Izbira in konstrukcija napovednih spremenljivk

Ljupčo Todorovski, Univerza v Ljubljani
http://kt.ijs.si/~ljupco/

Variable Selection and Ranking

The Problem of Variable Selection

- Algorithms for supervised machine learning
  - Designed to select the variables most appropriate for prediction
  - **In theory: more variables** should lead to **better predictions**
- But what really happens **in practice**?
  - Adding **irrelevant**/distracting **variables** leads to **worse predictions**
  - Empirically shown for various machine learning algorithms
- Decision trees
  - Variables on lower levels of the tree chosen on few data
  - This often leads to choosing irrelevant variables
- Nearest neighbors: number of examples necessary increases exponentially with the number of irrelevant variables

The Tasks of Variable Selection and Ranking

- Inputs
  - $X=\{X_1, ..., X_p\}$ is a set of p input variables (features)
  - Y is a single target variable (target)
  - Training data $D=(D_X, D_Y)$, where
    - D_X (Nxp matrix): N instances of observations of X
    - D_Y ($Nx1$ vector): observations of Y for the instances (rows)

X_1	X_2	...	X_p	Y
x_{11}	x_{12}	...	x_{1p}	y_1
x_{21}	x_{22}	...	x_{2p}	y_2
...	...		...	...
x_{N1}	x_{N2}	...	x_{Np}	y_N

- Find (the task of **variable/feature selection**)
  - Input variables X_i that are **relevant for predicting** the target Y
- Find (the task of **variable/feature ranking**)
  - The ranking of the input variables in X wrt their **relevance for predicting** the target Y
- Central notion: **variable relevance** for prediction

Measuring Variable Relevance

- Function r_D: $X \rightarrow R$
  - Takes an input variable V as an argument
  - $r_D(V)$ is a real number indicating the relevance of V for predicting Y
  - In the context of a given data set D
- Central question: how can we measure $r_D(V)$?
  - A trivial step from variable relevance to ranking: sort
- A taxonomy of measurement methods
  - Model-independent measures
    - Unsupervised (use only D_X)
    - Supervised (use also D_Y, i.e., the whole data set D)
  - Model-dependent measures (only supervised, use D)

Motivating and Illustrative Example: Arcene

- Arcene data set comes from UCI ML Repository
- Mass spectrometry data for cancer and healthy patients
  - Each input variable corresponds to the concentration of proteins having a given mass
  - Target corresponds to the patient health condition: cancer, healthy
- Data
  - **7000 real input variables, 3000 artificial irrelevant input variables**
  - 100 train examples to be used for variable selection
  - 100 validation examples to be used to evaluate variable sets
- **Identify classes of proteins with a certain mass** that can be used as **indicators of cancer**

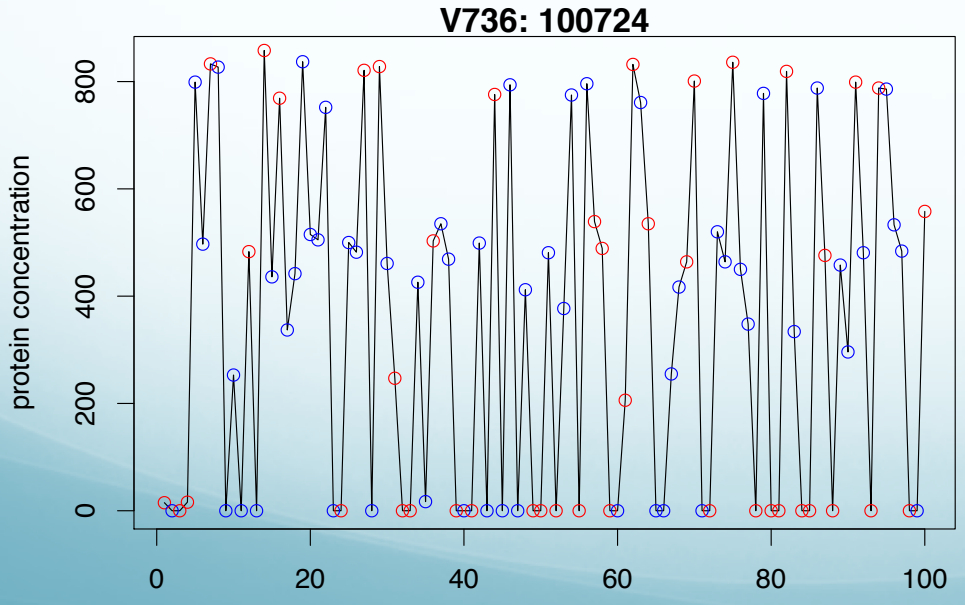
The Benefits of Variable Ranking/Selection

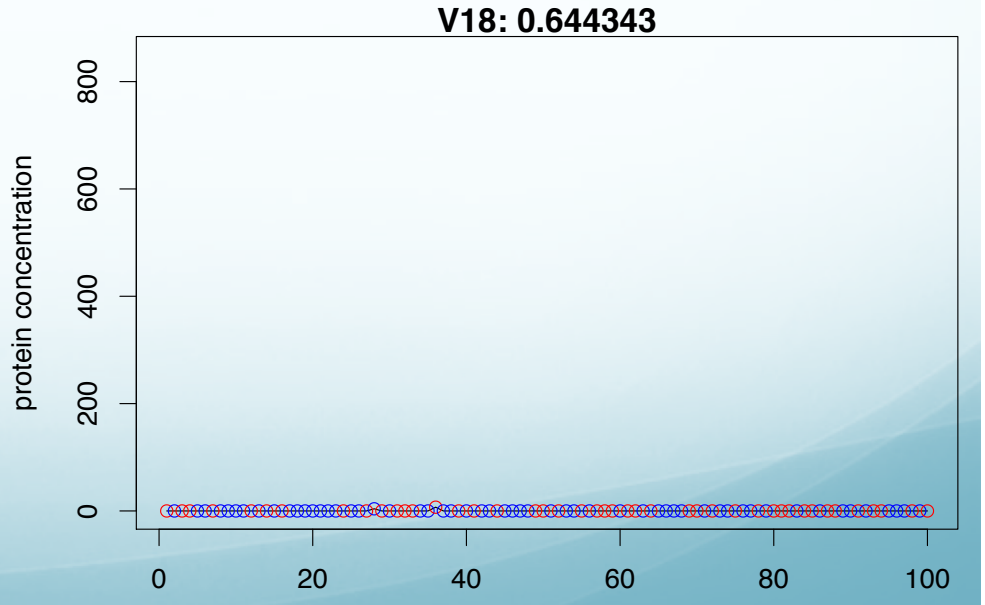
1. **Improve the predictive performance** of machine learning
   - Focusing on relevant variables improves the prediction
2. **Speed-up the learning** process
   - The computational complexity of machine learning algorithms often dependent on number of input variables
   - But, be careful: feature selection also takes time and space
3. **Improve the interpretability** of the learned models
   - Variable selection yields more compact predictive models, hence
   - Tighter focus on the most relevant variables

Methods for measuring variable relevance

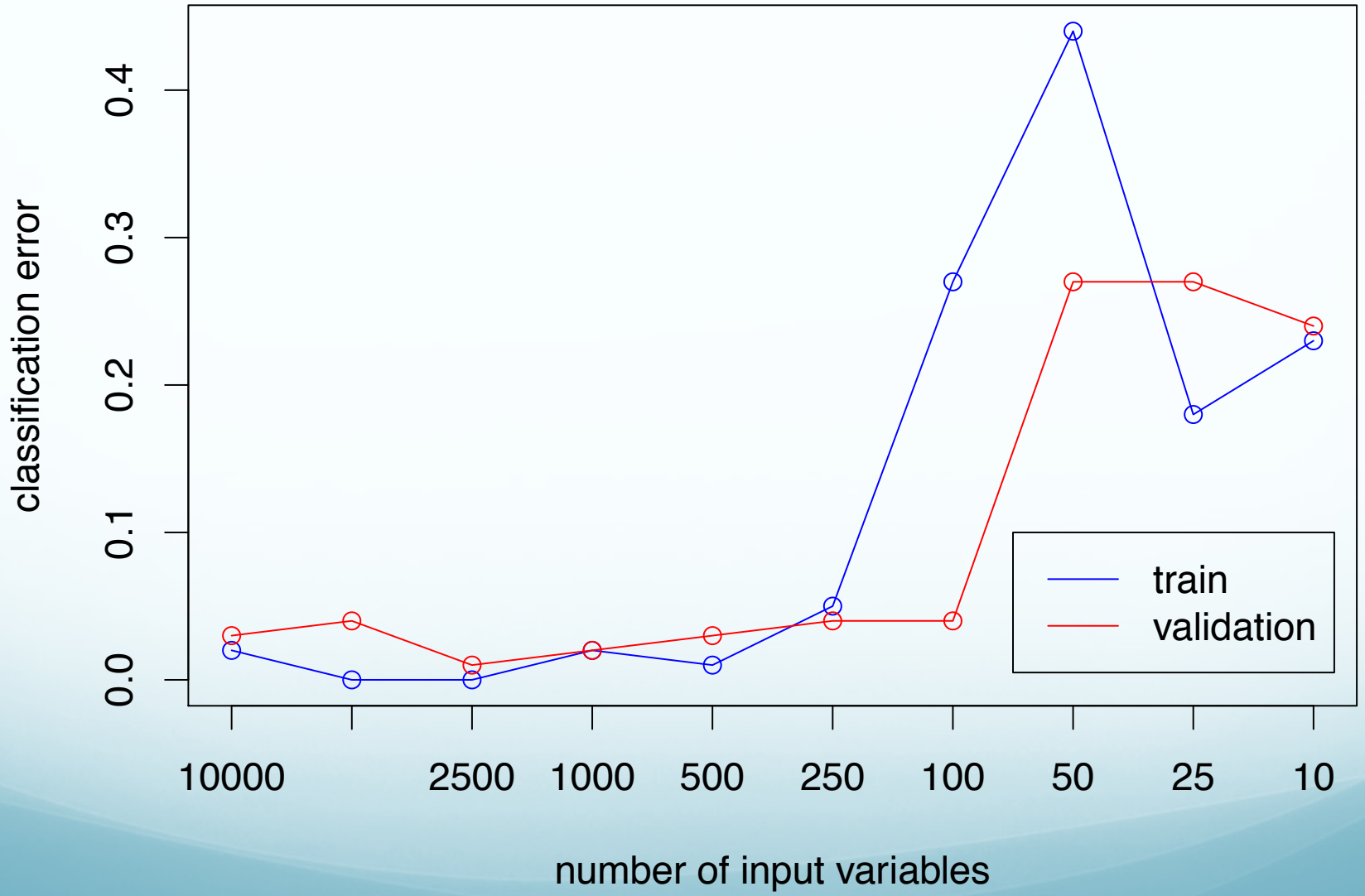
Unsupervised: Variable Variance

- Intuition: variable relevance correlated with variable variance
  - Low-variance variables does not hold predictive power
  - See the graph for the input variable V18 (right-hand side, below)
- Unbiased sample variance: $Var(V) = 1/(N-1) * \Sigma_i (v_i - Mean(V))$
  - Where $Mean(V) = 1/N * \Sigma_i v_i$





