Defect Content Estimation for Inspections:

Regression and Machine Learning

Frank Padberg

Universität Karlsruhe

Germany

Our Task

reliably estimate

the number of defects in a software document

from the outcome of an inspection!

Estimation Methods

- capture–recapture methods (Eick ea. ICSE 1992)
- curve–fitting methods (Wohlin ea. ICSE 1998)
- studies show that estimates are far too unreliable (Briand ea. TSE 2000, Biffl ea. ICSE 2001)
- interval estimate method (Padberg ICSE 2002)
- outperforms other methods on benchmark dataset

Interval Estimate Method

- use empirical data from past inspections for estimating
- stochastic model relates inspection outcome (the w_k and d) to the true number N of defects
- use that relation to estimate N for a new document from its inspection outcome

Regression Approach

- learn relationship between observable features of an inspection and true number of defects contained in the document

Regression Approach

- learn relationship between observable features of an inspection and true number of defects contained in the document
- view defect content estimation as a regression problem

Regression Approach

- learn relationship between observable features of an inspection and true number of defects contained in the document
- view defect content estimation as a regression problem
- again, need empirical database

Candidate Features

- derived from zero–one matrix
- use the w_k and d to get: TDD, AVE, MIN, MAX, STD
- example A1:

  (9, 7, 6, 13, 9, 6) and 23 yields

TDD	AVE	MIN	MAX	STD
23	8.3	6	13	2.4

Input Data for Linear Regression

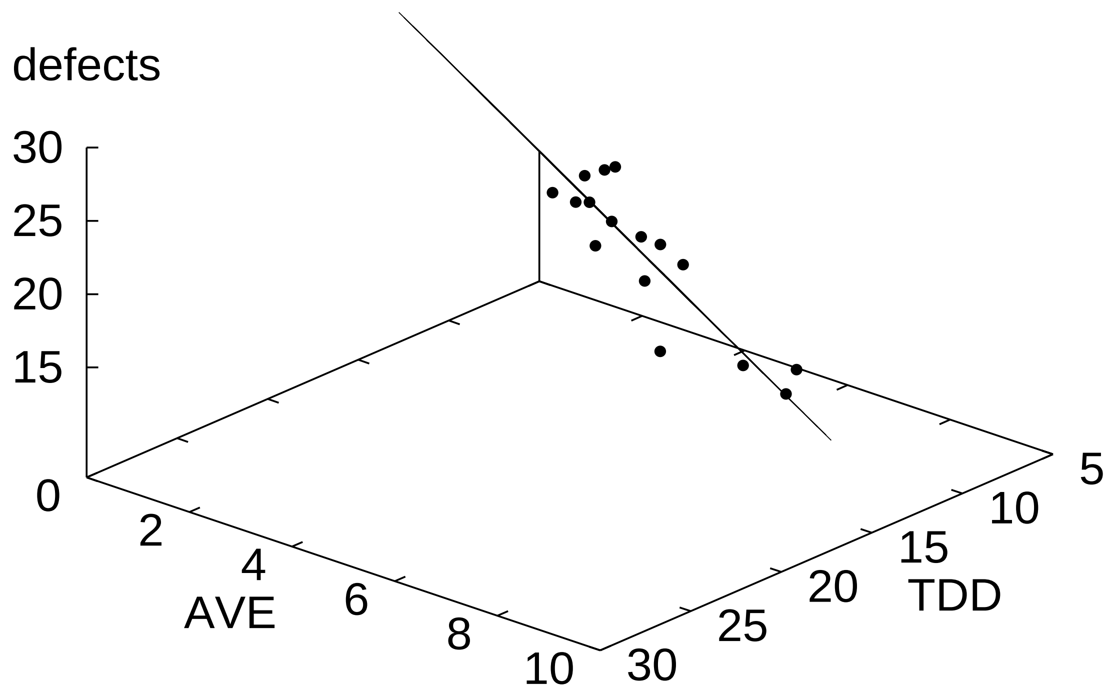
- correlation analysis yields ranking

$$\text{TDD} > \text{AVE} > \text{MIN} > \text{MAX} > \text{STD}$$

- some datapoints:

