

Non-linear models. Basis expansion.
Overfitting. Regularization.

Petr Pošík

Czech Technical University in Prague
Faculty of Electrical Engineering
Dept. of Cybernetics

Non-linear models	2
Question.....	3
Basis expansion.....	4
Two spaces.....	5
Remarks.....	7
How to evaluate a predictive model?	8
Question.....	9
Model evaluation.....	10
Training and testing error.....	12
Overfitting.....	13
Bias vs Variance.....	14
Crossvalidation.....	15
How to determine a suitable model flexibility.....	16
How to prevent overfitting?.....	17
Regularization	18
Question.....	19
Ridge.....	20
Lasso.....	21
Summary	22
Competencies.....	23

Question

When a linear model is not enough, ...

- A** we are doomed. There are no other methods than linear ones.
- B** we can easily use non-linear models. They are as easy to train as the linear methods.
- C** we must use non-linear models, but their training is much more demanding.
- D** we can use a special type of non-linear models which can be trained in the same way as the linear methods.

Basis expansion

a.k.a. **feature space straightening**.

Why?

- Linear decision boundary (or linear regression model) may not be flexible enough to perform accurate classification (regression).
- The algorithms for fitting linear models can be used to fit (certain type of) *non-linear models*!

How?

- Let's define a new multidimensional image space F .
- Feature vectors x are transformed into this image space F (new features are derived) using mapping Φ :

$$x \rightarrow z = \Phi(x),$$

$$x = (x_1, x_2, \dots, x_D) \rightarrow z = (\Phi_1(x), \Phi_2(x), \dots, \Phi_G(x)),$$

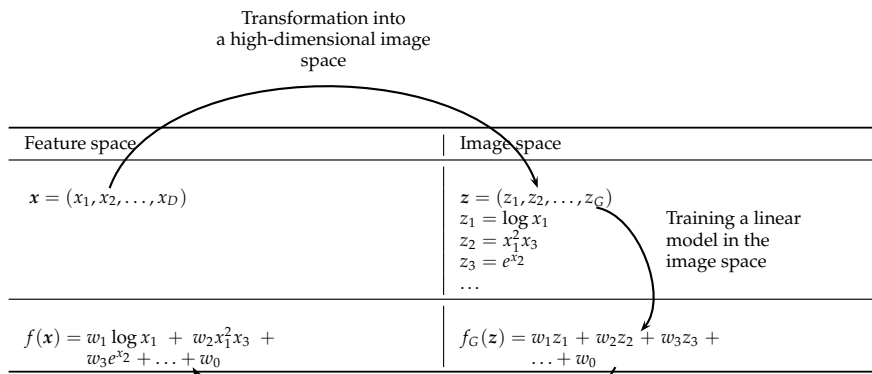
while usually $D \ll G$.

- In the image space, a linear model is trained. However, this is equivalent to training a non-linear model in the original space.

$$f_G(z) = w_1 z_1 + w_2 z_2 + \dots + w_G z_G + w_0$$

$$f(x) = f_G(\Phi(x)) = w_1 \Phi_1(x) + w_2 \Phi_2(x) + \dots + w_G \Phi_G(x) + w_0$$

