Unit 5. Feature transformation and feature selection

Artificial Intelligence and Learning

felipe.alonso@urjc.es

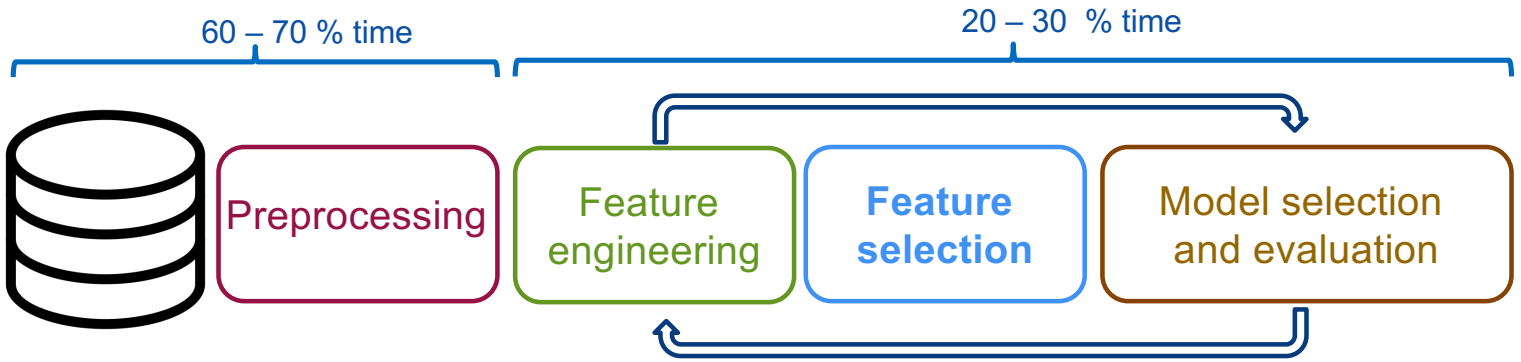
Bibliography

- Chapters 3 and 6 from "*An Introduction to Statistical Learning*", James et al.
- Scikit-learn documentation
- Additional resources (at the end of these slides)

Contents

Contents

5.1 The (*classic*) machine learning pipeline



- Noisy features/variables
- **Redundant variables(*)**
- Outliers
- Missing values
- Categorial variables

- Create a new set of features based on knowledge domain

- **Select a subset of "*most relevant*" features for bulding a ML model**

- Choosing free parameters
- Evaluate learning algorithm (CV, train-validation-test)

(*) highly correlated with the output, yet uncorrelated with each other

Contents

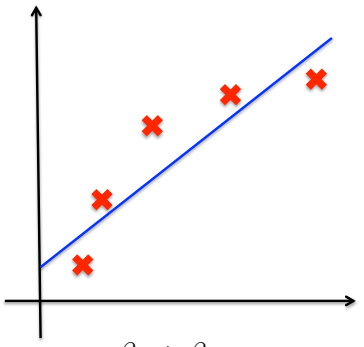
5.2. FS motivation

1. **Interpretability**
   - Simpler models are easier to understand
   - Better understanding of the underlying process that generates the data

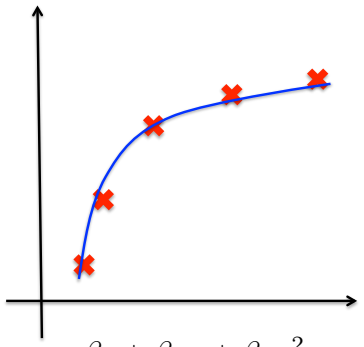
2. **Speed up algorithms training and output calculations**
   - Live / real environments

3. **Avoid overfitting**
   - ML algorithms using many features tends to overfit (increases the complexity of the model)
     - N = # examples
     - D = # dimensions or # features

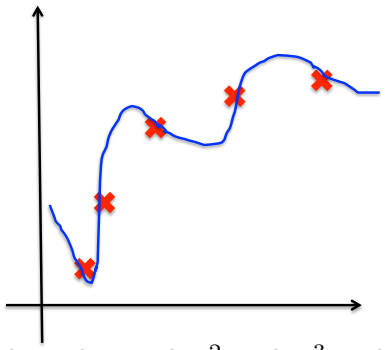
5.2 FS motivation: the problem of overfitting



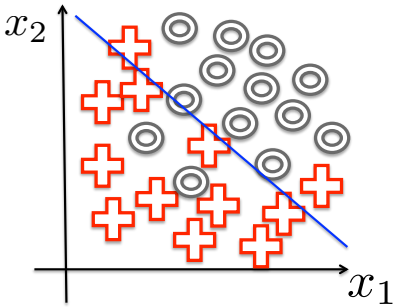
$\beta_0 + \beta_1 x$



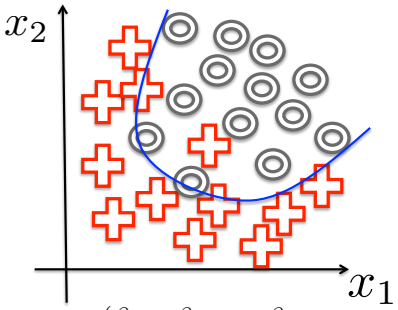
$\beta_0 + \beta_1 x + \beta_2 x^2$



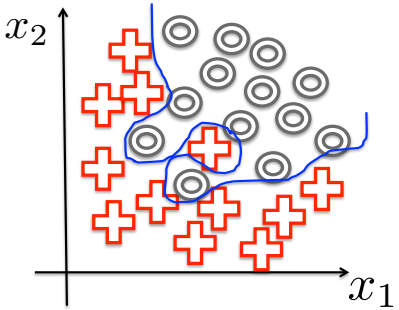
$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4$



$h_\beta(x) = g(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$



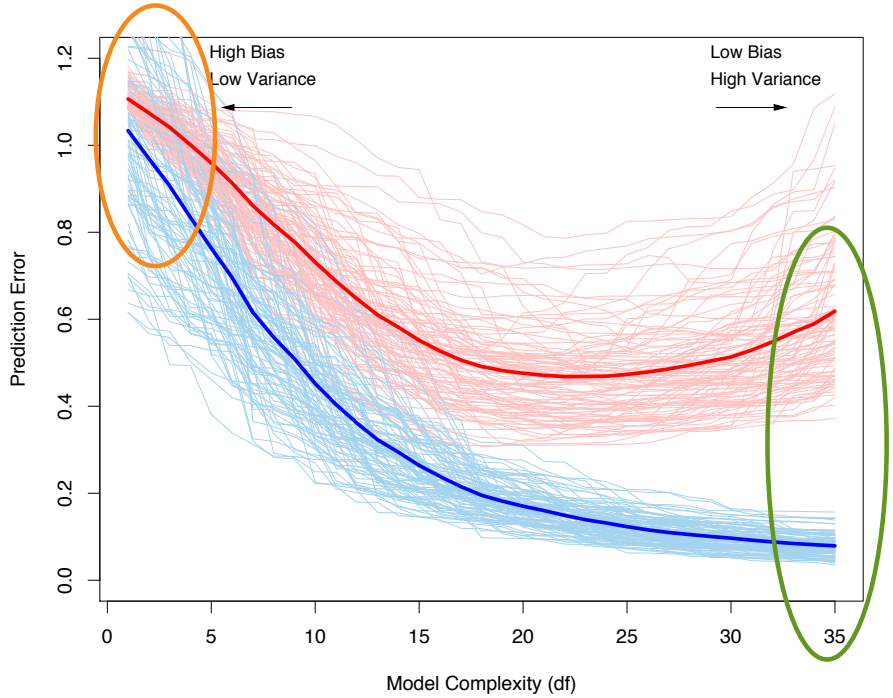
$g(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2)$



$g(\beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^2 x_2 + \beta_4 x_1^2 x_2^2 + \beta_5 x_1^2 x_2^3 + \beta_6 x_1^3 x_2 + \ldots)$

5.2 FS motivation: bias-variance trade off

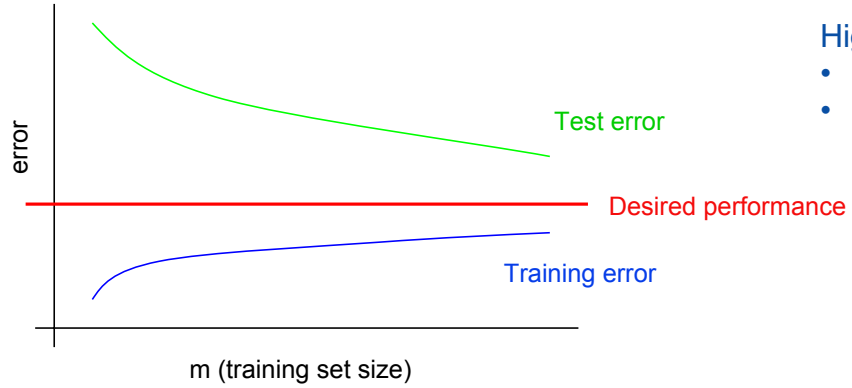
- Feature selection
- Regularization



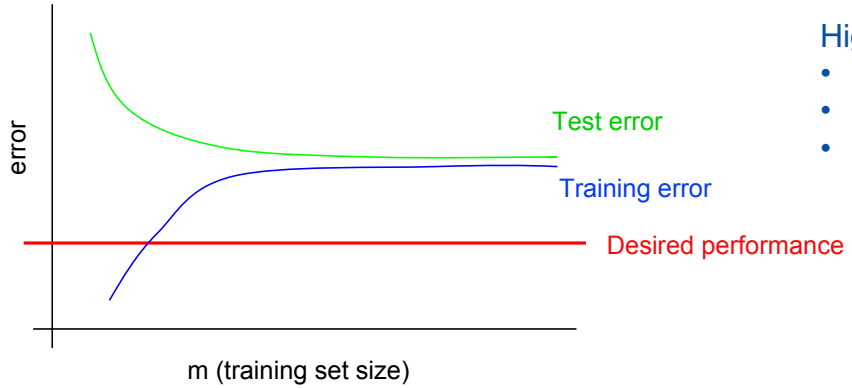
Small regularization parameter
Low dimensionality

Large regularization parameter
High dimensionality

5.2 FS motivation: learning curves



High variance:
- Gap between training and test error
- More data might help



High bias:
- Small gap between training and test error
- More data will not help
- Adding more features

Contents

5.2 Taxonomy: filter methods

- Evaluate the **relevance** of each variable by individually evaluating if each variable is important to discriminate the target.

- Variables are **ranked according to a predefined relevance score**, so that low-scored variables are removed.

- Selected variables constitute the input space of the ML algorithm.

- PROS:
  - Easy and fast

- CONS:
  - They do not consider the existence of interactions among features

5.2 Taxonomy: filter methods

REGRESSION

- Anova (correlation)
- Mutual information

CLASSIFICATION

- Anova (mean for class labels are different)
- Mutual information
- Chi^2 (categorical variables)

In scikit-learn they are called Univariate feature selection.

- For regression: `f_regression` , `mutual_info_regression`
- For classification: `chi2` , `f_classif` , `mutual_info_classif`

The methods based on F-test estimate the degree of linear dependency between two random variables. On the other hand, mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation.

5.2 Taxonomy: wrapper methods

- **Utilize a ML algorithm of interest as a black box to score subsets of variable according to their predictive power**

- **They require:**
  1. to define a classification algorithm
  2. A relevance criterion
  3. A searching procedure in the space of all possible subsets of features (usually heuristic)

- **Searching procedures:**
  - **Brute force**
  - Randomized: genetic algorithms, simulating annealing.
  - Greedy strategies: **backward**, **forward selection**

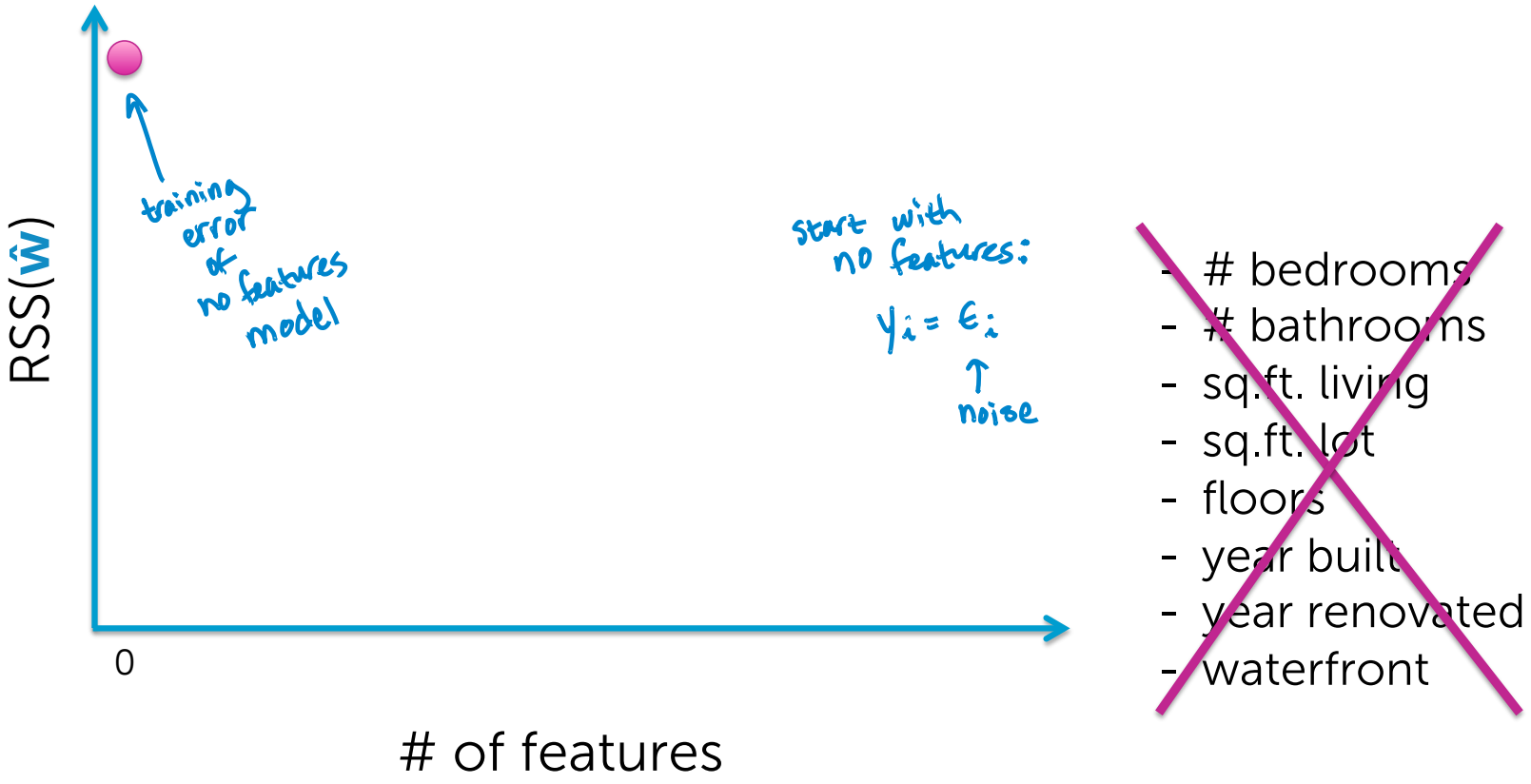
5.2 Taxonomy: wrapper methods – brute force

