Model Selection

Evgeny Burnaev

Skoltech, Moscow, Russia

Outline

1. Overfitting. Bias-variance decomposition
2. Model Quality Criteria
3. Linear Regression Model Selection
4. Regularization
5. Feature Selection

1. Overfitting. Bias-variance decomposition

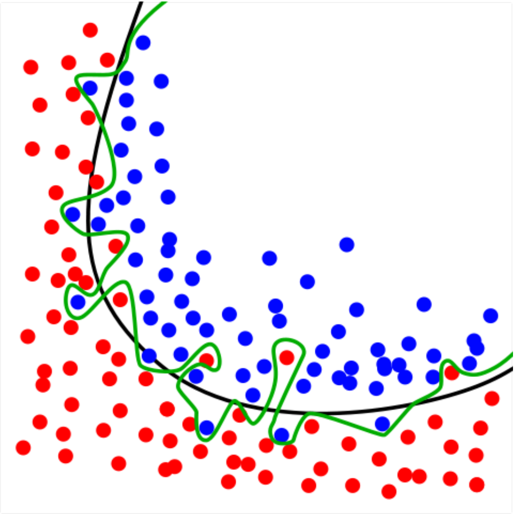
2. Model Quality Criteria

3. Linear Regression Model Selection

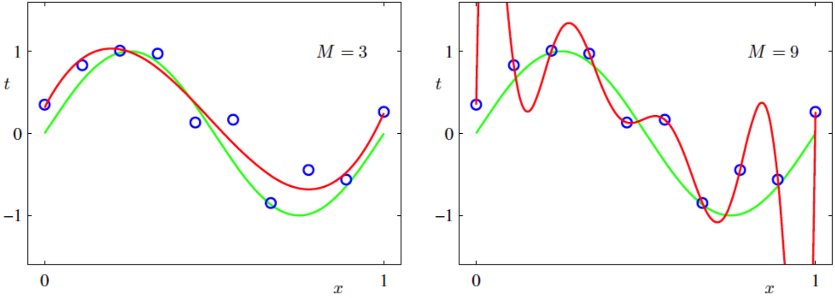
4. Regularization

5. Feature Selection

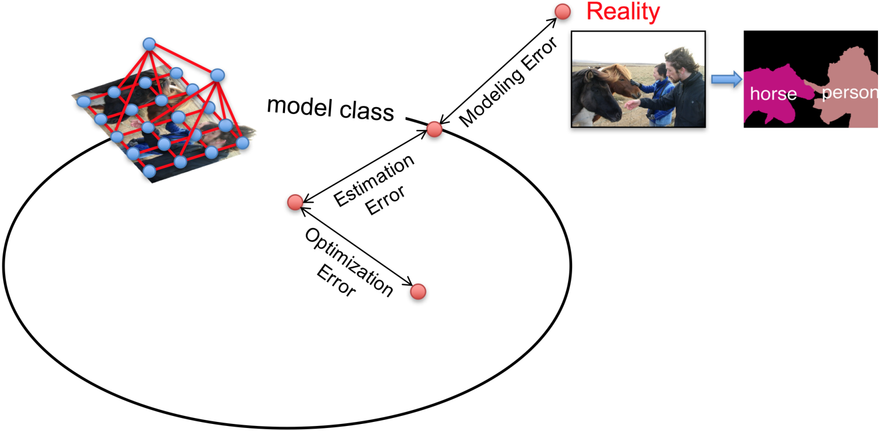
Overfitting: KNN



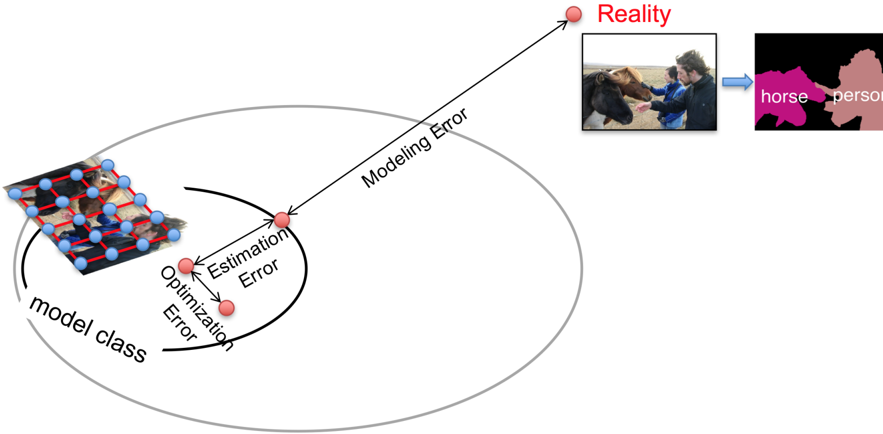
Overfitting: regression



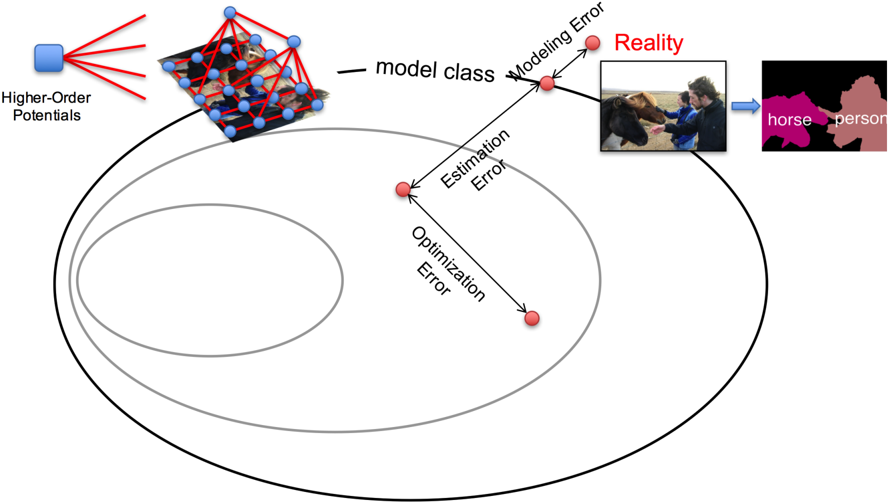
Error Decomposition I



Error Decomposition II



Error Decomposition III



Error Decomposition IV

- Approximation/Modeling Error
  — You approximated reality with a model
- Estimation Error
  — You tried to learn model with finite data
- Optimization Error
  — You were lazy and couldnt/didnt optimize to completion
- Bayes Error
  — Reality just sucks (i.e. there is a lower bound on error for all models, usually non-zero)

The Bias-Variance Decomposition

- Using regression as model, assume that

$$y = f(\mathbf{x}) + \varepsilon,$$

where $\mathbb{E}\varepsilon = 0$, $\mathrm{Var}(\varepsilon) = \sigma_\varepsilon^2$

- Then at an input point $\mathbf{x} = \mathbf{x}_0$

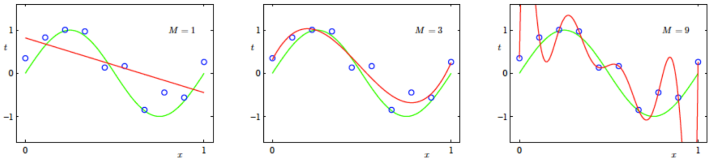
