

Statistical Natural Language Processing

ML intro & regression

Çağrı Çöltekin

University of Tübingen
Seminar für Sprachwissenschaft

Summer Semester 2018

Why machine learning?

- Majority of the modern computational linguistic tasks and applications are based on machine learning
 - Tokenization
 - Part of speech tagging
 - Parsing
 - ...
 - Speech recognition
 - Named Entity recognition
 - Document classification
 - Question answering
 - Machine translation
 - ...

Machine learning is ...

*The field of machine learning is concerned with the question of how to construct computer programs that automatically **improve with experience**.*

—Mitchell (1997)

Machine learning is ...

*The field of machine learning is concerned with the question of how to construct computer programs that automatically **improve with experience**.*

—Mitchell (1997)

*Machine Learning is the study of data-driven methods capable of mimicking, understanding and aiding **human and biological information processing tasks**.*

—Barber (2012)

Machine learning is ...

*The field of machine learning is concerned with the question of how to construct computer programs that automatically **improve with experience**.*
—Mitchell (1997)

*Machine Learning is the study of data-driven methods capable of mimicking, understanding and aiding **human and biological information processing tasks**.*
—Barber (2012)

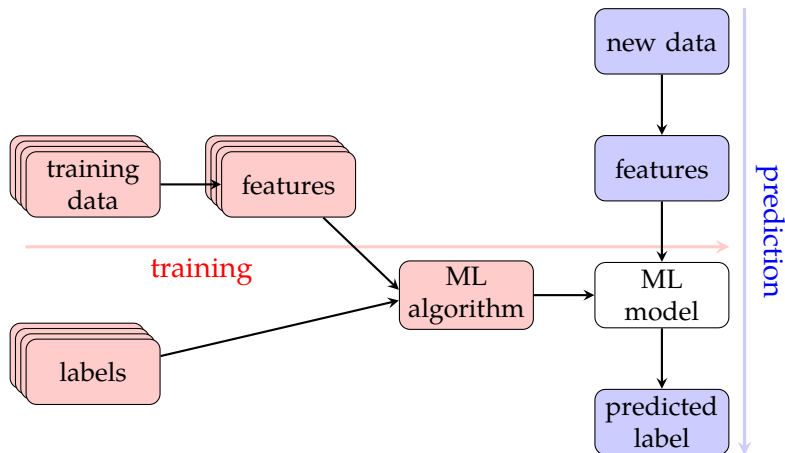
*Statistical learning refers to a vast set of tools for **understanding data**.*
—James et al. (2013)

Supervised or unsupervised

- Machine learning methods are often divided into two broad categories: *supervised* and *unsupervised*
- Supervised methods rely on *labeled* (or *annotated*) data
- Unsupervised methods try to find regularities in the data without any (direct) supervision
- Some methods do not fit any (or fit both):
 - *Semi-supervised* methods use a mixture of both
 - *Reinforcement learning* refers to the methods where supervision is indirect and/or delayed

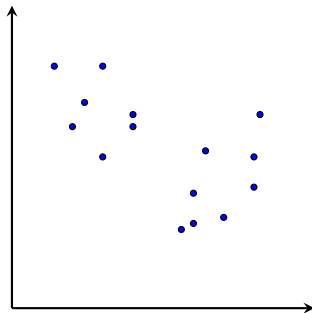
In this course, we will mostly discuss/use supervised methods.

Supervised learning



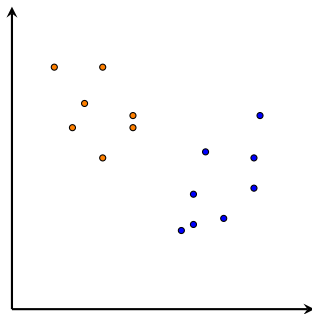
Unsupervised learning

- In unsupervised learning we do not have any labels
- The aim is discovering some 'latent' structure in the data
- Common examples include
 - Clustering
 - Density estimation
 - Dimensionality reduction
- In NLP, methods that do not require (manual) annotation are sometimes called unsupervised



Unsupervised learning

- In unsupervised learning we do not have any labels
- The aim is discovering some 'latent' structure in the data
- Common examples include
 - Clustering
 - Density estimation
 - Dimensionality reduction
- In NLP, methods that do not require (manual) annotation are sometimes called unsupervised