inspection	TDD	AVE	target
A1	23	8.3	30
B1	20	6.0	28
C1	10	3.2	18
D1	6	1.3	15

Regression Hyperplane

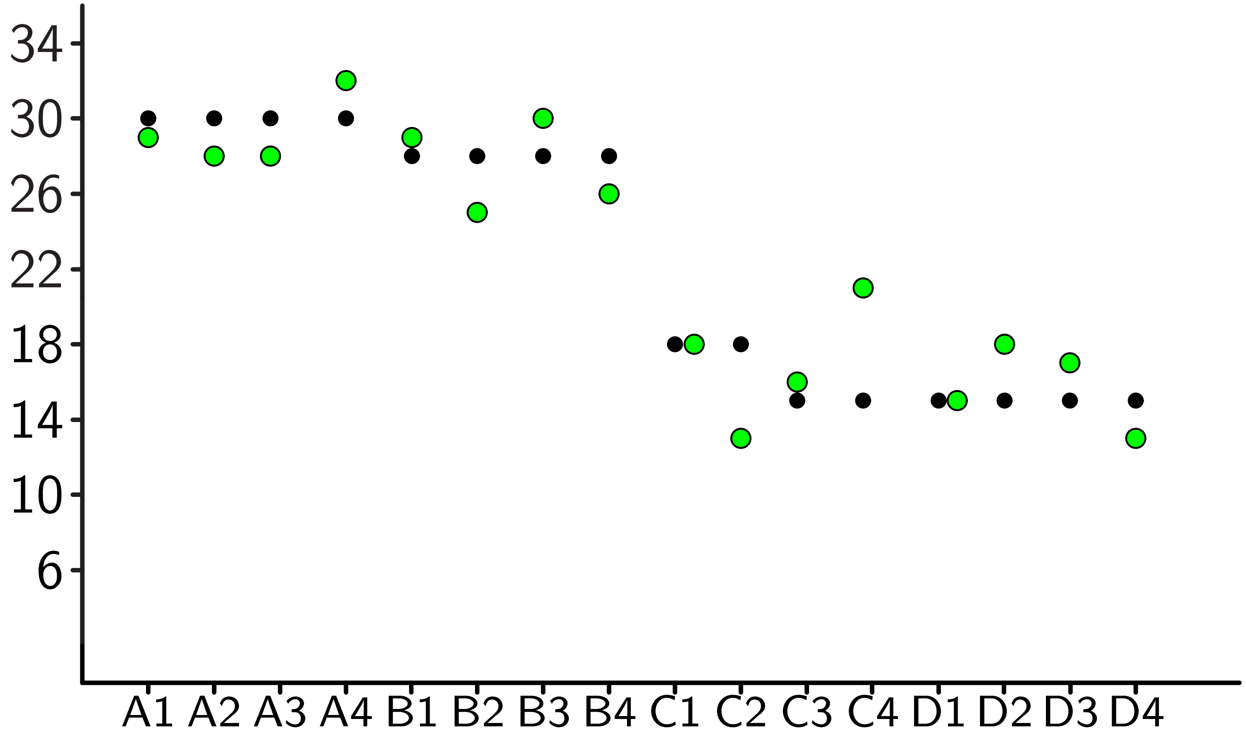


all 16 inspections

Jackknife Validation

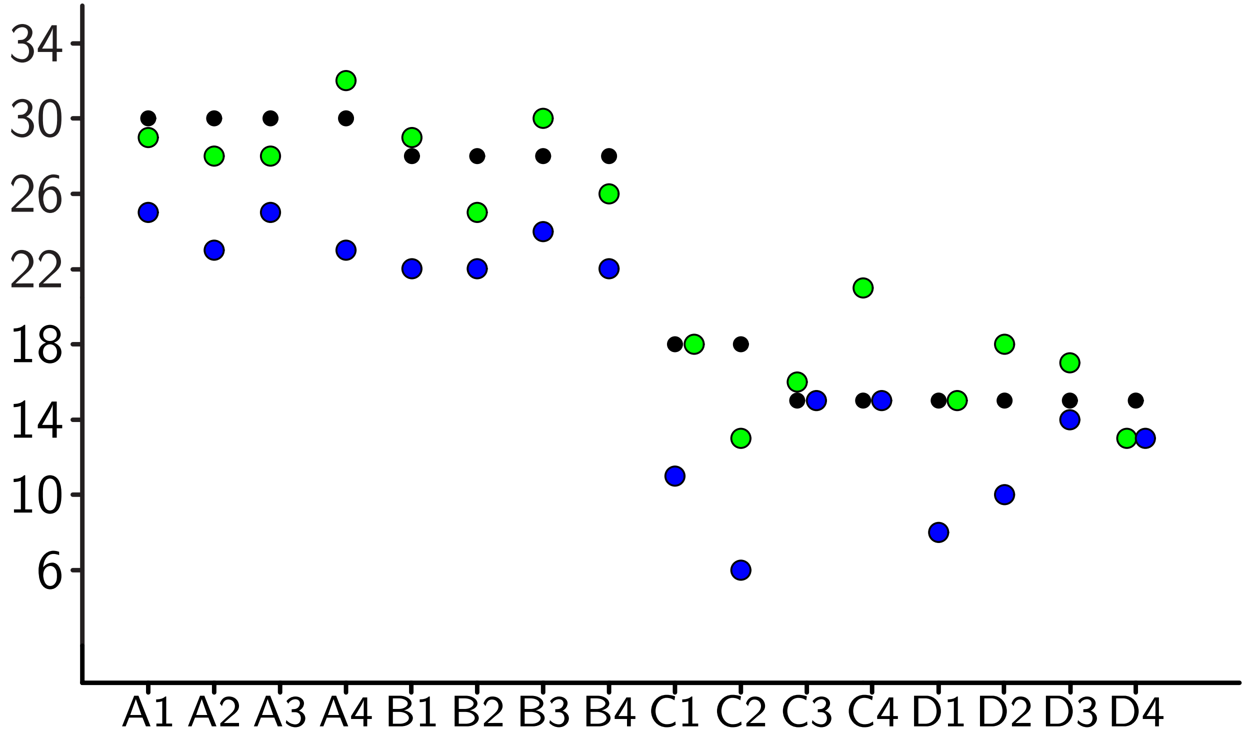
- leave out an inspection from the database
- compute the regression hyperplane using the remaining 15 inspections
- compute the regression estimate for the one inspection which was left out
- compare the estimate with the true value of the number of defects