$$\begin{aligned} Err(\mathbf{x}_0) &= \mathbb{E}\left[(y - \hat{f}(\mathbf{x}_0))^2 | \mathbf{x} = \mathbf{x}_0\right] \\ &= \sigma_\varepsilon^2 + \left[\mathbb{E}\hat{f}(\mathbf{x}_0) - f(\mathbf{x}_0)\right]^2 + \mathbb{E}\left[\hat{f}(\mathbf{x}_0) - \mathbb{E}\hat{f}(\mathbf{x}_0)\right]^2 \\ &= \sigma_\varepsilon^2 + \mathrm{Bias}^2\left(\hat{f}(\mathbf{x}_0)\right) + \mathbb{V}\left(\hat{f}(\mathbf{x}_0)\right) \\ &= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance} \end{aligned}$$

Bias-Variance Tradeoff

- **Bias**: difference between what you expect to learn and the truth
  - Measures how well you expect to represent true solution
  - Decreases with more complex model
- **Variance**: difference between what you expect to learn and what you learn from a from a particular dataset
  - Measures how sensitive learner is to specific dataset
  - Increases with more complex model

Linear Regression

- Example: polynomial regression $h(x) = \sum_{m=0}^{M} w_m x^m$



- Value of the optimal (ML) regression coefficients

	$m = 0$	$m = 1$	$m = 3$	$m = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

k-NN regression example

- Assume average of k nearest neighbors

$$
\begin{aligned}
Err(\mathbf{x}_0) &= \mathbb{E}\left[(y - \hat{f}(\mathbf{x}_0))^2 | \mathbf{x} = \mathbf{x}_0\right] \\
&= \sigma_\varepsilon^2 + \left[f(\mathbf{x}_0) - \frac{1}{k}\sum_{\mathbf{x}\in \mathrm{kNN}(\mathbf{x}_0)} f(\mathbf{x})\right]^2 + \frac{\sigma_\varepsilon^2}{k}
\end{aligned}
$$

- For small k, good fit (small bias), larger variance. For big k, more bias, less variance
- This is a model selection problem

What is Model Selection?

