# Linear Models

*Connectionist and Statistical Language Processing*

Frank Keller

keller@coli.uni-sb.de

Computerlinguistik
Universität des Saarlandes



# Overview

- classification vs. numeric prediction
- linear regression
- least square estimation
- evaluating a numeric model, correlation
- selecting a regression model
- linear regression for classification
- regression trees, model trees

Literature: Witten and Frank (2000: ch. 4, 6), Howell (2002: ch. 15).



# Numeric Prediction

An instance in the data set has the following general form:

$\langle a_{1,i}, a_{2,i}, \ldots, a_{k,i}, x_i \rangle$

where $a_{1,i}, \ldots, a_{k,i}$ are attribute values, and $x_i$ is the target value, for the $i$-th instance in the data set.

So far we have only seen *classification tasks*, where the target value $x_i$ is categorical (represents a class).

Techniques such as decision tree and Naive Bayes are not (directly) applicable if the target is numeric. Instead algorithms for *numeric prediction* can be used, e.g., *linear models*.



# Example

Predict CPU performance from configuration data:

| cycle time (ns) | memory min (kB) | memory max (kB) | cache (kB) | chan min | chan max | perfor- mance |
|---|---|---|---|---|---|---|
| 125 | 256 | 6000 | 256 | 16 | 128 | 198 |
| 29 | 8000 | 32000 | 32 | 8 | 32 | 269 |
| 29 | 8000 | 32000 | 32 | 8 | 32 | 220 |
| 29 | 8000 | 32000 | 32 | 8 | 32 | 172 |
| 29 | 8000 | 16000 | 32 | 8 | 16 | 132 |
| ... | | | | | | |
| 125 | 2000 | 8000 | 0 | 2 | 14 | 52 |
| 480 | 512 | 4000 | 32 | 0 | 0 | 67 |
| 480 | 1000 | 4000 | 0 | 0 | 0 | 45 |

# Linear Regression

Linear regression is a technique for numeric predictions that's widely used in psychology, medical research, etc.

**Key idea:** find a linear equation that predicts the target value $x$ from the attribute values $a_1, \dots, a_k$:

$$(1) \qquad x = w_0 + w_1 a_1 + w_2 a_2 + \dots + w_k a_k$$

Here, $w_1, \dots w_k$ are the *regression coefficients*, $w_0$ is called the *intercept*. These are the model parameters that need to be induced from the data set.



# Linear Regression

The regression equation computes the following *predicted value* $x_i'$ for the $i$-th instance in the data set.

$$(2) \qquad x_i' = w_0 + w_1 a_{1,i}, w_2 a_{2,i}, \dots, w_k a_{k,i} = w_0 + \sum_{j=1}^{k} w_j a_{j,i}$$

**Key idea:** to determine the coefficients $w_0, \dots w_k$, minimize $e$, the squared difference between the predicted and the actual value, summed over all $n$ instances in the data set:

$$(3) \qquad e = \sum_{i=1}^{n} (x_i - x_i')^2 = \sum_{i=1}^{n} \left( x_i - w_0 - \sum_{j=1}^{k} w_j a_{j,i} \right)^2$$

The method for this is called *Least Square Estimation* (LSE).



# Least Square Estimation

We demonstrate how LSE works with the simple case of $k = 1$, dropping the intercept $w_0$. The error equation (3) simplifies to (abbreviating $w_1 = w$ and $a_1 = a$):

$$(4) \qquad e = \sum_i (x_i - w a_i)^2 = \sum_i (x_i^2 - 2 w a_i x_i + w^2 a_i^2)$$

Now differentiate the error equation in (4) with respect to $w$:

$$(5) \qquad \frac{\partial e}{\partial w} = \sum_i (-2 a_i x_i + 2 w a_i^2) = -2 \sum_i a_i x_i + 2w \sum_i a_i^2$$

The derivative is the *slope* of the error function. The slope is zero at all points at which the function has a minimum.