Lot size
Single Family
Year built
Last sold price
Last sale price/sqft
Finished sqft
Unfinished sqft
Finished basement sqft
floors
Flooring types
Parking type
Parking amount
Cooling
Heating
Exterior materials
Roof type
Structure style

Dishwasher
Garbage disposal
Microwave
Range / Oven
Refrigerator
Washer
Dryer
Laundry location
Heating type
Jetted Tub
Deck
Fenced Yard
Lawn
Garden
Sprinkler System
⋮

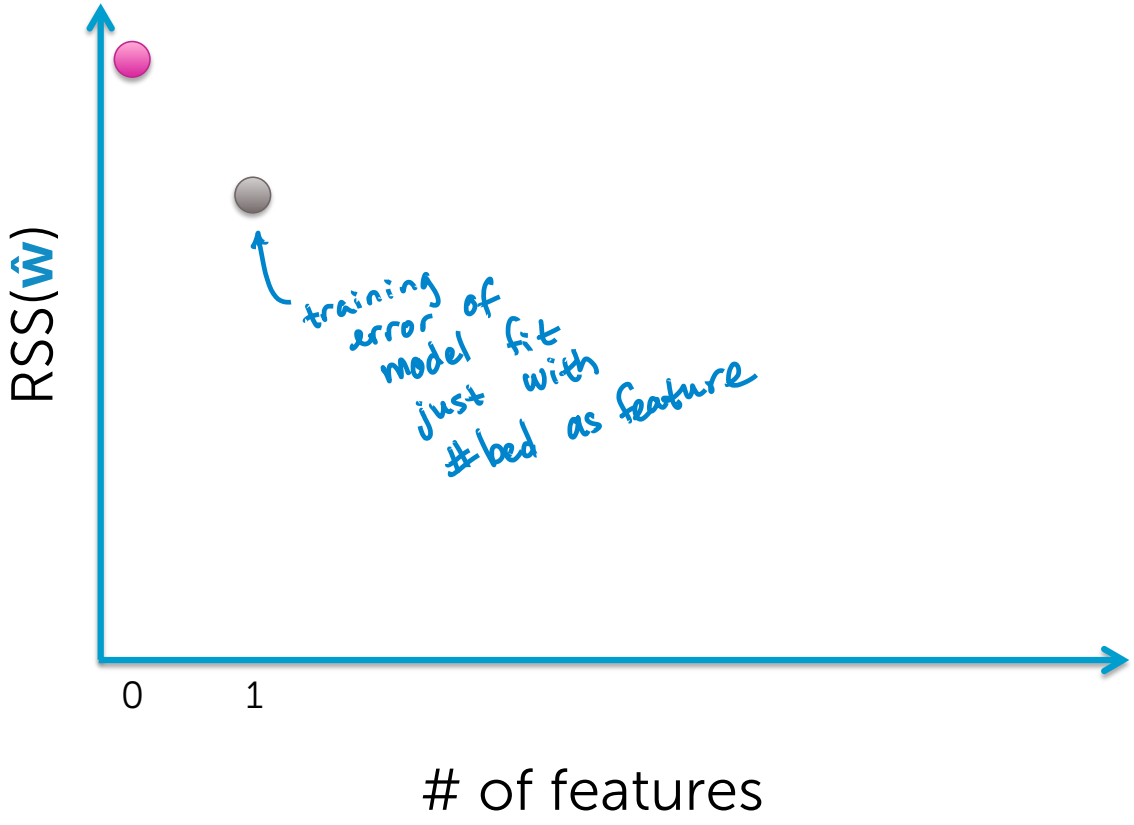
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – brute force



Source: Machine learning course. Emily Fox and Carlos Guestrin

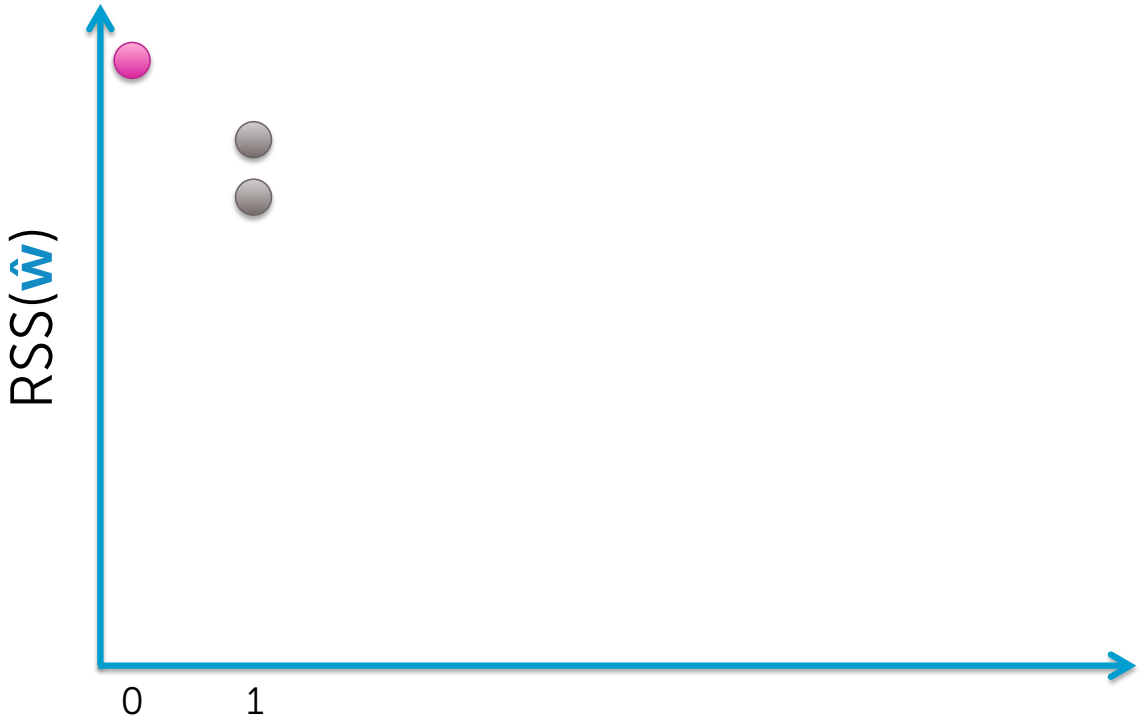
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

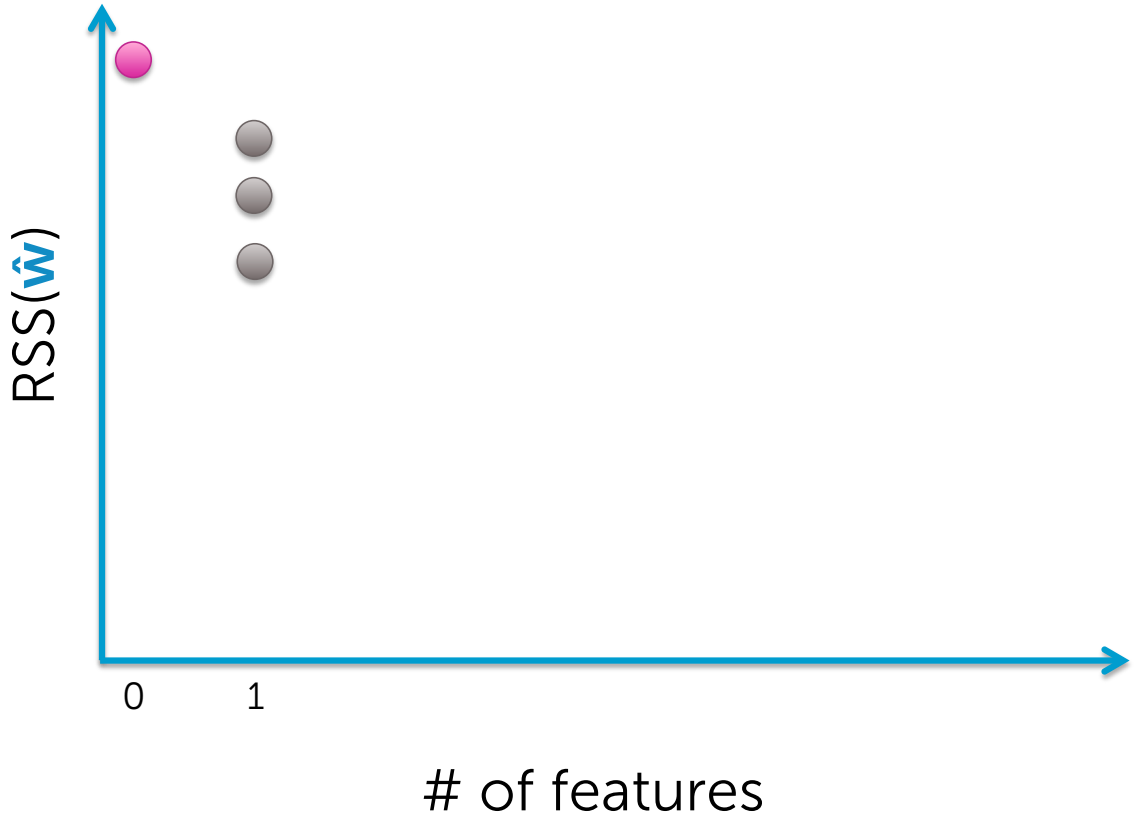
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

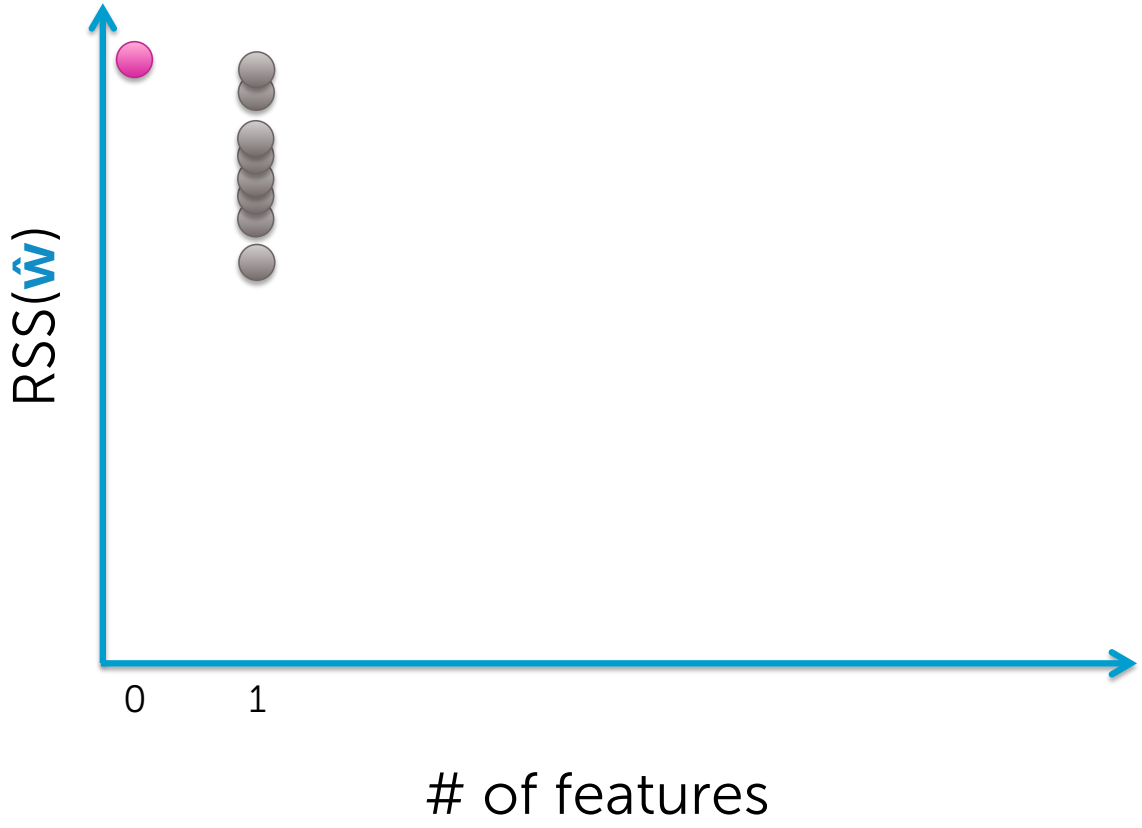
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

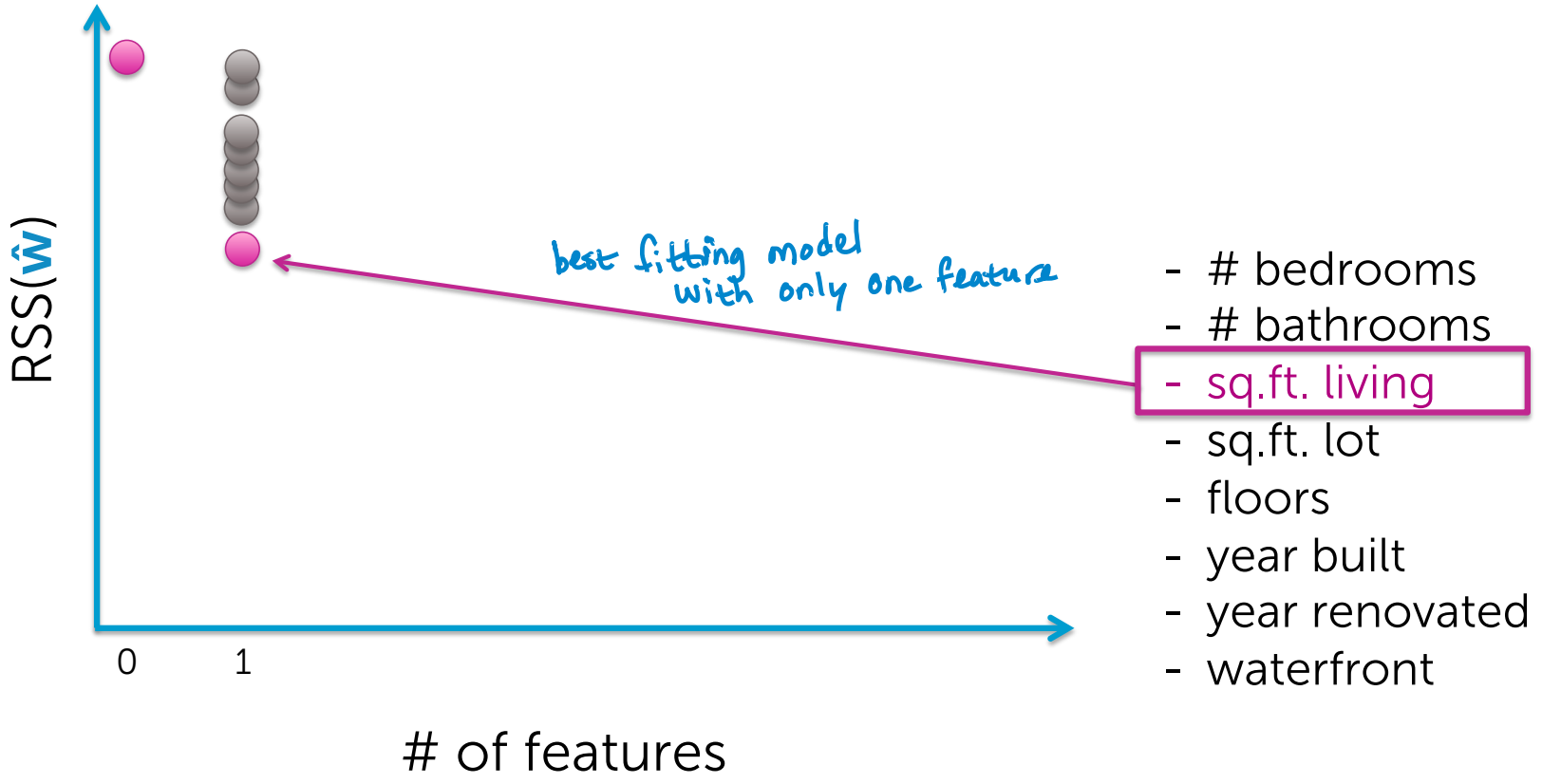
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