Performance Improvement: C4.5



The Issue of Redundant Variables

- Used in combination with relevance measures
  - Select the pair of the most correlated variables (correlation matrix)
  - From the two features in the pair, remove the less relevant feature
  - Repeat until the highest observed correlation is above a threshold
- Resolves the issue of redundant variables
  - **Redundant variable can be relevant** for prediction, but only when considered independently from the other input variables
  - At the same time, **redundant variable is highly correlated with** (or dependent on) **other variables**
- **Redundant variables** also **harmful to predictive performance**

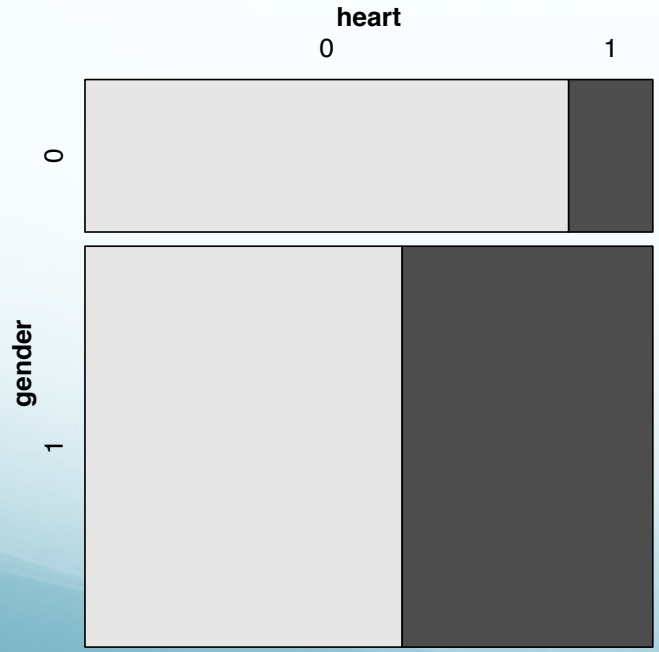
Supervised Univariate Methods

- **Focus on the relationship between *V* and *Y***
  - **Ignore other input variables from *X***: measure the degree of association (correlation) between *V* and *Y*
  - Supervised, model-independent, make use of D_V and D_Y
- The method choice depends on **the types of *V* and *Y***
- Classification task (nominal target *Y*)
  - Nominal *V*: χ^2 statistic or mutual information
  - Numeric *V*: independent two-sample (Welch) t-test or ANOVA
- Regression task (numeric target *Y*)
  - Nominal *V*: independent two-sample (Welch) t-test or ANOVA
  - Numeric *V*: correlation coefficient

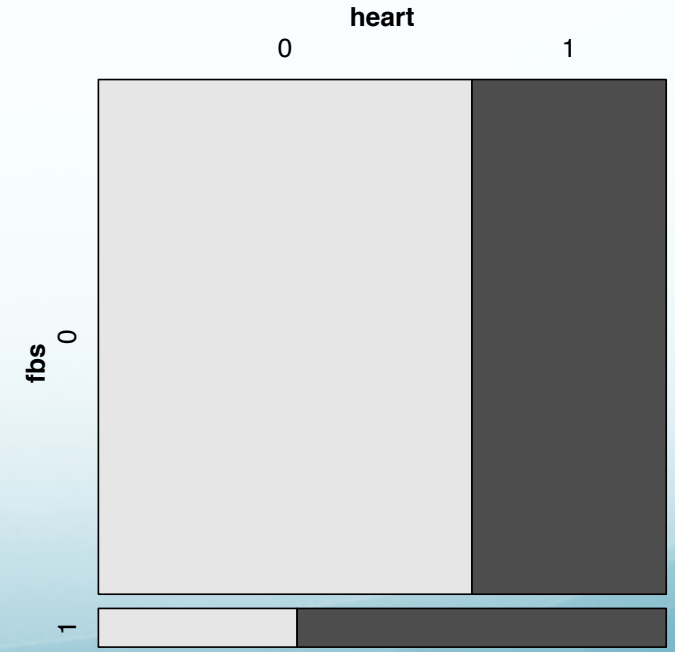
Univariate Methods: Nominal/Nominal

- Alternative measures of **association between nominal variables**
  - Contingency tables and χ^2 statistic
  - Entropy (H) and mutual information $MI(V, Y) = H(V) + H(Y) - H(V, Y)$

X–squared: 20.62333
MI: 0.05912517

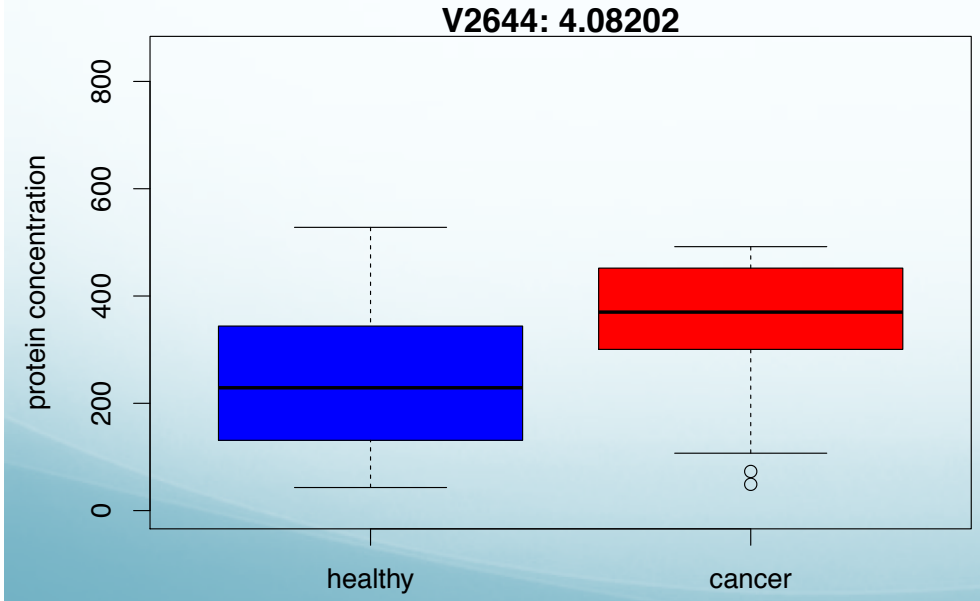


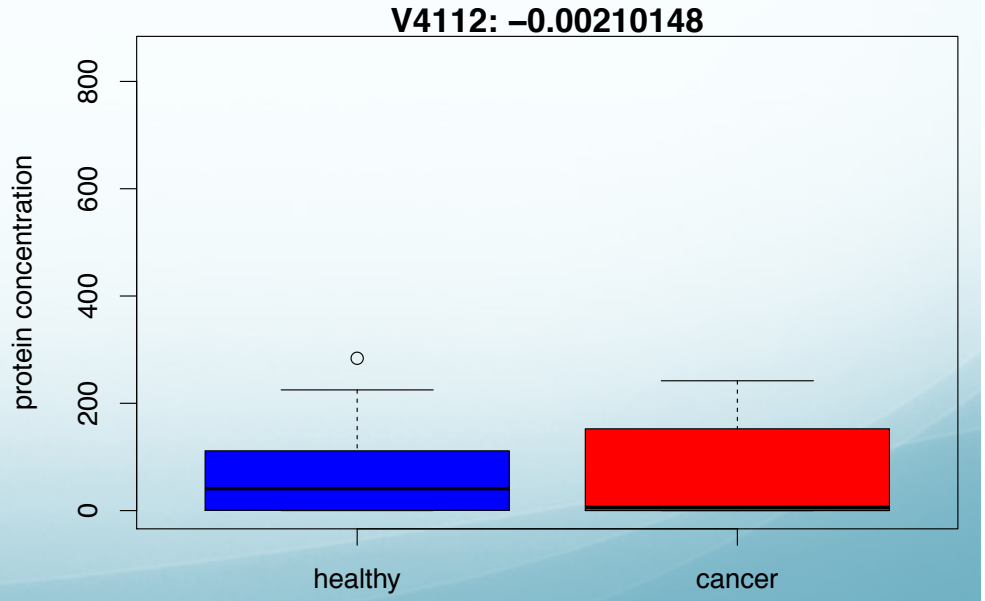
X–squared: 6.347929
MI: 0.01580802



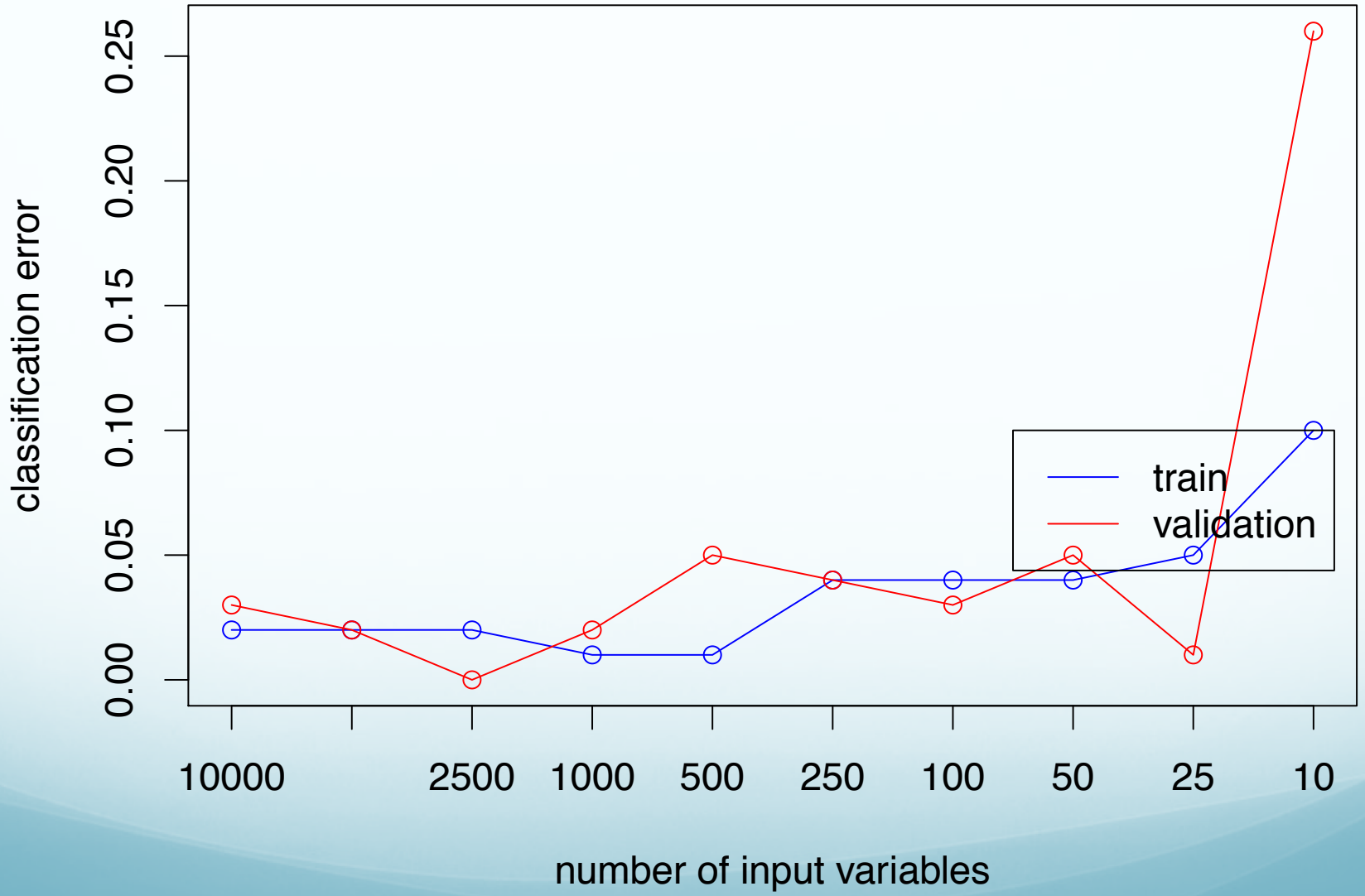
Univariate Methods: Numeric/Nominal and Nominal/Numeric