# Least Square Estimation

To minimize the squared error for the data set, we therefore set the derivative in (5) equal to zero:

$$(6) \qquad -2 \sum_i a_i x_i + 2w \sum_i a_i^2 = 0$$

By resolving this equation to $w$, we obtain a formula for computing the value of $w$ that minimizes the error:

$$(7) \qquad w = \frac{\sum_i a_i x_i}{\sum_i a_i^2}$$

This formula can be generalized to regression equations with more than one coefficient.

# Example

Sample data set:

| $a$ | $x$ |
|---|---|
| 1 | 2 |
| 2 | 5 |
| $-1$ | $-2$ |
| 5 | 8 |

Use Least Square Estimation to compute $w$ for this data set:

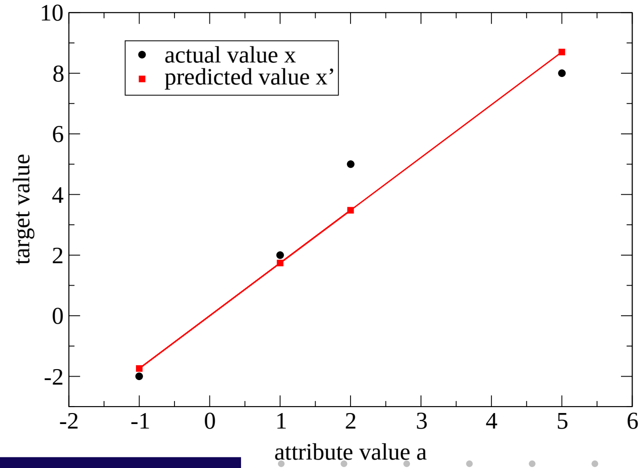
$$w = \frac{\sum_i x_i a_i}{\sum_i a_i^2} = \frac{1 \cdot 2 + 2 \cdot 5 + (-1)(-2) + 5 \cdot 8}{1^2 + 2^2 + (-1)^2 + 5^2} = 1.74 \tag{8}$$

Regression equation: $x' = wa = 1.74a$



# Evaluating a Numeric Model

The fit of a regression model can be visualized by plotting the predicted data values against the actual values.






# Evaluating a Numeric Model

A suitable numeric measure for the fit of a linear model is the *mean squared error:*

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - x_i')^2 \tag{9}$$

Intuitively, this represents how much the predicted values diverge from the actual values on average.

Note that the MSE is the quantity the LSE algorithm minimizes.



# Example

Compute the mean squared error for the sample data set and the regression equation $x' = 1.74a$:

| $a$ | $x$ | $x'$ | $(x - x')^2$ |
|---|---|---|---|
| 1 | 2 | 1.74 | 0.068 |
| 2 | 5 | 3.48 | 1.346 |
| $-1$ | $-2$ | $-1.74$ | 0.068 |
| 5 | 8 | 8.70 | 0.490 |

$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - x_i')^2 = \frac{1}{4}(0.068 + 1.346 + 0.068 + 0.490) =$ 0.646

# Correlation Coefficient

The correlation coefficient $r$ measures the *degree of linear association* between predicted and the actual values:

$$r = \frac{S_{PA}}{S_P S_A} \tag{10}$$

$$S_{PA} = \frac{\sum_{i=1}^{n}(x'_i - \overline{x'})(x_i - \overline{x})}{n-1} \tag{11}$$

$$S_P = \sqrt{\frac{\sum_{i=1}^{n}(x'_i - \overline{x'})^2}{n-1}} \quad S_A = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{12}$$

Here $\overline{x}$ and $\overline{x'}$ are the means of the actual and predicted values, $S_P$ and $S_A$ their *standard deviations*. $S_{PA}$ is the *covariance* of the actual and predicted values.