Linear Regression Estimates



jackknife error of 11 percent

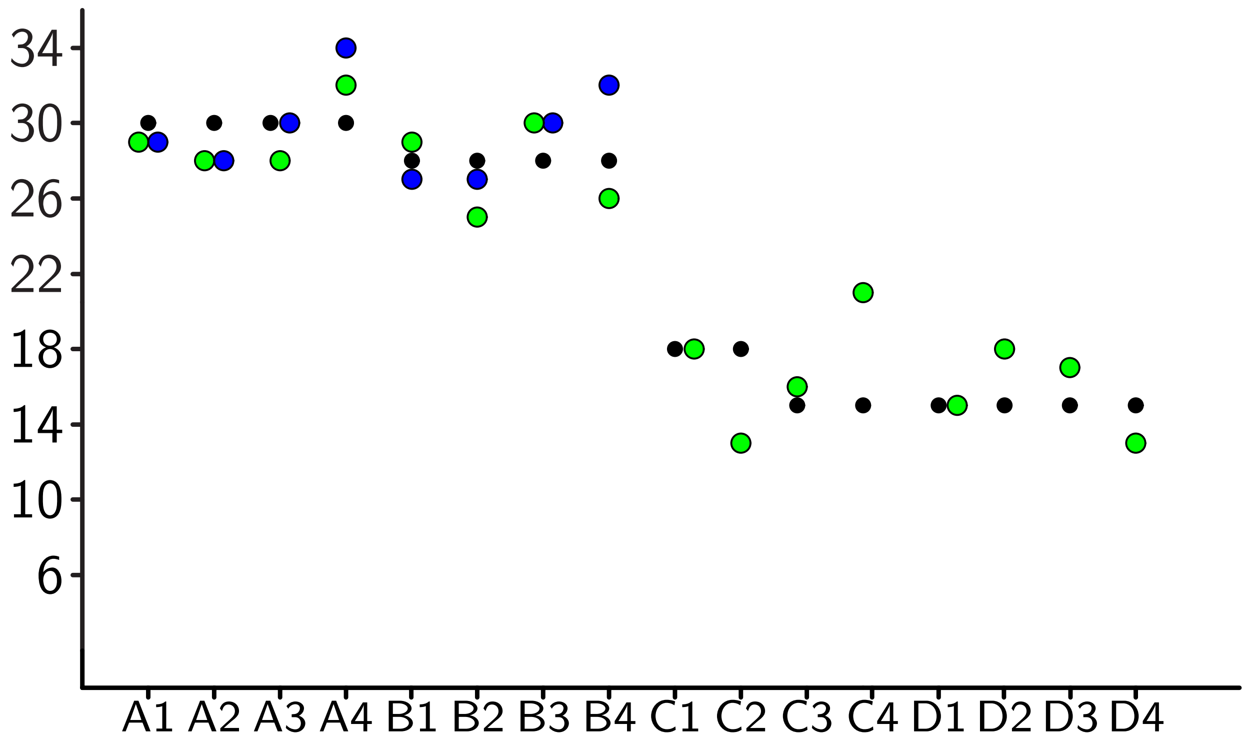
Linear Regression versus Capture–Recapture



clearly outperforms capture–recapture

(11 percent versus 24)