- Given a set of models $\mathcal{H} = \{H_1, \ldots, H_K\}$, choose the model **expected to do the best on the test data**
- $\mathcal{H}$ may consist of
  1. Same learning model with different complexities of hyperparameters
     - Nonlinear regression: polynomials with different degress
     - k-Nearest Neighbors: Different choices of k
     - Decision Trees: Different choices of the number of levels/leaves
     - SVM: Different choices of the misclassification penalty hyperparameter C
     - Regularized Models: Different choices of the regularization parameter
     - Kernel based Methods: Different choices of kernels, etc.
  2. Different learning models (e.g., SVM, KNN, DT, etc.)
- Note: Usually considered in supervised learning contexts but unsupervised learning also faces this issue (e.g., “how many clusters” when doing clustering)

Model Selection

- Occam's razor: among competing hypotheses, the one with the fewest assumptions should be selected
- Too much variables/parameters ⇒ significant prediction variance and small bias on the training sample, and vice versa
- We have two interrelated problems
  - to estimate value of a target function, characterizing generalization ability of the considered model
  - select an optimal model w.r.t. to the constructed accuracy criterion

1. Overfitting. Bias-variance decomposition
2. Model Quality Criteria
3. Linear Regression Model Selection
4. Regularization
5. Feature Selection

Model Selection Problem Statement

- Input:
  - — Training sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, $\mathbf{x} \in X$, $y \in Y$
  - — $H_n = \{h : X \to Y\}$ is a hypothesis set, $n = 1, 2, \dots$
  - — $\mathcal{L}_n : (X \times Y)^m \to H_n$ is a learner, $n = 1, 2, \dots$
- Output: obtain a learner with the best generalization ability
- Examples:
  - — Select the best model H_n (model selection)
  - — Select the best learner $\mathcal{L}_n$ for a given model H (e.g. hyperparameters tuning)
  - — Select subset of features $\mathbf{x}_J = (x_j, j \in J)$ from the available features $\mathbf{x} = (x_1, \dots, x_N)$, i.e. the learner $\mathcal{L}$ uses only features $\mathbf{x}_J$

Empirical Error

- $L(h(\mathbf{x}), y)$ is a loss for a pair $(\mathbf{x}, y)$ and a model h
- $\hat{R}(h; S_m) = \frac{1}{m}\sum_{i=1}^{m} L(h(\mathbf{x}_i), y_i)$ is a loss of h on S_m
- Empirical Training Error

  $$\hat{R}_{\mathcal{L}}(S_m) = \hat{R}(h; S_m),\ h(\cdot) = \mathcal{L}(S_m)$$

  This error is a biased estimate of the generalization risk
- Empirical Test Error is estimated using a hold-out test sample S^t

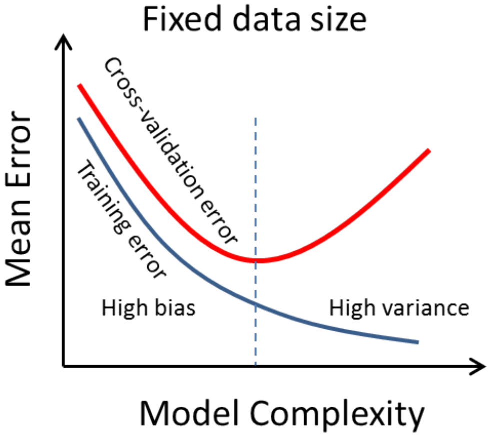
  $$\hat{R}_{\mathcal{L}}(S_m; S^t) = \hat{R}(h; S^t),\ h(\cdot) = \mathcal{L}(S_m)$$

  — either we need an additional test set S^t

  — or we have to divide S_m into a train and a validation sets (results depend on this division)