- Welch t-test of whether two populations have different means
  - **Different means** indicate **high predictive**/discriminative **power**
  - Populations based on the value of the nominal variable
  - If the domain of the nominal variable contains more than two values, alternative tests are to be used (e.g., one-way analysis of variance)





Performance Improvement: C4.5

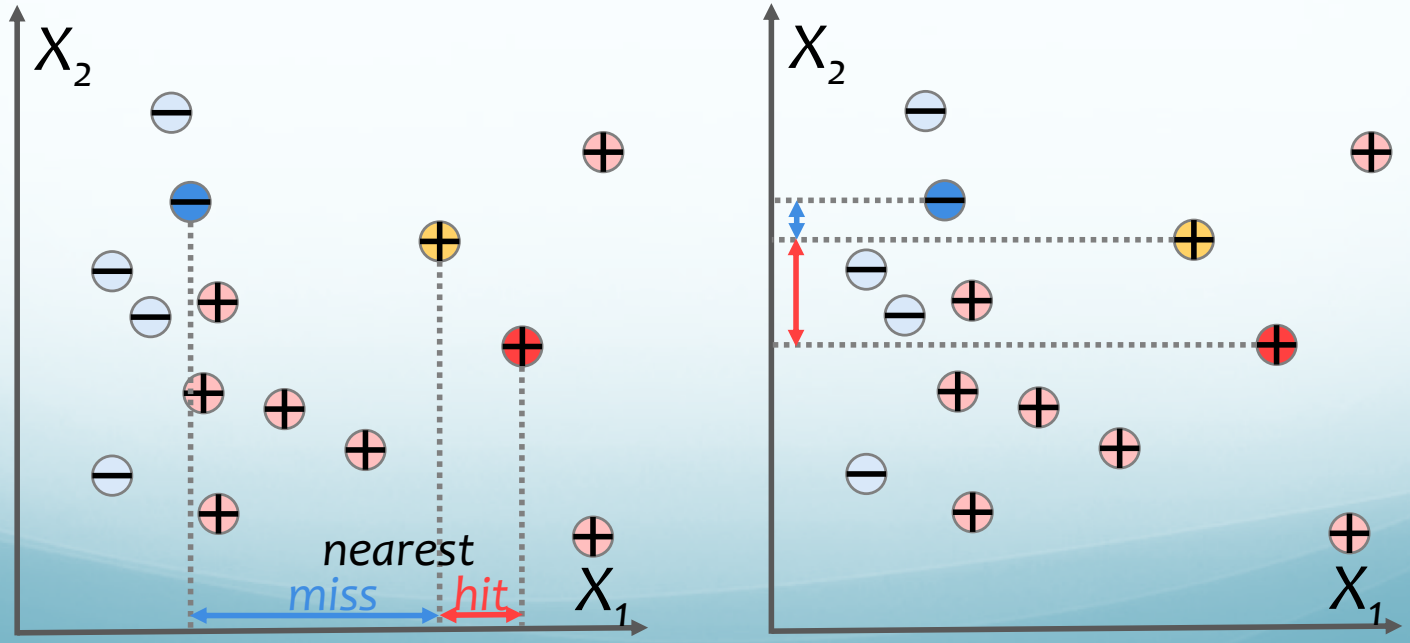


Univariate Methods: Numeric/Numeric

- **Measure the linear correlation** between *V* and *Y*
  - Pearson $R_D(V)=r_P(V,Y)=E[(V-E[V])(Y-E[Y])] / (E[(V-E[V])^2] E[(Y-E[Y])^2])$
- Or **alternatives more robust to non-normal distributions**
  - **Spearman** $R_D(V) = r_S(V,Y) = r_P(ranks(V), ranks(Y))$
  - **Kendall** $R_D(V) = r_K(V,Y) = 2(n_C-n_D) / (n (n-1))$, where
    - n_C: number of concordant pairs of examples
    - n_D: number of dis-concordant pairs of examples
- We **often use the squared values**
  - Since we are not interested in the direction of the relationship (positive, negative), but only to its degree

Model-Dependent Methods: Relief

- The basic idea comes from the nearest-neighbor methods
  - Observe the nearest neighbors of an example
  - If the **nearest neighbor from the same class has the same (or very similar) value of *V***, it indicates a high relevance of V; and vice versa
  - If the **nearest neighbor from the other class has a (very) different value of *V***, it also indicates a high relevance of V; and vice versa



Relief: the Algorithm

- Initialize all the variable relevancies to $r_D(V)=0$
- Repeat m times, where the value of m is user provided
  - Select a random example R from the training set D
  - Find the **nearest hit in *D***: example H closest to R, such that $y_H = y_R$
  - Find the **nearest miss in *D***: example M closest to R, such that $y_M \neq y_R$
  - Update $r_D(V) = r_D(V) - diff(x_{RV}, x_{HV})/m + diff(x_{RV}, x_{MV})/m$
- The value of $diff(x_{IV},x_{JV})$ is calculated as follows
  - For numeric V it equals $|x_{IV} - x_{JV}| / (max(V) - min(V))$
  - For nominal V it equals 1, if $x_{IV} = x_{JV}$, and 0 otherwise

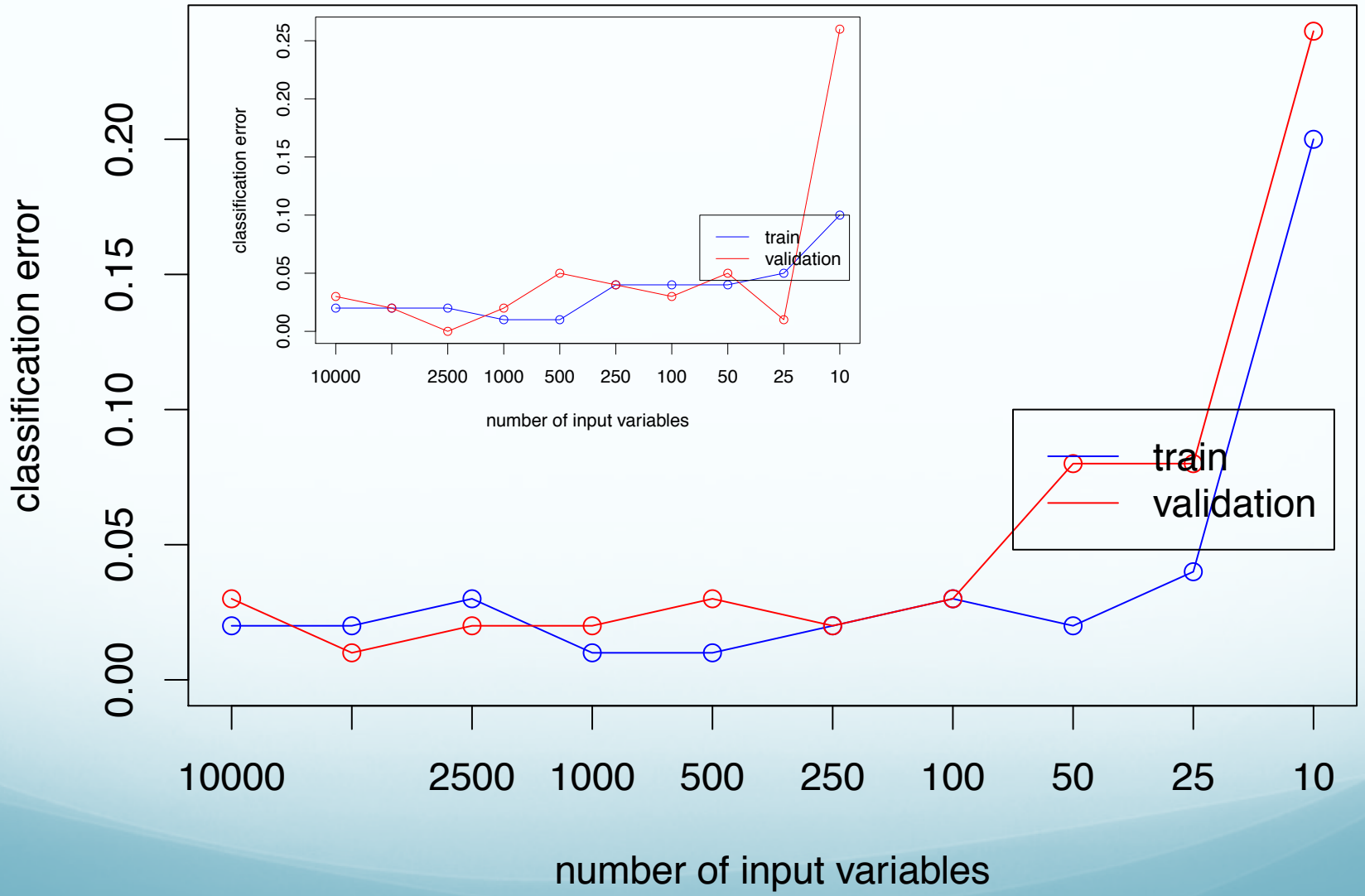
More Than Two Classes: ReliefF

- We **find multiple misses of *R***: one for each target value $y_M \neq y_R$
  - So the update for the misses, i.e., $diff(x_{RV}, x_{MV})$ in the algorithm, becomes the weighted sum

$$\Sigma_{yM \neq yR}\, p(y_M) / (1 - p(y_R)) * diff(x_{RV}, x_{MV})$$

  - $p(Y_l)$ is the probability/frequency of the value y_l in the training data D
- Further generalization of Relief towards **improved robustness**
  - Instead of finding a single nearest miss and hit
  - We **find the *k* nearest misses and hits**, and calculate the average *diff*
  - Where the value of *k* is user defined
- It has been shown that $r_D(V)$ is the difference of two cond probs
  - An example has a different value of *V* for the nearest misses
  - An example has a different value of *V* for the nearest hits

Performance Improvement: C4.5



Towards Regression: RReliefF

- Let us introduce several conditional probabilities
  - p_{diffV} = P(different value of V | nearest neighbors)
  - p_{diffY} = P(nearest misses | nearest neighbors)
  - $p_{diffY|diffV}$ = P(nearest misses | diff value of V in the nearest neighbors)
- Then $r_D(V) = p_{diffV|misses} - p_{diffV|hits}$
  - Where by applying the Bayes rule we obtain the estimates
  - $p_{diffV|misses} = p_{diffY|diffA} * p_{diffV} / p_{diffY}$
  - $p_{diffV|hits} = (1 - p_{diffY|diffV}) * p_{diffV} / p_{diffY}$
- For regression, we estimate p_{diffY} from the probability **distribution of y_N–y_R for the examples N in the neighborhood of R**