Two coordinate systems

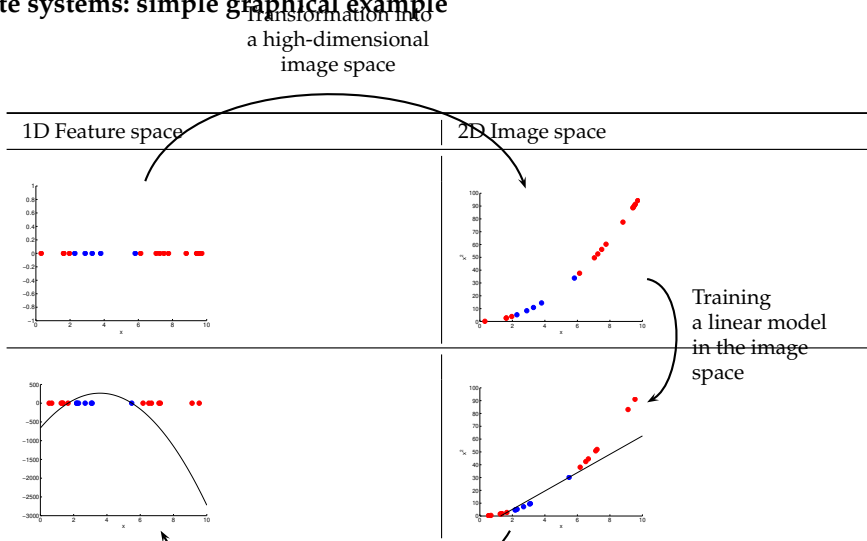


P. Pošík © 2020

Artificial Intelligence – 5 / 23

Non-linear model in the feature space

Two coordinate systems: simple graphical example



P. Pošík © 2020

Artificial Intelligence – 6 / 23

Non-linear model in the feature space

Basis expansion: remarks

Advantages:

- Universal, generally usable method.

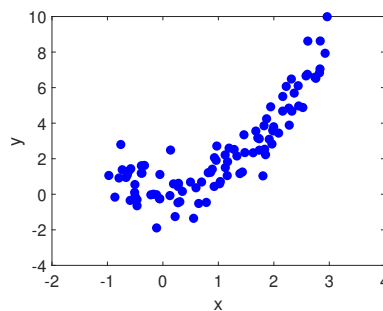
Disadvantages:

- We must define what new features shall form the high-dimensional space F .
- The examples must be really transformed into the high-dimensional space F .
- When too many derived features are used, the resulting models are prone to overfitting (see next slides).

For certain type of algorithms, there is a method how to perform the basis expansion without actually carrying out the mapping!
(See the next lecture.)

How to evaluate a predictive model?

Question



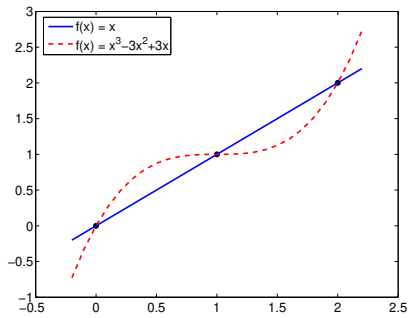
Which of the following models will have the smallest training error if fitted to the above data?

- A** $\hat{y} = w_0 + w_1x$
- B** $\hat{y} = w_0 + w_1x + w_2x^2$
- C** $\hat{y} = w_0 + w_1x + w_2x^2 + w_3x^3$
- D** $\hat{y} = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4$

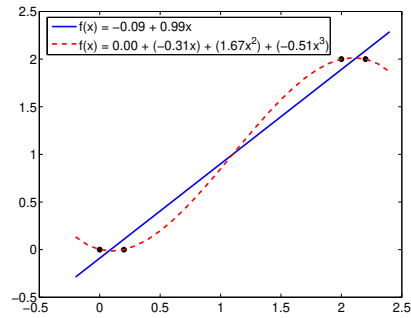
Model evaluation

Fundamental question: What is a good measure of “model quality” from the machine-learning standpoint?

- We have various measures of model error:
 - For regression tasks: MSE, MAE, ...
 - For classification tasks: misclassification rate, measures based on confusion matrix, ...
- Some of them can be regarded as finite approximations of the *Bayes risk*.
- Are these functions *good approximations* when measured on the same data that were used for training?



Using MSE only, both models are equivalent!!!



Using MSE only, the cubic model is better than linear!!!