Linear Regression versus Interval Estimates

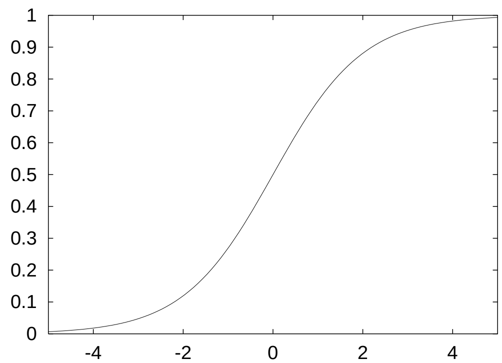


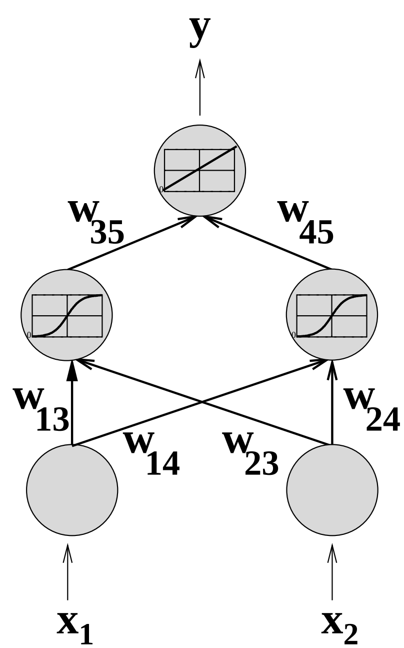
similar performance on one half of the dataset

(7 percent each)

Non-Linear Regression: Neural Networks

$$\mathrm{logist}\,(\,x\,) \;=\; \frac{1}{1\;+\;e^{-x}} \qquad\qquad s_i \;=\; \mathrm{logist}\Big(\sum_j w_{ji}\cdot s_j\Big)$$





Neural Network Methodology

- determine a set of candidate features
- select an appropriate subset (feature selection)
- train different neural networks on the dataset
- select the best neural network (model selection)

Input Data for Non-Linear Regression

- non-linear feature selection yields ranking

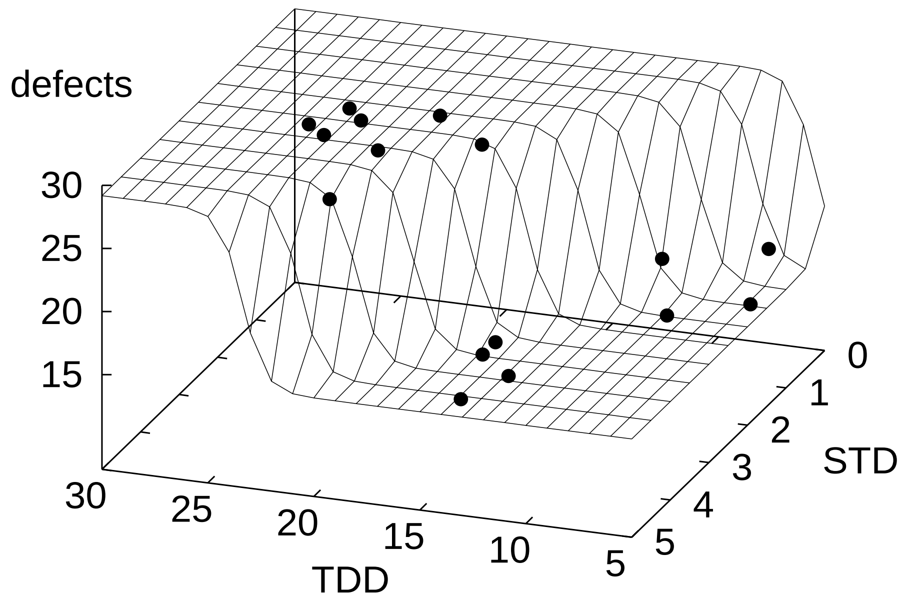
$$\text{TDD} > \text{STD} > \text{MAX} > \text{MIN} > \text{AVE}$$

- STD instead of AVE

- some training patterns:

