Breiman, L (2001). Statistical modeling: The two cultures

Statistical learning reading group

Alexander Ly

Psychological Methods
University of Amsterdam

Amsterdam, 27 October 2015

Overview

Definition of machine learning

Arthur Samuel (1959)

Field of study that gives computers the ability to learn without being explicitly programmed

Tom M. Mitchell (1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

History

Evolved from
- Artificial intelligence
- Pattern recognition

Success stories (from 1990s onwards):
- Spam filters
- Optical character recognition
- Natural language processing (Search engines)
- Recommender systems
- Netflix challenge (2006) won by AT&T labs research

History

Success stories (from 1990s onwards):

- Spam filters
- Optical character recognition
- Natural language processing (Search engines)
- Recommender systems
- Netflix challenge (2006) won by AT&T labs research

Summary:

- Success correlated with the rise of the internet and reinvented statistics.
- Machine learning terms for statistical concepts
- Ignored by most statisticians, except for the Breiman and Tibshirani, Hastie, Friedman (Efron, Stanford school)

History

Summary:

- Success correlated with the rise of the internet and reinvented statistics.
- Machine learning terms for statistical concepts
- Ignored by most statisticians, except for the Breiman and Tibshirani, Hastie, Friedman (Efron, Stanford school)

Statistics	Machine learning
Estimation	Learning
Data point	Example/Instance
Regression	Supervised learning
Classification	Supervised learning
Covariate	Feature
Response	Label

Breiman 1928 – 2005

Career: 1954 PhD in probability, 1960s Consulting, 1980s onwards professor at UC Berkeley

Breiman's consulting experience

- Prediction problems
- Live with the data before modelling
- Solution should be either an algorithmic or a data model.
- Predictive accuracy is the criterion for the quality of the model
- Computers necessary in practice

Examples of algorithmic techniques/models: Classification and regression trees, bagging, random forest. In the paper Breiman focusses on supervised learning

Problem statement: Supervised learning

Problem setting

Based on n pairs of data $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$, where x_i are features and y_i are correct labels, predict future labels y_{new} given x_{new}.

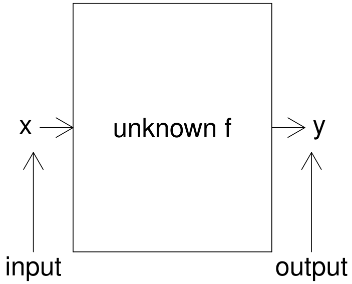
Example (classification):

- Features: $x_1 =$ (nose, mouth). Label: $y_1 =$ yes, face
- Features: $x_2 =$ (doorbell, hinge). Label: $y_2 =$ no, face

Problem statement: Supervised learning

Problem setting

Based on n pairs of data $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$, where x_i are features and y_i are correct labels, predict future labels y_{new} given x_{new}.



Problem statement: Supervised learning

Problem setting

Based on n pairs of data $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$, where x_i are features and y_i are correct labels, predict future labels y_{new} given x_{new}.

$$y = f(x) + \epsilon$$

For prediction: discover the unknown f that relates features (covariates) to labels (dependent variables).

Goal: Machine learning

Predict future instances based on already observed data

- Prediction based on past data.
- Example: "Based on previous data, I predict that it will rain tomorrow"
- Give a prediction accuracy.
- Example: "Based on previous data, I'm 67% sure that it will rain tomorrow"
- Use data models or even algorithmic models to discover the relationship f provided by nature

Machine learning vs "standard" statistical inference culture

Machine learning

- Goal: Prediction of new data (learn f from the data)
- Approach: Data are always right, there are no models, only algorithms
- Passive: Data are already collected
- "Big" data

"Standard" approach in psychology

- Goal: evaluation of theory (evaluate a known f)
- Approach: The model is right, the data could be wrong. Evaluate theory by comparing two models
- Active: Confirmatory analysis based on the model: Design of experiments, power analysis, etc etc
- "Small" data

"Standard" approach in psychology

- Goal: evaluation of theory (evaluate a known *f*)
- ✓ Many time no theory, most research has an exploratory flavour (80% according to Lakens)
- Approach: The model is right, the data could be wrong. Evaluate theory by comparing two models
- ✓ What if the model is wrong?
- Active: Confirmatory analysis based on the model: Design of experiments, power analysis, etc etc
- ✓ Design of experiments, power analysis, etc etc wrong if the model assumption is wrong
- "Small" data
- ✓ Mechanical turk, fMRI data, genetics, cito, OSF, international collaboration many labs, etc.

