A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms

Dietrich Wettschereck
David W. Aha
Takao Mohri

Presented by Colin Prinn

The index - a brief outline

1. Introduction
2. Context
   2.1 Lazy Learning Algorithms
   2.2 The K-nn classifier
   2.3 Scope for our review
3. A framework for feature weighting methods
   3.1 Bias: Performance vs Preset
      3.1.1 Performance bias
         3.1.1.1 Online optimisers
         3.1.1.2 Batch optimisers
      3.1.2 Preset bias
         3.1.2.1 Conditional probabilities
         3.1.2.2 Class projection
         3.1.2.3 Mutual Information
   3.2 Weight Space: Weighting vs Selection
   3.3 Representation: Given vs Transformed
   3.4 Generality: Global vs Local
   3.5 Knowledge: None vs Domain Specific
4. Comparative evaluation
   4.1 Selected algorithms
   4.2 Selected datasets
   4.3 Methodology and Initial Results
   4.4 Summary: Trends and their evaluation
      4.4.1 Intra-model comparisons
         4.4.1.1 Performance Bias methods
         4.4.1.2 Preset bias methods 4.4.2

Summary
5. Discussion and Implications
6. Related Work
   6.1 Similar Studies
   6.2 Performance vs Preset Biases
   6.3 Information Theory
   6.4 Instance Weighting
   6.5 Alternative architectures for lazy algorithms
7. Conclusions
8. Acknowledgements
9. References

"Space constraints prevent a more detailed discussion"

In other words...

This paper provides us with

- An overview of k-NN variants and their approaches to optimize distance calculations.
- A framework with which we may contrast and compare lazy learning feature weight setting algorithms
- An emperical comparison of a selection of k-NN variants across a variety of data sets
- A review of observations and some trends which hint toward optimal use cases

k-NN has more parameters than k

k-NN eliminates parametrisation by incorporating invariants into it's design. Examples of these parameters are:

- The cost of incorrect classification is constant across cases.
- The mean and variance of feature values is not incorporated into distance calculation.
- Equal weights are applied to each feature independent of frequency, range or inter-feature dependence

The aim of this paper is to evaluate the effect of variations on these parameters with respect to efficiency and optimality.

A framework for feature weighting methods

Dimension	Possible Values
Bias	{ Performance, Preset }
Weight Space	{ Continuous, Binary }
Representation	{ Given, Transformed }
Generality	{ Global, Local }
Knowledge	{ Poor, Intensive }

Feature weighting bias - Performance

- Performance Bias algorithms adjust feature weights by evaluating feature values.
  - Online classifiers
    - perform a single pass through the dataset.
    - detect irrelevant features.
    - fail to detect redundant or interacting features.
  - Batch Optimizers
    - repeatedly process case instances
    - some use knowledge of a functions' gradient to increase learning times.

Feature weighting bias - Preset

- Do not use feedback from classifier to assign weights
  - conditional probability
    - weight features based on correlation
  - class projection
    - weight features based on distribution
  - mutual information
    - weight features according to which features define the class
    - calculated using the frequency of the class and the frequency of the feature value in the training set
    - discretises continuous feature values

Weight space reduction
- feature selection

- Assigns binary weights to features.
- Can significantly increase learning times.
- Effective for cases with redundant or irrelevant features
- Feature selection methods
  - induced decision trees
  - random mutation - hill climbing
  - parallel search
  - beam search with stepwise selection
  - stepwise feature removal in oblivious decision trees

Representational Transformations

Using case features as given may not always be the best use of the given features.

Feature transformation can be achieved either using domain knowledge to transform given features into combined or ignored features, or using algorithms to determine correlated and irrelevant features.

Representation transformations can reduce the retrieval time and increase accuracy and may also reveal feature correlations.

Weight Generality

Feature weights need not always be applied globally. Weights may differ locally instead of remaining constant across the entire instance set.

Local weights may differ at the feature level, in that weights differ as a function of the features' value
or
Local weights may differ across case instances as a function of the distribution of feature values.

Two draw backs are noted. Local weighting is sensitive to noisy training data and distinct local distance functions may obscure useful feature information.

Knowledge: None vs Domain Specific

The final category of feature weighting alternatives is to apply some domain knowledge to the task of feature selection and prioritisation.

The Experiment !

Due to space constraints, the authors only tested a subset of lazy learning algorithms from the Performance based feature weighting and Preset feature weighting categories.

The authors then selected 14 datasets to evaluate the algorithms. 10 of the datasets provide a selection of controlled conditions.

Name	Type
k-NN	Control
RELIEF-F	Online performance optimiser
k-NN vsm	Batch performance optimiser
CCF	Preset conditional probability
VDM	Preset class projection
MVDM	Preset class Projection
MI	Preset mutual information

The results

	Feature Weight Learning Algorithm						
	Control	Performance Bias Method		Preset Bias Method			
Dataset	none	Relief-F	k-NN$_{VSM}$	CCF	VDM	MVDM	MI
Banded	83.0±0.4	**11.2**	**12.8**	**12.8**	**12.8**	**12.8**	**10.8**
Sinusoidal	74.2±0.8	**5.9**	**14.4**	**-9.1**	**-9.1**	**-9.2**	**-4.6**
Gauss-band	78.3±0.5	**8.6**	**16.6**	**14.9**	**15.5**	**17.5**	**12.1**
Parity	67.3±0.1	**32.7**	**32.7**	1.3	1.7	1.9	2.2
LED-7 Display	72.7±0.4	-1.0	0.0	-1.5	-1.4	-1.3	-1.2
LED-7+17B	52.5±0.5	**19.2**	**15.5**	**9.2**	**19.4**	**18.9**	**19.4**
LED-7+17C	68.8±0.6	**3.3**	**2.0**	**-5.6**	**2.0**	**2.3**	**3.6**
Waveform-21	82.1±0.4	0.3	-0.5	**-6.1**	**-3.7**	**-3.9**	0.5
Waveform-40	81.3±0.9	1.7	1.2	**-3.4**	-0.7	-0.4	1.0
Cleveland	82.4±0.8	-0.5	0.0	-1.3	0.2	0.7	-0.6
Hungarian	82.6±0.7	**-2.5**	-0.4	-0.1	0.1	0.0	0.1
Voting	92.6±0.7	**2.9**	**2.5**	1.0	**2.1**	**2.1**	**2.0**
Isolet	84.2±0.3	0.4	**1.9**	**-1.1**	**-3.9**	**1.6**	**1.6**
NETtalk	69.6±0.2	**9.2**	**6.6**	**7.7**	**10.0**	**12.1**	**9.7**

Trends

1. Preset bias methods performed poorly on the sinusoidal task due to improper discretisation. When corrected, they performed of the order of three standard deviations above control.
2. Performance differences between performance and preset bias methods for the parity task indicate that performance bias methods provide higher accuracy when dealing with interacting features.
3. Feature weighting methods have a substantially higher learning rate. Performance bais methods more so than preset.
4. Feature weighting is superior to feature selection.

Conclusions

The authors developed a lazy learning algorithm feature weighting categorisation framework.

They used their framework to compare a selection of algorithms and provided a review of their relative strengths.

The paper goes on to list a collection of future research directions which may be initiated using this framework.