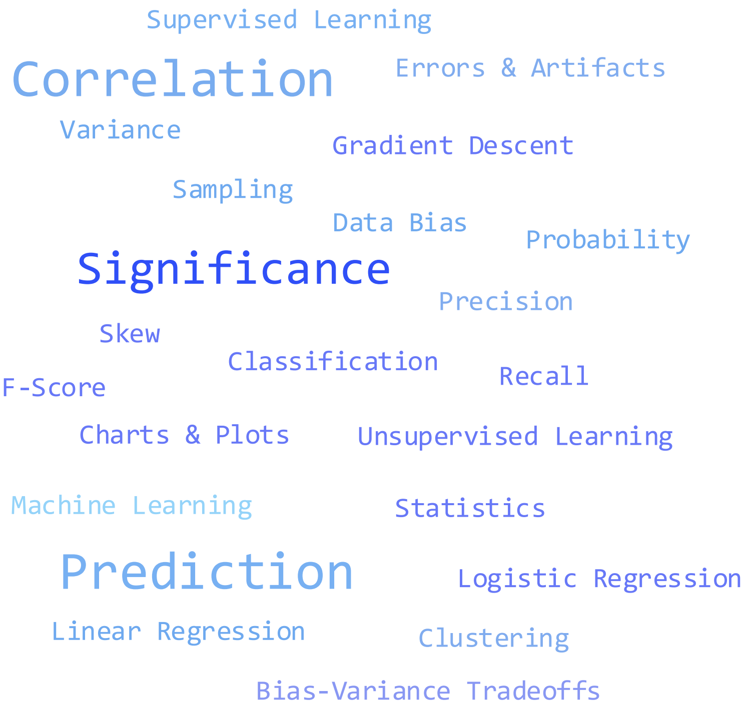


Data Science 1:
Introduction to Data Science

Linear Regression & Gradient Descent

Winter 2025

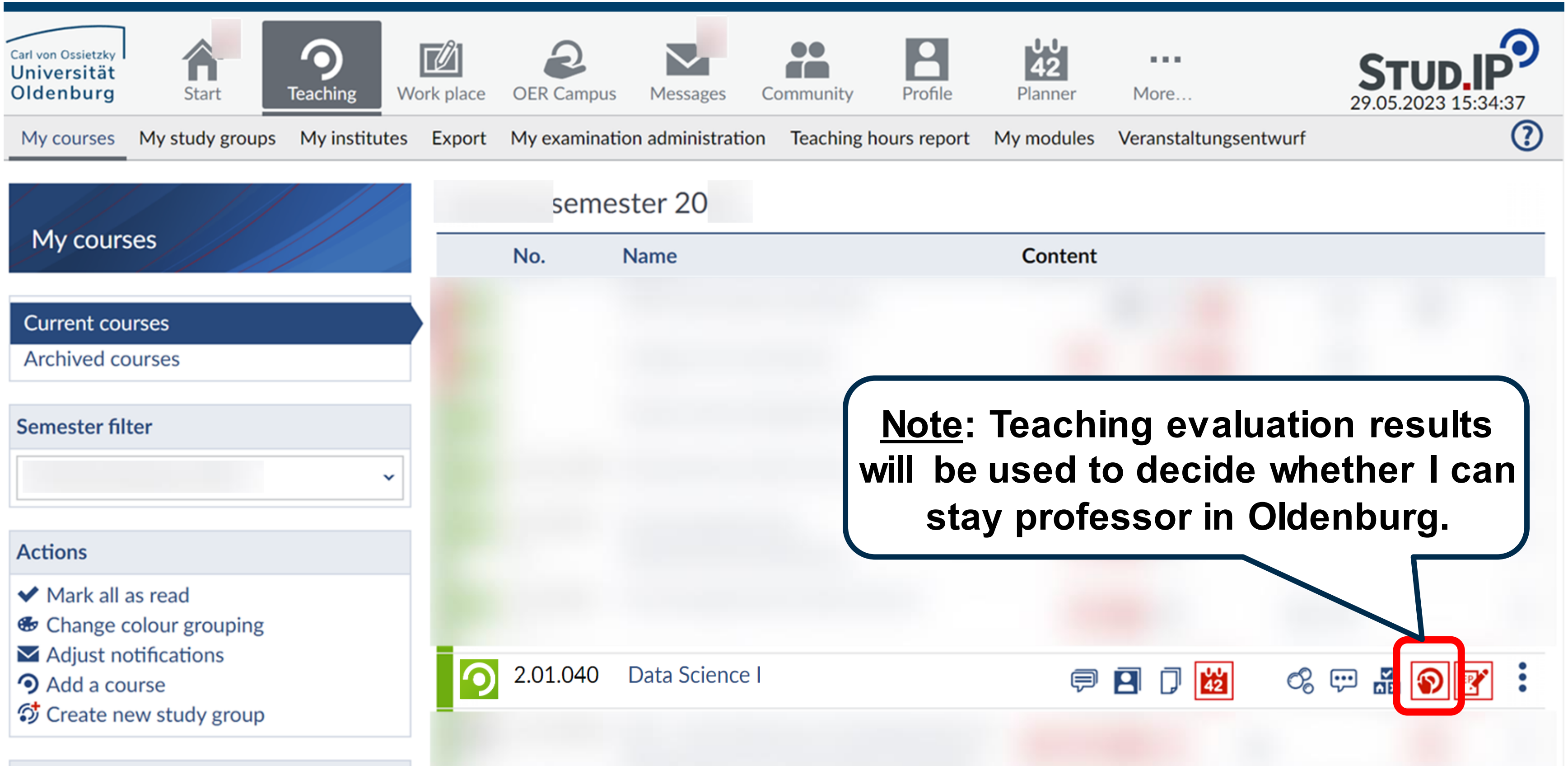
Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