# Example

Compute the correlation coefficient for the example data set:

$\overline{x} = 3.25 \quad \overline{x'} = 3.14$

$$\begin{aligned} S_{PA} &= ((1.74-3.14)(2-3.25)+(3.48-3.14)(5-3.25)+ \\ &\quad (-1.74-3.14)(-2-3.25)+(8.70-3.14)(8-3.25))/3 \\ &= 18.13 \\ S_P^2 &= ((1.74-3.14)^2+(3.48-3.14)^2+ \\ &\quad (-1.74-3.14)^2+(8.70-3.14)^2)/3 = 18.93 \\ S_A^2 &= ((2-3.25)^2+(5-3.25)^2+ \\ &\quad (-2-3.25)^2+(8-3.25)^2)/3 = 18.25 \end{aligned}$$

$r = 18.13/(\sqrt{18.93} \cdot \sqrt{18.25}) = 0.975$

$r^2 = 0.951$



# Correlation Coefficient

Some important properties:

- The correlation coefficient $r$ ranges from $1.0$ (perfect correlation) to $0$ (no correlation) to $-1.0$ (negative correlation).
- Intuitively, $r$ expresses how well the data points fit on the *straight line* described by the regression model.
- We can test if $r$ is *significant*. Null hypothesis: there is no linear relationship between predicted and actual values.
- We can also compute $r^2$, which represents the *amount of variance accounted for* by the regression model.



# Partial Correlations

We can compute the *multiple correlation coefficient* that tells us how well the full regression model (with all attributes) fits the target values.

We can also compute the correlation between the values of a single attribute and the target values.

However this is not very useful as attributes can be *intercorrelated*, i.e., they correlate with each other (colinearity).

We need to compute the *partial correlation coefficient*, which tells us how much variance is *uniquely* accounted for by an attribute once the other attributes are partialled out.

# Selecting a Regression Model

We want to build a regression model that only contains the attributes that are predictive. Several methods to achieve this:

- *All subsets:* compute models for all subsets of attributes and chose the one with the highest multiple $r$.
- *Backward elimination:* compute a model for all attributes, and then eliminate the one with the lowest partial $r$. Iterate until the multiple $r$ deteriorates.
- *Forward selection:* compute a model consisting only of the attributes with the highest partial $r$. Then add the next best attribute. Stop when the multiple $r$ doesn't improve.

Different model selection algorithm can yield different models.



# Testing on Unseen Data

We compute the regression weights and perform the model selection on the *training data*.

To evaluate the resulting model, we compute model fit on unseen *test data* using LSE or the correlation coefficient.

Techniques for testing on unseen data (see last lecture):

- *Holdout:* set aside a random sample of the data set for testing, train on the rest.
- *$k$-fold crossvalidation:* split the data in $k$ random partition and test on each one in turn.
- *leave-one-out:* set $k$ to the number of instances in the data set, i.e., test on each instance separately.



# Linear Regression for Classification

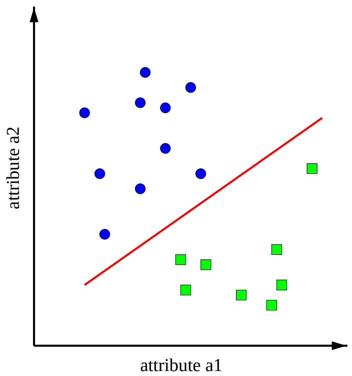
Regression can be applied to *classification:*

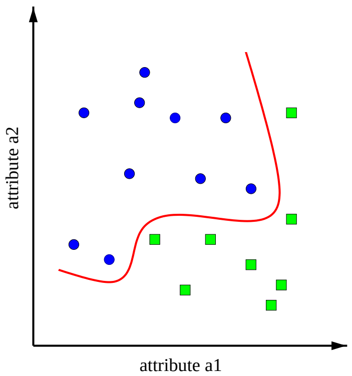
- Perform a separate regression on each class, set the target value to $1$ if an instance is in the class, and $0$ if it is not in the class.
- The regression equation then approximates the *membership function* for the class ($1$ for members, $0$ for non-members).
- To classify a new instance, compute the regression value for each membership function, and assign the new instance the class with the highest value.
- This procedure is called *multiresponse linear regression*.



# Linear Separability

Linear regression approximates a linear function. This means that the classes have to be *linearly separable*.




For many interesting problems, this is not the case.

# Regression Trees

*Regression trees* are decision trees for numeric attributes. The leaves are not labeled with classes, but with the *mean of the target values* of the instances classified by a given branch.