Breiman's: Critique on the data modelling approach

Wrong presumption

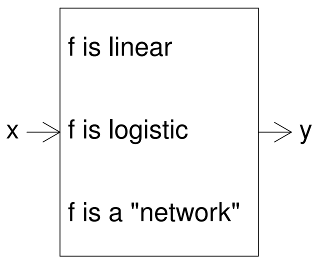
Let data be generated according to the data model *f*, where *f* is linear/logistic regression/...

Example: Uncritical use of linear regression to, for instance, bimodal data.

Breiman's: Critique on the data modelling approach

Wrong presumption

Let data be generated according to the data model *f*, where *f* is linear/logistic regression/...



Breiman's: Critique on the data modelling approach

Wrong presumption

Let data be generated according to the data model *f*, where *f* is linear/logistic regression/...

Focus on modelling the sampling distribution of the error not the *f*:

$$\underbrace{y}_{\text{obs}} - \overbrace{f}^{\text{known}}(\underbrace{x}_{\text{obs}}) = \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Example: in ANOVA, sums of squared error, R^2, etc. If the errors are big, it is implied that the theory is bad. To quantify "big" require sampling distribution.

Breiman's: Critique on the data modelling approach

Wrong presumption

Let data be generated according to the data model f, where f is linear/logistic regression/...

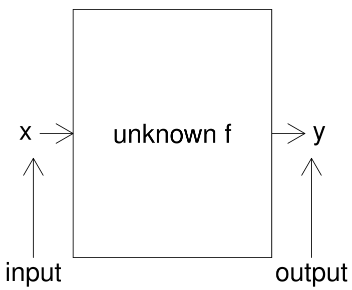
- To calculate sampling distributions stuck with simple models (Linear)
- Conclusion are about the model mechanism, not about nature's mechanism
- If model is a poor emulation of nature, conclusions will be wrong.

Breiman's experience from consulting

- The classical (linear) models are tractable, but typically yield bad predictions
- Live with the data before modelling
- Algorithmic models are also good
- Predictive accuracy on test set is the criterion for how good the model is
- Computers are important

Recall setting

Problem:



Recall setting

Problem: With which f does nature generate data

$$y = f(x) + \epsilon$$

Goal:

- Prediction based on past data. Learn (estimate) unknown f
- Give a prediction accuracy.

"Live with the data before modelling". Data split in three parts:

- Training set to learn f from the data

Recall setting

Problem: With which f does nature generate data

$$y = f(x) + \epsilon$$

Goal:

- Prediction based on past data. Learn (estimate) unknown f
- Give a prediction accuracy.

"Live with the data before modelling". Data split in three parts:

- Training set to learn f from the data
- Validation set to do model selection

Recall setting

Problem: With which f does nature generate data

$$y = f(x) + \epsilon$$

Goal:

- Prediction based on past data. Learn (estimate) unknown f
- Give a prediction accuracy.

"Live with the data before modelling". Data split in three parts:

- Training set to learn f from the data
- Validation set To do model selection
- Test set to estimate the prediction accuracy

General learning procedure

Recall data set consists of n pairs, say,

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, \begin{pmatrix} x_4 \\ y_4 \end{pmatrix}, \begin{pmatrix} x_5 \\ y_5 \end{pmatrix}, \begin{pmatrix} x_6 \\ y_6 \end{pmatrix}, \begin{pmatrix} x_7 \\ y_7 \end{pmatrix}, \begin{pmatrix} x_8 \\ y_8 \end{pmatrix}, \begin{pmatrix} x_9 \\ y_9 \end{pmatrix}, \begin{pmatrix} x_{10} \\ y_{10} \end{pmatrix},$$

Corresponding formula

$$y = f(x) + \epsilon$$

General learning procedure

Randomly select training samples, say,

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, \begin{pmatrix} x_4 \\ y_4 \end{pmatrix}, \begin{pmatrix} x_5 \\ y_5 \end{pmatrix}, \begin{pmatrix} x_6 \\ y_6 \end{pmatrix}, \begin{pmatrix} x_7 \\ y_7 \end{pmatrix}, \begin{pmatrix} x_8 \\ y_8 \end{pmatrix}, \begin{pmatrix} x_9 \\ y_9 \end{pmatrix}, \begin{pmatrix} x_{10} \\ y_{10} \end{pmatrix}$$

- Training set is used to learn f from the data.