Supervised learning

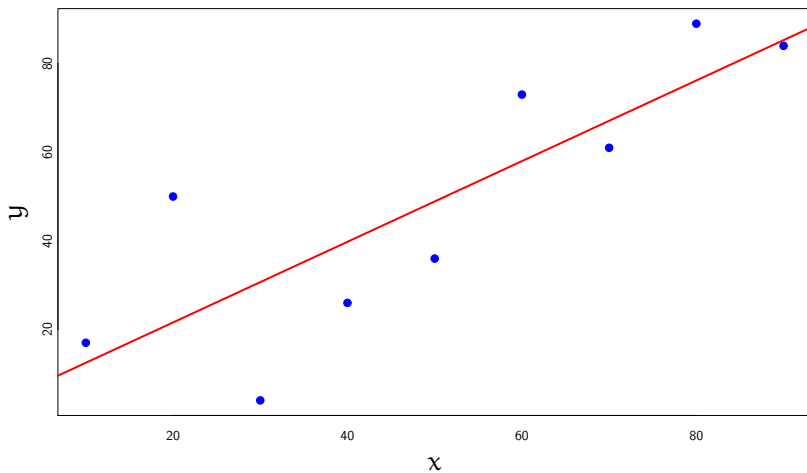
two common settings

An ML algorithm is called

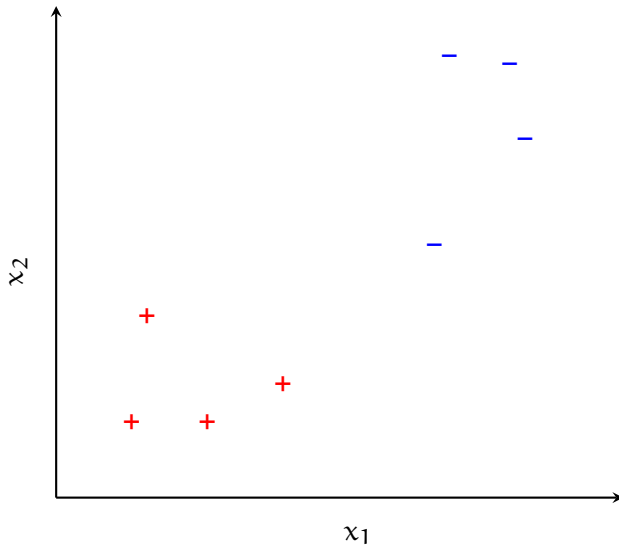
regression if the outcome to be predicted is a numeric
(continuous) variable

classification if the outcome to be predicted is a categorical
variable

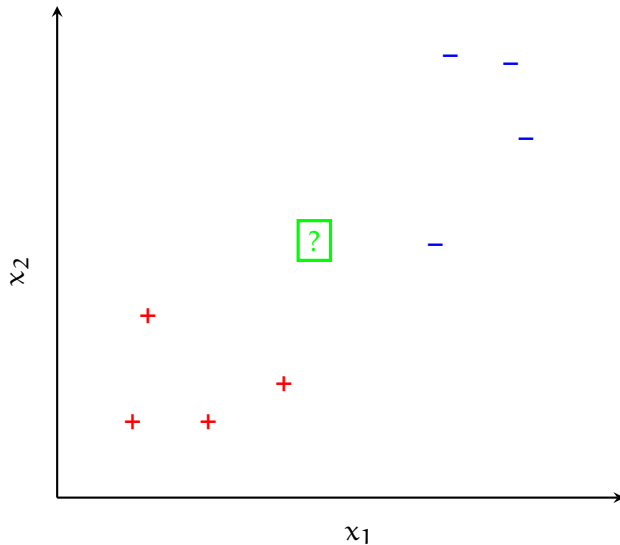
Regression



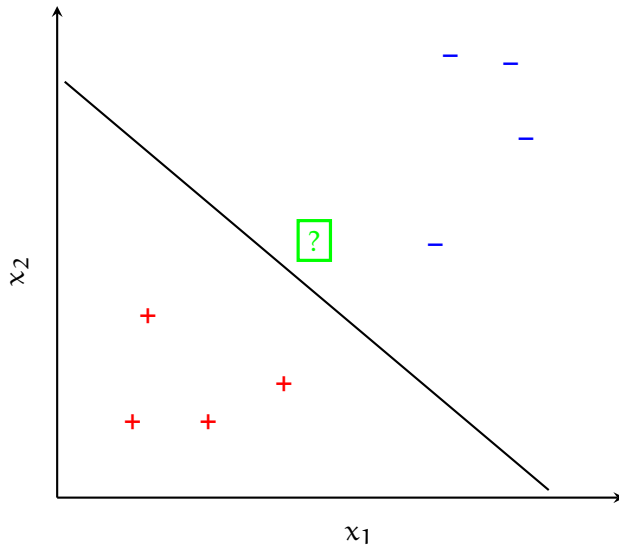
Classification



Classification



Classification



ML topics we will cover in this course

- (Linear) Regression (today)
- Classification (perceptron, logistic regression)
- Evaluation ML methods / algorithms
- Unsupervised learning
- Sequence learning
- Neural networks / deep learning

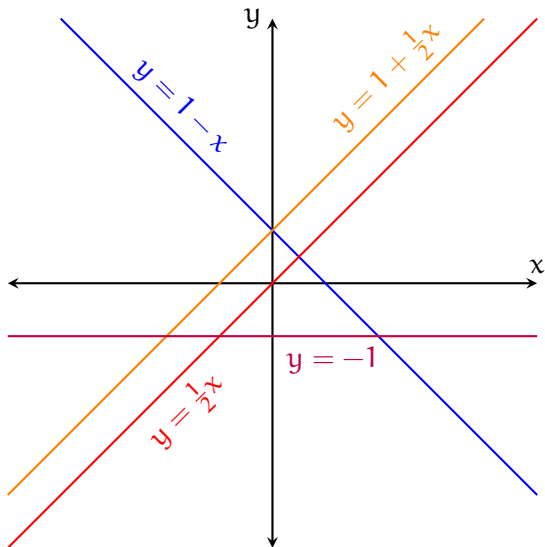
Regression

- Regression is a (supervised) method for predicting the value of a continuous response variable based on a number of predictors
- We estimate the conditional expectation of the outcome variable given the predictor(s)
- It is the foundation of many models in statistics and machine learning
- If the outcome is a label, the problem is called classification

The linear equation: a reminder

$$y = a + bx$$

- a (intercept) is where the line crosses the y axis.
- b (slope) is the change in y as x is increased one unit.