Model-Dependent: Transparent Models

- Find the relevance of *V* using a transparent predictive model *M*
  - Build a predictive model *M* on the training data *D*
  - **Read out the relevance of the input variables from *M***
- How we read out the relevancies from a given model?
- For **linear models**
  - The **weights of the variables** in the linear regression
  - Alternatively, take into account the confidence intervals and p-values
- For (ensembles) of **decision trees**
  - The **average impurity reduction obtained in the decision nodes** that **use the particular variable of interest *V* as a splitting variable**
  - The number of internal nodes that use *V* as a splitting variable

Model-Dependent: Error Difference

- Again, the relevance of *V* is measured using a **strong, but not necessarily transparent predictive model**
- Compare the predictive errors of two models built on
  - The training data $D=(D_X, D_Y)$; model *M*
  - The modified data $D'=(D'_X, D_Y)$, where variable V is randomized (*M'*)
  - The values in the column *V* of D_X is permuted to obtain $D_{X'}$
- The **larger the difference**, the **more important is *V* for prediction**
- $r_D(V) = E_{D'}(M') - E_D(M)$, where
  - $E_D(M)$ is the estimate of the predictive error of the model *M* on *D*
  - In bagging and random forests, we can use the out-of-bag estimates
  - The error is estimated using cross-validation or bootstrap

Model-Dependent: Wrapper Approach

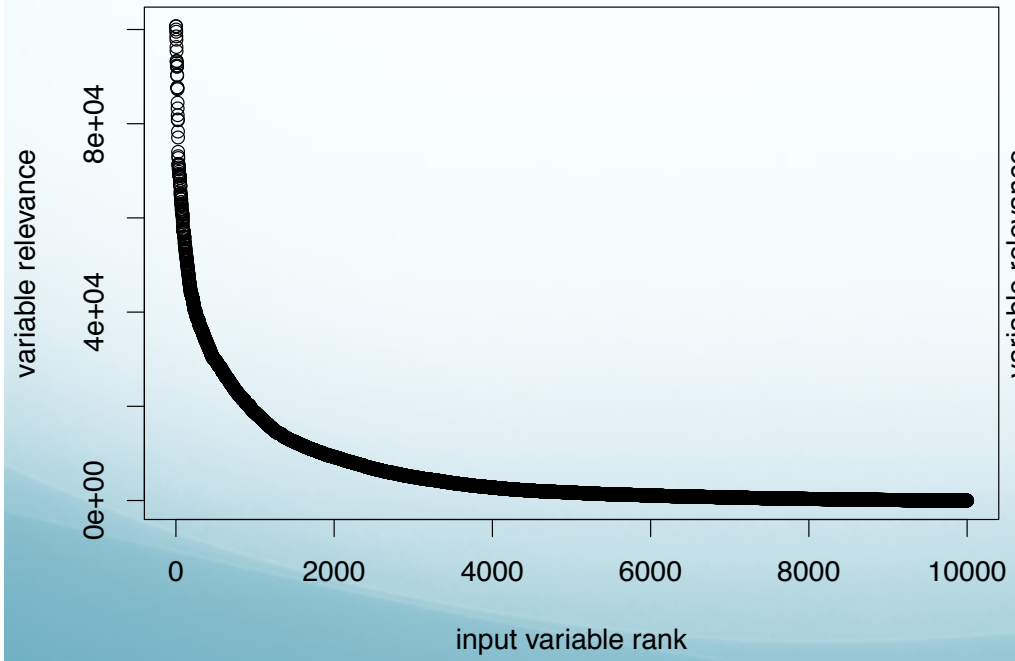
- When the prediction method is chosen in advance
  - We are interested in the relevance wrt the chosen method
  - So we use the chosen method to estimate the relevance
- Similar to Error Difference (previous slide)
  - Compare the predict. errors of **two models built with and without *V***
  - Again: the larger the difference, the larger the relevance of V
  - $r_D(V) = E_{D'}(M') - E_D(M)$; errors cross-validated or bootstrapped

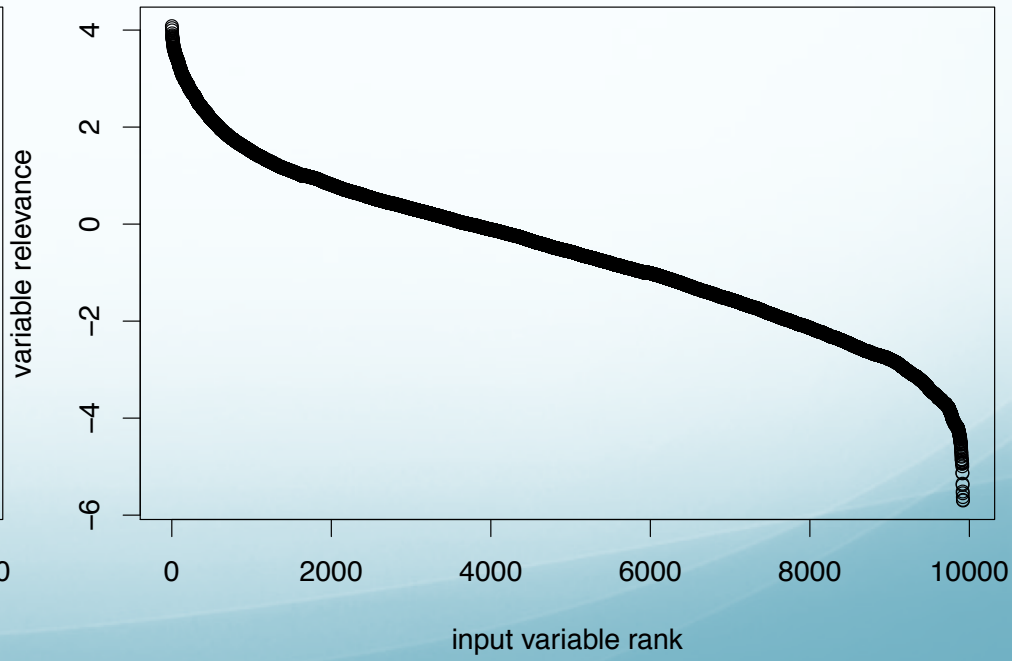
From variable ranking to variable selection

Filter Approach: A Quick (and Dirty) Solution

- **Threshold-based** filter: decide upon the relevance threshold *T*
  - Select features with **relevance higher than *T***
  - Often used threshold value for variance: *T=0*
  - Recommended upper bound for *T* for Relief: $1/sqrt(\alpha \cdot m)$, where α the probability of accepting an irrelevant feature as relevant
- **Number-based filter**: decide upon the number of features *q*
  - **The *q* top-ranked features** wrt the relevance measure selected
- Filter methods often use a plot of the ranking relevance
  - The x-axis: features ranked wrt decreasing values of $r_D(V)$
  - The y-axis: feature relevance
- **Computationally efficient**: no computational overload

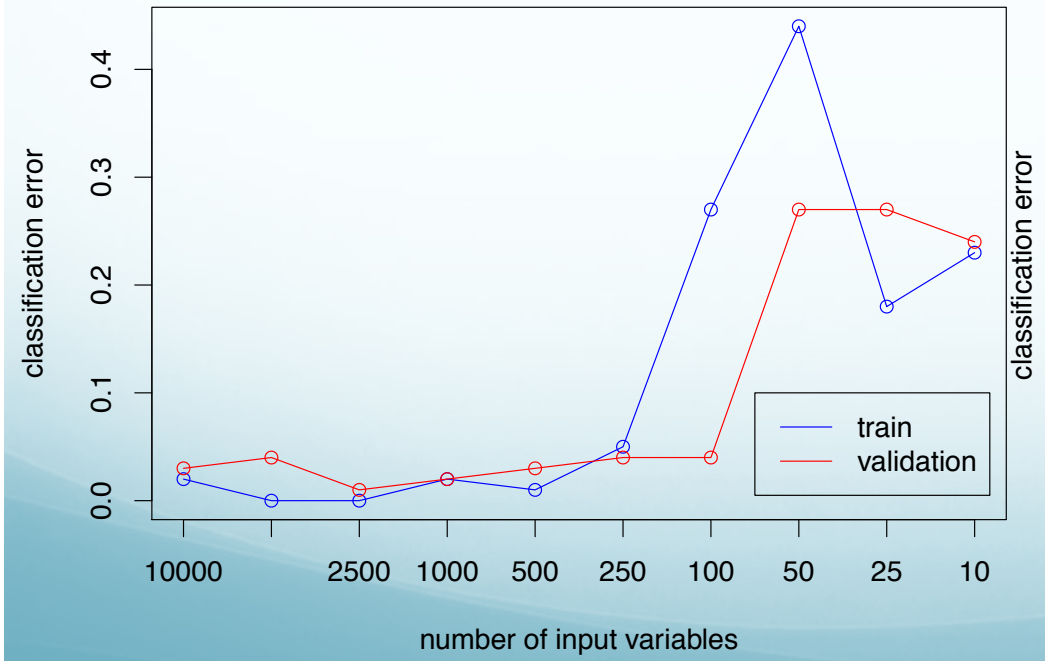
Ranked Relevance Plots: Variance and T-Test