Fill in the true $x_{\text{train},i}, y_{\text{train},i}$

$$y_{\text{train},i} = f(x_{\text{train},i}) + \epsilon$$

find that f for which the loss between

$$\frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \text{Loss}\Big(y_{\text{train},i}, f(x_{\text{train},i})\Big)$$

is smallest. Call the minimiser f_{trained}

General learning procedure

Use the other samples as test samples, say,

$$\binom{x_1}{y_1}, \binom{x_2}{y_2}, \binom{x_3}{y_3}, \binom{x_4}{y_4}, \binom{x_5}{y_5}, \binom{x_6}{y_6}, \binom{x_7}{y_7}, \binom{x_8}{y_8}, \binom{x_9}{y_9}, \binom{x_{10}}{y_{10}}$$

- Training set is used to learn f from the data.
- Test set is used to derive the prediction accuracy.

Fill in f_{trained} and apply it to x to yield $y_{\text{implied}} = f_{\text{trained}}(x_{\text{test},i})$ and compare the estimate the error between y_{implied} with true $y_{\text{test},i}$,

$$\epsilon_{\text{estim}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \text{Loss}\Big(y_{\text{test},i}, f_{\text{trained}}(x_{\text{test},i})\Big)$$

The error ϵ_{estim} so estimated serves as the prediction accuracy.

General learning procedure

Generalise to a new instance

$$\binom{x_1}{y_1}, \ldots, \binom{x_9}{y_9}, \binom{x_{10}}{y_{10}}, \binom{x_{\text{new}}}{\ldots}$$

- Training set is used to learn f from the data.
- Test set is used to derive the prediction accuracy.

Final answer for yet unseen features x_{new} use to generate y_{new}

$$y_{\text{new}} = \overbrace{f_{\text{trained}}(x_{\text{new}})}^{y_{\text{implied}}} \pm \underbrace{\epsilon_{\text{estim}}}_{\text{Generalisation error}}$$

Remarks

- Only assumption is that the data $\binom{x_i}{y_i}$ are iid, that is, they are generated with the same (true, fixed, but unknown) f^*.
- The data are assumed to be true
- No, specification of the Loss function. The loss function replaces the assumptions on the error (typically, Gaussian error in "standard" statistics)
- No, specification of the collection $\mathcal{F}$ of functions f that we believe to be viable

Loss functions

For categorical data, typically, zero-one (all or nothing) loss:

$$\text{Loss}\Big(y_i, f(x_i)\Big) = \begin{cases} 0 \text{ if } y_i = f(x_i) \\ 1 \text{ if } y_i \neq f(x_i) \end{cases}$$

Note the hard rule, here observation y_i "supervise" the learning.

Loss functions

For categorical data, typically, zero-one (all or nothing) loss:

$$\text{Loss}\Big(y_i, f(x_i)\Big) = \begin{cases} 0 \text{ if } y_i = f(x_i) \\ 1 \text{ if } y_i \neq f(x_i) \end{cases}$$

Note the hard rule, here observation y_i "supervise" the learning.
For continuous data, typically, mean-squared error loss:

$$\text{Loss}\Big(Y, f(x)\Big) = E\Big[Y - f(x)\Big]^2$$

Here, E is the expectation (average) with respect to the true relationship f^* between X and Y.

Loss functions

For categorical data, typically, zero-one (all or nothing) loss:

$$\text{Loss}\Big(y_i, f(x_i)\Big) = \begin{cases} 0 \text{ if } y_i = f(x_i) \\ 1 \text{ if } y_i \neq f(x_i) \end{cases}$$

Note the hard rule, here observation y_i "supervise" the learning.
For continuous data, typically, mean-squared error loss:

$$\text{Loss}\Big(Y, f(x)\Big) = \frac{1}{n}\sum_{i=1}^{n}\Big[y_i - f(x_i)\Big]^2$$

Here, the expectation E is replaced with the empirical average with respect to the data being true.

Bias-Variance trade-off and overfitting

For continuous data, typically, mean-squared error loss:

$$\text{Loss}\Big(Y, f(x)\Big) = E\Big[Y - f(x)\Big]^2$$

Here, E is the expectation (average) with respect to the true f^*.

Bias-Variance trade-off and overfitting

For continuous data, typically, mean-squared error loss:

$$\begin{aligned}\text{Loss}\Big(Y, f(x)\Big) &= E\Big[Y - f(x)\Big]^2 \\ &= \text{Var}(f(x)) + \Big[\text{Bias}(f(x))\Big]^2 + \text{Var}(\epsilon)\end{aligned}$$

Both Var and E, thus, Bias, are with respect to the true f^*.