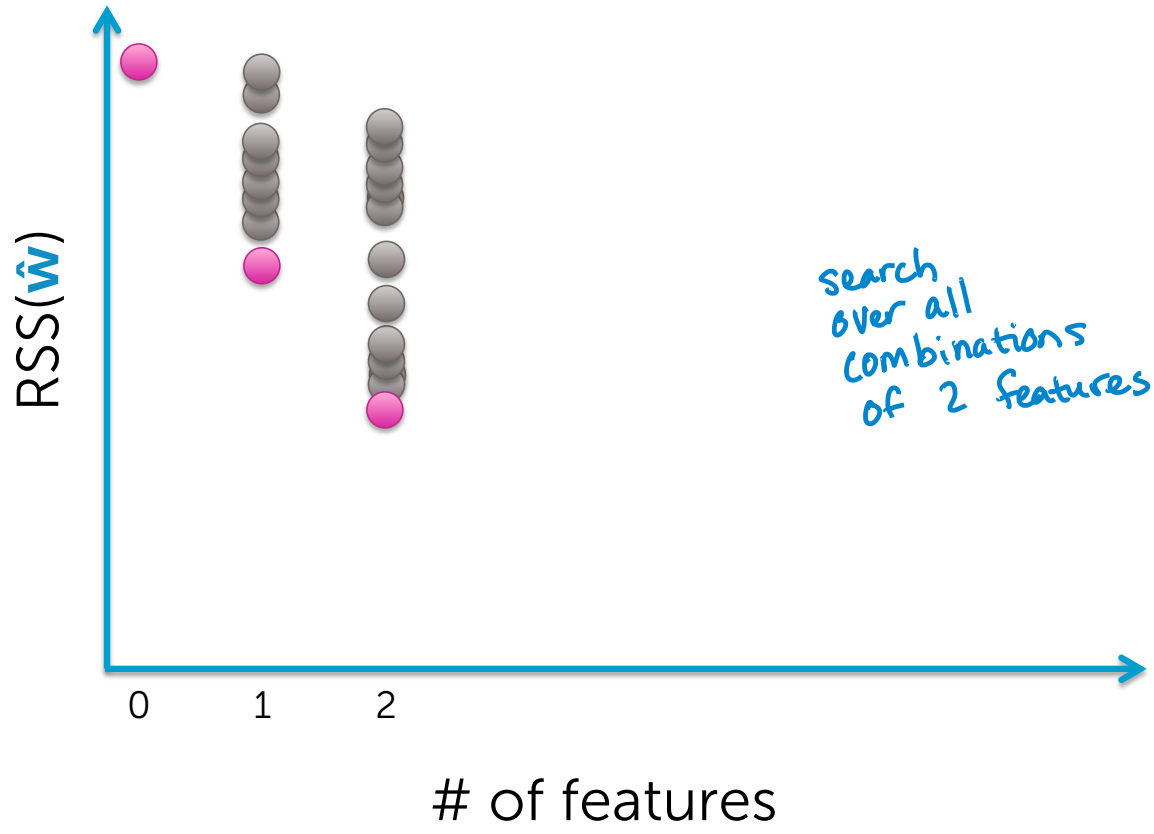
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – brute force



Source: Machine learning course. Emily Fox and Carlos Guestrin

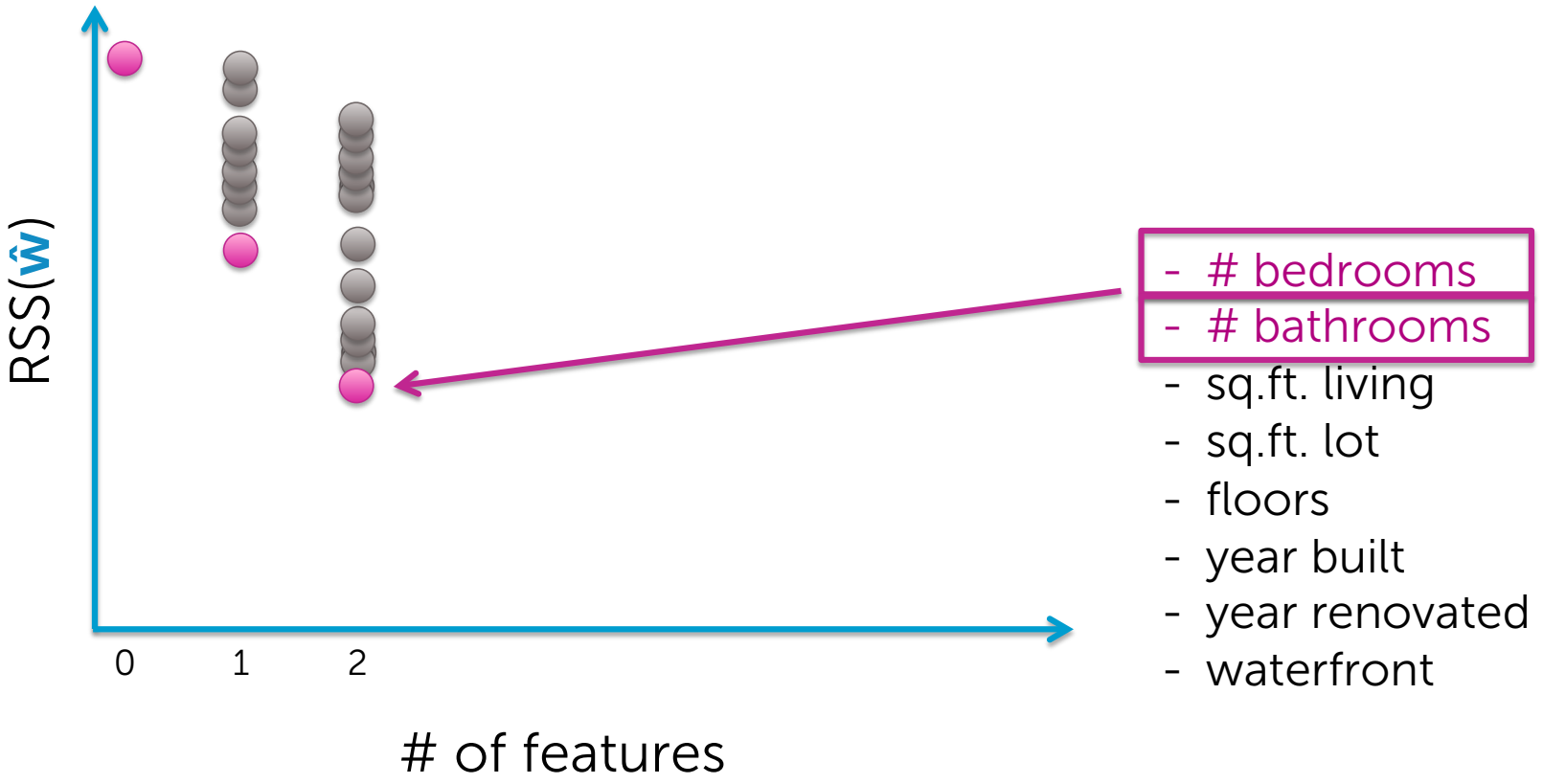
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

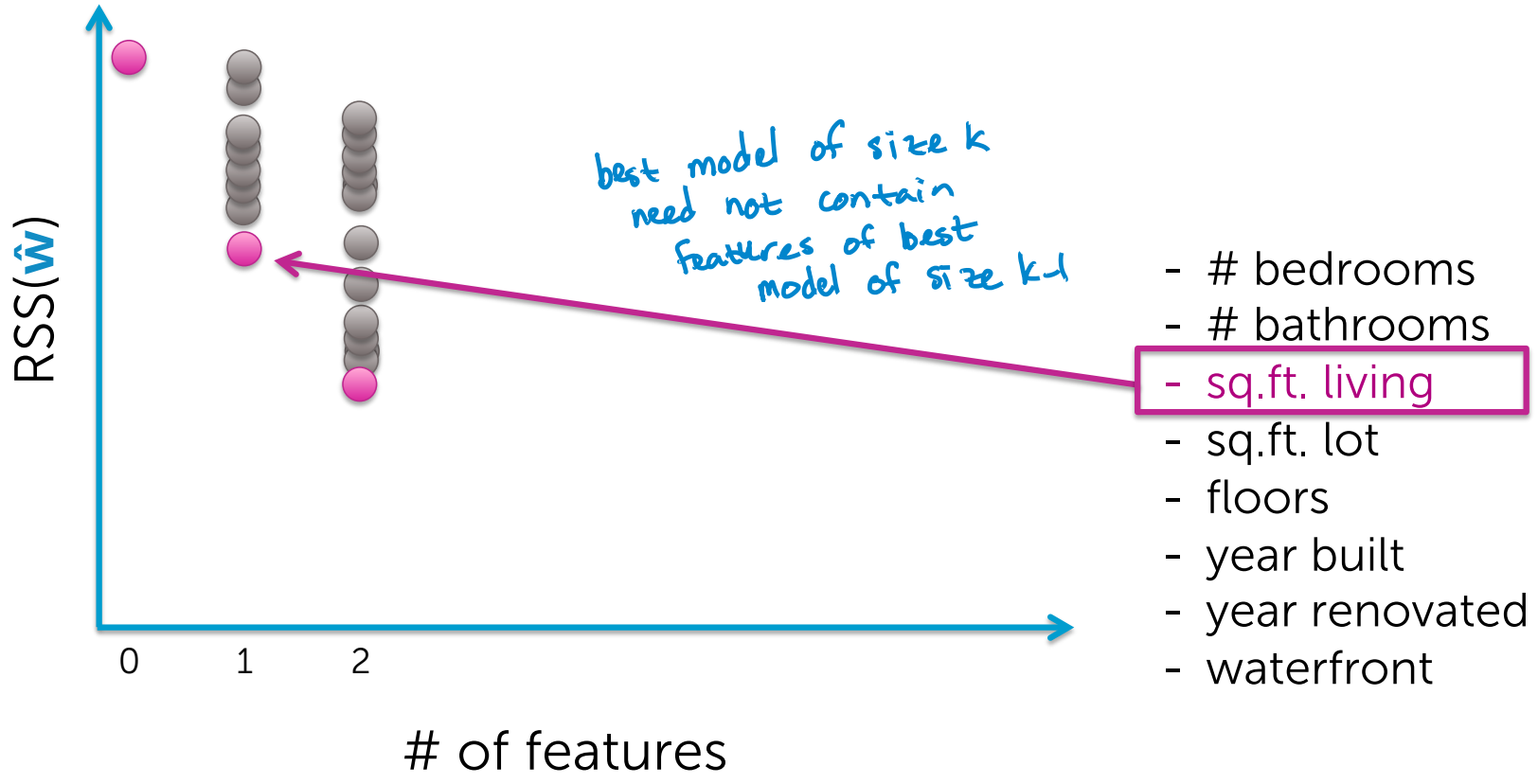
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – brute force



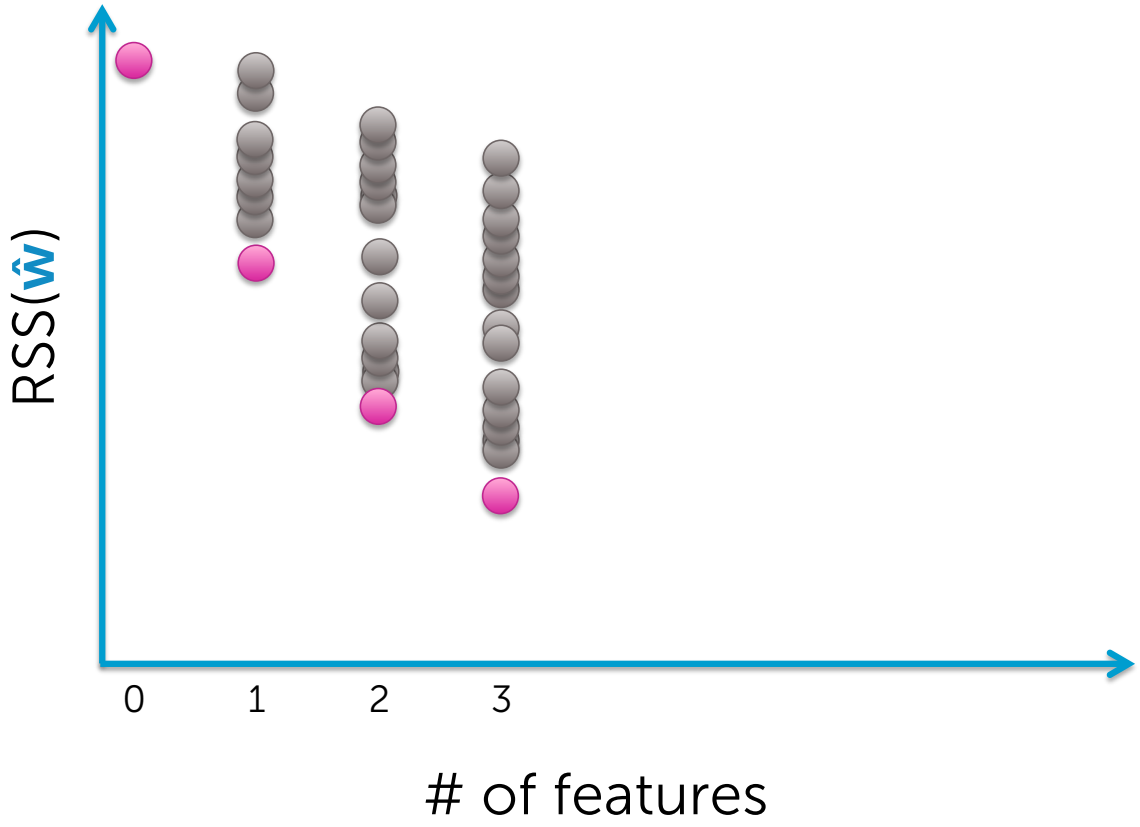
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – brute force