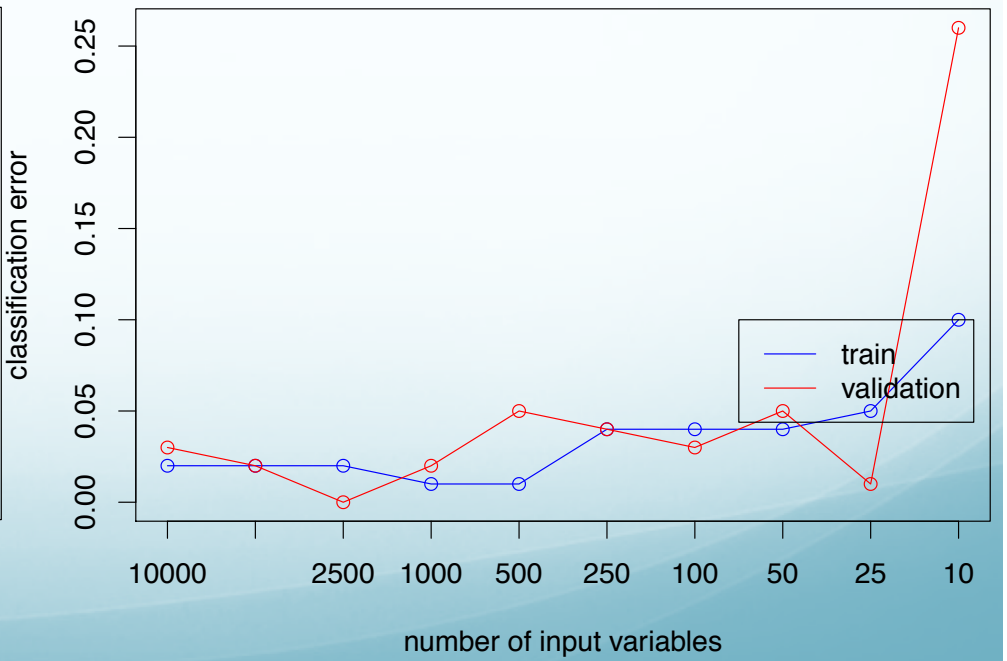




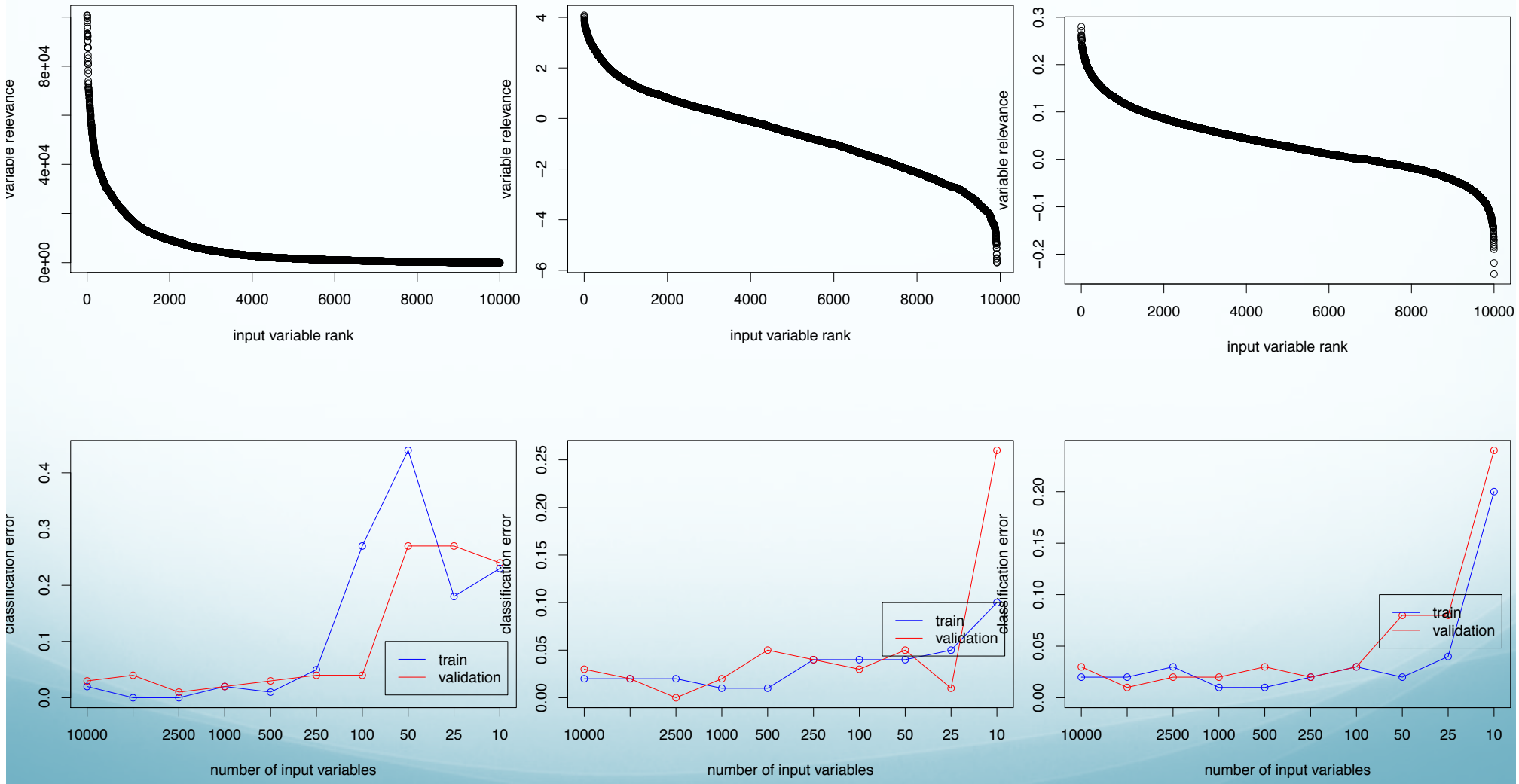
The Error of C4.5: Variance and T-Test

- Alternatively, we can observe the error of a selected predictor
  - And select the the filter threshold based on the plots below
  - Variance (left-hand side): 250 features selected
  - T-test (right-hand side): 50 or 25 features selected
  - **Computationally expensive:** numerous runs of the predictor method

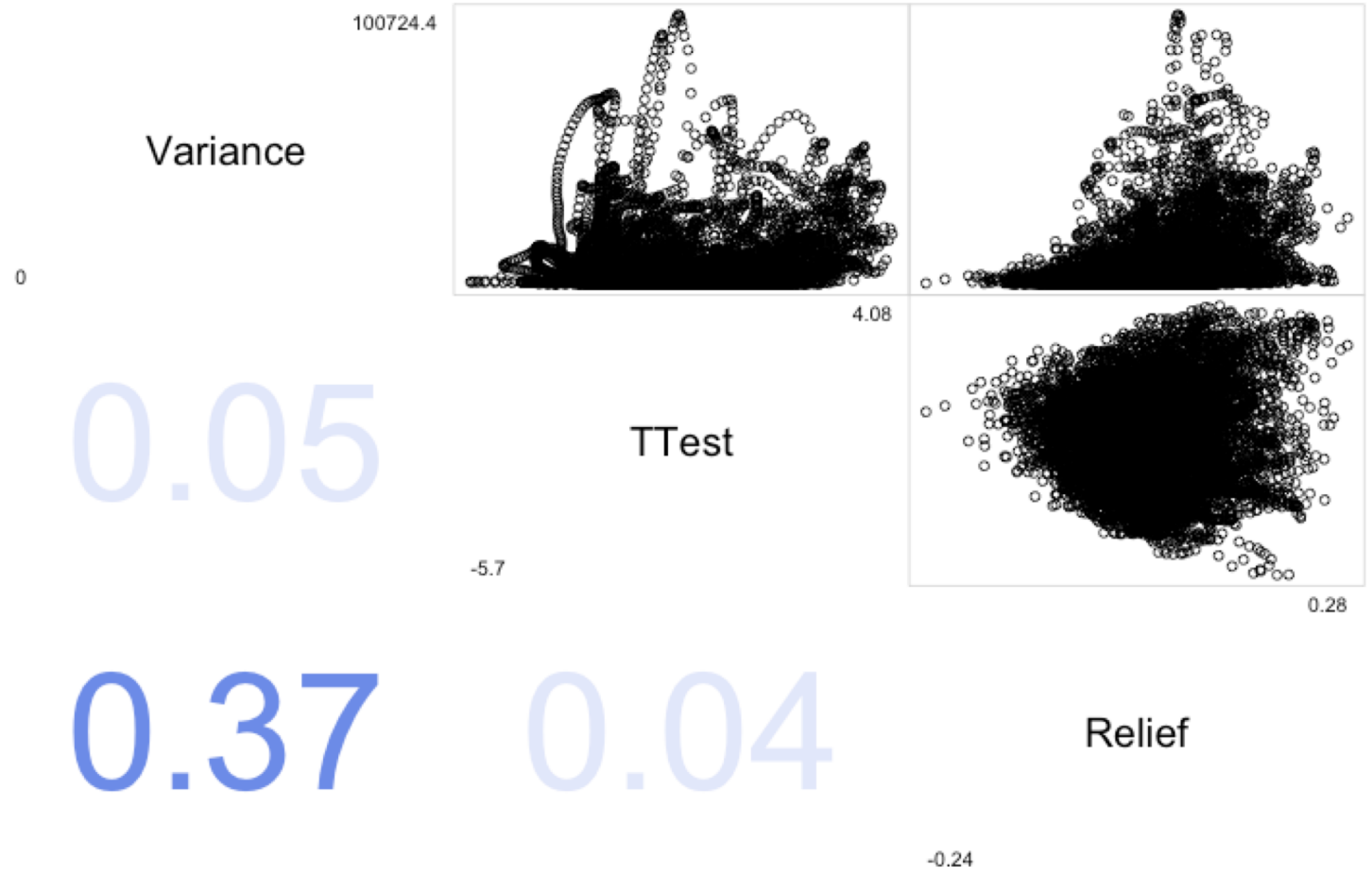




Relevance Measures Comparison



Relevance Measures Pairwise Correlation



Take-Home Messages

1. **Variable relvance and variable selection closely related tasks**

2. **Many methods for measuring variable relevance**
   - From unsupervised and univariate model-independent methods
   - To supervised methods using predictive models
   - **No relevance measure provides an ultimate relevance scoring:** the selection of the optimal relevance measure is data dependent

3. **Variable selection is often approached as a search task**
   - With an exponentially complex search space
   - Incomplete search methods applied with relevance as heuristics
   - Most common approaches: forward selection and **backward (recursive feature) elimination**

Literature Overview: Textbooks

- Witten IH, Frank E, Hall MA (2011) *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier.
  - *7.1: Attribute Selection*
- Kuhn M, Johnson K (2013) *Applied Predictive Modeling*. Springer.
  - *18: Measuring Predictor Importance*
- Flach P (2008) *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press.
  - *10.3 Feature Construction and Selection*

Literature Overview: Articles

- Vipin K, Sonajharia M (2014) Feature Selection: A Literature Review. Smart Computing Review 4(3). DOI: 10.6029/smartcr.2014.03.007211
- Guyon I, Elisseeff A (2003) An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3: 1157–1182.
- Reshef, ND et al (2011) Detecting Novel Associations in Large Data Sets. Science 334: 1518–1524.
- Robnik-Šikonja M, Kononenko I (2003) Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning Journal* 53: 23–69.
  - Kira K, Rendell LA (1992) A practical approach to feature selection. In *Proceedings of the Nineth International Conference on Machine learning (pp. 249–256)*. Morgan Kaufmann.