The simple linear model

$$y_i = a + bx_i + \epsilon_i$$

y is the *outcome* (or response, or dependent) variable. The index i represents each unit observation/measurement (sometimes called a 'case')

x is the *predictor* (or explanatory, or independent) variable

a is the *intercept* (called *bias* in the NN literature)

b is the *slope* of the regression line.

a and b are called *coefficients* or *parameters*

$a + bx$ is the *deterministic* part of the model. It is the model's prediction of y (\hat{y}), given x

ϵ is the **residual**, error, or the variation that is not accounted for by the model. Assumed to be normally distributed with 0 mean

Notation differences for the regression equation

$$y_i = a + bx_i + \epsilon_i$$

Notation differences for the regression equation

$$y_i = \alpha + \beta x_i + \epsilon_i$$

- Sometimes, Greek letters α and β are used for intercept and the slope, respectively

Notation differences for the regression equation

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- Sometimes, Greek letters α and β are used for intercept and the slope, respectively
- Another common notation to use only b , β , θ or w , but use subscripts, 0 indicating the intercept and 1 indicating the slope

Notation differences for the regression equation

$$y_i = w_0 + w_1 x_i + \epsilon_i$$

- Sometimes, Greek letters α and β are used for intercept and the slope, respectively
- Another common notation to use only b , β , θ or w , but use subscripts, 0 indicating the intercept and 1 indicating the slope
- In machine learning it is common to use w for all coefficients (sometimes you may see b used instead of w_0)

Notation differences for the regression equation

$$y_i = \hat{w}_0 + \hat{w}_1 x_i + \epsilon_i$$

- Sometimes, Greek letters α and β are used for intercept and the slope, respectively
- Another common notation to use only b , β , θ or w , but use subscripts, 0 indicating the intercept and 1 indicating the slope
- In machine learning it is common to use w for all coefficients (sometimes you may see b used instead of w_0)
- Sometimes coefficients wear hats, to emphasize that they are estimates

Notation differences for the regression equation

$$y_i = \mathbf{w} \mathbf{x}_i + \epsilon_i$$

- Sometimes, Greek letters α and β are used for intercept and the slope, respectively
- Another common notation to use only b , β , θ or w , but use subscripts, 0 indicating the intercept and 1 indicating the slope
- In machine learning it is common to use w for all coefficients (sometimes you may see b used instead of w_0)
- Sometimes coefficients wear hats, to emphasize that they are estimates
- Often, we use the vector notation for both input(s) and coefficients: $\mathbf{w} = (w_0, w_1)$ and $\mathbf{x}_i = (1, x_i)$

Estimating model parameters: reminder

In least-squares regression, we find

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_i (y_i - \hat{y}_i)^2$$

In general, we define an objective (or loss) function $J(\mathbf{w})$ (e.g., negative log likelihood), and minimize it with respect to the parameters

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{w})$$

Then,

- take the derivative of $J(\mathbf{w})$
- set it to 0
- solve the resulting equation(s)

Least-squares regression

$$y_i = \underbrace{w_0 + w_1 x_i}_{\hat{y}_i} + \epsilon_i$$

Least-squares regression

$$y_i = \underbrace{w_0 + w_1 x_i}_{\hat{y}_i} + \epsilon_i$$

- Find w_0 and w_1 , that minimize the prediction error:

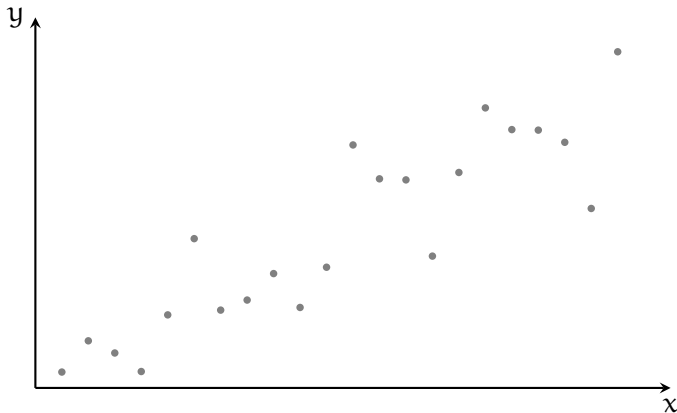
$$J(\mathbf{w}) = \sum_i \epsilon_i^2 = \sum_i (y_i - \hat{y}_i)^2 = \sum_i (y_i - (w_0 + w_1 x_i))^2$$

- We can minimize $J(\mathbf{w})$ analytically

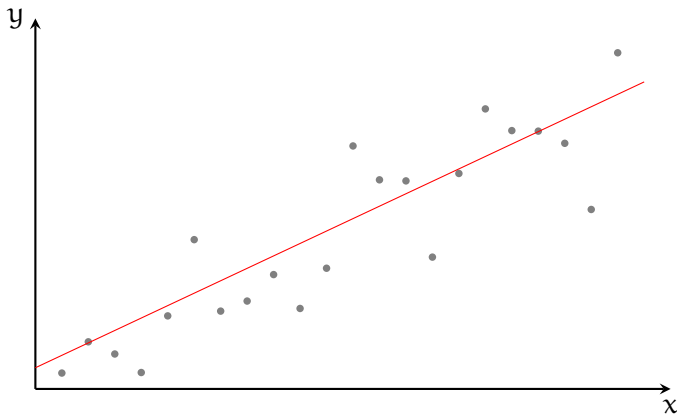
$$w_1 = r \frac{sd_y}{sd_x} \qquad w_0 = \bar{y} - w_1 \bar{x}$$

* See appendix for the derivation.