Bias-Variance trade-off and overfitting

For continuous data, typically, mean-squared error loss:

$$\underbrace{\mathrm{Var}(f(x)) + \left[\mathrm{Bias}(f(x))\right]^2}_{\text{Structural error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Unavoidable error}}$$

Recall that f_{trained} is from minimising this loss. The more complicated a candidate f, the smaller the unavoidable error, as everything is seen as structural. Problem: overfitting.

Overfitting

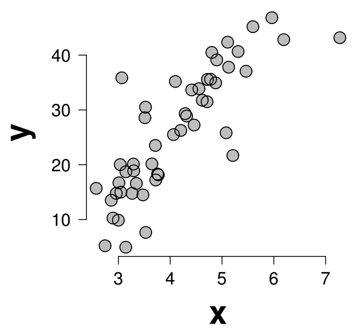
For continuous data, typically, mean-squared error loss:

$$\underbrace{\mathrm{Var}(f(x)) + \left[\mathrm{Bias}(f(x))\right]^2}_{\text{Structural error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Unavoidable error}}$$

Overfitting occurs if the fs under consideration are too complicated. An over complicated f generalises badly and is recognised by low bias, high variance.

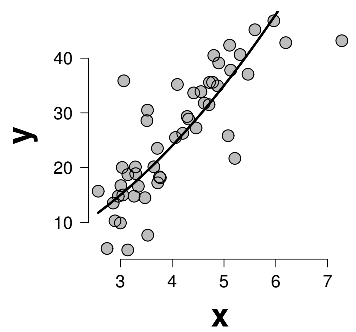
Overfitting: Over complicated f: low bias, high variance.

True data generating $f^*(x) = x^2 + 2x + \epsilon$, where ϵ is a Laplace distribution (thicker tails than normal).
Raw data:



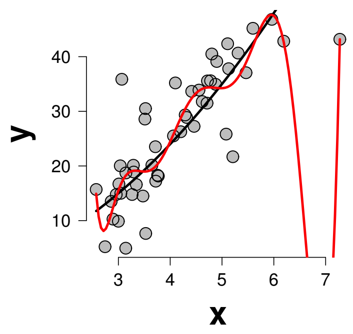
Overfitting: Over complicated f: low bias, high variance.

Raw data and true function f^*:



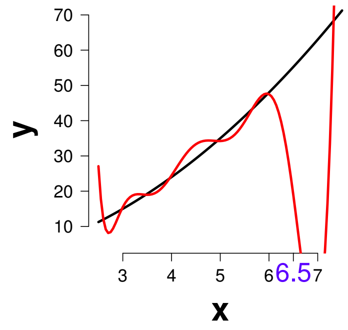
Overfitting: Over complicated f: low bias, high variance.

Fitted with a polynomial of order nine.



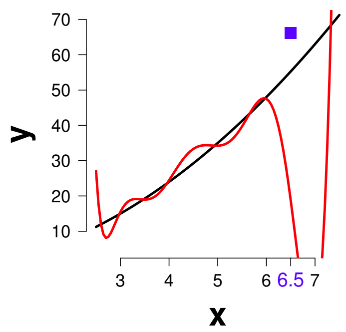
Overfitting: Over complicated f: low bias, high variance.

Observe new test sample/instance $x_{\text{test}} = 6.5$



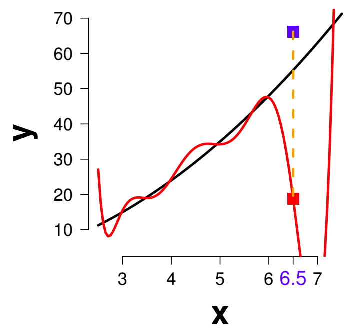
Overfitting: Over complicated f: low bias, high variance.

Observe new test sample/instance $x_{\text{test}} = 6.5$ and $y_{\text{test}} = 66$

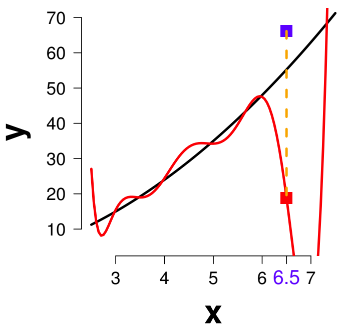


Overfitting: Over complicated f: low bias, high variance.

Large loss between y_{test} and y_{implied}:



Overfitting: Over complicated f: low bias, high variance.



Example conclusion: your expected survival years are 12 years $\pm$ 50 years. Meaningless prediction.