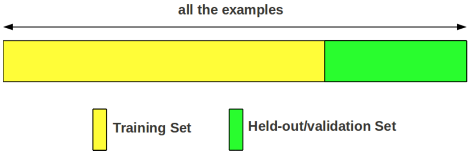
Train vs. Test Error

- Train Error decreases w.r.t. increasing model complexity
- Test Error increases w.r.t. increasing model complexity



Hold-out data

- Set aside a fraction (say $10\% - 20\%$) of the training data
- This part becomes our hold-out data (validation or development data)

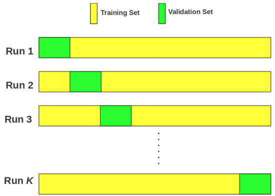


- Remember: hold-out data is NOT the test data
- Train each model using the remaining training data
- Evaluate error on the hold-out data
- Choose the model with the smallest hold-out error
- Problems:
  - — Wastes training data, so typically used when we have plenty of training data
  - Hold-out data may not be good if there was an unfortunate split (use random splitting!)

Cross-Validation

K-fold Cross-Validation

- Create K equal sized partitions of the training data
- Each partition has m/K examples
- Train using $K-1$ partitions, validate on the remaining partitions
- Repeat the same K times, each with a different validation partition



- Finally, choose the model with smallest average validation error
- Usually K is chosen as 10

Cross-Validation

$M \times K$-fold Cross-Validation

- We divide the training sample M times into K equal sized partitions
- Results of all M K-fold cross-validations are aggregated (e.g. averaged)
- By increasing M we can improve accuracy
- Each object is used in a test set M times
- We can construct confidence intervals using results of M repetitions

Bootstrapping

- Given: a set of m examples
- Idea: Sample m elements from this set with replacement
  - An already sampled element could be picked again
- Use this new sample as the training data
- Use the set of examples not selected as the validation data
- For large m, training data consists of about only 63% unique examples
- Training data is inherently small → error estimate may be pessimistic
- Use the following equation to compute the expected model error

$$\hat{R} = 0.632 \times R_{test} + 0.368 \times R_{train}$$

Model Consistency

- If a hypothesis set H is appropriate, then models $h = \mathcal{L}(S)$, constructed for different subsets $S \subset S_m$ should be "similar"
- E.g. we can divide S_m M-times into two parts $\{S_m^{1,i} \cup S_m^{2,i}\}$, $i = 1, 2, \ldots, M$ and calculate

$$\Delta_M(H, \mathcal{L}; S_m) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{m} \sum_{j=1}^{m} |\mathcal{L}(S_m^{1,i})(\mathbf{x}_j) - \mathcal{L}(S_m^{2,i})(\mathbf{x}_j)|$$

- Problems:
  - — Sample size is two times smaller
  - — Computational complexity is two times bigger

1. Overfitting. Bias-variance decomposition
2. Model Quality Criteria
3. Linear Regression Model Selection
4. Regularization
5. Feature Selection

Notations

- Training sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, $\mathbf{x} \in X$, $y \in Y$
- $\mathbf{X} = \{\mathbf{x}_i, i = 1, \ldots, m\}$ is a design matrix
- We consider a linear model $h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b$, $\mathbf{w} \in \mathbb{R}^N$, $\mathbf{x} \in \mathbb{R}^N$ in a stochastic white noise setting
- Let $J \subseteq \{1, \ldots, N\}$ be a subset of features from $\mathbf{x}$ we use to construct a linear model
- We denote by
  - $\mathbf{X}_J$ a submatrix of the full feature matrix $\mathbf{X}$, selected according to the specified subset of feature
  - $\mathbf{w}_J$ linear model coefficients, corresponding to $\mathbf{X}_J$, $\hat{\mathbf{w}}_J$ are their estimates by the least squares method
  - $\hat{h}_J(\mathbf{x}) = \hat{\mathbf{w}}_J^{\mathrm{T}} \cdot \mathbf{x}_J + \hat{b}$ a regression function, $\hat{y}_i(J) = \hat{h}_J(\mathbf{x}_i)$

Regression Risk

- Risk of a prediction (in-sample error)

$$R(J) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}(\hat{y}_i(J) - y_i^*)^2,$$

where y_i^* is a newly randomly generated y_i (with independently generated noise value) for the same $\mathbf{x}_i$

- The problem is to select J, such that $R(J)$ is small
- Risk estimate on the training set is equal to

$$\hat{R}_{\text{tr}}(J) = \frac{1}{m}\sum_{i=1}^{m} \left(\hat{y}_i(J) - y_i\right)^2$$