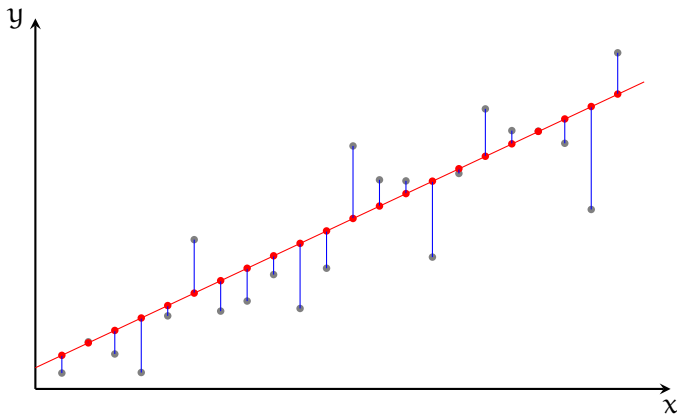
Visualization of least-squares regression



Visualization of least-squares regression



Visualization of least-squares regression



What is special about least-squares?

- Minimizing MSE (or SS_R) is equivalent to MLE estimate under the assumption $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- Working with 'minus log likelihood' is more convenient

$$J(\mathbf{w}) = -\log \mathcal{L}(\mathbf{w}) = -\log \prod_i \frac{e^{-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$$

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} (-\log \mathcal{L}(\mathbf{w})) = \arg \min_{\mathbf{w}} \sum_i (y_i - \hat{y}_i)^2$$

- There are other error functions, e.g., absolute value of the errors, that can be used (and used in practice)
- One can also estimate regression parameters using Bayesian estimation

Short digression: minimizing functions

In least squares regression, we want to find w_0 and w_1 values that minimize

$$J(\mathbf{w}) = \sum_i (y_i - (w_0 + w_1 x_i))^2$$

- Note that $J(\mathbf{w})$ is a *quadratic* function of $\mathbf{w} = (w_0, w_1)$
- As a result, $J(\mathbf{w})$ is *convex* and have a single extreme value
 - there is a unique solution for our minimization problem
- In case of least squares regression, there is an analytic solution
- Even if we do not have an analytic solution, if our error function is convex, a search procedure like *gradient descent* can still find the *global minimum*

Measuring success in Regression

- *Root-mean-square error (RMSE)*

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2}$$

measures average error in the units compatible with the outcome variable.

- Another well-known measure is the *coefficient of determination*

$$R^2 = \frac{\sum_i^n (\hat{y}_i - \bar{y})^2}{\sum_i^n (y_i - \bar{y})^2} = 1 - \left(\frac{\text{RMSE}}{\sigma_y} \right)^2$$

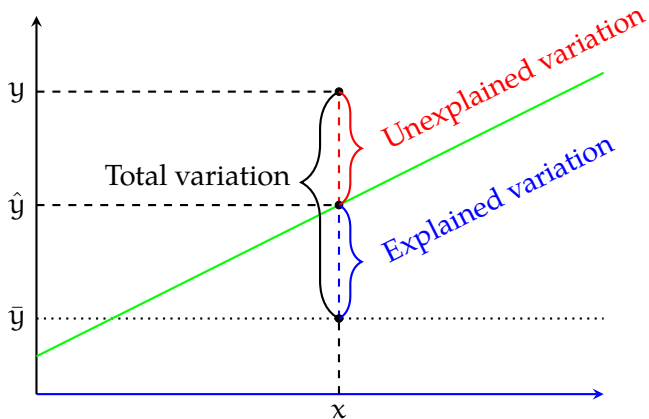
Assessing the model fit: r^2

We can express the variation explained by a regression model as:

$$\frac{\text{Explained variation}}{\text{Total variation}} = \frac{\sum_i^n (\hat{y}_i - \bar{y})^2}{\sum_i^n (y_i - \bar{y})^2}$$

- This value is the square of the correlation coefficient
- The range of r^2 is $[0, 1]$
- $100 \times r^2$ is interpreted as 'the percentage of variance explained by the model'
- r^2 shows how well the model fits to the data: closer the data points to the regression line, higher the value of r^2

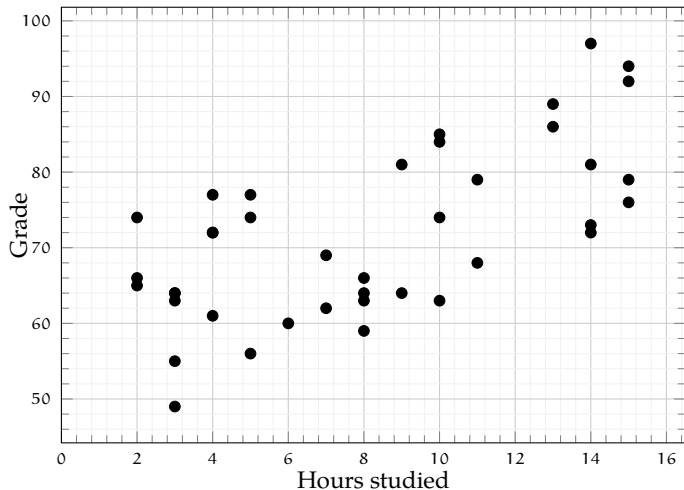
Explained variation



$$\begin{aligned}
 \text{Total variation} &= \text{Unexplained variation} + \text{Explained variation} \\
 y - \bar{y} &= y - \hat{y} + \hat{y} - \bar{y}
 \end{aligned}$$

A hands-on exercise

Draw a regression line over the plot

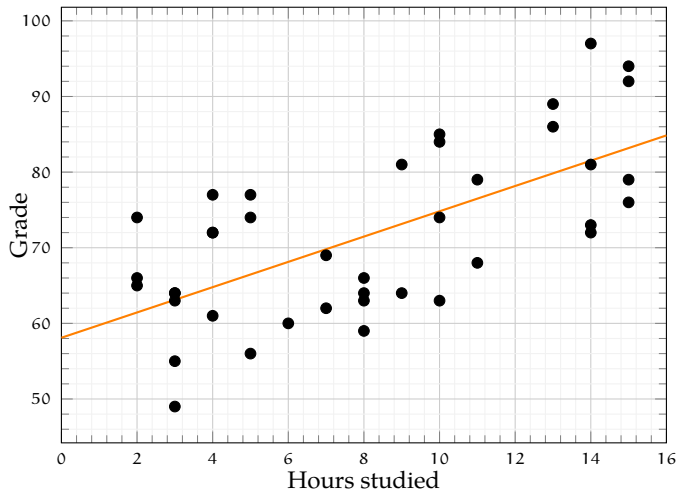


A hands-on exercise (cont.)

