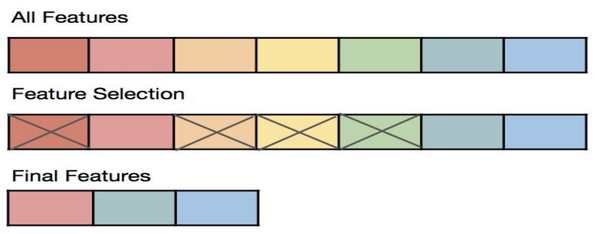
Agnostic Feature Selection

{p.ossenkopp, m.lumpe, niklas.blume, niels.nuthmann}@stud.uni-hannover.de

1. Feature Selection

- Feature Selection is the process of selecting a subset of relevant features for use in model construction.
- Reasons for feature selection:
  - simplification of models to make them easier to interpret
  - shorter training times
  - avoid the curse of dimensionality
  - enhanced generalization by reducing overfitting
- The central premise when using a feature selection technique is that the data contains some features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.
- In difference to feature extraction, which creates new features from functions of the original features, feature selection returns a subset of the features.
- Possible applications:
  - Reduce text data
  - DNA microarray



Source: 1)

2. Experimental Setting

- Hyperparameters
  - Hidden layer size: Chosen by ID estimator; Adam Optimizer; Mean squared error loss; Learning rate: 0.01; Epochs: 150; Regularization factor: 1
- Metrics
  - K-means clustering accuracy (ACC) and coefficient determination (R^2 score)

Dataset	# Samples	# Features	# Classes	ID	Data Type
arcene	200	10000	2	40	Medical
Isolet	1560	617	26	9	Sound processing
ORL	400	1024	40	6	Face image
pixraw10P	100	10000	10	4	Face image
ProstateGE	102	5966	2	23	Medical
TOX171	171	5748	4	15	Medical
warpPie10P	130	2400	10	3	Face image
Yale	165	1024	15	10	Face image
BASEHOCK	1993	4862	2	PCA	Text
COIL20	1440	1024	50	PCA	Object Images
GLI-85	85	22283	2	PCA	Microarray, biological

3. AgnoS

- Problems with Vanilla Autoencoder (AE) for feature selection:
  - How to extract the feature importance from the model?
  - How to estimate the hidden layer size?
  - How to prevent the model from focusing on redundant features?

- Solutions for the above problems (introducing AgnoS):
  - Use a score function that extracts the feature importance from the trained AE
  - Use the Poisson model to estimate the intrinsic dimension (ID) of the feature space
  - Use one of three regularizers to enforce sparsity in the model

Algorithm AgnoS

Input : Feature set $F = \{f_1, \ldots, f_D\}$
Parameter : λ
Output : Ranking of features in F
Normalize each feature with zero and unit variance.
Estimate intrinsic dimension ID of F.
Initialize neural network with $d = ID$ neurons in the hidden layer.
Repeat

Backpropagate $L(F)$

until convergence

Rank features by decreasing scores with $Score(f_i)$

$$AgnosW : L(F) = \sum_{i}^{D} \|\hat{f}_i - f_i\|_2^2 + \lambda \sum_{i=1}^{D} \|W_{i,\cdot}\|_2$$

$$Score_W(f_i) = \|W_{i,\cdot}\|_\infty$$

$$AgnosG : L(F) = \sum_{i}^{D} \|\hat{f}_i - f_i\|_2^2 + \lambda \sum_{i=1}^{D} \sqrt{\sum_{k=1}^{n} \sum_{j=1}^{d} \left(\frac{\partial \phi_j}{\partial f_i}(x_k)\right)^2}$$

$$Score_G(f_i) = \max_{j \epsilon [1,\ldots,d]} \sum_{k=1}^{n} \left(\frac{\partial \phi_j}{\partial f_i}(x_k)\right)^2$$

$$AgnosS : L(F) = \sum_{i}^{D} \|\hat{f}_i - f_i\|_2^2 + \lambda \sum_{i=1}^{D} |a_i|$$