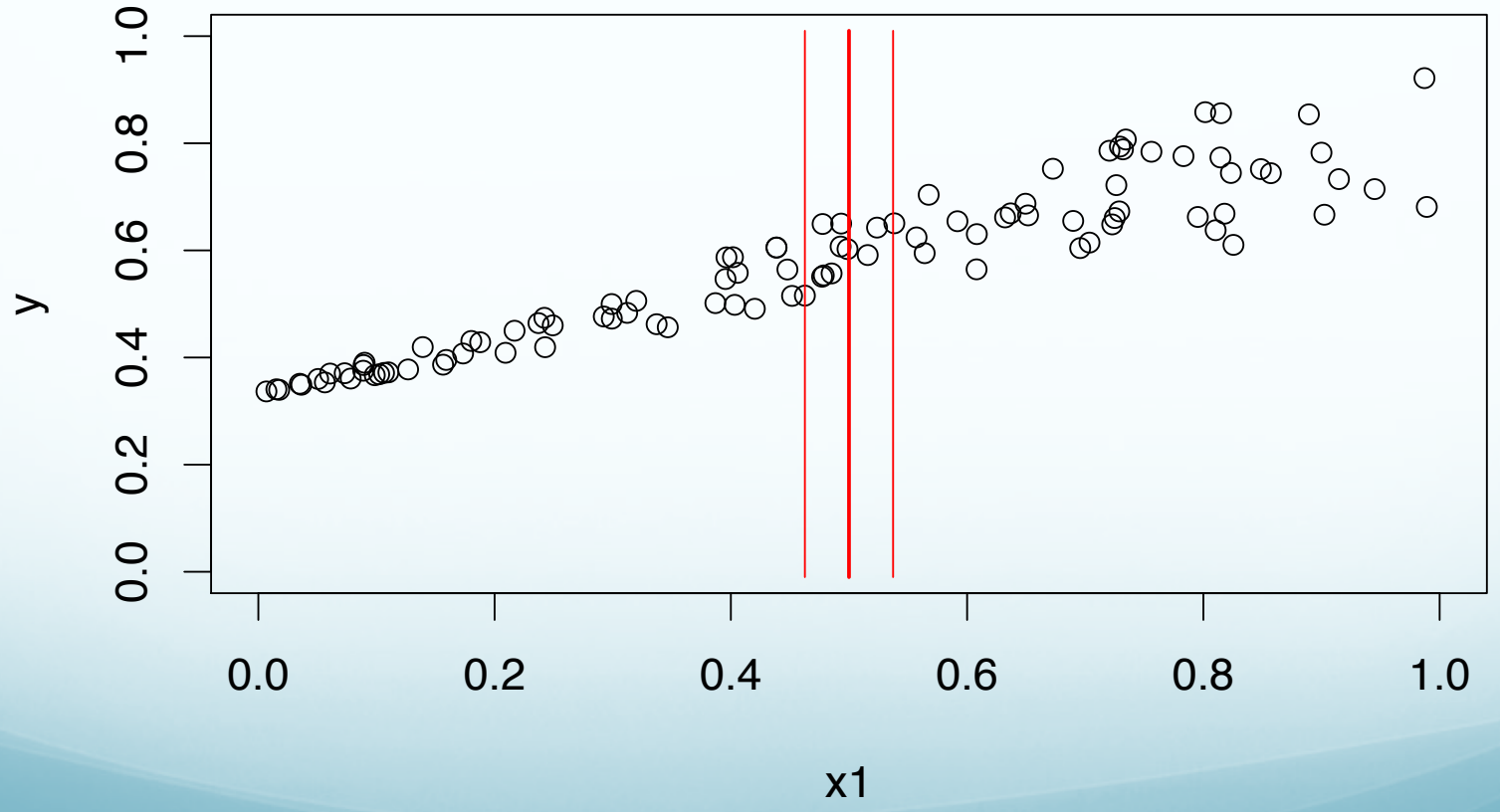
Dimensionality Reduction

Why Dimensionality Reduction?

- Dimensionality reduction is an approach to resolve the issue known as the **curse of dimensionality**
- Let us consider the "ideal", **Bayesian otimal predictive model**
  - $f^*(X) = E[Y|X=x]$
  - It is ideal, since it **minimizes the mean-square loss** $E[(Y-f(X))^2|X=x]$
- In practice given a data set $D=(D_X, D_Y)$
  - We esitmate $f^*(x_o) = E[Y|X=x_o] = mean(y_i|X=x_o)$
  - Practical problem: small number of examples in D, such that $X=x_o$
  - Practical estimate: $f^*(x_o) = mean(y_i|X: |X-x_o|<\varepsilon)$
- **How large ε we need** to obtain a **reliable estimate**?

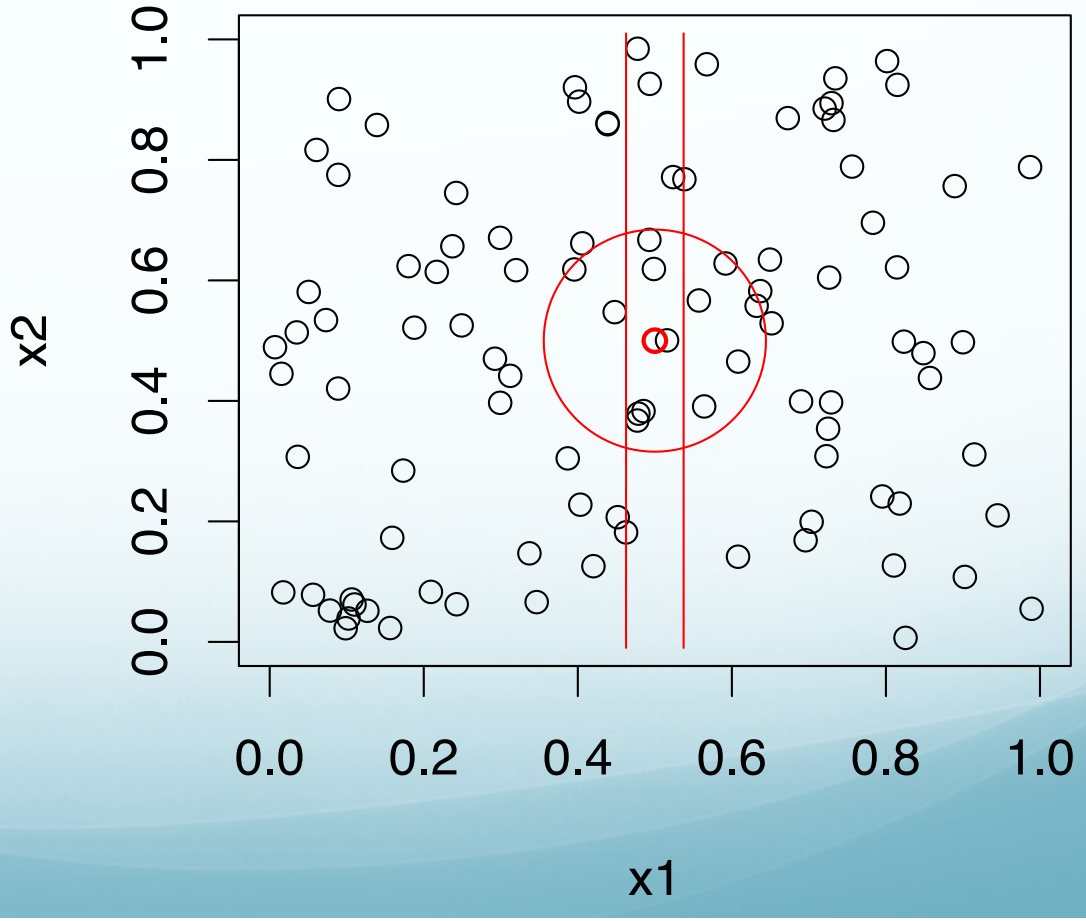
Bayesian Predictive Model: One Dimension

- Given 100 examples randomly drawn from [0,1]
  - **We need** the value of ε=0,04 (**8% of the whole range**)



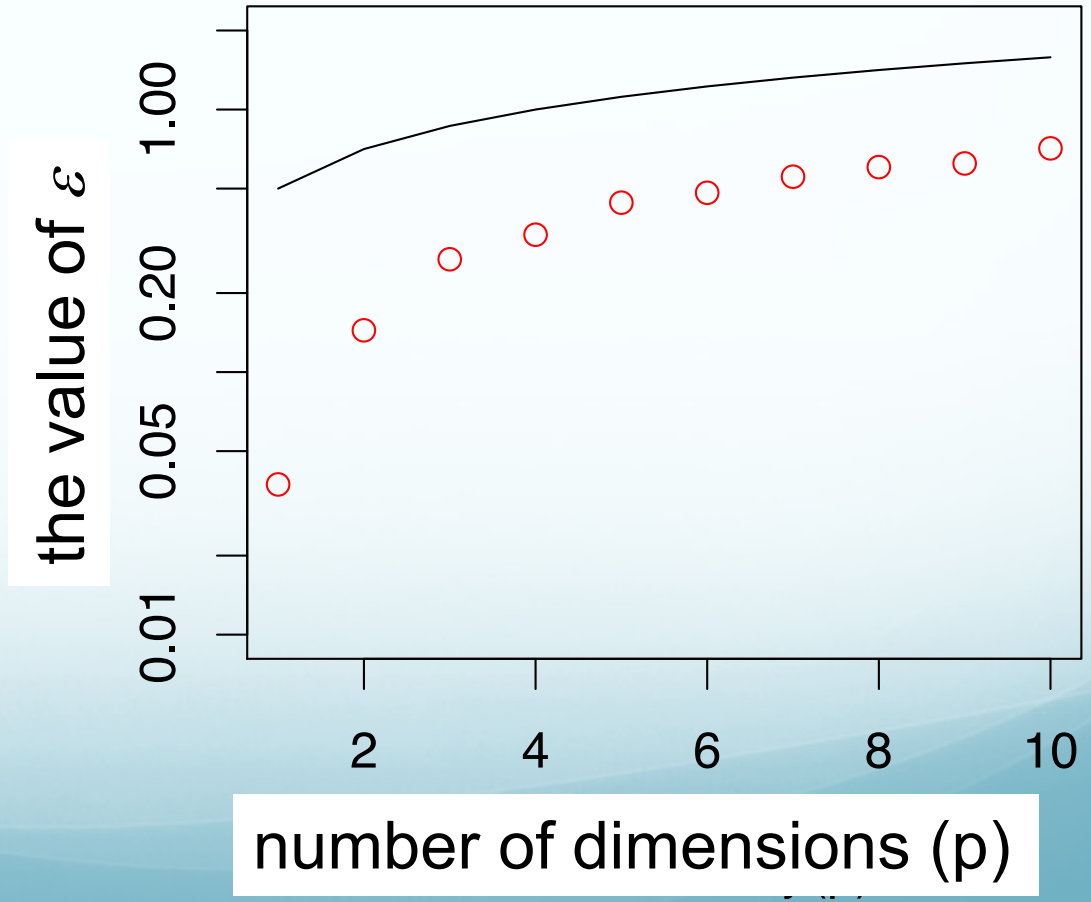
Bayesian Model: Two Dimensions

- We need the value of $\varepsilon=0.15$
  - Which is almost ***4*** **times the value needed in one dimension**
  - Is it still **considered a neighborhood of x_0?**



The Curse of Dimensionality

- For seven-dimensional problem we need ε=0.5
  - Obvious problem: with **increasing number of dimensions it is impossible to obtain a representative neighborhood**



The Task of Dimensionality Reduction

X_1	X_2	…	X_p
x_{11}	x_{12}	…	x_{1p}
x_{21}	x_{22}	…	x_{2p}
…	…		…
x_{N1}	x_{N2}	…	x_{Np}

- Given
  - A set of p input variables $X=\{X_1, \ldots X_p\}$
  - A data set D_X (Nxp matrix): N instances of observations of X
  - Note the lack of Y and D_Y (target): **unsupervised setting**
- Find **low-dimensional representation D_Z of the original space D_X**
  - D_z, where $Z=\{Z_1, \ldots Z_q\}$ and $q<p$ (often $q<<p$)
  - **Typically** $Z = XW$, i.e., the **transformation of X to Z is linear**