inspection	TDD	STD	target
A1	23	2.4	30
B1	20	1.7	28
C1	10	1.5	18
D1	6	1.4	15

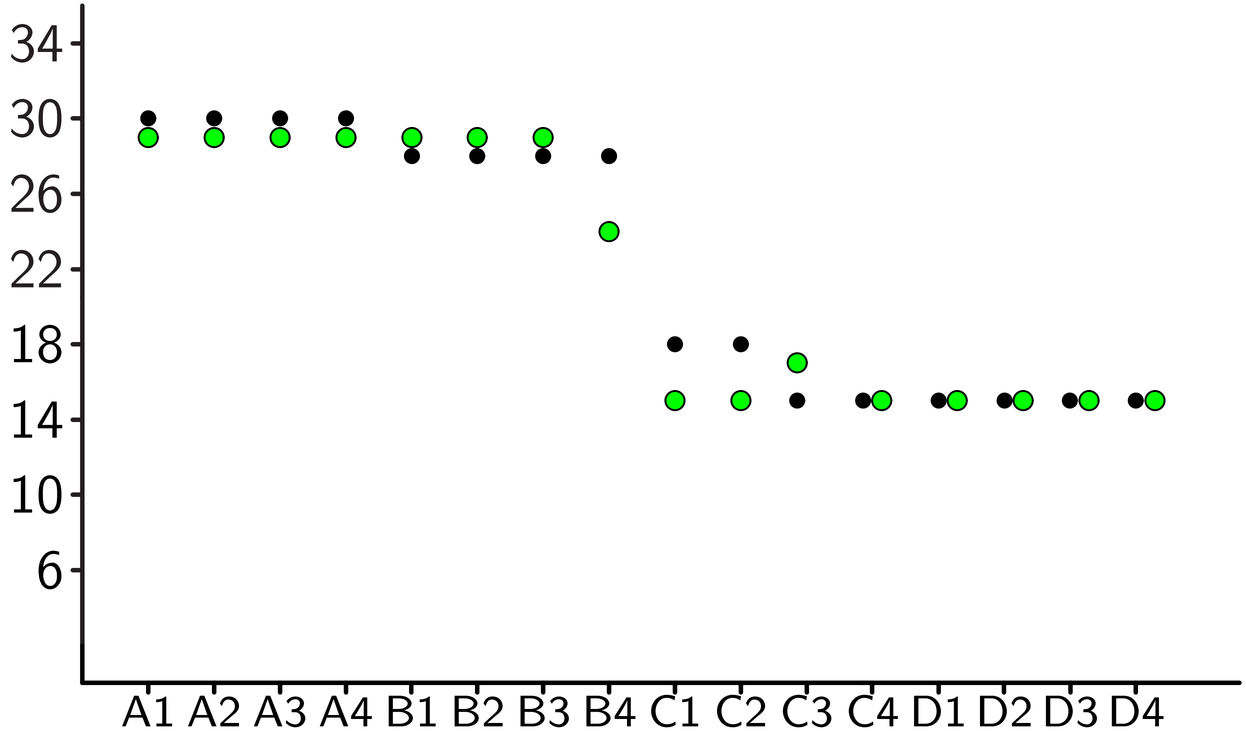
Non-Linear Regression Surface



neural network with two hidden units in one layer

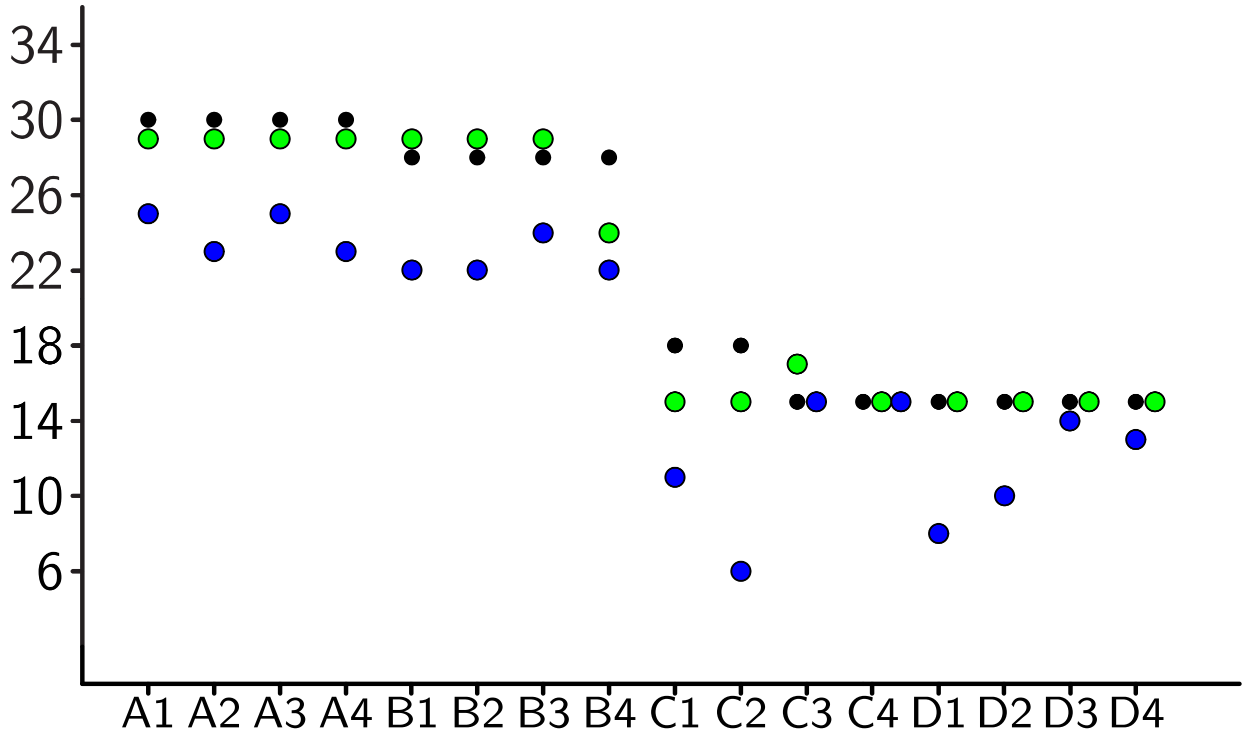
