithm
  - the penalty $\lambda$ in penalized regression
  - the number of components in PLS-DA
- It is common practice to choose the value of the parameter through **internal cross-validation**.

# Internal cross-validation (CV)

- Error rates are estimated via external CV corresponding to partition $\mathcal{S} = \cup S_k$.
- In each learning set $\mathcal{S} \setminus S_k$:
  - Internal CV is performed with different values $\theta_1, \ldots, \theta_m$ of the parameter.
  - The value $\theta^*$ yielding the lowest error rate is selected.
  - $\theta^*$ is used for model selection based on $\mathcal{S} \setminus S_k$.
- **In internal CV, error rates are calculated, but the goal is only to determine $\theta^*$, not to estimate the error rates.**

# Research question

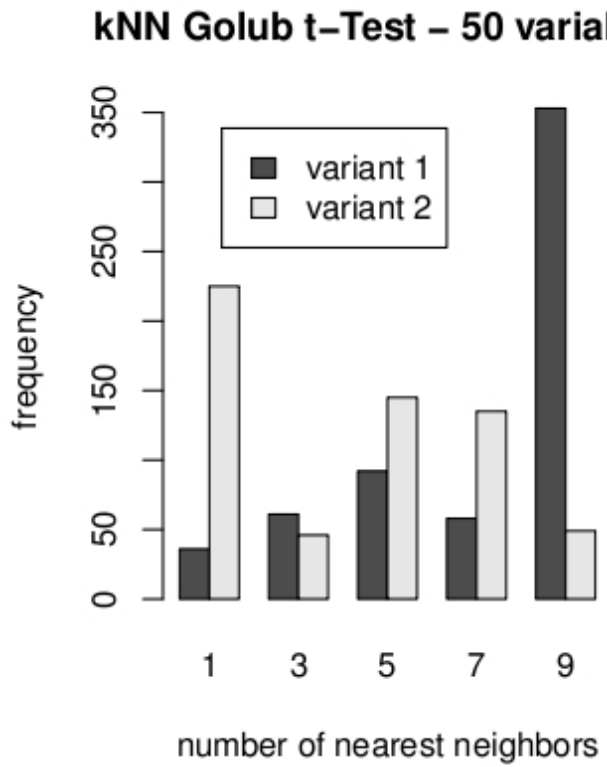
Should we perform variable selection before internal CV (V1) or repeat variable selection for each *internal* CV iteration (V2)?

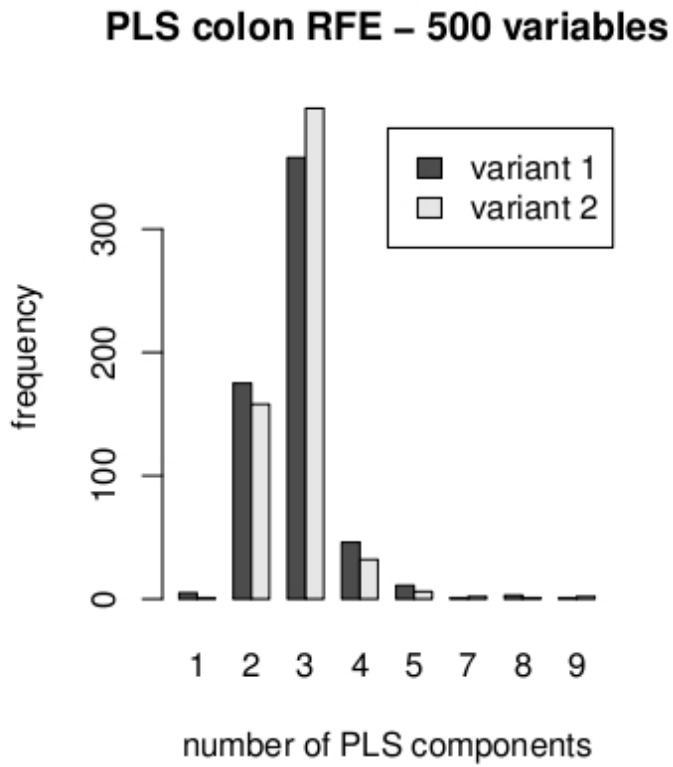
- For external CV, variable selection must *always* be repeated for each iteration, but for internal CV the answer is not obvious.
- V2 is time consuming: for example, in LOO-CV, variable selection has to be performed $n \times (n-1)$ times.

# Our empirical study

- Two real data microarray sets
- Two classification methods: *kNN* and PLS+LDA
- Two variable selection methods: t-statistic and RFE
- 100 times 5-fold-CV for error estimation (external CV)
- 5 times 3-fold-CV for parameter tuning (internal CV)

# Result 1: V2 selects more complex models than V1



# Result 2: The error rates of V1 and V2 are similar

| kNN | | Golub data | | | | colon cancer data | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | t-test | | RFE | | t-test | | RFE | |
| | | V1 | V2 | V1 | V2 | V1 | V2 | V1 | V2 |
| 20 genes | mean | 7.8% | 7.4% | 5.8% | 6.1% | 16.8% | 18.8% | 21.6% | 23.3% |
| | std. dev. | 2.6% | 2.8% | 2.5% | 2.9% | 1.9% | 2.4% | 3.3% | 4.1% |
| 50 genes | mean | 5.9% | 5.5% | 1.9% | 2.2% | 16.4% | 19.9% | 16.9% | 18.5% |
| | std. dev. | 2.4% | 2.7% | 1.8% | 1.7% | 1.6% | 1.9% | 3.3% | 3.0% |

No clear difference between V1 and V2 in terms of error rate (variances are high!)