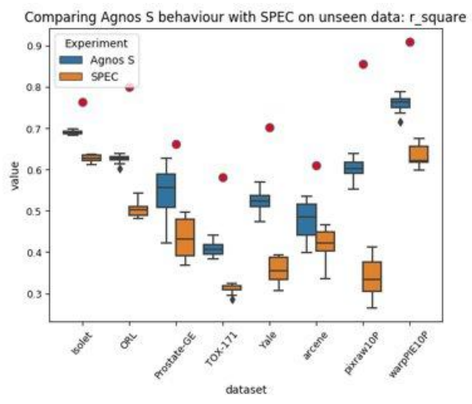
$$Score_S(f_i) = |a_i|$$

Site notes:

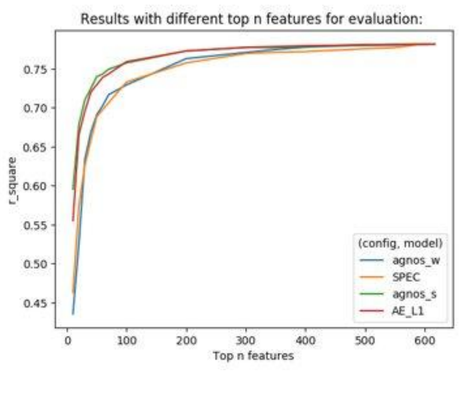
- W : Matrix of encoder weights
- f_i : Feature at position i
- $\hat{f}_i$: Reconstructed feature i
- λ : Regularization coefficient
- D : Dimensionality of the feature space
- $\widehat{ID}$: Estimated intrinsic dimensionality
- W_i : Weight vector for feature i
- ϕ : Output vector of the encoder function with dimensionality d
- x_k : Sample in row k of the dataset
- a : Slack variable vector with dimensionality D

4. Results

- We managed to achieve comparable results to the paper using AgnoS-S and AgnoS-W
- The introduced R^2 score is less susceptible to variations
- **Experiment 1** - comparing AgnoS-S to SPEC
  - AgnoS-S performs slightly better but has more variance
  - But AgnoS-S takes a lot longer to train than its counterpart
- **Experiment 2** - comparing AgnoS-S to SPEC with unseen data
  - Complex algorithms are worse at generalizing
  - Accuracy is now more in favor of SPEC
  - R^2 is still better when using AgnoS-S
- **Experiment 3** - comparing AgnoS-S to random feature selection
  - Random feature selection was very close in terms of accuracy and R^2 score and in some cases it was even better than the AgnoS-S values
- **Experiment 4** - comparing different scores on top n features
  - In the paper only the top 100 features are selected for evaluation
  - So we tried using a different amount of top selected features
- Drawback of AgnoS is the long runtime
  - Added early stopping reduce it
- A L1 regularized autoencoder performs worse than AgnoS-S
- A plain autoencoder performs pretty similar to AgnoS-S and better than AgnoS-W



Experiment 2



Experiment 4

- Two plots for each experiment - one for accuracy and one for R^2 score
- The red circle in the boxplot shows the results in the paper

5. Conclusion

- Random feature selection is not as bad as it seemed to be
- It is hard to replicate paper results just using the paper as guidance
- The R^2 scores are pretty similar for every method on one dataset, this raises doubts for its information value

- It makes sense to question everything explained by the authors, as one can learn a lot more and have different points of view

Leibniz Universität Hannover

Sources:

1) https://en.wikipedia.org/wiki/Feature_selection
2) https://ecmlpkdd2019.org/downloads/paper/744.pdf

Agnostic Feature Selection repository:

- https://nielsmitie.github.io/DM_Lab/
- https://github.com/Nielsmitie/DM_Lab

Check out our website!

By Philip Ossenkopp, Niels Nuthmann, Maximilian Lumpe, Niklas Blume