all 16 inspections

Neural Network Estimates



jackknife error of 6 percent

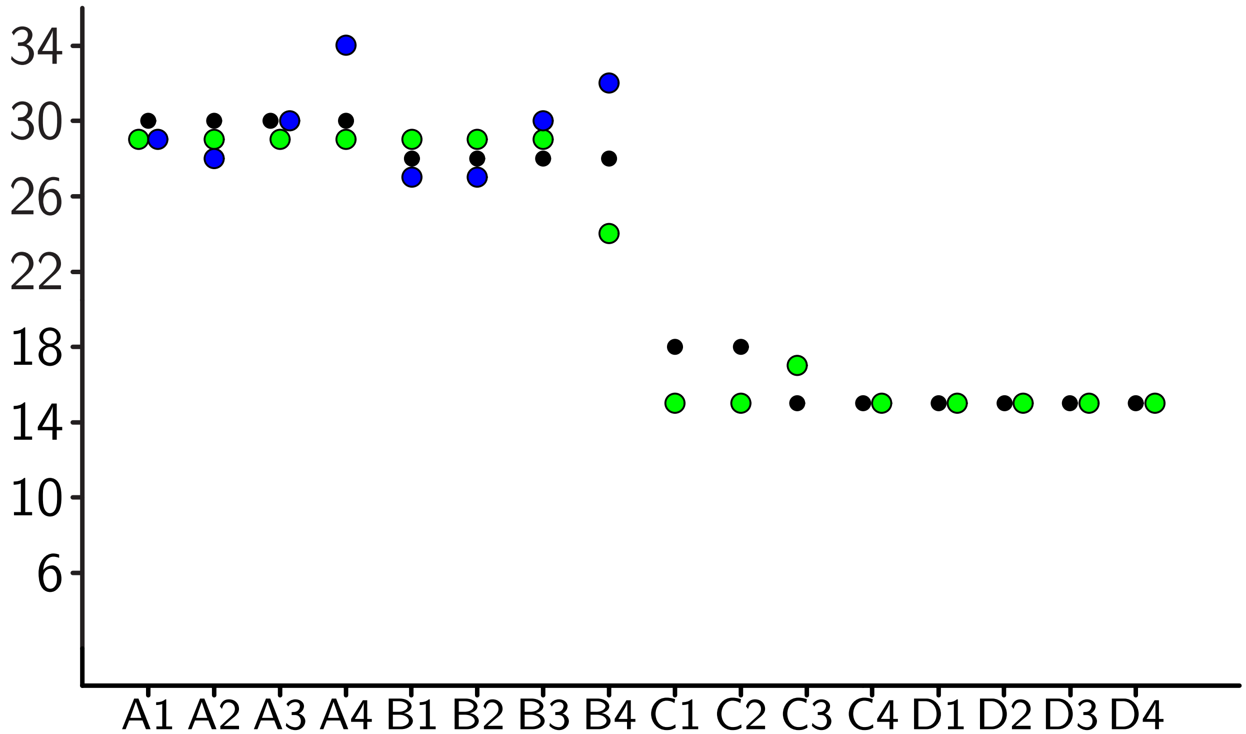
Neural Networks versus Capture–Recapture



clearly outperforms capture–recapture

(6 percent versus 24)