# Why does V2 lead to more complex models?

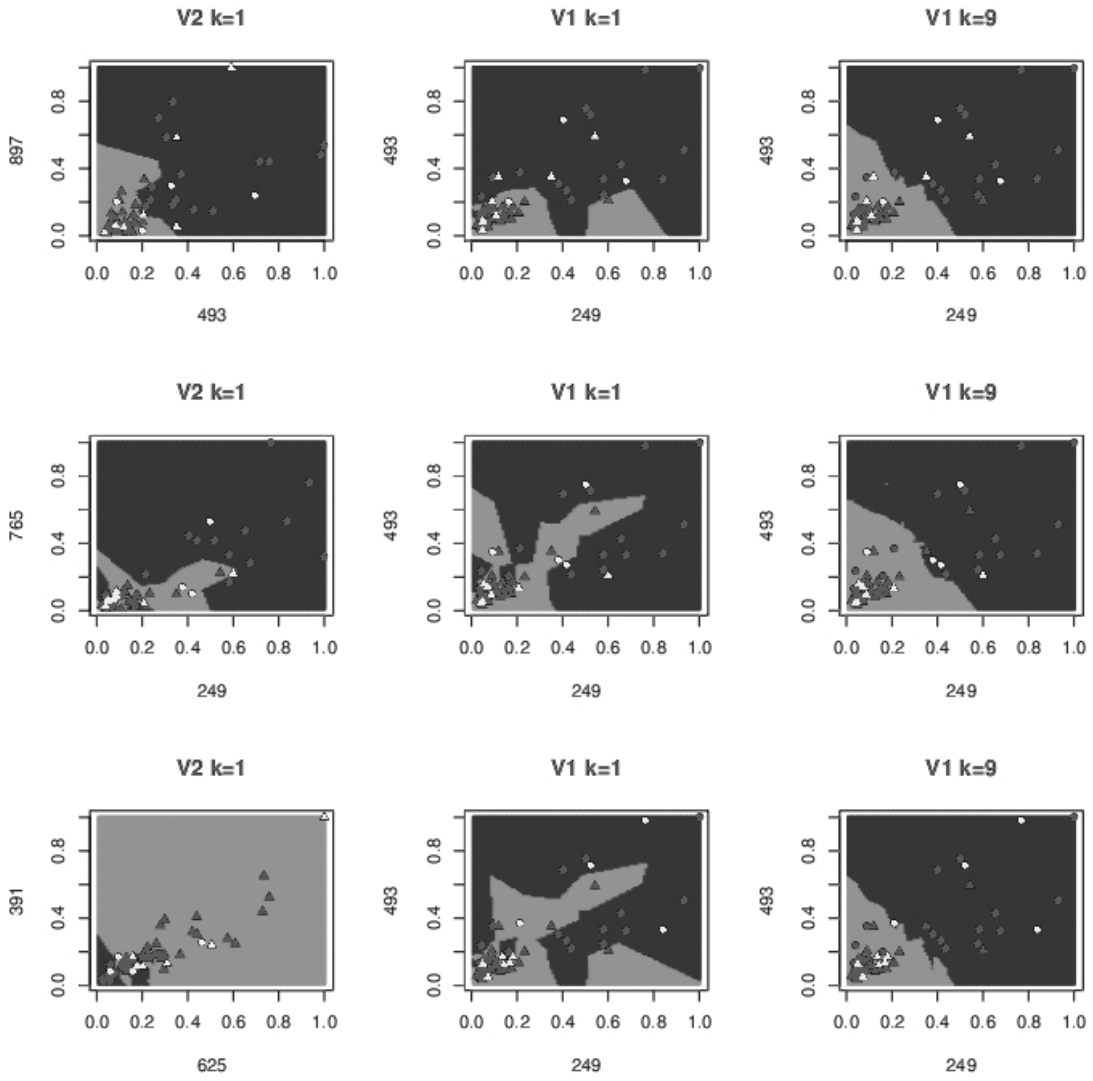
- In V1 the variables are selected based on the external learning set $\mathcal{S} \setminus \mathcal{S}_k$.
- In V2 the variables are selected based the smaller learning set $(\mathcal{S} \setminus \mathcal{S}_k) \setminus \mathcal{S}_{kj}$, on which the models are fit in internal CV.

$\longrightarrow$ In V2 the variables better discriminate the two classes in the learning set $(\mathcal{S} \setminus \mathcal{S}_k) \setminus \mathcal{S}_{kj}$ than in V1.

$\longrightarrow$ In V2 complex models perform better.

$\longrightarrow$ In V1 complex models are fit to "bad variables" and thus lead to worse results.

# Why does V2 lead to more complex models?



# Further remarks

- V2 possibly leads to *too complex models*: since the internal learning sets are small, it is easier to find variables that separate the classes perfectly (and lead to comparatively good performance for complex models).
- A problem of V2 is that the parameter is chosen based on sets of variables but applied to another set of variables.
- A problem of V1 is that, for well-separated data sets, all parameter values yield an error rate of 0%
  $\longrightarrow$ no tuning is performed in this case.

# Conclusion and outlook

- No definitive answer in terms of error rate
- V2 is more intuitive but has some inconveniences and is time consuming.
- **Outlook:** Methods with intrinsic variable selection (such as lasso) are implicitly based on V2. Do they also lead to too complex models?