- What is the regression equation?
- What is the expected grade for a student who did not study at all?
- What is the expected grade for a student who studied 12 hours?
- What is the expected grade for a student who studied 40 hours?

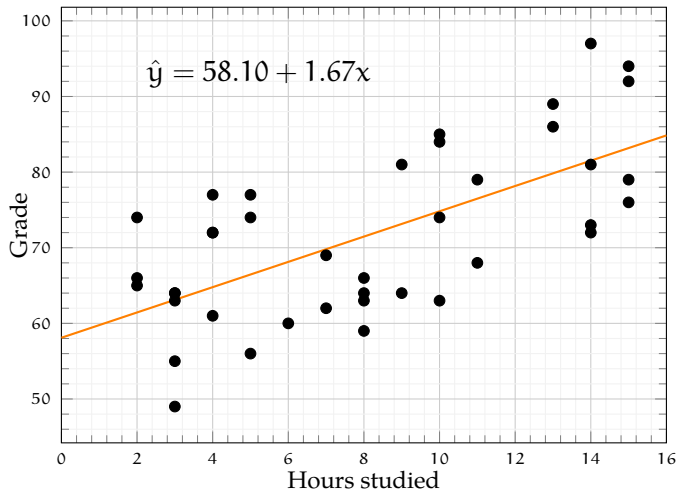
A hands-on exercise

The regression line



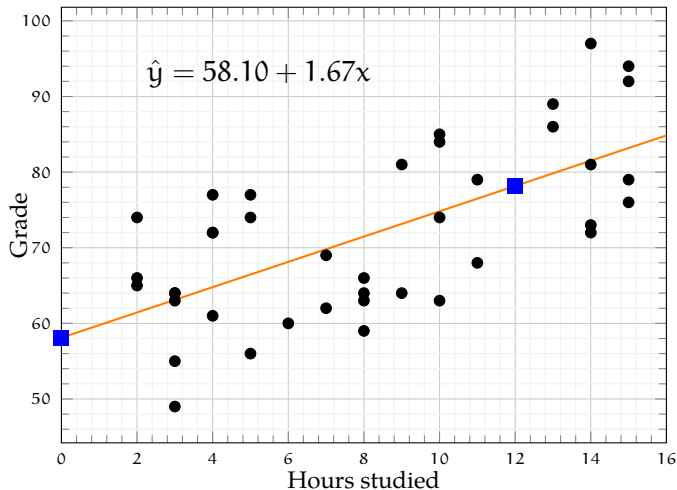
A hands-on exercise

The regression line



A hands-on exercise

The regression line



Regression with multiple predictors

$$y_i = \underbrace{w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \dots + w_k x_{i,k}}_{\hat{y}} + \epsilon_i = \mathbf{w} \mathbf{x}_i + \epsilon_i$$

w_0 is the intercept (as before).

$w_{1..k}$ are the coefficients of the respective predictors.

ϵ is the error term (residual).

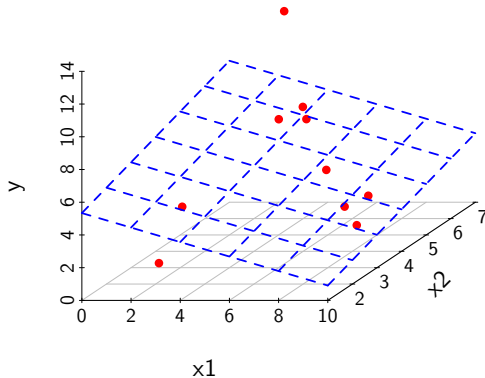
- using vector notation the equation becomes:

$$y_i = \mathbf{w} \mathbf{x}_i + \epsilon_i$$

where $\mathbf{w} = (w_0, w_1, \dots, w_k)$ and $\mathbf{x}_i = (1, x_{i,1}, \dots, x_{i,k})$

It is a generalization of simple regression with some additional power and complexity.

Visualizing regression with two predictors



Input/output of liner regression: some notation

A regression with k input variables and n instances can be described as:

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,k} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,k} \end{bmatrix}}_{\mathbf{X}} \times \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_k \end{bmatrix}}_{\mathbf{w}} + \underbrace{\begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

Estimation in multiple regression

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

We want to minimize the error (as a function of \mathbf{w}):

$$\begin{aligned}\epsilon^2 = J(\mathbf{w}) &= (\mathbf{y} - \mathbf{X}\mathbf{w})^2 \\ &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2\end{aligned}$$

Our least-squares estimate is:

$$\begin{aligned}\hat{\mathbf{w}} &= \arg \min_{\mathbf{w}} J(\mathbf{w}) \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T\end{aligned}$$

Note: the least-squares estimate is also the maximum likelihood estimate under the assumption of normal distribution of errors.