- **Theorem**: $\mathbb{E}(\hat{R}_{\text{tr}}(J)) < R(J)$ and

$$\text{bias}(\hat{R}_{\text{tr}}(J)) = \mathbb{E}(\hat{R}_{\text{tr}}(J)) - R(J) = -\frac{2}{m}\sum_{i=1}^{m}\text{Cov}(\hat{y}_i, y_i)$$

C_p Mallow

- It can be proved, that in the linear case

$$2\sum_{i=1}^{m}\mathrm{Cov}(\hat{y}_i, y_i) \sim 2|J|\hat{\sigma}^2,$$

  where $\hat{\sigma}^2$ is an estimate of an output noise standard deviation σ^2, obtained using residuals on the training set, calculated by fitting the model

- Thus, we get C_p Mallow statistics, representing asymptotically unbiased estimate of the regression risk

$$\hat{R}(J) = \hat{R}_{\mathrm{tr}}(J) + 2\frac{\hat{\sigma}^2}{m}|J|$$

  The second term here penalizes complexity

AIC & BIC

- AIC (Akaike Information Criterion) provides estimate of the risk in case of more general models. It has the form

  $$L_J - |J|,$$

  where

  — L_J is a model log-likelihood
  — $|J|$ is a number of model parameters
- AIC is equivalent to Mallow C_p in case of linear regression model with a Gaussian noise
- BIC (Bayesian Information Criterion) is equal to

  $$L_J - |J| \log m$$

LOO CV

- Another possibility to estimate risk: leave-one-out cross-validation

$$\hat{R}_{CV}(J) = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_{(-i)}\right)^2,$$

where $\hat{y}_{(-i)}$ is a prediction, obtained by a model, constructed using a sample $S_m \setminus \{(\mathbf{x}_i, y_i)\}$

- Increase computational efficiency using formula

$$\hat{R}_{CV}(J) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{y_i - \hat{y}_i(J)}{1 - U_{ii}(J)}\right)^2$$

$$U(J) = \mathbf{X}_J(\mathbf{X}_J^{\mathrm{T}}\mathbf{X}_J)^{-1}\mathbf{X}_J^{\mathrm{T}}$$

1. Overfitting. Bias-variance decomposition

2. Model Quality Criteria

3. Linear Regression Model Selection

4. Regularization

5. Feature Selection

Learning Guarantees

- Upper bound on a probability of over-training, valid for any sample S_m, rather general hypothesis class H and learner $\mathcal{L}$:

$$\mathbb{P}\left(\hat{R}(h; S^t) - \hat{R}(h; S_m) \geq \varepsilon\right) \leq \delta(\varepsilon, H),\ h(\cdot) = \mathcal{L}(S_m)$$

- Then for any S_m, H and $\mathcal{L}$, and $\delta \in (0, 1)$ with a probability not less than $(1 - \delta)$ we get that

$$\hat{R}(h; S^t) \leq \hat{R}(h; S_m) + \varepsilon(\delta, H)$$

- Corrected Empirical Risk

$$\hat{R}(h; S_m) + \varepsilon(\delta, H) \rightarrow \min_{h,H}$$

Regularization

- Regularization penalizes complex models H

$$\hat{R}_{\text{pen}}(h; S_m) = \hat{R}(h; S_m) + \text{pen}(H)$$

- Let us consider linear models $H = \{h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\text{T}} \cdot \mathbf{x})\}$ (classification) or $H = \{h(\mathbf{x}) = (\mathbf{w}^{\text{T}} \cdot \mathbf{x})\}$ (regression)
- Then
  - A) L_2-regularization $\text{pen}(H) = \lambda \sum_{j=1}^{N} w_j^2$
  - B) L_1-regularization $\text{pen}(H) = \lambda \sum_{j=1}^{N} |w_j|$
  - C) L_0-regularization $\text{pen}(H) = \lambda \sum_{j=1}^{N} 1_{w_j \neq 0}$
- AIC $\&$ BIC are special cases of L_0-regularization

Bayesian interpretation of Regularization

- We consider linear regression model with a Gaussian i.i.d. noise
- Log-likelihood of S_m has the form

$$l(\mathbf{w}) = m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \mathbf{w}^{\mathrm{T}} \mathbf{x}_i)^2$$