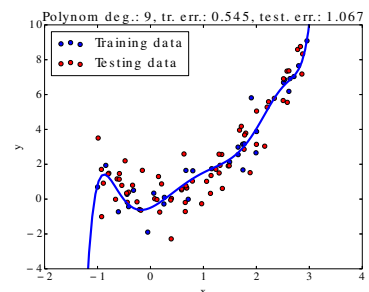
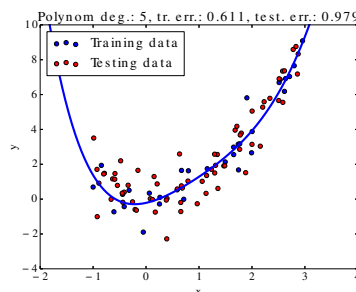
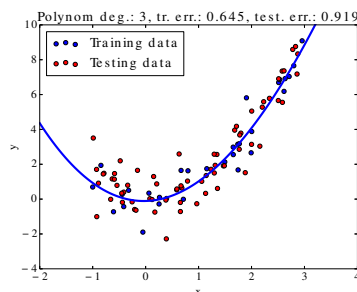
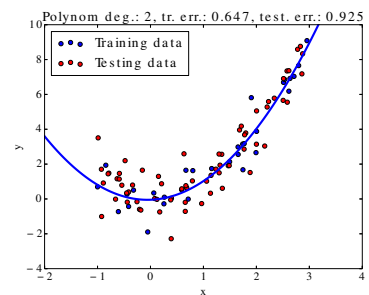
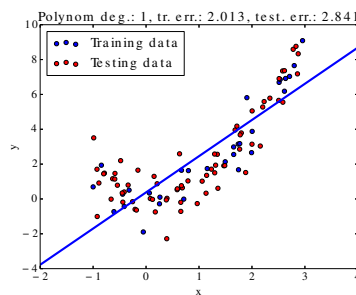
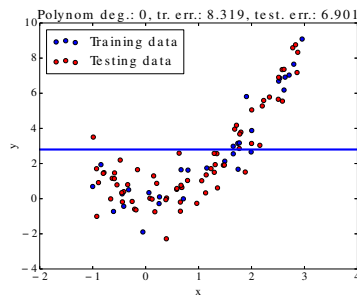
A basic method of evaluation is *model validation on a different, independent data set* from the same source, i.e. on **testing data**.

Validation on testing data

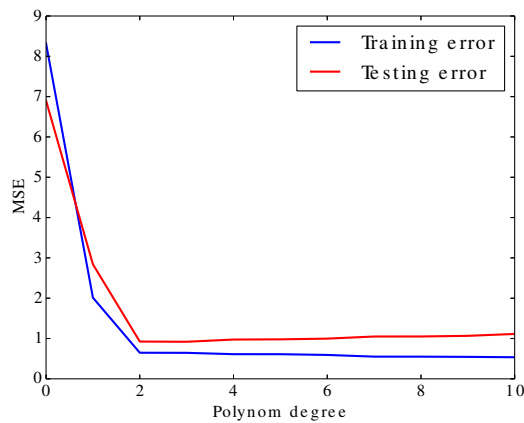
Example: Polynomial regression with varying degree:

$$X \sim U(-1, 3)$$

$$Y \sim X^2 + N(0, 1)$$



Training and testing error



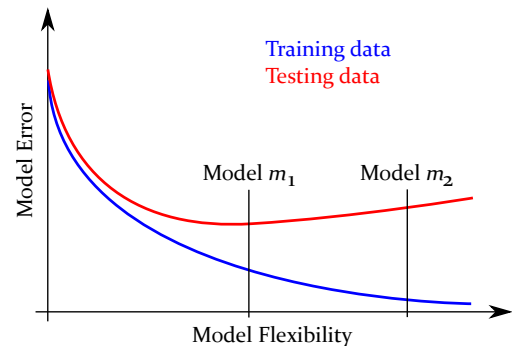
- The *training error* decreases with increasing model flexibility.
- The *testing error* is minimal for certain degree of model flexibility.

Overfitting

Definition of overfitting:

- Let M be the space of candidate models.
- Let $m_1 \in M$ and $m_2 \in M$ be 2 different models from this space.
- Let $\text{Err}_{\text{Tr}}(m)$ be an error of the model m measured on the training dataset (training error).
- Let $\text{Err}_{\text{Tst}}(m)$ be an error of the model m measured on the testing dataset (testing error).
- We say that m_2 is overfitted if there is another m_1 for which

$$\text{Err}_{\text{Tr}}(m_2) < \text{Err}_{\text{Tr}}(m_1) \wedge \text{Err}_{\text{Tst}}(m_2) > \text{Err}_{\text{Tst}}(m_1)$$

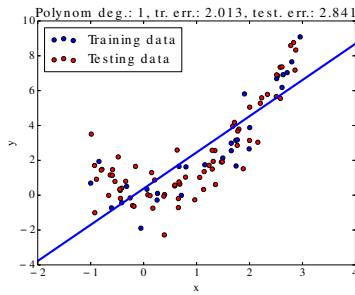


- “When overfitted, the model works well for the training data, but fails for new (testing) data.”
- Overfitting is a general phenomenon *affecting all kinds of inductive learning* of models with tunable flexibility.