Categorical predictors

- Categorical predictors are represented as multiple binary coded input variables
- For a binary predictor, we use a single binary input. For example, (1 for one of the values, and 0 for the other)

$$x = \begin{cases} 0 & \text{for male} \\ 1 & \text{for female} \end{cases}$$

- For a categorical predictor with k values, we use $k - 1$ predictors (various coding schemes are possible). For example, for 3-values

$$x = \begin{cases} (0, 0, 1) & \text{neutral} \\ (0, 1, 0) & \text{negative} \\ (1, 0, 0) & \text{positive} \end{cases} \quad \text{one-hot coding}$$

$$x = \begin{cases} (0, 0) & \text{neutral} \\ (0, 1) & \text{negative} \\ (1, 0) & \text{positive} \end{cases} \quad \text{'treatment' encoding}$$

Dealing with non-linearity

- Least squares works, because the loss function is linear with respect to parameter w
- Introducing non-linear combinations of inputs does not affect the estimation procedure. The following are still linear models

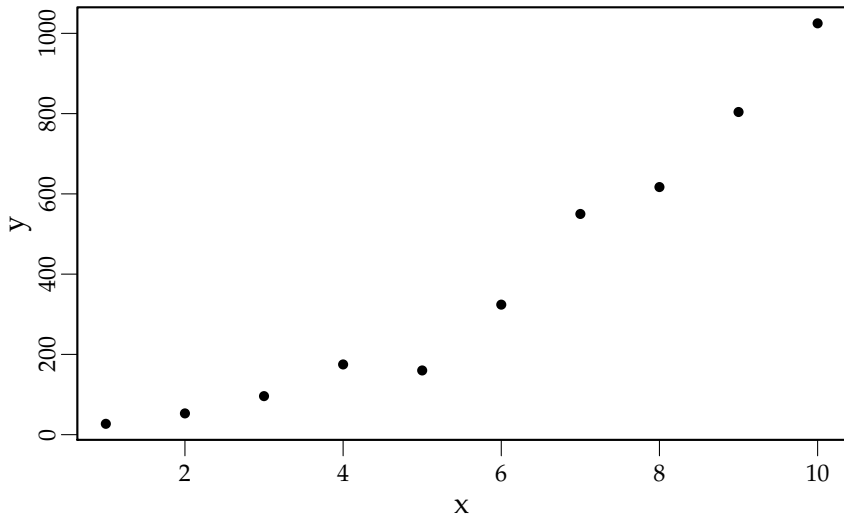
$$y_i = w_0 + w_1 x_i^2 + \epsilon_i$$

$$y_i = w_0 + w_1 \log(x_i) + \epsilon_i$$

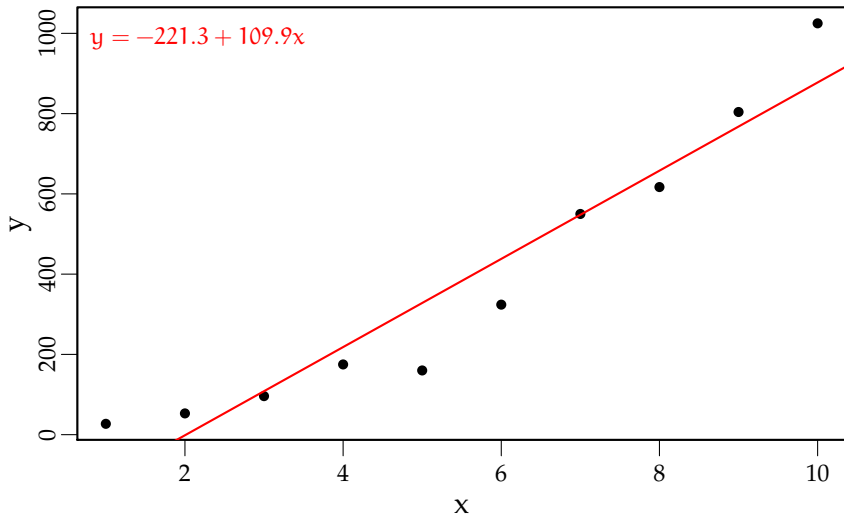
$$y_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + w_3 x_{i,1} x_{i,2} + \epsilon_i$$

- These *transformations* allow linear models to deal with some non-linearities
- In general, we can replace input x by a function of the input(s) $\Phi(x)$. $\Phi()$ is called a *basis function*