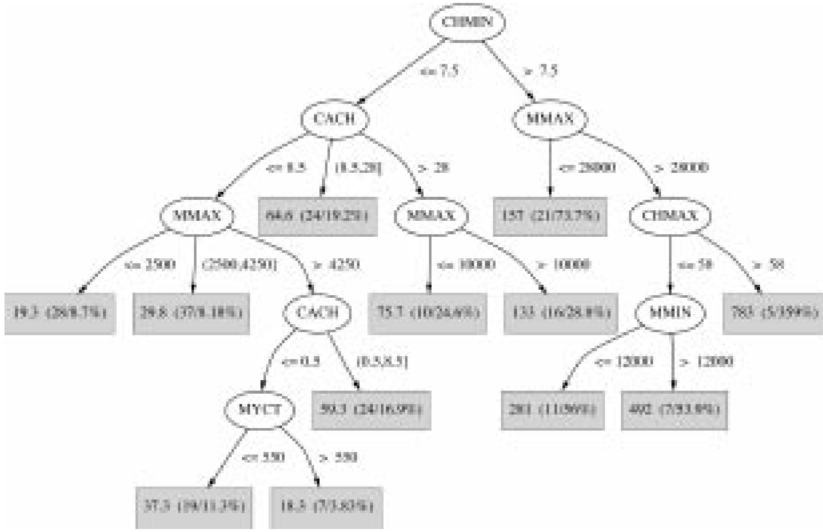
To construct a regression tree, chose splitting attributes to minimize intrasubset variation for each branch. Maximize *standard deviation reduction* (instead of information gain):

$$\text{SDR} = \sigma_T - \sum_i \frac{|T_i|}{|T|}\sigma_{T_i} \tag{13}$$

Where $T$ is the set of instances classified at a given node, $T_1, \ldots, T_i$ are the subset that $T$ is split into, and $\sigma$ is the standard deviation.



# Example






# Model Trees

*Model trees* are regression trees that have linear regression models at their leaves, not just numeric values.

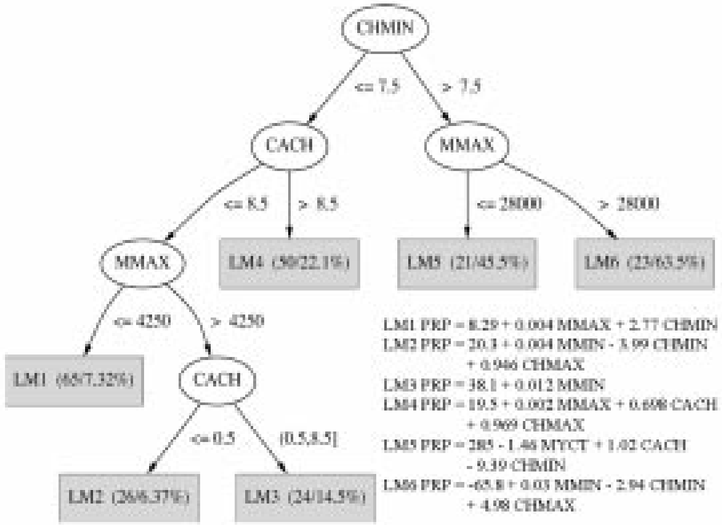
Induction algorithm:

- Induce a regression tree using standard deviation reduction as the splitting criterion.
- Prune back the tree starting from the leaves.
- For each leaf construct a regression model that accounts for the instances classified by this leaf.

Model trees get around the problem of linear separability by *combining several regression models*.



# Example



# Summary

Linear regression models are used for *numeric prediction*. They fit a *linear equation* that combines attributes values to predict a numeric target attribute.

*Least square estimation* can be used to determine the coefficients in the regression equation so that the difference between predicted and actual values is minimal.

A numeric model can be evaluates using the mean squared error or the *correlation coefficient*.

Regression models can be used for classification either directly in *multiresponse regression* or in combination with decision trees: *regression trees, model trees*.



# References

Howell, David C. 2002. *Statistical Methods for Psychology*. Pacific Grove, CA: Duxbury, 5th edn.

Witten, Ian H., and Eibe Frank. 2000. *Data Mining: Practical Machine Learing Tools and Techniques with Java Implementations.* San Diego, CA: Morgan Kaufmann.