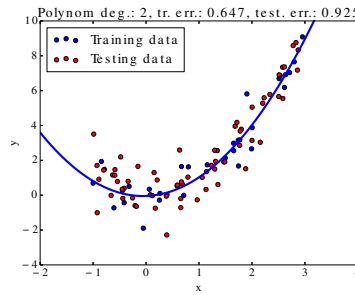
We want models and learning algorithms with a good **generalization ability**, i.e.

- we want models that encode *only the relationships valid in the whole domain*, not those that learned the specifics of the training data, i.e.
- we want algorithms able to find *only the relationships valid in the whole domain* and ignore specifics of the training data.

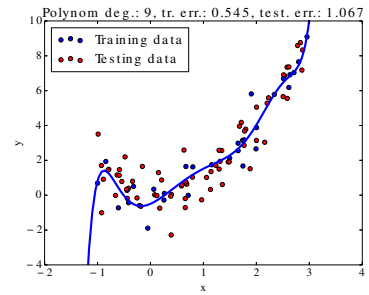
Bias vs Variance



High bias:
model not flexible enough
(Underfit)



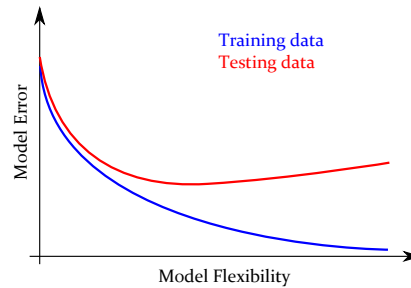
"Just right"
(Good fit)



High variance:
model flexibility too high
(Overfit)

High bias problem:

- $Err_{Tr}(h)$ is high
- $Err_{Tst}(h) \approx Err_{Tr}(h)$



High variance problem:

- $Err_{Tr}(h)$ is low
- $Err_{Tst}(h) \gg Err_{Tr}(h)$

Crossvalidation

How to estimate the true error of a model on new, unseen data?

Simple crossvalidation:

- Split the data into training and testing subsets.
- Train the model on training data.
- Evaluate the model error on testing data.

K-fold crossvalidation:

- Split the data into k folds (k is usually 5 or 10).
- In each iteration:
 - Use $k - 1$ folds to train the model.
 - Use 1 fold to test the model, i.e. measure error.

Iter. 1	Training	Training	Testing
Iter. 2	Training	Testing	Training
Iter. k	Testing	Training	Training

- Aggregate (average) the k error measurements to get the final error estimate.
- Train the model on the whole data set.

Leave-one-out (LOO) crossvalidation:

- $k = |T|$, i.e. the number of folds is equal to the training set size.
- Time consuming for large $|T|$.

How to determine a suitable model flexibility

Simply test models of varying complexities and choose the one with the best testing error, right?

- The testing data are used here to *tune a meta-parameter* of the model.
- *The testing data* are used to train (a part of) the model, thus essentially *become part of training data*.
- The error on testing data is *no longer an unbiased estimate* of model error; it underestimates it.
- A new, separate data set is needed to estimate the model error.

Using simple crossvalidation:

1. *Training data*: use cca 50 % of data for model building.
2. *Validation data*: use cca 25 % of data to search for the suitable model flexibility.
3. Train the suitable model on training + validation data.
4. *Testing data*: use cca 25 % of data for the final estimate of the model error.

Using k -fold crossvalidation

1. *Training data*: use cca 75 % of data to find and train a suitable model using crossvalidation.
2. *Testing data*: use cca 25 % of data for the final estimate of the model error.

The ratios are not set in stone, there are other possibilities, e.g. 60:20:20, etc.

How to prevent overfitting?

1. **Feature selection**: Reduce the number of features.
 - Select manually, which features to keep.
 - Try to identify a suitable subset of features during learning phase (many feature selection methods exist; none is perfect).
2. **Regularization**:
 - Keep all the features, but reduce the magnitude of their weights w .
 - Works well, if we have a lot of features each of which contributes a bit to predicting y .

Question

Which of the following methods will NOT help with overfitting:

- A Choosing only a subset of input variables.
- B Increasing the number of training data.
- C Penalizing the size of model parameters.
- D Basis expansion, i.e., deriving new features from the old ones.