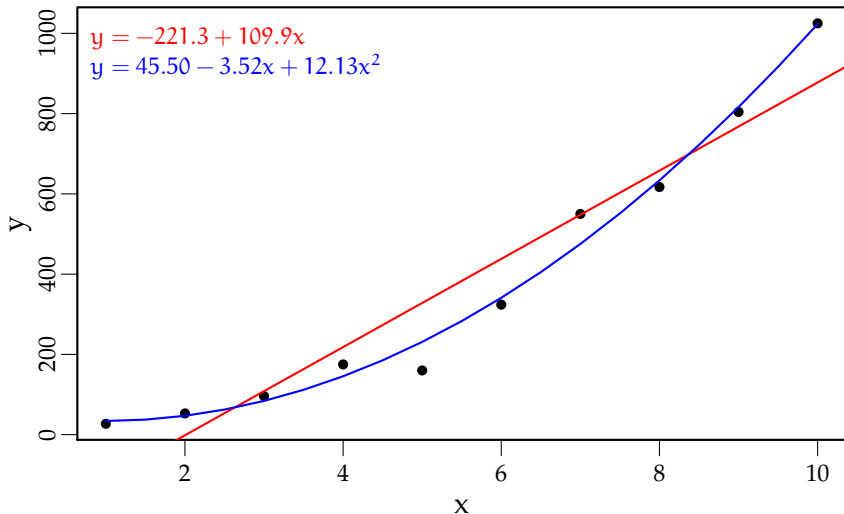
Example: polynomial basis functions



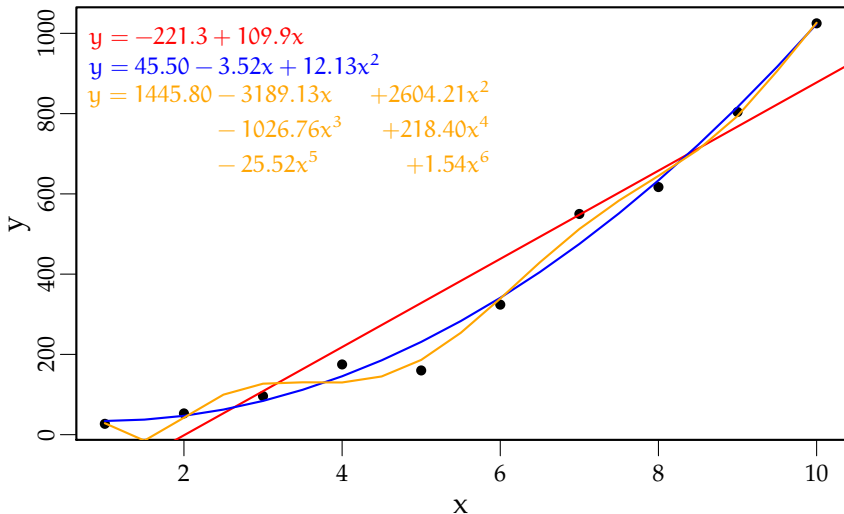
Example: polynomial basis functions



Example: polynomial basis functions



Example: polynomial basis functions



Regularized parameter estimation

- To avoid overfitting and high variance, one of the common methods is *regularization*
- With regularization, in addition of minimizing the cost function, we simultaneously constrain the possible parameter values
- For example, the regression estimation becomes:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_i (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2$$

Regularized parameter estimation

- To avoid overfitting and high variance, one of the common methods is *regularization*
- With regularization, in addition of minimizing the cost function, we simultaneously constrain the possible parameter values
- For example, the regression estimation becomes:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_i (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 + \lambda \sum_{j=1}^k w_j^2$$

- The new part is called the regularization term, where λ is a *hyperparameter* that determines the effect of the regularization.
- In effect, we are preferring small values for the coefficients
- Note that we do not include w_0 in the regularization term

L2 regularization

The form of regularization, where we minimize the regularized cost function,

$$J(\mathbf{w}) + \lambda \|\mathbf{w}\|$$

is called L2 regularization.

- Note that we are minimizing the L2-norm of the weight vector
- In statistic literature this L2-regularized regression is called *ridge regression*
- The method is general: it can be applied to other ML methods as well
- The choice of λ is important
- Note that the scale of the input becomes important

L1 regularization

In L1 regularization we minimize

$$J(\mathbf{w}) + \lambda \sum_{j=1}^k |w_j|$$

- The additional term is the L1-norm of the weight vector (excluding w_0)
- In statistic literature the L1-regularized regression is called *lasso*
- The main difference from L2 regularization is that L1 regularization forces some values to be 0 – the resulting model is said to be ‘sparse’

Regularization as constrained optimization

L1 and L2 regularization can be viewed as minimization with constraints

L2 regularization

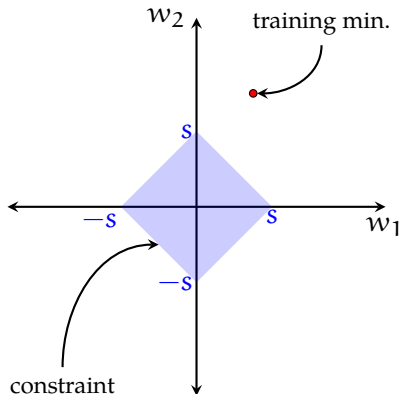
Minimize $J(\mathbf{w})$ with constraint $\|\mathbf{w}\| < s$

L1 regularization

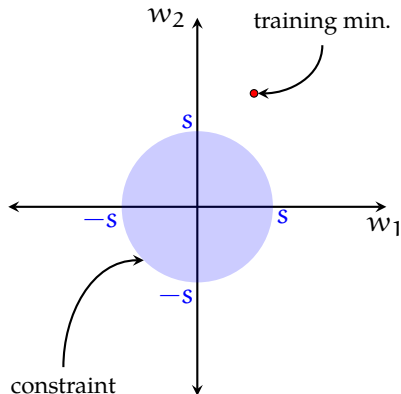
Minimize $J(\mathbf{w})$ with constraint $\|\mathbf{w}\|_1 < s$