Source: Machine learning course. Emily Fox and Carlos Guestrin

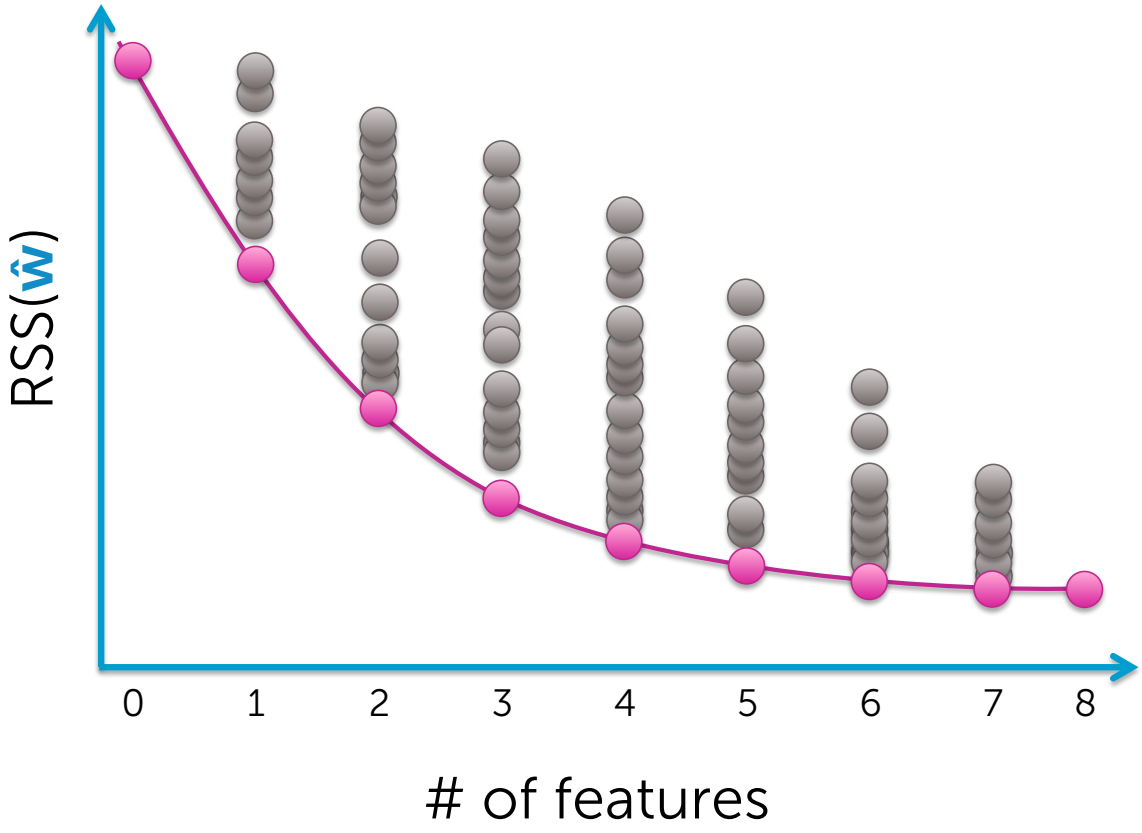
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

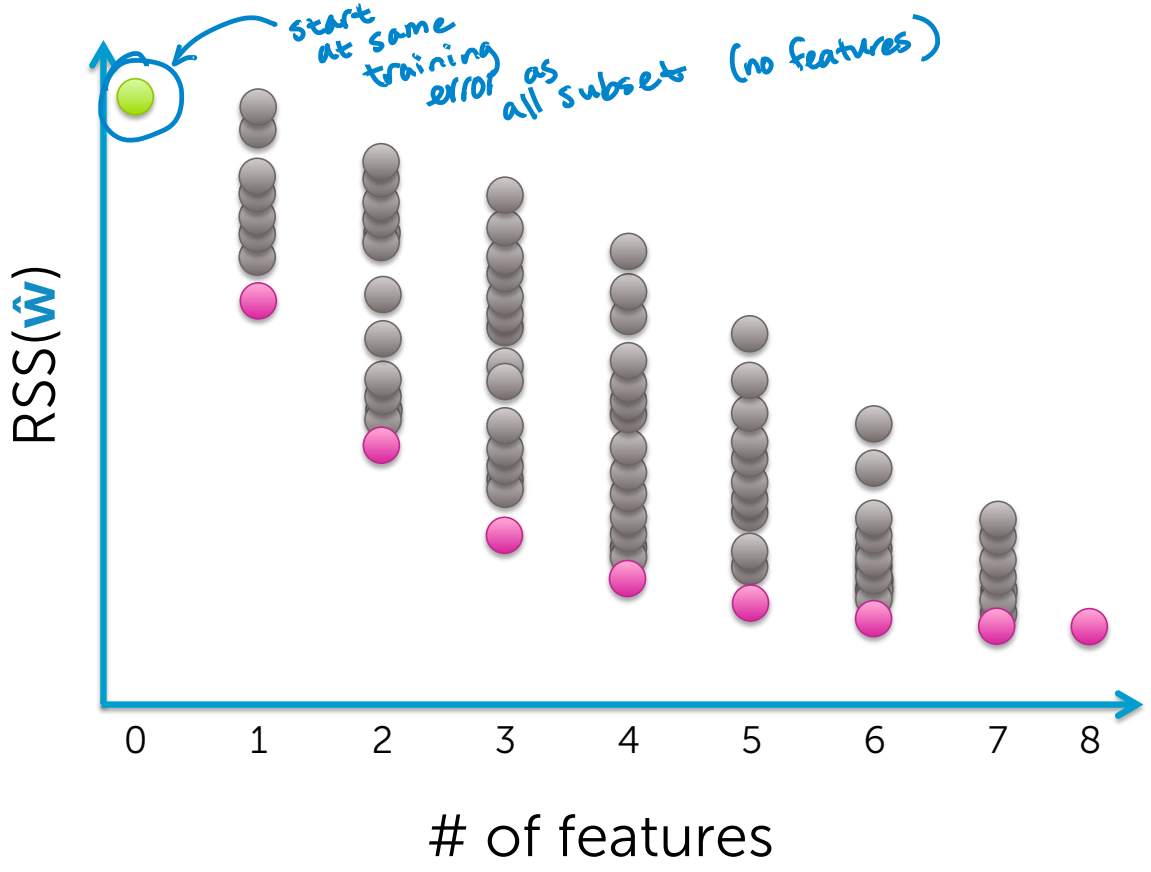
5.2 Taxonomy: wrapper methods – brute force



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

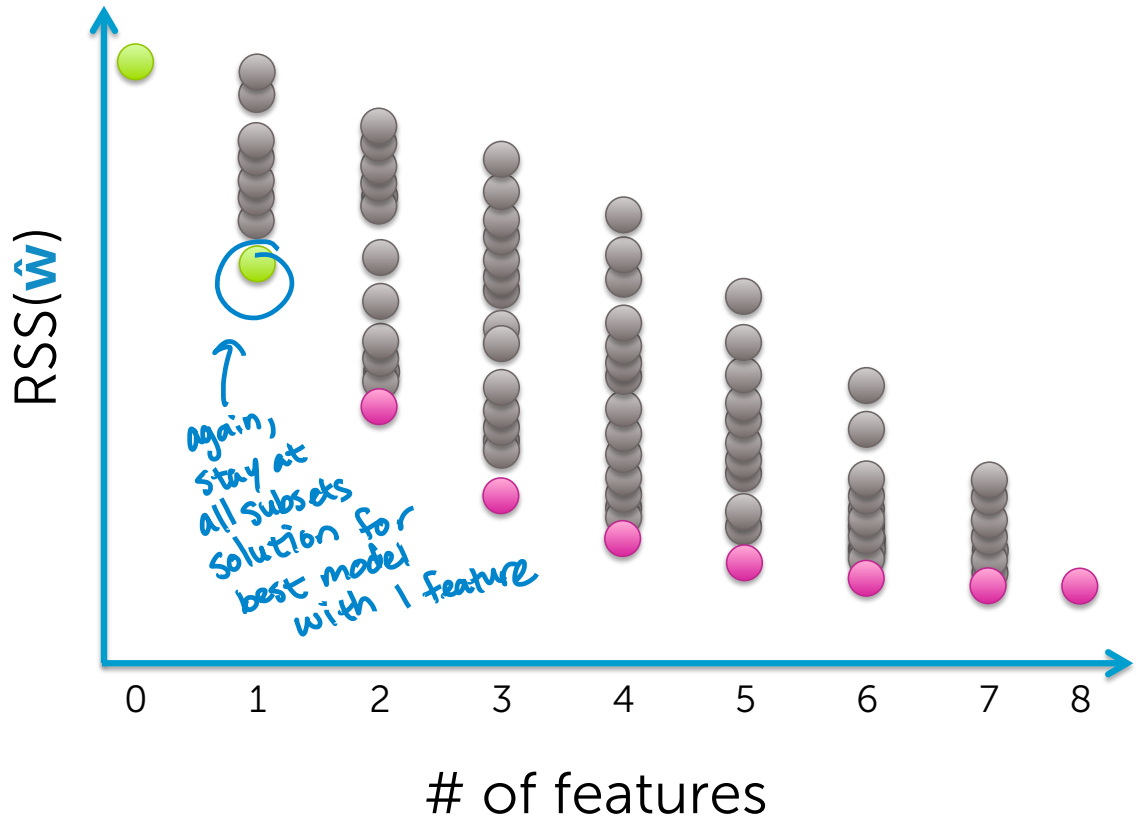
5.2 Taxonomy: wrapper methods – forward



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

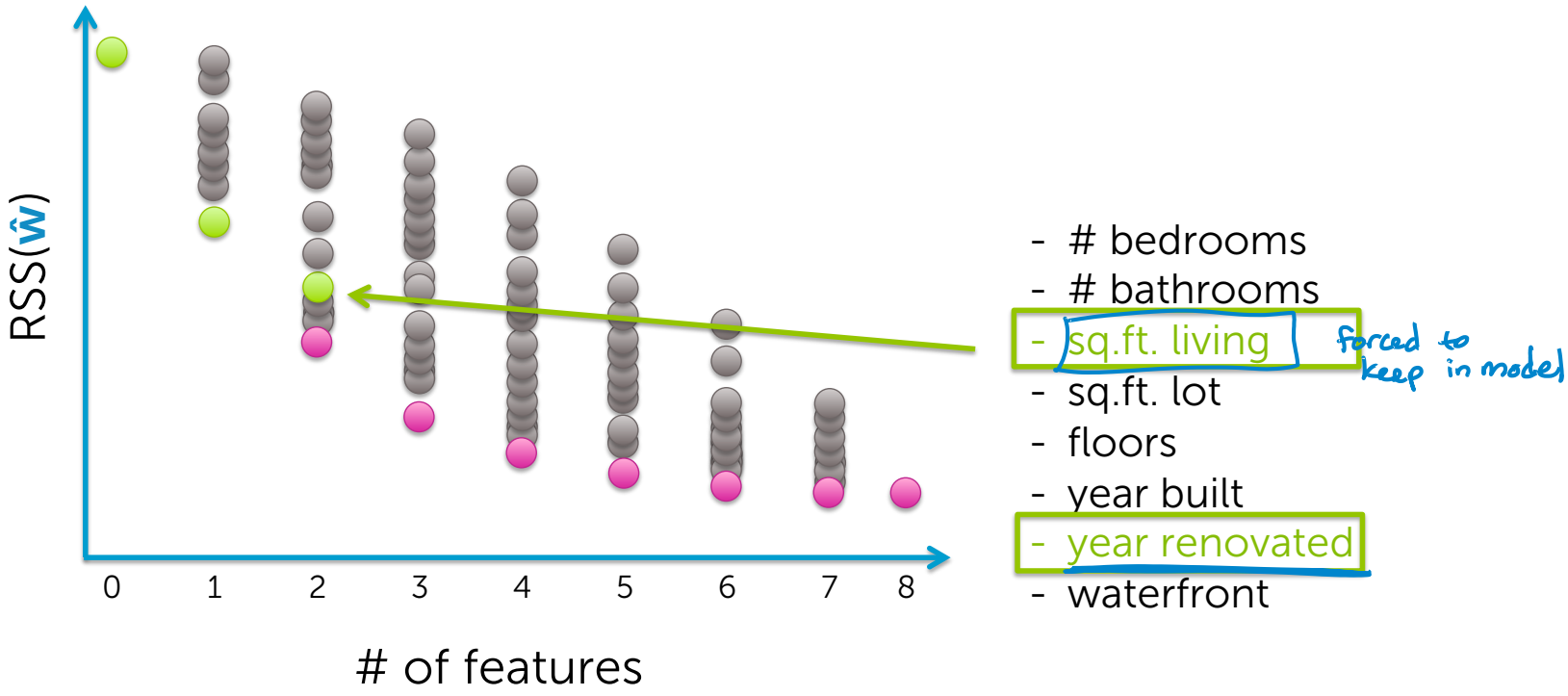
5.2 Taxonomy: wrapper methods – forward



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

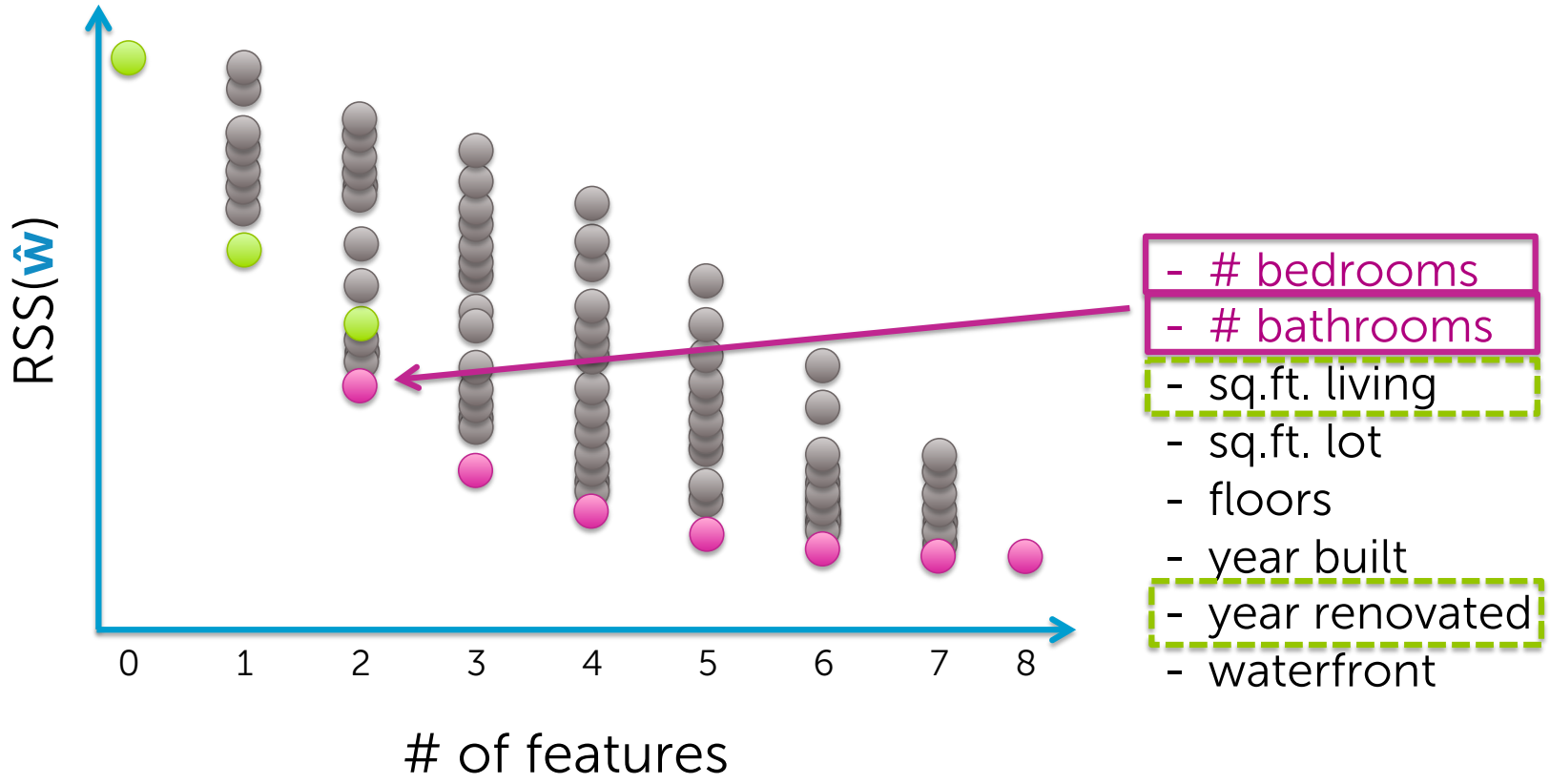
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – forward