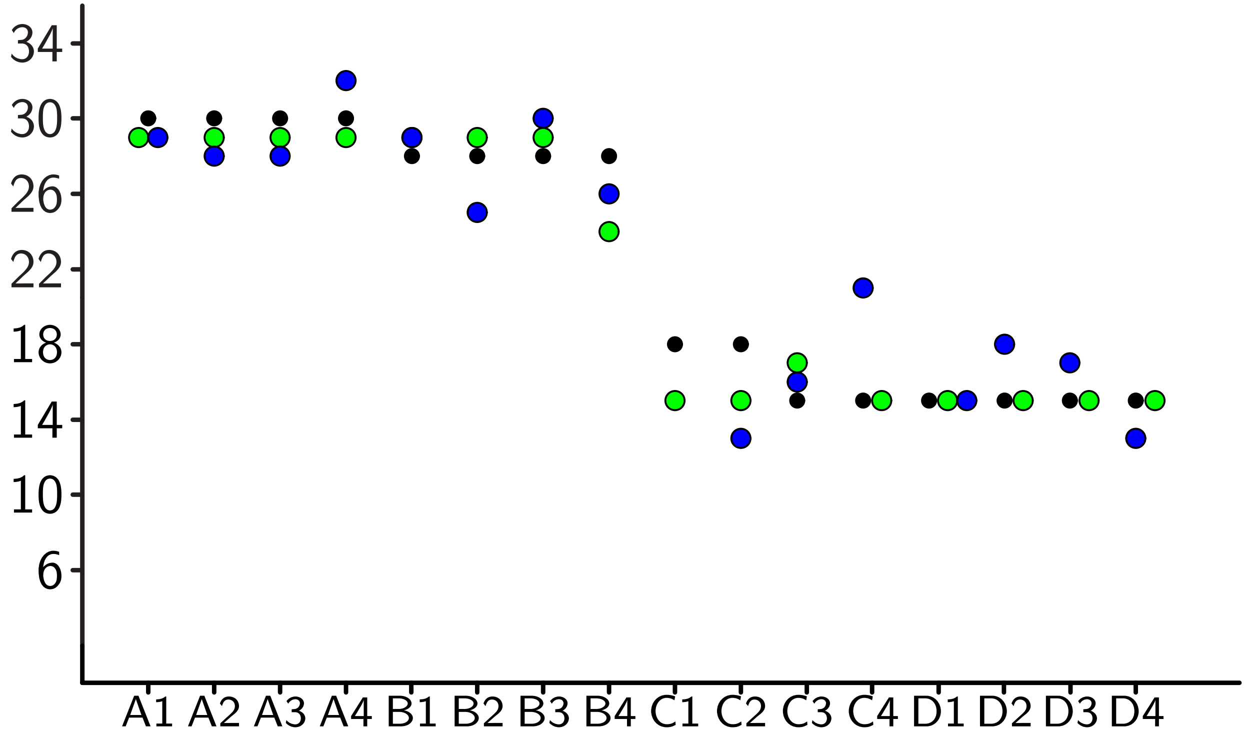
Neural Networks versus Interval Estimates



similar performance on one half of the dataset

(5 percent versus 7)

Neural Networks versus Linear Regression



outperforms linear regression

(6 percent versus 11, smaller variance)

Neural Network Advantages

- much flexibility when fitting to data
- detects non-linearity in the data
- gives guidelines which features to use
- works well even with small datasets
- automatically adapts to different document types and sizes

Neural Network Topology

- number of inputs
- number of hidden layers
- number of units in hidden layers
- connections between layers

Training a Neural Network

- fit regression function to training data
- non-linear optimization process (choose weights to minimize error on training data)
- might get caught in local minimum
- train networks with different initial weights

Model Selection

- good generalization (predictive power) is more important than a small training error
- can use cross-validation on additional dataset
- we use model evidence (Bayesian technique)
- model evidence works well if network is small