Visualization of regularization constraints

L1 regularization

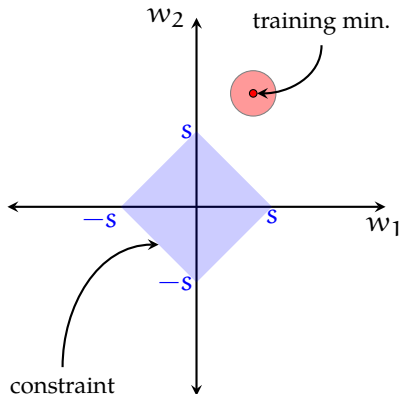


L2 regularization

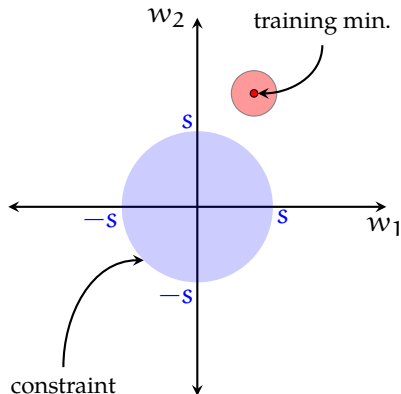


Visualization of regularization constraints

L1 regularization

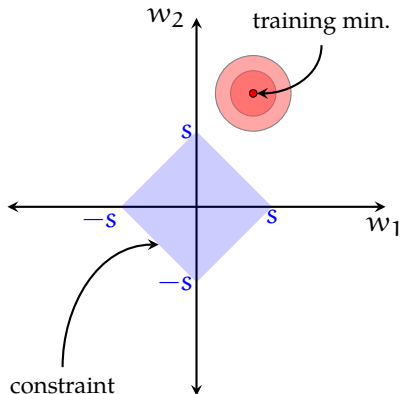


L2 regularization

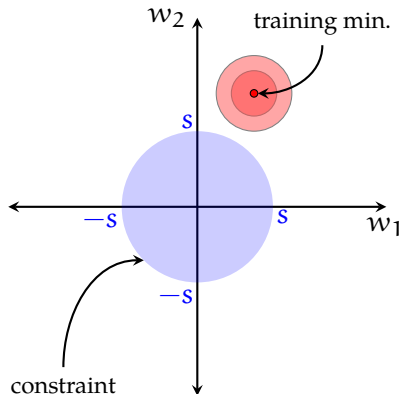


Visualization of regularization constraints

L1 regularization

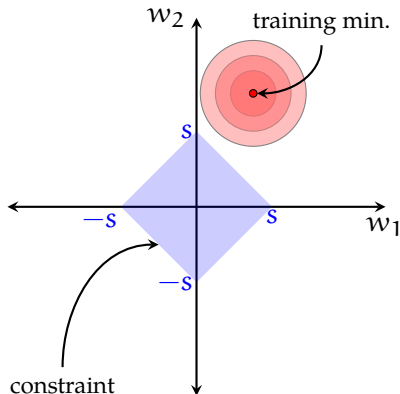


L2 regularization

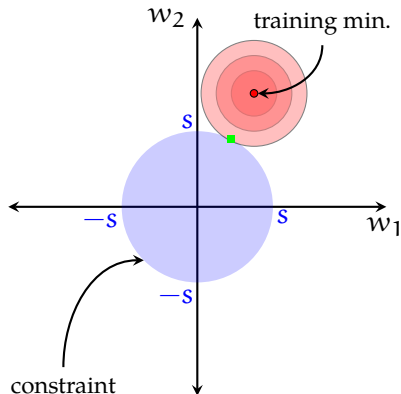


Visualization of regularization constraints

L1 regularization

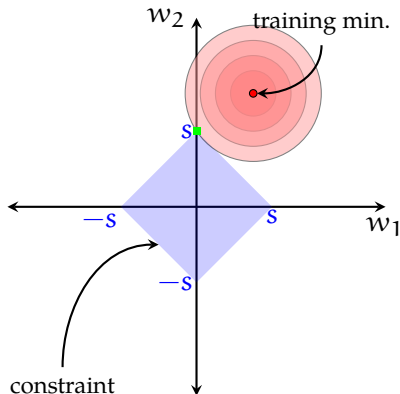


L2 regularization

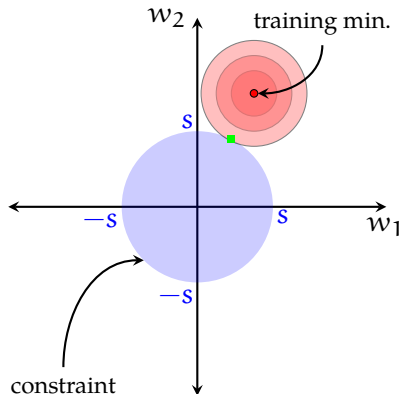


Visualization of regularization constraints

L1 regularization



L2 regularization



Regularization: some remarks

- Regularization prevents overfitting and reduces variance
- The *hyperparameter* λ needs to be determined
 - best value is found typically using a *grid search*, or a *random search*
 - it is tuned on an additional partition of the data, *development set*
 - **development set cannot overlap with training or test set**
- The regularization terms can be interpreted as *priors* in a Bayesian setting
- Particularly, L2 regularization is equivalent to a normal prior with zero mean

Summary

What to remember:

- Supervised vs. unsupervised learning
- Regression vs. classification
- Linear regression equation
- Least-square estimate
- MSE, r^2
- non-linearity & basis functions
- L1 & L2 regularization (lasso and ridge)

Next:

Mon classification

Wed exercises

Fri classification / ML evaluation

Additional reading, references, credits

- Hastie, Tibshirani, and Friedman (2009) discuss introductory bits in chapter 1, and regression on chapter 3 (sections 3.2 and 3.4 are most relevant to this lecture)
- Jurafsky and Martin (2009) has a short section (6.6.1) on regression
- You can also consult any machine learning book (including the ones listed below)



Barber, David (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press. ISBN: 9780521518147.



Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second. Springer series in statistics. Springer-Verlag New York. ISBN: 9780387848587. URL: <http://web.stanford.edu/~hastie/ElemStatLearn/>.



James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York. ISBN: 9781461471387. URL: <http://www-bcf.usc.edu/~gareth/ISL/>.

Additional reading, references, credits (cont.)



Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. second. Pearson Prentice Hall. ISBN: 978-0-13-504196-3.



Mitchell, Thomas (1997). *Machine Learning*. 1st. McGraw Hill Higher Education. ISBN: 0071154671,0070428077,9780071154673,9780070428072.