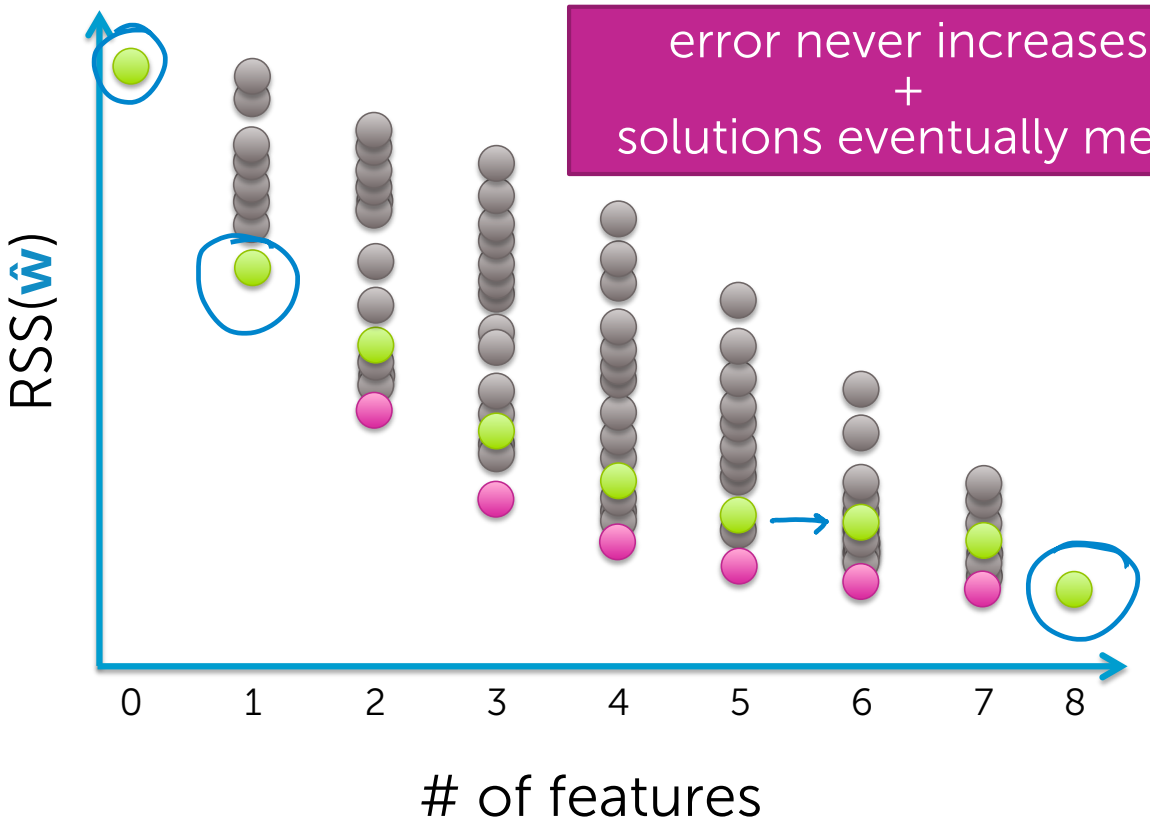
Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – forward



Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods – forward



- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront

Source: Machine learning course. Emily Fox and Carlos Guestrin

5.2 Taxonomy: wrapper methods

- **Brute Force complexity: $O(2^D)$**
  - D = 20 $\Rightarrow$ $2^{20} \approx$ 1e6 possibilities
- **Forward selection complexity: $O(D^2)$**
- **How to choose the model?**
  - CV techniques, do it well.
- **PROS:**
  - ML algorithm as a black box, universal and simple solution.
- **CONS:**
  - The selected features depend on the ML that was used
  - For each subset evaluation, a new model must be created i.e. the algorithm needs to be trained and tested for each subset to obtain its performance.

5.2 Taxonomy: embedded methods

1. **Incorporate the feature selection as part of the training process**
   - These methods rely on the internal design of the learning algorithm for intrinsically selecting features

2. **Together with backward and forward selection techniques, nested/hybrid methods constitute very efficient schemes for FS:**
   - Recursive Feature Elimination (RFE): the search procedure is guided by estimating changes in the objective function (e.g., classifier performance) for different subsets of features

3. **Examples:**
   - Decision/regression trees (measure of importance)
   - L1-regularized algorithms (LASSO, L1-SVM, L1-logistic regression)

   - SCALE YOUR DATA!!!

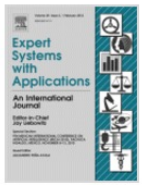
Contents

5.2 Biomedical examples

ELSEVIER

Expert Systems with Applications

Volume 39, Issue 2, 1 February 2012, Pages 1956–1967



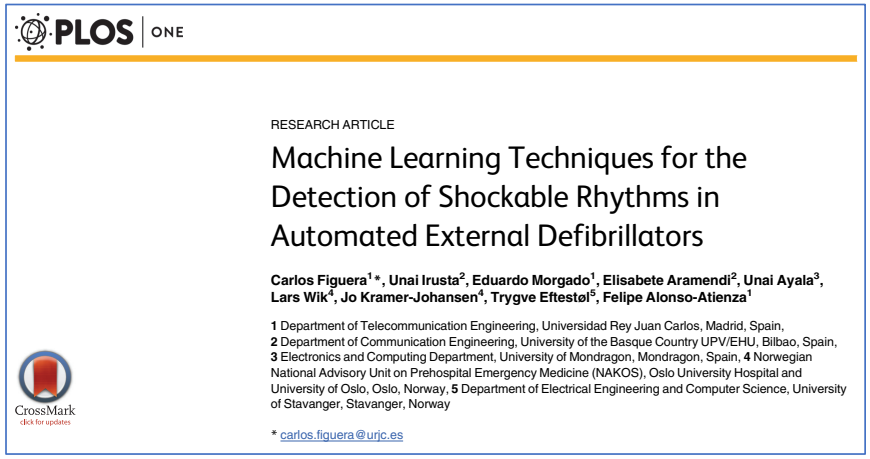
Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection

Felipe Alonso-Atienza[a], José Luis Rojo-Álvarez[a], Alfredo Rosado-Muñoz[b], Juan J. Vinagre[a], Arcadi García-Alberola[c], Gustavo Camps-Valls[b]

Detection of Life-Threatening Arrhythmias Using Feature Selection and Support Vector Machines

Felipe Alonso-Atienza*, Eduardo Morgado, Lorena Fernández-Martínez, Arcadi García-Alberola, and José Luis Rojo-Álvarez, *Senior Member, IEEE*

5.2 Biomedical examples



PLOS ONE

RESEARCH ARTICLE

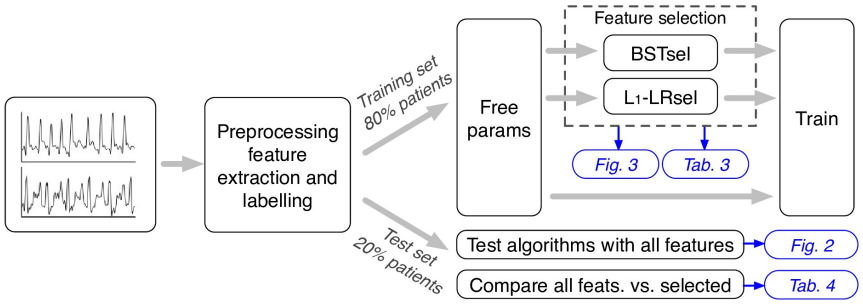
Machine Learning Techniques for the Detection of Shockable Rhythms in Automated External Defibrillators

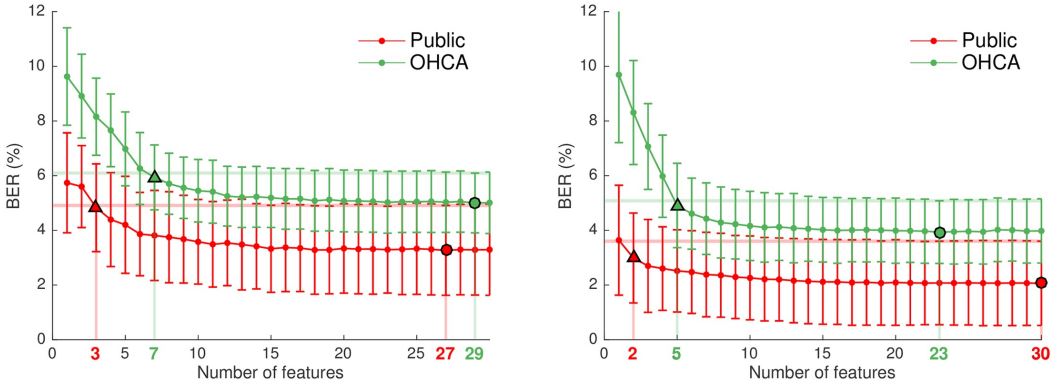
Carlos Figuera[1]*, Unai Irusta[2], Eduardo Morgado[1], Elisabete Aramendi[2], Unai Ayala[3], Lars Wik[4], Jo Kramer-Johansen[4], Trygve Eftestøl[5], Felipe Alonso-Atienza[1]

1 Department of Telecommunication Engineering, Universidad Rey Juan Carlos, Madrid, Spain, **2** Department of Communication Engineering, University of the Basque Country UPV/EHU, Bilbao, Spain, **3** Electronics and Computing Department, University of Mondragon, Mondragon, Spain, **4** Norwegian National Advisory Unit on Prehospital Emergency Medicine (NAKOS), Oslo University Hospital and University of Oslo, Oslo, Norway, **5** Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway

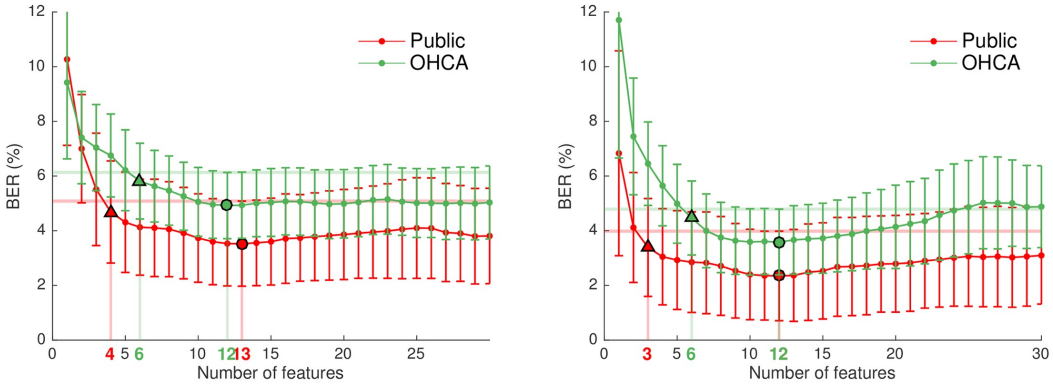
* carlos.figuera@urjc.es

CrossMark click for updates





(a) BSTsel algorithm for 4-s (left) and 8-s (rigth) segments



(b) L_1-LRsel algorithm for 4-s (left) and 8-s (rigth) segments

Contents

5.3 Feature extraction

Differs from feature selection in two ways:

- Create new features, instead of selecting a subset of them
- Do not consider class labels, just the data

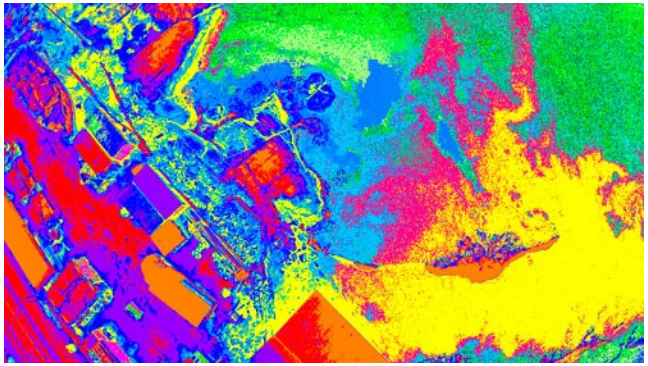
1. **Feature transformation** to support effective learning
   - Log, sqrt transformations
   - Encoding: categorical variables into numerical ones

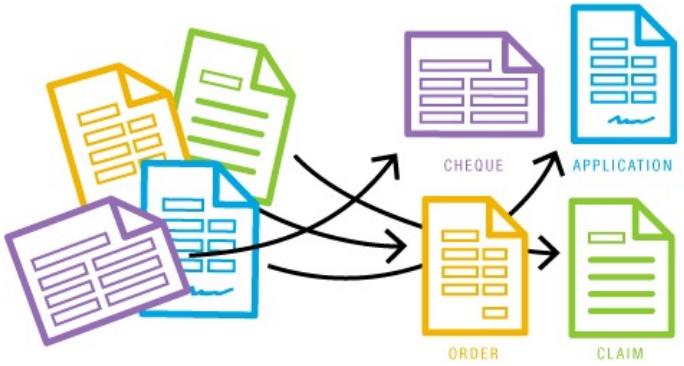
2. **Feature engineering:** create new features based on expertise domain knowledge
   - Applying signal and image processing techniques for ECG, diagnostic images, etc.
   - Interaction terms: `sales = b0 + b1*youtube + b2*facebook + b3*(youtube*facebook)`