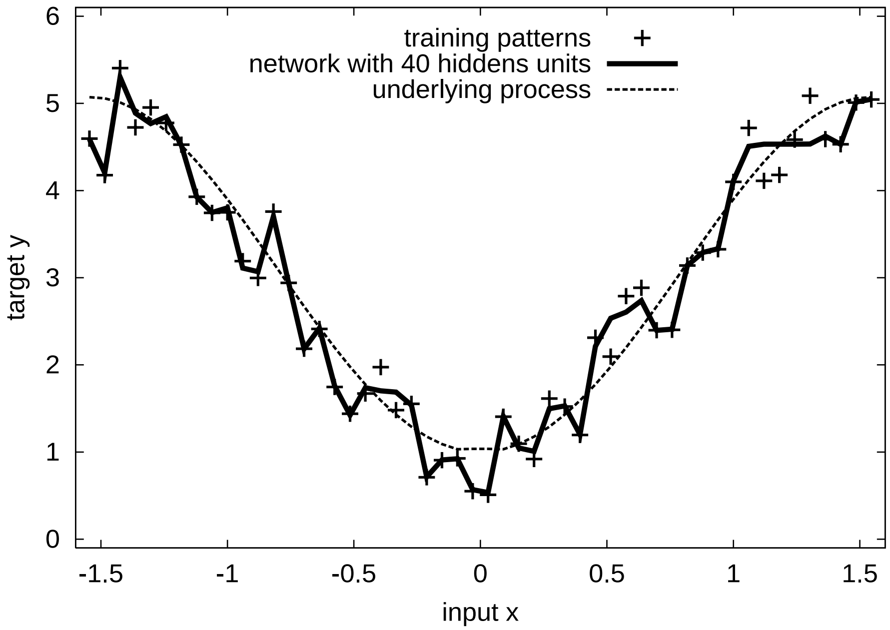
Empty Space Phenomenon

features	patterns
1	4
2	19
3	67
4	223
5	768
6	2790

maximum number of features that can be used

depends on number of training patterns available

Overfitting



good fit to training patterns, but

underlying smooth process poorly approximated

Technical Countermeasures

- Empty Space Phenomenon

  $\longrightarrow$ follow Silverman's rule of thumb

  $\longrightarrow$ apply feature selection

  $\longrightarrow$ we use mutual information

Technical Countermeasures

- Empty Space Phenomenon

  $\longrightarrow$ follow Silverman's rule of thumb

  $\longrightarrow$ apply feature selection

  $\longrightarrow$ we use mutual information

- Overfitting

  $\longrightarrow$ prefer small networks

  $\longrightarrow$ prefer networks with small weights

  $\longrightarrow$ use regularization during training

Mutual Information

$$\mathrm{H}(T) \;-\; \mathrm{H}(T \mid X) \;=$$

$$\iint p(x, t) \cdot \log \frac{p(x, t)}{p(x)\, p(t)}$$

- measures stochastic dependence between target T and feature X
- detects non-linear dependencies

Regularization

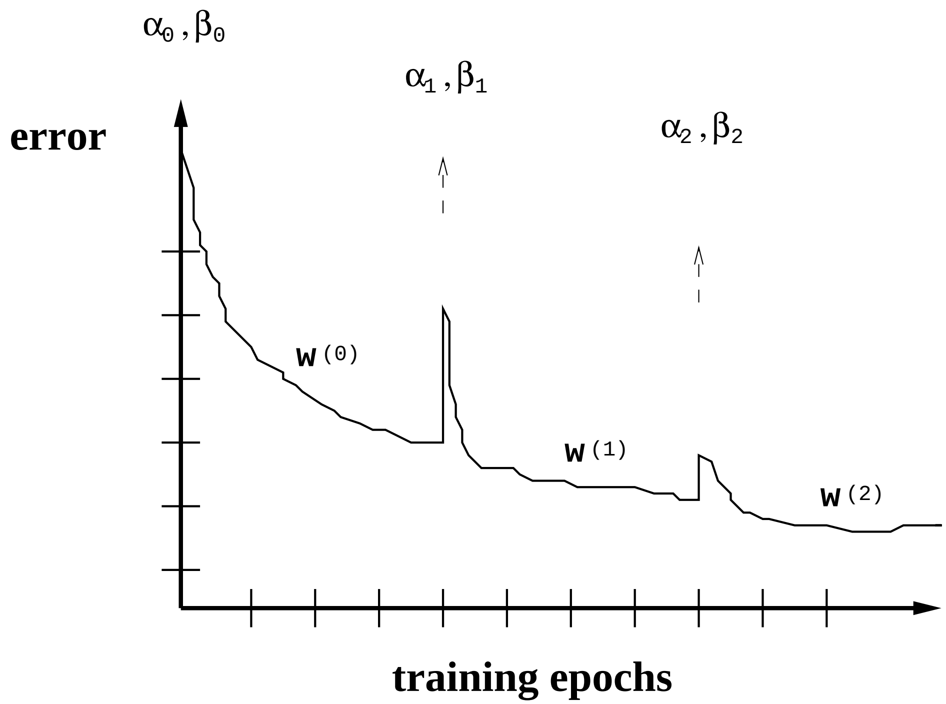
- prefer networks with small weights w_{ji}

- minimize regularized error

$$\beta \cdot \mathsf{E}_{\mathsf{train}} \; + \; \alpha \cdot \sum {w_{ji}}^2$$

- α and β are additional parameters

Iterative Training Procedure



alternate between optimizing the weights w_{ji}

and updating the parameters α, β

Results

Method	mean error	max error
Capture–Recapture	24 %	67 %
Detection Profile	36 %	113 %
Linear Regression	11 %	40 %
Interval Estimates	(7 %)	(14 %)
Neural Networks	6 %	17 %

all three novel approaches are promising

need more empirical data for validation

Regression Approach Summary

- uses empirical data from past inspections
- linear regression
- neural networks as non-linear regression
- outperforms existing methods
- see Ragg, Padberg, Schoknecht ICANN 2002

Let's Try This, Too !