- Let us assume that

$$\mathbf{w} \sim \mathcal{N}(0, \tau^2 \mathbb{I})$$

- Posterior distribution of $\mathbf{w}$ has the form

$$\begin{aligned} p(\mathbf{w}|S_m) &\propto p(S_m|\mathbf{w}) p(\mathbf{w}) \\ &= \mathrm{C} \cdot \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \mathbf{w}^{\mathrm{T}} \mathbf{x}_i)^2 \right\} \exp\left\{ -\frac{\mathbf{w}^{\mathrm{T}}\mathbf{w}}{2\tau^2} \right\} \end{aligned}$$

Bayesian interpretation of Regularization

- Log-posterior

$$\begin{aligned} l_{MAP}(\mathbf{w}|S_m) &= -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 - \frac{1}{2\tau^2}\sum_{k=1}^{N} w_k^2 \\ &= -\frac{m}{2\sigma^2}\left(\frac{1}{m}\sum_{i=1}^{m}(y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i)^2 + \frac{\sigma^2}{m\tau^2}\sum_{k=1}^{N} w_k^2\right) \\ &= -\frac{m}{2\sigma^2}\left(\hat{R}(h; S_m) + \lambda\|\mathbf{w}\|^2\right),\ \lambda = \frac{\sigma^2}{m\tau^2} \end{aligned}$$

- Thus MAP estimate is the same as L_2-regularized estimate

1. Overfitting. Bias-variance decomposition

2. Model Quality Criteria

3. Linear Regression Model Selection

4. Regularization

5. Feature Selection

Feature Selections

Why Feature Selection?

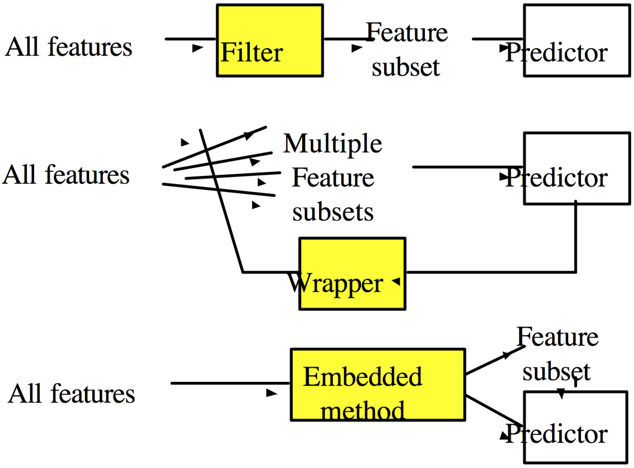
- Some algorithms scale (computationally) poorly with increased dimension
- Irrelevant features can confuse some algorithms
- Redundant features adversely affect regularization
- Removal of features can increase (relative) margin (and generalization)
- Reduces data set and resulting model size
- Note: Feature Selection is different from Feature Extraction
    - — The latter transforms original features to get a small set of new features
    - — Dimensionality Reduction is a type of Feature Extraction

Feature Selection Methods

- Methods agnostic to the learning algorithm
  - Preprocessing based methods
    - A) E.g., remove a binary feature if it's ON in very few or most examples
  - Filter Feature Selection methods
    - A) Use some ranking criteria to rank features
    - B) Select the top ranking features
  - Wrapper Methods (keep the learning algorithm in the loop)
    - A) Requires repeated runs of the learning algorithm with different set of features
    - B) Can be computationally expensive

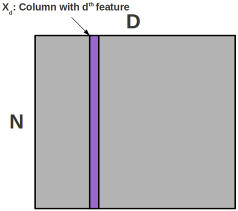
Scheme of Feature Selection Methods

Filters,Wrappers, and Embedded methods



Filter Feature Selection

- Uses heuristics but is much faster than wrapper methods



- Correlation Critera: Rank features in order of their correlation with the labels

$$r(x_d, y) = \frac{\mathrm{cov}(x_d, y)}{\sqrt{\mathrm{var}(x_d)\mathrm{var}(y)}}$$

- Mutual Information Criteria:

$$MI(x_d, y) = \sum_{x_d} \sum_{y} p(x_d, y) \log \frac{p(x_d, y)}{p(x_d)p(y)}$$

Wrapper Methods

Two types: Forward Search and Backward Search

- Forward Search
  - — Start with no features
  - — Greedily include the most relevant feature and estimate model's error on a renewed feature set
  - — Stop when selected the desired number of features
- Backward Search
  - — Start with all the features
  - — Greedily remove the least relevant and estimate model's error on a renewed feature set
  - — Stop when selected the desired number of features
- Inclusion/Removal criteria uses cross-validation