Model complexity

- True data generation was done with a polynomial of degree 2, polynomial of degree 9 is too complex. Hence, the collection of candidate fs should be restricted to those fs with max degree 2.
- In reality, don't know the "complexity" of the true. How to choose the collection of candidates $\mathcal{F}$?

Two types of model complexity

- The structure of f. Linear, polynomial (but still a summation of terms, thus, linear) <- nothing new same as in statistics

Two types of model complexity

- The structure of f. Linear, polynomial (but still a summation of terms, thus, linear). Neural networks (non-linear), support vector machines, generalised additive models, kernel smoothing, splines, reproducing kernel Hilbert space, regression trees.
- Number of features $n < p$.

"Standard" setting

- The functions fs in $\mathcal{F}$ are linear
- More data than features $p < n$.

Regularisation

- The functions fs in $\mathcal{F}$ are linear
- More features than data $n < p$. Hazard of overfitting
- Control the complexity with additional tuning parameter λ (for instance, lasso) Hence, $\mathcal{F} = \mathcal{F}_\lambda$

Example:

$$f_\lambda(x) = \underbrace{\theta_0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_p x^p}_{\text{Over complicated structure}} + \underbrace{\lambda}_{\text{tuning parameter}} |\theta| \quad (1)$$

Here $\lambda|\theta|$ acts as a penalty for complexity. How to tune λ?

Regularisation

- The functions fs in $\mathcal{F}$ are linear
- More features than data $n < p$. Hazard of overfitting
- Control the complexity with additional tuning parameter λ (for instance, lasso) Hence, $\mathcal{F} = \mathcal{F}_\lambda$

Example:

$$f_\lambda(x) = \underbrace{\theta_0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_p x^p}_{\text{Over complicated structure}} + \underbrace{\lambda}_{\text{tuning parameter}} |\theta| \quad (1)$$

Here $\lambda|\theta|$ acts as a penalty for complexity. Tune λ with validation set aka repeat the general procedure many times with different (fixed) λ.

General learning procedure

Split data into three sets.

- Training set is used to learn f from the data.
- Validation set to tune λ (model selection).
- Test set is used to derive the prediction accuracy.

Partition λ. For the lasso, $\lambda = 0$ no regularisation, for $\lambda = \infty$ only the constant function is viable. Say, $\lambda = 0, 2, 2^2, \ldots, 2^{10}$. For each fixed λ follow the general procedure f_λ.

General learning procedure

- Training set is used to learn f from the data.
- Validation set to tune λ (model selection).
- Test set is used to derive the prediction accuracy.

Say, $\lambda = 0, 2, 2^2, \ldots, 2^{10}$.
We get

$$\mathcal{F} = f_{0,\text{trained}}, f_{2,\text{trained}}, f_{2^2,\text{trained}}, \ldots, f_{2^{10},\text{trained}} \quad (2)$$

General learning procedure

- Training set is used to learn f from the data.
- Validation set to tune λ (model selection).
- Test set is used to derive the prediction accuracy.

Say, $\lambda = 0, 2, 2^2, \ldots, 2^{10}$.
Pick $f_{\lambda,\text{trained}}$ with lowest average loss on the validation set.

$$f_{\text{val}} = \arg\min \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} \text{loss}\left(f_{\lambda,\text{trained}}(x_{i,\text{val}}), y_{i,\text{val}}\right) \quad (2)$$

General learning procedure

- Training set is used to learn f from the data.
- Validation set to tune λ (model selection).
- Test set is used to derive the prediction accuracy.

Say, $\lambda = 0, 2, 2^2, \ldots, 2^{10}$.
Estimate prediction accuracy by averaging the average loss on the test set

$$\epsilon_{\text{est}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \text{loss}\left(f_{\text{val}}(x_{i,\text{test}}), y_{i,\text{test}}\right) \quad (2)$$

- Gave general idea of supervised learning. Role of training, validation and test set
- Which specific classes of $\mathcal{F}$ to take. This seminar discusses a couple of them: Neural networks (non-linear), support vector machines, generalised additive models, kernel smoothing, splines, reproducing kernel Hilbert space, regression trees.
- How to minimise? Technicallities
- Selection method based on select the best. Other methods, bagging and boosting

Further organisation

- Changed name from machine learning reading group to statistical learning seminar
- Reading groups die
- Seminar does not require everyone to read everything
- Requires a small peak in preparation of a talk
- Not necessary to understand everything. Can be practical and theoretical
- Still good to read things in advanced. Also website with youtube clips are available.