3. **Dimensionality reduction:** transform data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data
   - PCA: linear transformation
   - t-sne, U-MAP: manifold learning
   - Autoencoders: Deep Learning configuration
   - Embeddings: a mapping from discrete objects, such as words, to vectors of real numbers (word2vec)

5.3 Dimensionality reduction: motivation

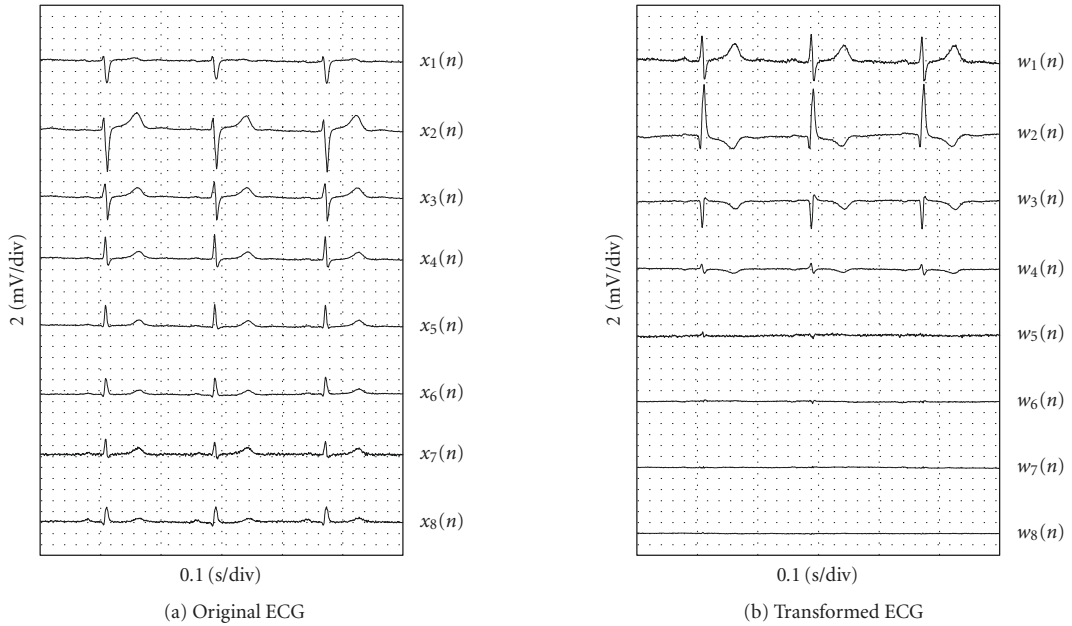
- Many applications use high-dimensional data which might contain **redundant information**, and therefore we can reduce data dimensions to **avoid the curse of dimensionality**





5.3 Dimensionality reduction: motivation

1. **Data compression**
   a. Compress the data
   b. Speed-up ML algorithms

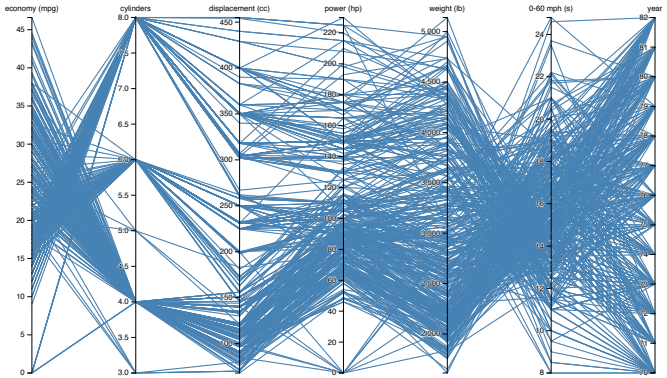


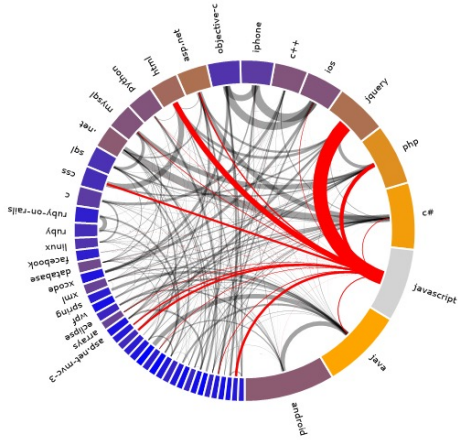
- Wearable tech.
- IoT

[F Castells et al. EURASIP Journal on Advances in Signal Processing, 2007]

5.3 Dimensionality reduction: motivation

2. **Visualization**
   - To better understand our data
   - How do we visualize 50-dimensional data?



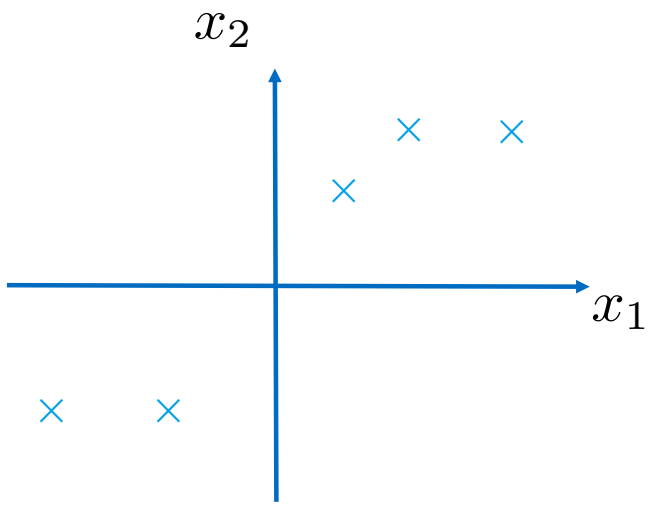


5.3 Dimensionality reduction: idea

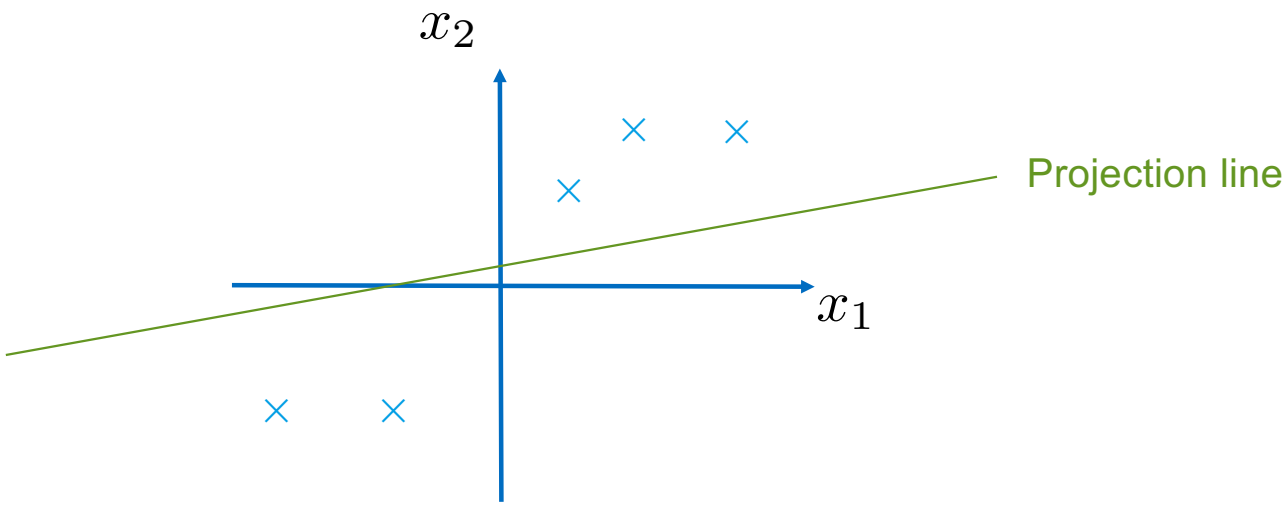
- **Idea:**
  - Given data points in d-dimensional space
  - **Project** them into a lower k-dimensional space while preserving as much information as possible

    - Can you describe yourself using 2 words?

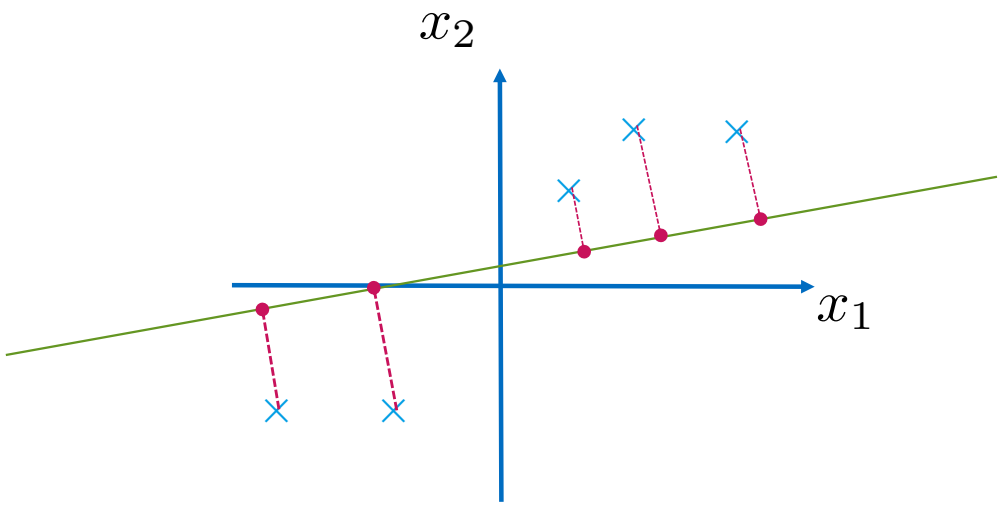
5.3 Dimensionality reduction: project your data



5.3 Dimensionality reduction: project your data



5.3 Dimensionality reduction: project your data



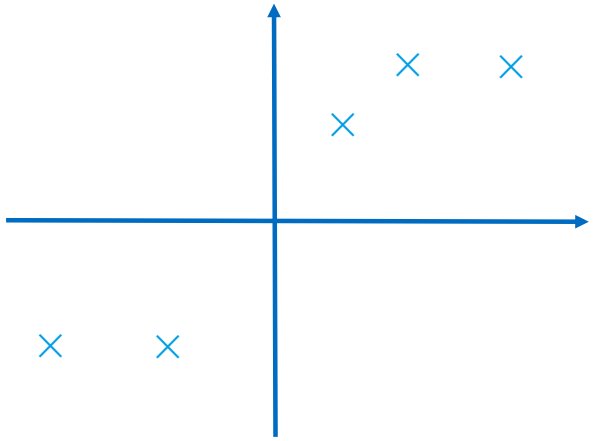
Projection error

5.3 Dimensionality reduction: project your data

Contents

5.3 Principal Component Analysis

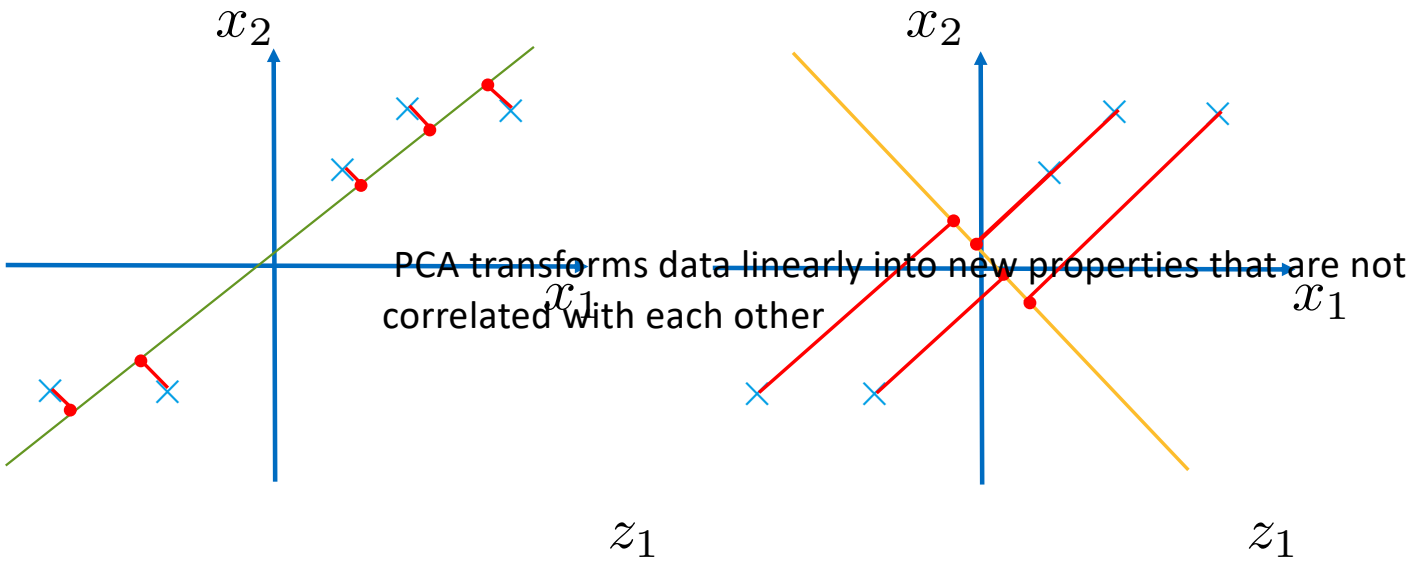
Suppose you want to reduce/project the following data



Which projection line would you choose?