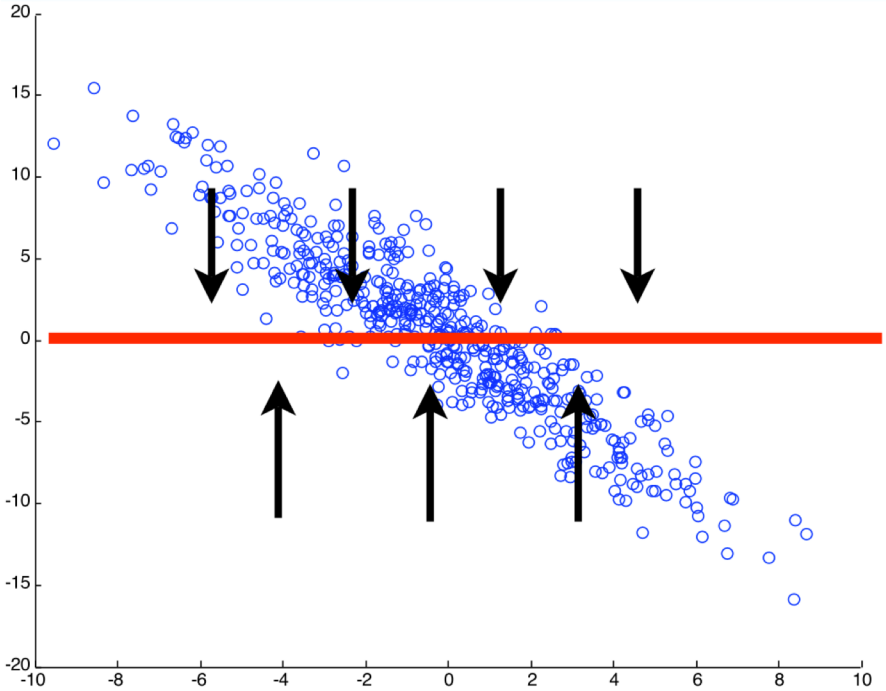
DR Methods

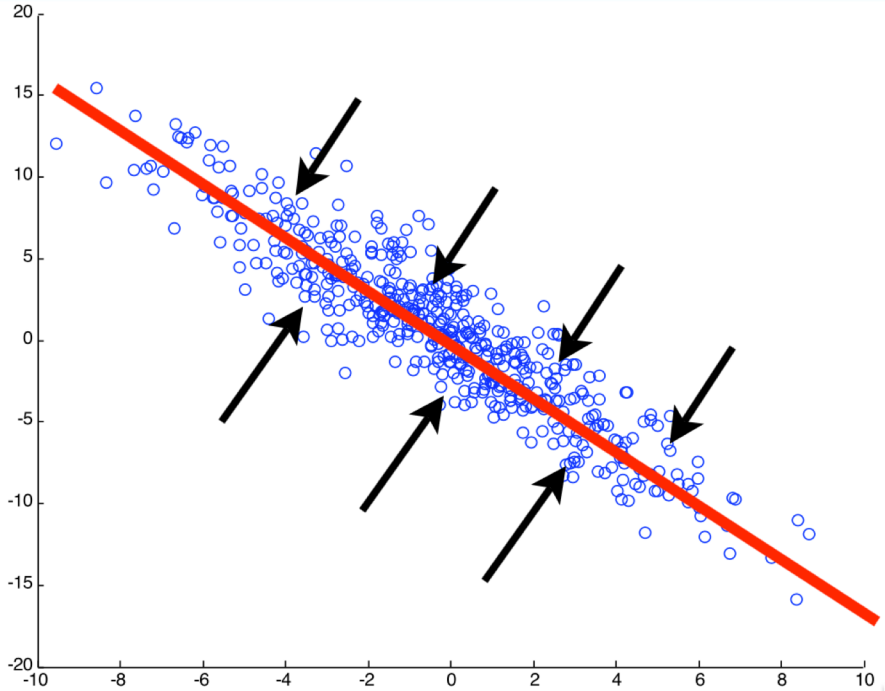
- Two important dimensions for distinguishing DR methods

1. What is the **objective** for the transformation *W* and space *Z*?
   - Interesting: **captures** most of the original data **variance**
   - Approximate: can **accurately reconstruct** the **original data**
   - **Distance-preserving**: distances between transformed instances resembles the distances between the original ones
2. What are the **constraints** for the transformation *W*?
   - Have to be orthogonal, i.e., $W^T W = I$
   - Have to be non-negative, i.e., $W_{ij} >= 0$

DR vs Feature Selection

- **Feature selection** is one approach to dimensionality reduction





- **Dimensionality reduction** methods produce
  - A lower-dim **space** that **is not perpendicular to the original one**
  - In an **unsupervised setting** (no target variable *Y* in the data)

The Benefits of Dimensionality Reduction

Resemble the ones of variable selection and ranking

1. **Improve the predictive performance** of machine learning
   - Focusing on lower-dimensional space improves the prediction
   - Remember the curse of dimensionality slides?
2. **Speed-up the learning** process
   - The computational complexity of machine learning algorithms often dependent on number of input variables
   - But, be careful: dimensionality reduction also takes time and space
3. **Improve the interpretability** of the learned models
   - Dimensionality reduction yields more compact predictive models

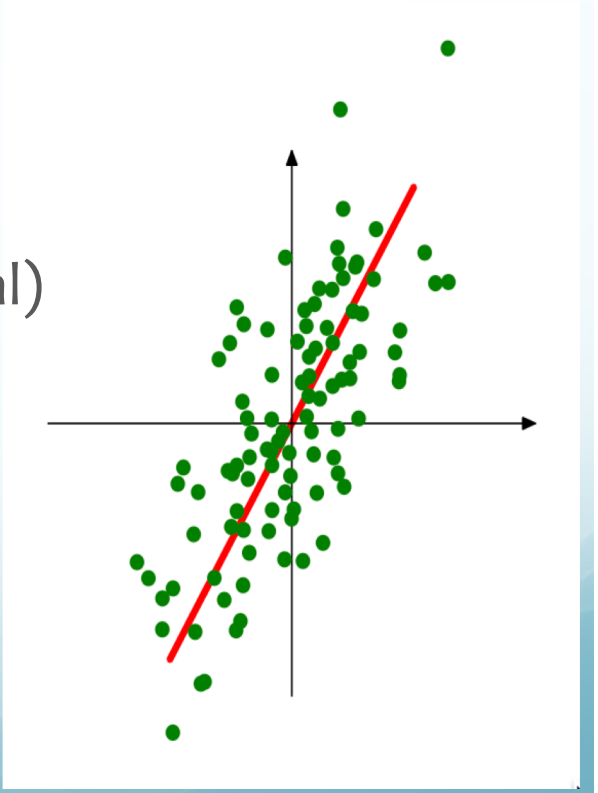
Motivating and Illustrative Example: Arcene

- Arcene data set comes from UCI ML Repository
- Mass spectrometry data for cancer and healthy patients
  - Each input variable corresponds to the concentration of proteins having a given mass
  - Target corresponds to the patient health condition: cancer, healthy
- Data
  - **7000 real input variables, 3000 artificial irrelevant input variables**
  - **100 train examples** to be used for variable selection
  - **100 validation examples** to be used to evaluate variable sets
- **Identify classes of proteins with a certain mass** that can be used as **indicators of cancer**

Principal component analysis (PCA)

Principal Component Analysis (PCA)

- Find a new *p*-dimensional space *Z*
  - **Dimensions** of *Z* should **explain as much data variance as possible:** the dimensions are referred to as principal components
  - Transformation matrix *W* is **orthogonal,** $W^TW=I$
- Let us do the math for the first dimension *w*
  - Projection of *X* to *w* is *Xw*
  - If ***X* is zero-centered,** its variance is (proportional) to $||Xw||^2 = (Xw)^T(Xw) = w^T(X^TX)w$
  - The maximum value is the largest eigenvalue of X^TX, where *w* is the corresponding eigenvector
- PCA is the **eigendecomposition of X^TX**



PCA: Eigenvalues and Eigenvectors

- $X^TX = W\Lambda^2W^T$
  - W is the (pxr) matrix of the eigenvectors (principal components)
  - Λ is the (rxr) diagonal matrix of the eigenvalues (explained variance)
  - r is the rank of the covariance matrix X^TX
- ***W*** is the **transformation matrix**
  - From the original to the transformed space: $Z = XW$
  - And vice-versa: $X = ZW^T$
- Note that W allows for **complete reconstruction of *X*** from Z
  - $(XW)W^T = X(WW^T) = X$, since W is an orthogonal matrix

PCA: DR and Approximating *X*

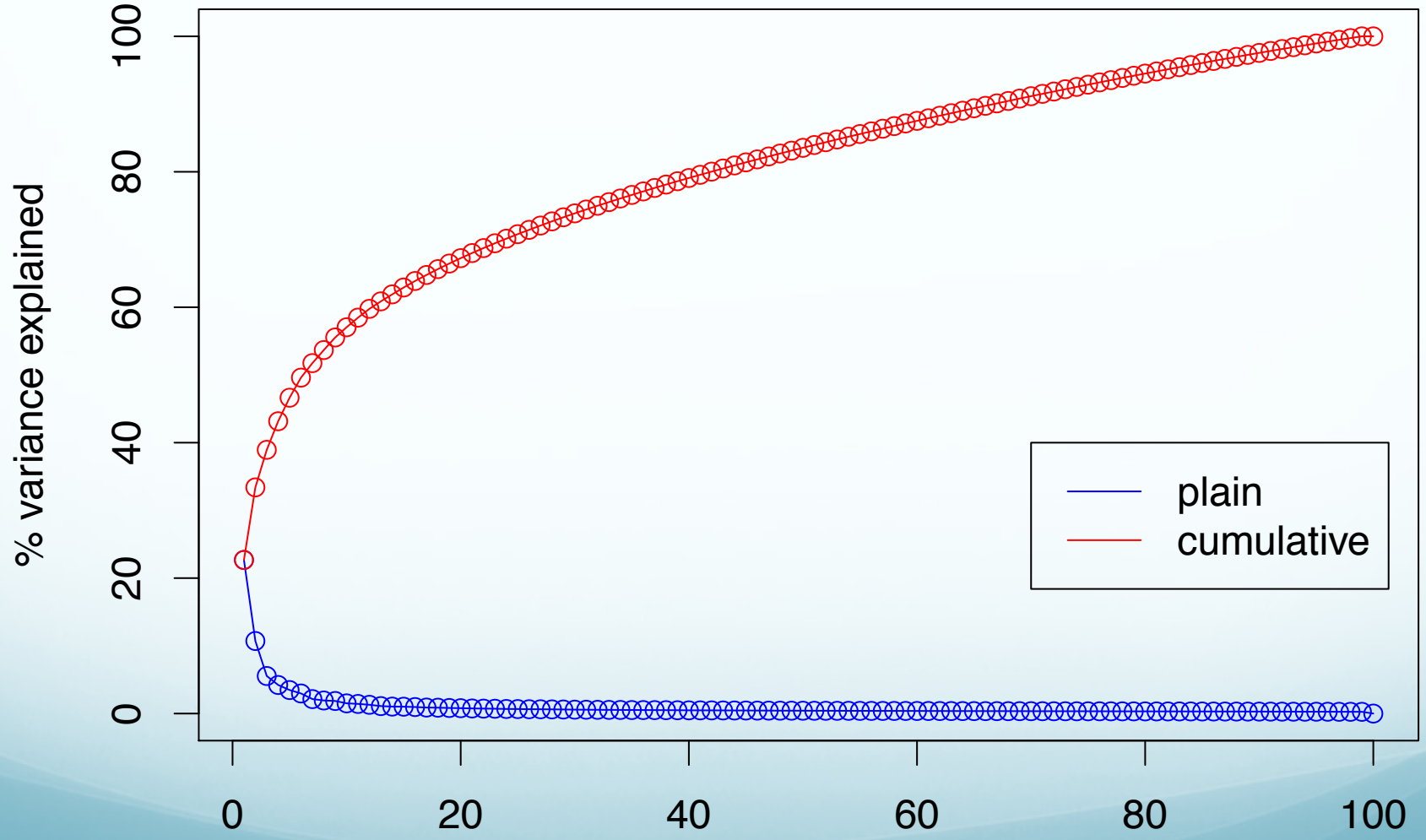
- What about the dimensionality reduction?
  - **Keep only the *q* top (largest) principal components**
  - W_R is a *pxq* matrix with the first *q* columns of *W*
- Transformation to the lower-dim space $Z=XW_R$
- Note that **complete reconstruction** with W_R is **not possible**
  - ZW_R^T is only an approximation of *X*
- Alternative perspective on PCA: it minimizes $||X - XWW^T||$
  - Both **captures variance** and **minimizes reconstruction error**

PCA: How Many Principal Components?