Ridge regularization (a.k.a. Tikhonov regularization)

Ridge regularization penalizes the size of the model coefficients:

- Modification of the optimization criterion:

$$J(w) = \frac{1}{|T|} \sum_{i=1}^{|T|} (y^{(i)} - h_w(x^{(i)}))^2 + \alpha \sum_{d=1}^D w_d^2.$$

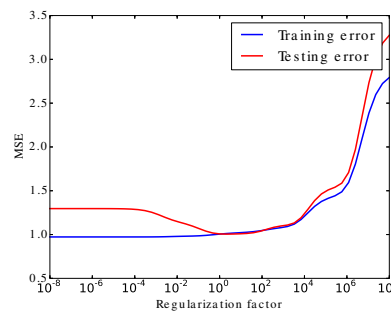
- The solution is given by a modified Normal equation

$$w^* = (X^T X + \alpha I)^{-1} X^T y$$

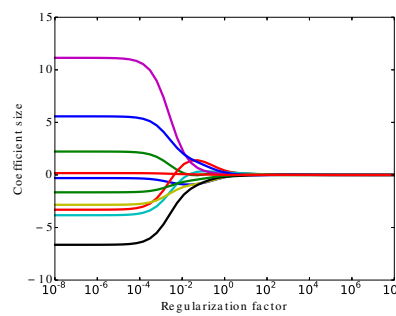
- As $\alpha \rightarrow 0$, $w^{\text{ridge}} \rightarrow w^{\text{OLS}}$.
- As $\alpha \rightarrow \infty$, $w^{\text{ridge}} \rightarrow 0$.

OLS - ordinary least squares. Just a simple multiple linear regression.

Training and testing errors as functions of regularization parameter:



The values of coefficients (weights w) as functions of regularization parameter:



Lasso regularization

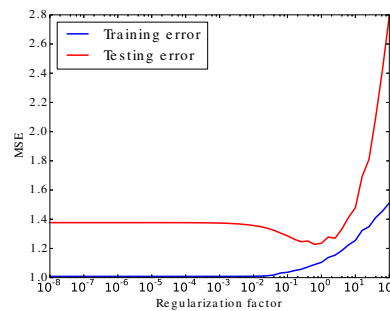
Lasso regularization penalizes the size of the model coefficients:

- Modification of the optimization criterion:

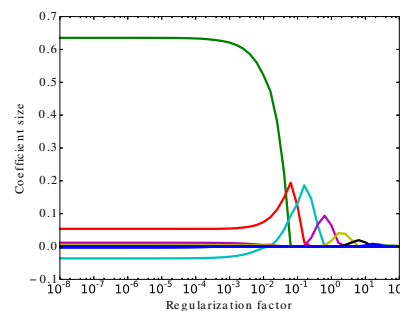
$$J(\mathbf{w}) = \frac{1}{|T|} \sum_{i=1}^{|T|} \left(y^{(i)} - h_{\mathbf{w}}(\mathbf{x}^{(i)}) \right)^2 + \alpha \sum_{d=1}^D |w_d|.$$

- As $\alpha \rightarrow \infty$, Lasso regularization *decreases the number of non-zero coefficients*, effectively also performing *feature selection* and creating *sparse models*.

Training and testing errors as functions of regularization parameter:



The values of coefficients as functions of regularization parameter:



Summary

22 / 23

Competencies

After this lecture, a student shall be able to ...

1. explain the reason for doing basis expansion (feature space straightening), and describe its principle;
2. show the effect of basis expansion with a linear model on a simple example for both classification and regression settings;
3. implement user-defined basis expansions in certain programming language;
4. list advantages and disadvantages of basis expansion;
5. explain why the error measured on the training data is not a good estimate of the expected error of the model for new data, and whether it under- or overestimates the true error;
6. explain basic methods to get unbiased estimate of the true model error (testing data, k-fold crossvalidation, LOO crossvalidation);
7. describe the general form of the dependency of training and testing errors on the model complexity/flexibility/capacity;
8. define overfitting;
9. discuss high bias and high variance problems of models;
10. explain how to proceed if a suitable model complexity must be chosen as part of the training process;
11. list 2 basic methods for overfitting prevention;
12. describe the principles of ridge (Tikhonov) and lasso regularizations and their effects on the model parameters.