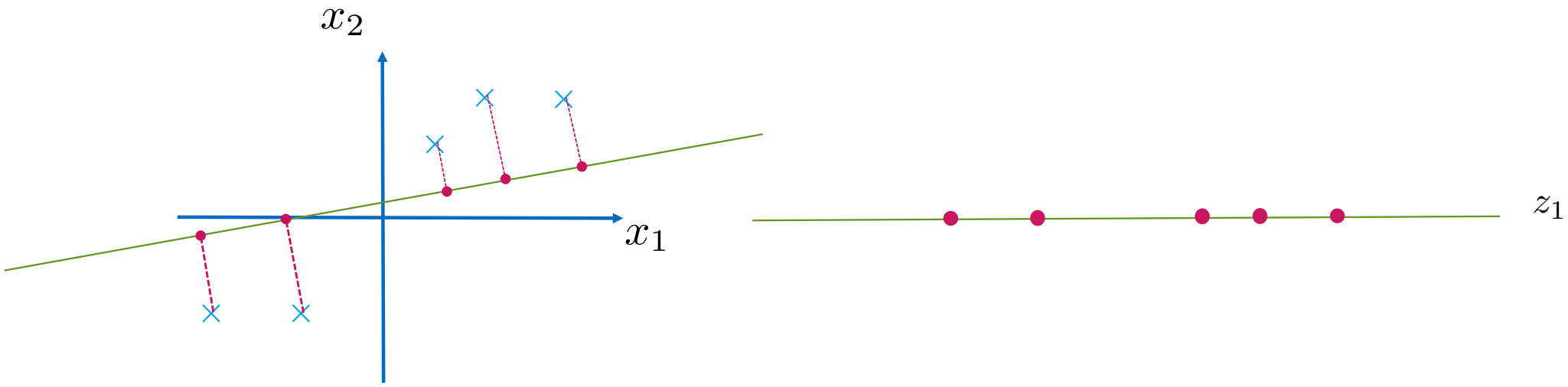
- **PCA chooses the line (direction) that minimizes the projection error**
- **Or equivalently, the line that maximizes the variance of the projected data**

5.3 PCA: projection error



PCA transforms data linearly into new properties that are not correlated with each other

5.3 PCA: orthonormal transformation



PCA linearly transforms data into new properties that are **not correlated** with each other

5.3 PCA: formulation

1. Preprocess your data (mean normalization and feature scaling)

$$\mathbf{x}_j = \frac{\mathbf{x}_j - \bar{\mathbf{x}}_j}{\sigma_j} \Rightarrow \mathbf{X}$$

2. Compute the covariance matrix

$$\Sigma = \frac{1}{N}(\mathbf{X}^T \cdot \mathbf{X})$$

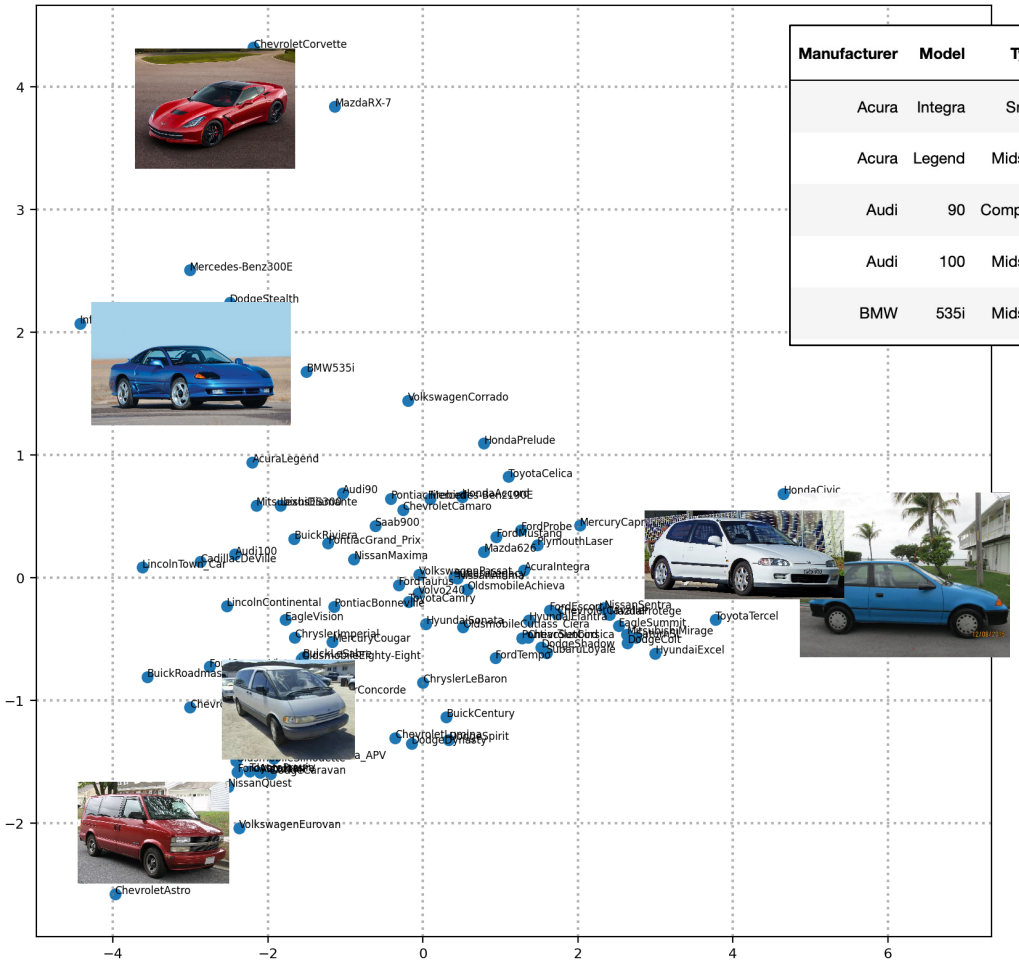
3. Compute the eigenvectors of the covariance matrix

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathrm{svd}(\Sigma)$$

4. Principal components are the k eigenvectors with largest eigenvalues

$$\mathbf{z} = \mathbf{X} \cdot \mathbf{U}[:, 0:k]$$

5.3 PCA example: data table to 2D

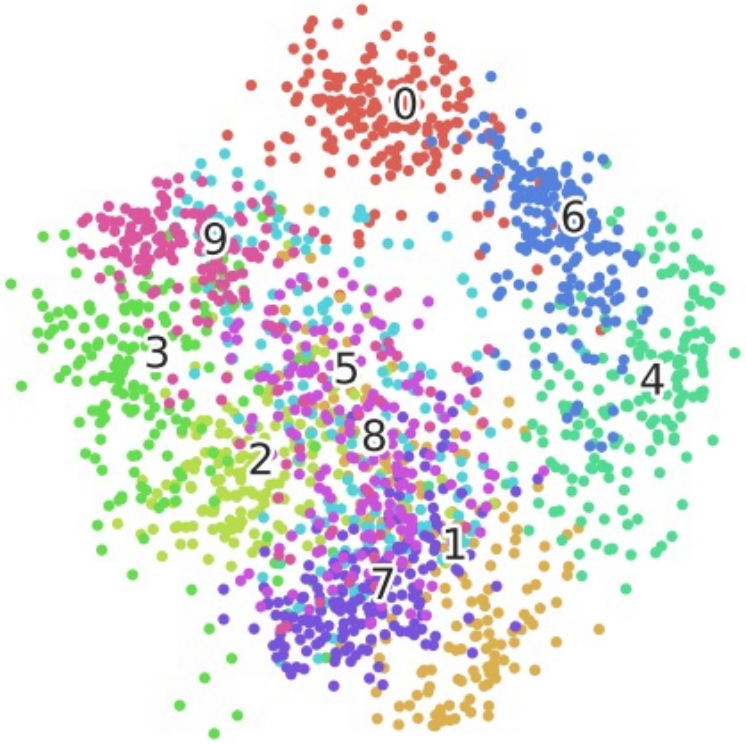


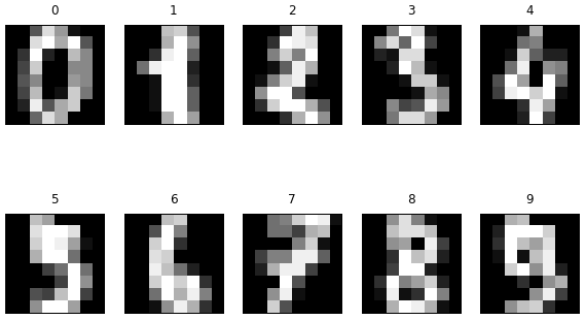
Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	...	Passengers	Length	Wheelbase	Width	Turn.circle	Rear
Acura	Integra	Small	12.9	15.9	18.8	25	31	None	...	5	177	102	68	37	
Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	...	5	195	115	71	38	
Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	...	5	180	102	67	37	
Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver & Passenger	...	6	193	106	70	37	
BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	...	4	186	109	69	39	

Any car can be represented in a 2D space
- 2D orthogonal space
- Axis with no-meaning

Cars93 dataset: https://www.tandfonline.com/doi/full/10.1080/10691898.1993.11910459

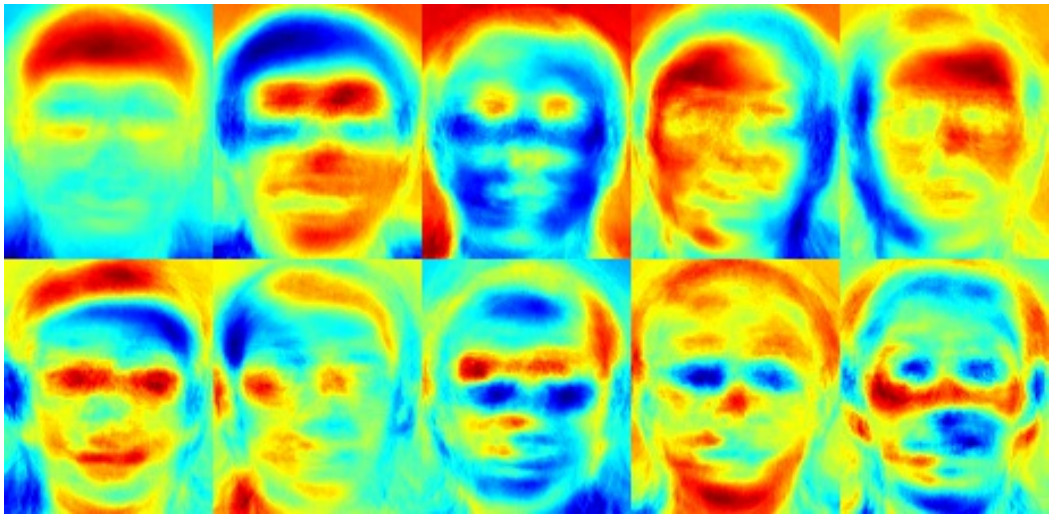
5.3 PCA example: visualization from 64D to 2D





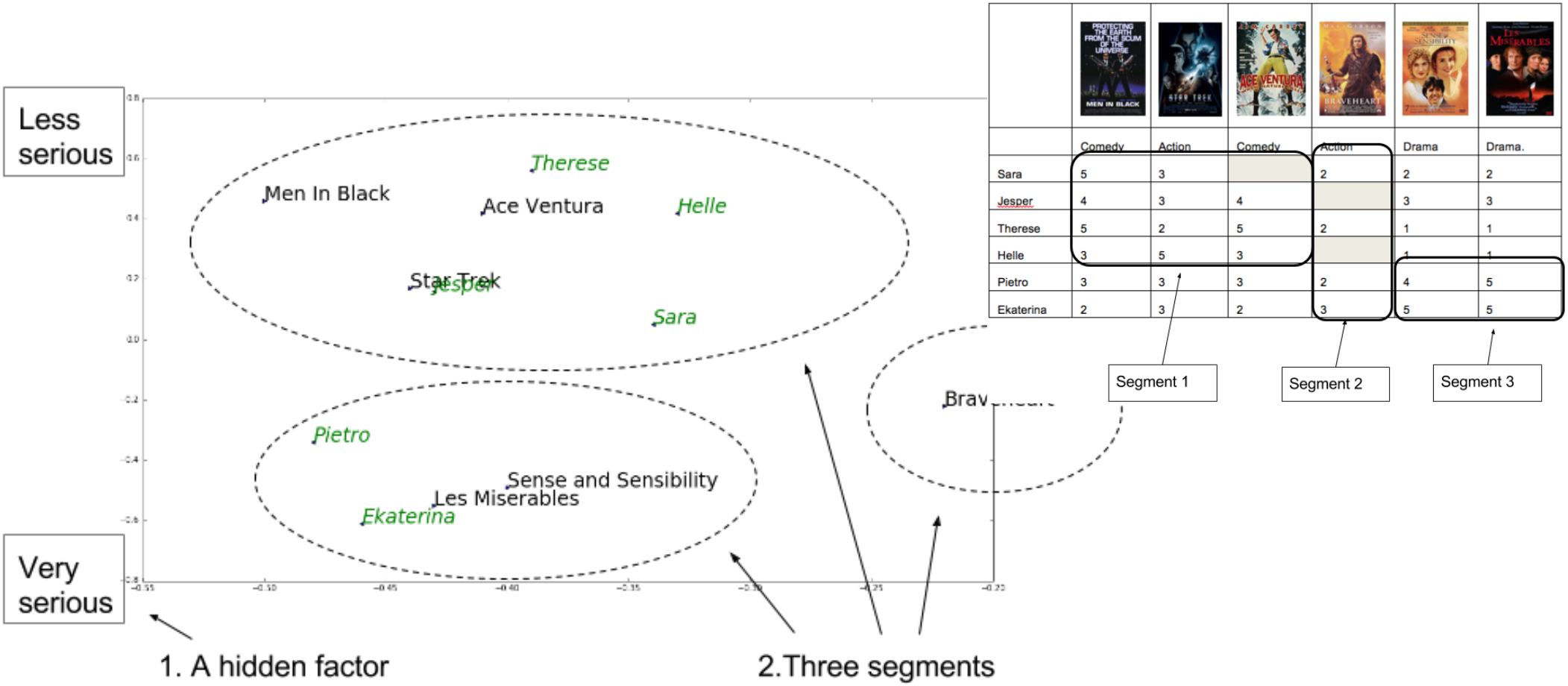
5.3 PCA example: singular value decomposition

- **Projects the data into orthogonal directions (U)**



eigenfaces: any face can be expressed as a linear combination of eigenfaces

5.3 SVD example



	Comedy	Action	Comedy	Action	Drama	Drama.
Sara	5	3		2	2	2
Jesper	4	3	4		3	3
Therese	5	2	5	2	1	1
Helle	3	5	3		1	1
Pietro	3	3	3	2	4	5
Ekaterina	2	3	2	3	5	5

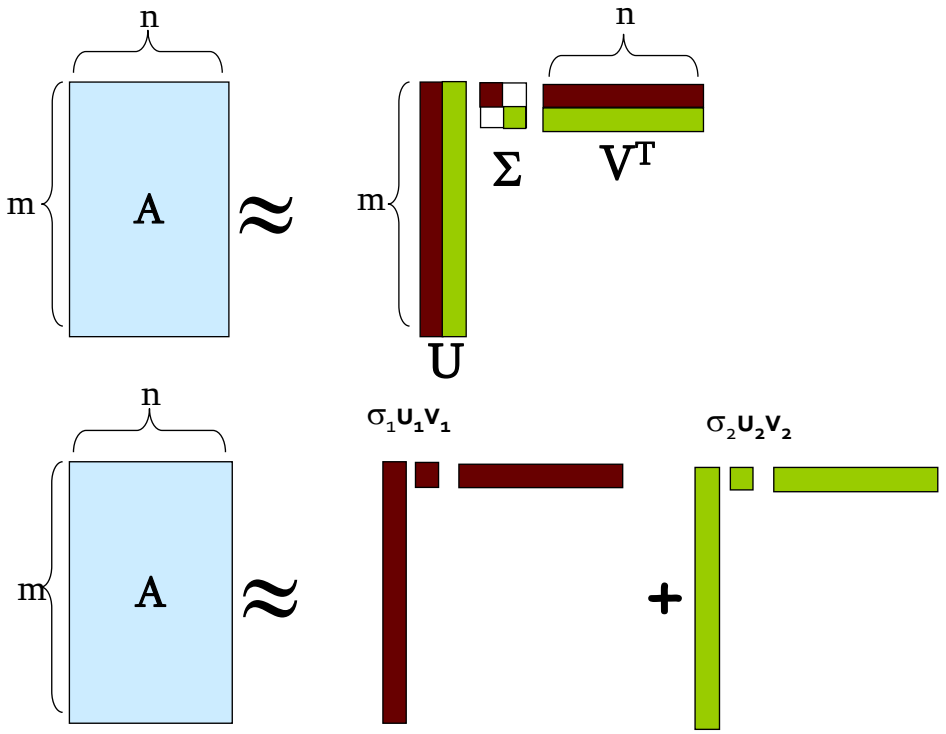
Segment 1

Segment 2

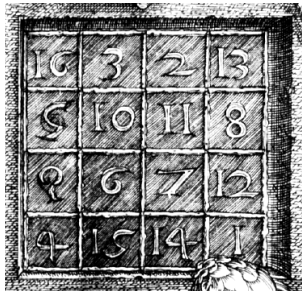
Segment 3

5.3 PCA: singular value decomposition

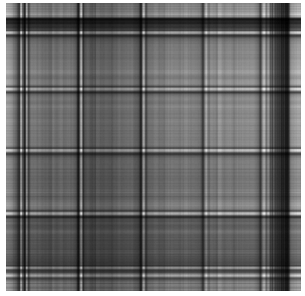
$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