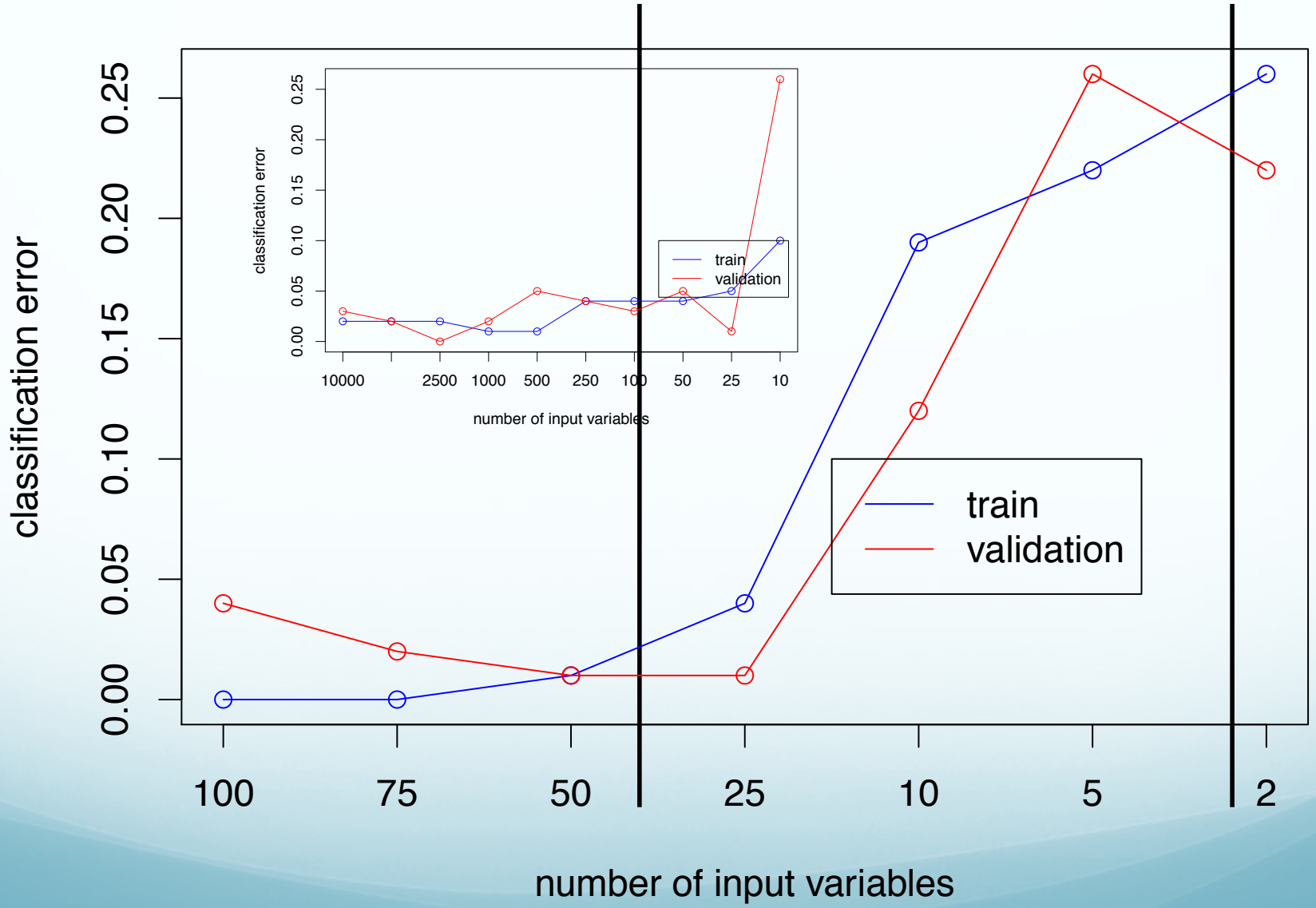
- The usual **heuristics** used
  - Select the first *q* components, that **explain 80% of the total variance**
  - Select the components with the **individual contribution to the variance explanation larger than 5%**
  - Plot a **scree plot** (x-axis: principal component index/rank, y-axis: contribution to the variance explanation) and look for a **knee point**
- When **using PCA in the predictive setting**
  - We can **select *q* using forward-selection or bakward elimination**, i.e.,
  - Perform variable (i.e., components) ranking and selection to obtain the transformed, lower-dimenisonal space

PCA: Scree Plot, Arcene data set

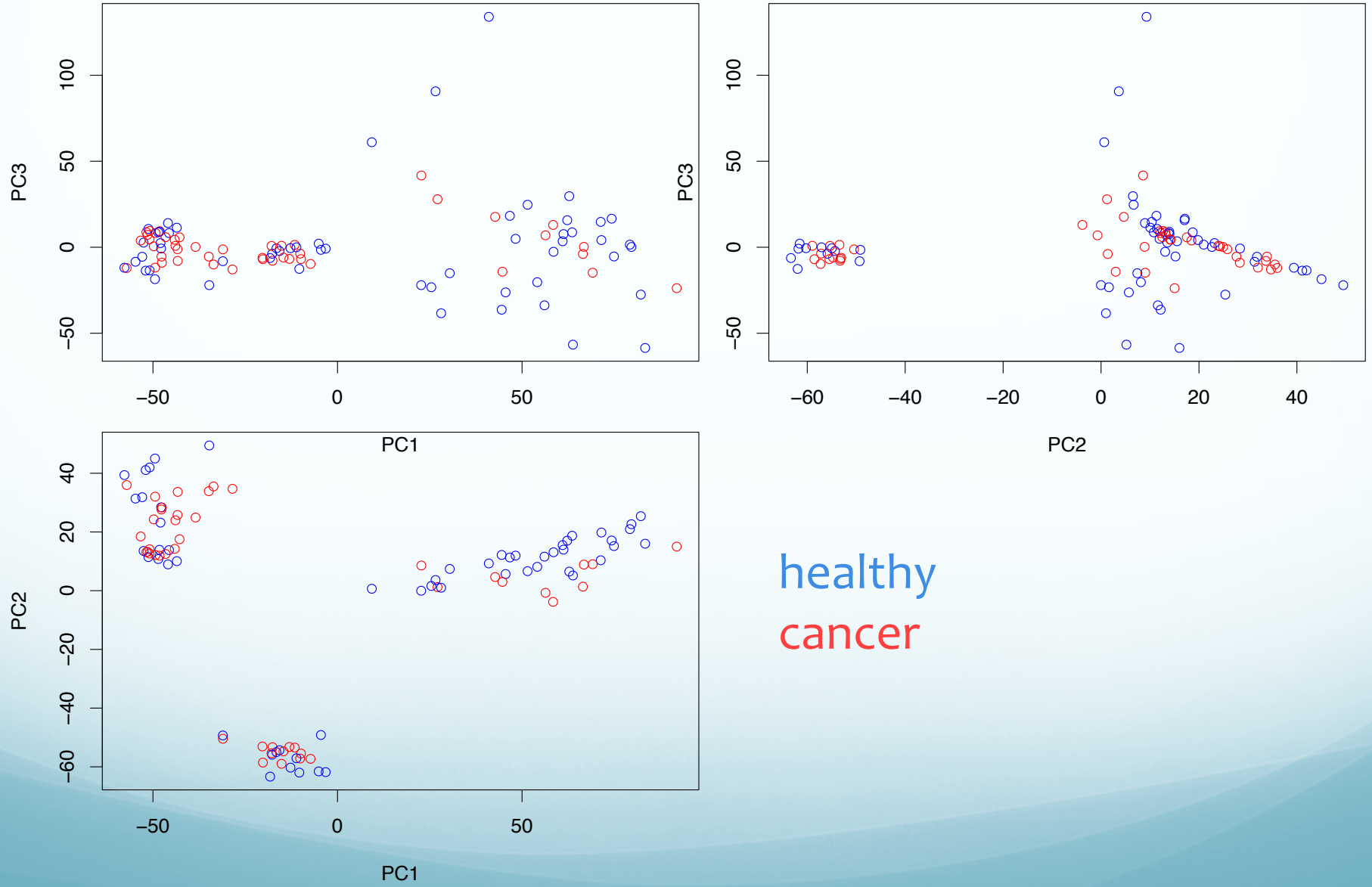
- 80% tot.var.expl.: 42 components, 5% var.expl. and first knee: 3



PCA: Performance Improvement, C4.5



PCA: Visualization of the 3-Top Components



healthy

cancer

PCA: Singular Value Decomposition (SVD)

- $X^TX = W\Lambda^2W^T = W\Lambda(U^TU)\Lambda W^T = (W\Lambda U^T)(U\Lambda W^T) = (U\Lambda W^T)^T(U\Lambda W^T)$
  - In other words $X = U\Lambda W^T$
- $X = U\Lambda W^T$ is referred to as a singular value decomposition of X
- **More efficient way of computing the principal components**
  - The computational complexity of SVD for the q top-ranked principal components is $O(Npq)$
  - As opposed to the computational complexity of eigendecomposition, which is $O(Np^2)$
  - An important difference in cases when $q<<p$

Take-Home Messages

- **PCA is a general framework for dimensionality reduction**
  - Both: **caputures** data **variabnce** and **minimizes reconstruction loss**
  - Can be **efficently computed using SVD**
  - Variable selection task: how many components?

Literature Overview

- James G, Witten D, Hastie T, Tibshirani R (2013) *An Introduction to Statistical Learning with Applications in R*. Springer.
  - *10.2: Principal Components Analysis*
- Flach P (2008) *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press.
  - *10.3 Feature Construction and Selection*
- Singh AP, Gordon GJ (2008) A Unified View of Matrix Factorization Models. In *Proceedings of the The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD (pp. 358–373)*. Springer.

Thanks for Your Attention!

Questions and Discussion