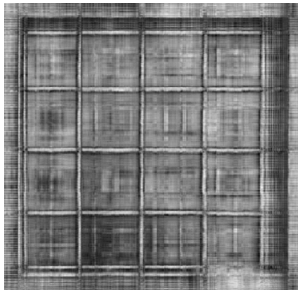
5.3 PCA: reconstruction



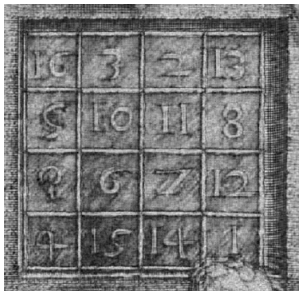
Original
359 x 371



k = 1



k = 10

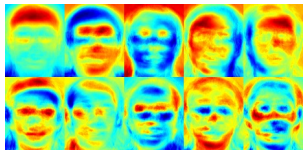


k = 40

$$\mathbf{z} = \mathbf{X} \cdot \mathbf{U}[:, 0:k]$$

$$\hat{\mathbf{X}} = \mathbf{z} \cdot \mathbf{U}[:, 1:k]^T$$

5.3 PCA: reconstruction



5.3 Advice for applying PCA

1. **PCA should be fit on the training set, and transform the validation/test set accordingly**

2. **For data compression, use k to keep 99% of the variance**

3. **For data visualization, use k=2 or k=3**

4. **Not use PCA to prevent overfitting, use regularization instead**
   - PCA does not use the information contained in the target variable

5. **Check the performance of your ML algorithm with the original data before transforming them using PCA**

5.3 PCA: conclusions

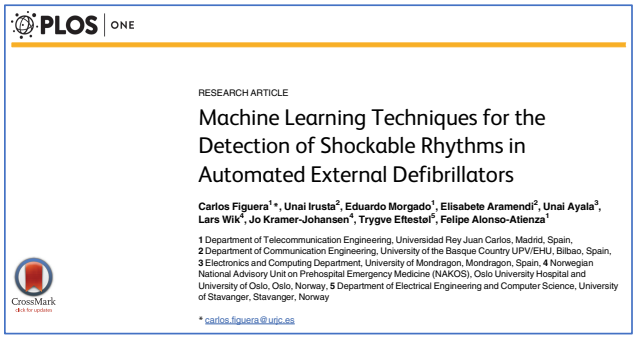
- **Most popular dimensionality reduction technique**
- **Linear transformation that projects the data into orthogonal directions**
  - ICA: statistically independent projections
  - LDA: best separating projections (supervised)
  - PLS: reduce/decompose both X (predictors) and y (target) to explain correlation between X and y.

- **Careful: preprocess your data**

- **Pros:**
  - Convex problem: no local minima (exact)
  - Non-iterative
- **Cons:**
  - Restricted to linear variability in high-dimensional data
    - Kernel PCA

Contents

5.3 Biomedical applications



PLOS ONE

RESEARCH ARTICLE

Machine Learning Techniques for the Detection of Shockable Rhythms in Automated External Defibrillators

Carlos Figuera[1]*, Unai Irusta[2], Eduardo Morgado[1], Elisabete Aramendi[2], Unai Ayala[3], Lars Wik[4], Jo Kramer-Johansen[4], Trygve Eftestøl[5], Felipe Alonso-Atienza[1]

1 Department of Telecommunication Engineering, Universidad Rey Juan Carlos, Madrid, Spain, 2 Department of Communication Engineering, University of the Basque Country UPV/EHU, Bilbao, Spain, 3 Electronics and Computing Department, University of Mondragon, Mondragon, Spain, 4 Norwegian National Advisory Unit on Prehospital Emergency Medicine (NAKOS), Oslo University Hospital and University of Oslo, Oslo, Norway, 5 Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway

* carlos.figuera@urjc.es

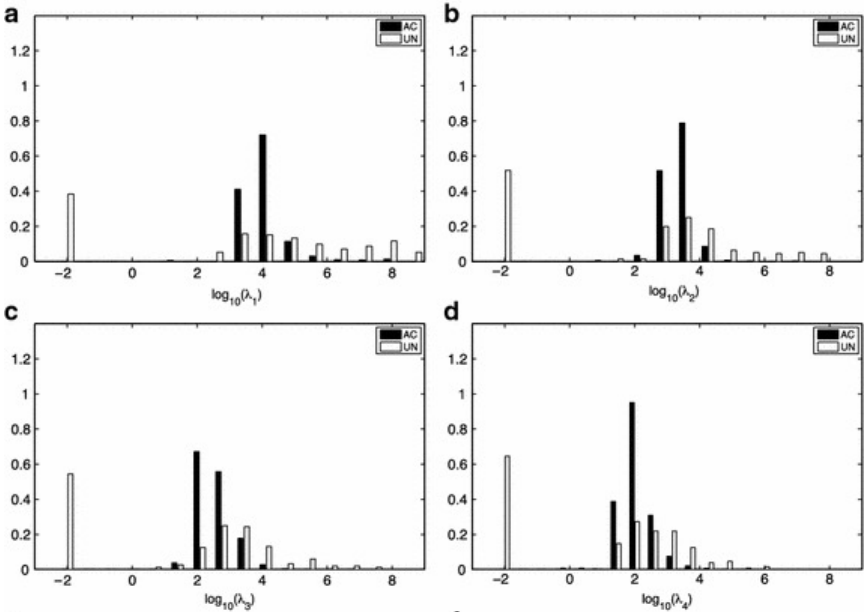
		Public (4-s and 8-s)		OHCA (4-s and 8-s)				Public (4-s and 8-s)		OHCA (4-s and 8-s)	
Feature		**Se/Sp**	**Se/Sp**	**Se/Sp**	**Se/Sp**	**Feature**		**Se/Sp**	**Se/Sp**	**Se/Sp**	**Se/Sp**
bCP	[27]	94.8/97.8	96.0/98.7	95.3/90.0	94.4/91.2	A2	[6, 11]	85.5/91.8	85.8/93.1	71.2/83.1	76.2/81.6
x1	[23]	95.6/96.3	95.8/96.5	93.8/91.1	94.7/89.5	TCI	[7, 11]	86.8/74.9	86.5/80.7	87.5/73.2	90.0/79.3
HILB	[12, 33]	96.5/93.3	95.8/93.7	93.8/88.7	92.4/87.3	x4	[23]	77.7/93.7	79.2/93.2	66.3/89.5	72.4/85.9
SamEn	[30]	94.9/91.6	96.6/92.1	91.3/89.9	91.5/91.2	Li	[29]	82.3/77.6	94.9/86.2	74.3/69.6	85.3/81.6
bWT	[27]	96.1/90.8	95.9/93.6	91.3/87.9	95.6/86.7	bW	[27]	90.6/88.5	93.5/88.9	80.1/60.1	86.2/55.8
PSR	[12, 33]	96.3/91.3	95.6/92.5	90.9/88.1	91.2/86.9	A3	[6, 11]	79.0/85.9	85.2/83.7	77.8/68.0	70.9/79.0
Count2	[10]	93.2/88.1	93.9/96.1	90.4/87.1	89.1/94.3	CM	[9, 11]	84.5/63.3	83.7/67.9	80.7/79.4	87.4/78.2
x2	[23]	95.0/95.0	92.8/96.0	90.4/87.1	87.9/85.6	M	[6, 11]	82.2/81.3	80.7/86.6	76.6/68.4	72.9/73.1
TCSC	[15]	95.3/91.0	97.1/92.4	91.5/81.4	92.4/83.0	Frqbin	[13, 17]	81.4/66.2	82.1/67.3	89.9/69.7	90.0/73.7
MAV	[14]	95.8/90.4	97.1/92.4	91.5/81.4	92.4/83.0	x5	[23]	86.6/78.9	89.5/78.9	87.4/41.3	88.2/40.3
Count3	[10]	90.3/85.5	94.6/90.6	86.5/84.1	92.1/87.6	CVbin	[13, 17]	91.8/47.2	89.0/48.8	88.7/55.3	90.9/56.0
vFleak	[4, 11]	94.4/93.1	96.2/92.7	78.7/87.4	83.2/85.2	abin	[13, 17]	92.3/46.6	90.6/47.1	89.0/54.9	90.9/56.0
Kurt	[13, 17]	96.3/87.4	96.9/87.8	91.2/76.3	87.6/80.1	x3	[23]	83.8/55.4	80.1/60.4	79.6/52.2	79.4/53.8
Count1	[10]	82.6/82.9	90.3/89.4	86.9/72.2	90.0/82.5	Exp	[11]	58.7/66.5	84.0/66.2	47.1/34.2	83.8/62.1
Expmod	[11]	86.5/78.1	90.0/77.9	87.1/83.7	90.6/81.9	A1	[6, 11]	14.1/92.9	14.2/93.6	15.4/79.0	14.4/77.7

5.3 Biomedical applications

RESEARCH Open Access

Quality estimation of the electrocardiogram using cross-correlation among leads

Eduardo Morgado[1], Felipe Alonso-Atienza[1], Ricardo Santiago-Mozos[1], Óscar Barquero-Pérez[1], Ikaro Silva[2], Javier Ramos[1*] and Roger Mark[2]



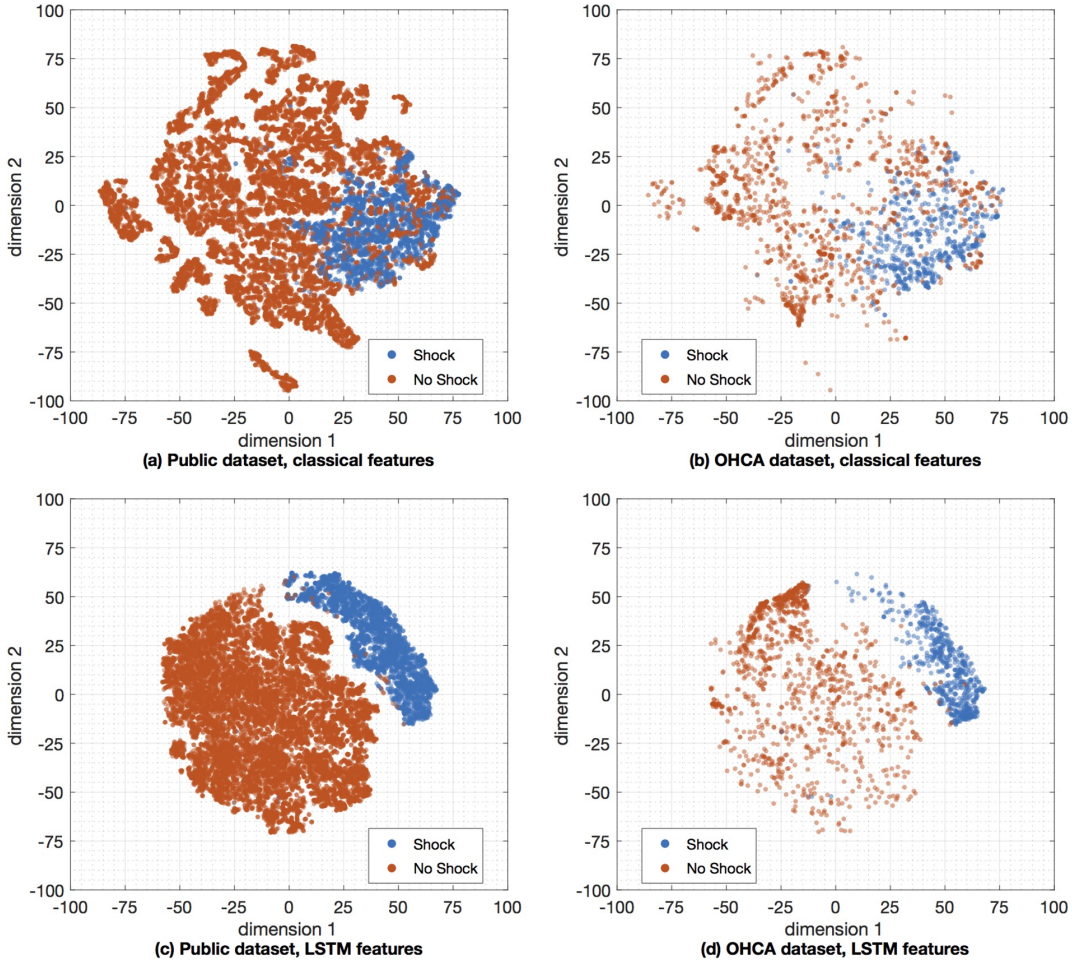
5.3 Biomedical applications

PLOS ONE

RESEARCH ARTICLE

Mixed convolutional and long short-term memory network for the detection of lethal ventricular arrhythmia

Artzai Picon[1*], Unai Irusta[2**], Aitor Álvarez-Gila[1], Elisabete Aramendi[2], Felipe Alonso-Atienza[3,4], Carlos Figuera[3,4], Unai Ayala[5], Estibaliz Garrote[1], Lars Wik[6], Jo Kramer-Johansen[6], Trygve Eftestøl[7]



Additional resources I

Pros and cons of dimensionality reduction

Visualize Nigel Goddard video

https://media.ed.ac.uk/media/Pros+and+cons+of+dimensionality+reduction/1_xo8l1cfm

Dimensionality Reduction - Feature Extraction & Selection

Cognitive Class

https://www.youtube.com/watch?v=AU_hBML2H1c

Feature selection and feature extraction

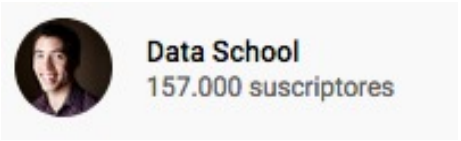
Visualize Nigel Goddard video

https://media.ed.ac.uk/media/Feature+selection+and+feature+extraction/1_71vdt2cd

Additional resources II

How do I select features for Machine Learning?

https://www.youtube.com/watch?v=YaKMeAlHgqQ



Feature Selection Techniques Easily Explained | Machine Learning

https://www.youtube.com/watch?v=EqLBAmtKMnQ