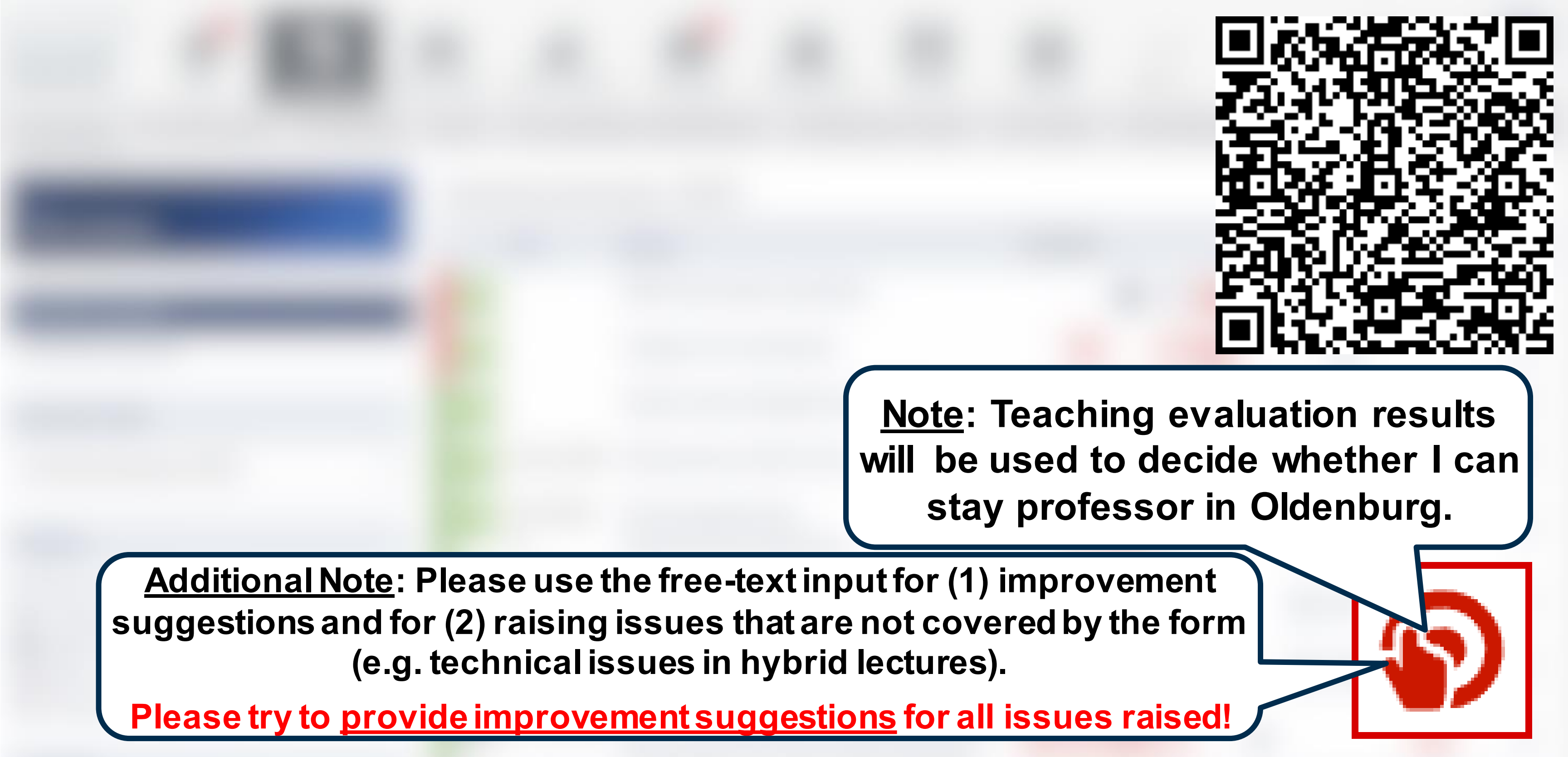
Lecture slides based on content from “The Data Science Design Manual” (Steven Skiena, 2017) and associated course materials generously made available online by the author at https://www3.cs.stonybrook.edu/~skiena/data-manual/.

Special thanks to Professor Skiena for sharing these valuable teaching resources!

Before We Start: Teaching Evaluation!



Before We Start: Teaching Evaluation!



<u>Note</u>: Teaching evaluation results will be used to decide whether I can stay professor in Oldenburg.

<u>Additional Note</u>: Please use the free-text input for (1) improvement suggestions and for (2) raising issues that are not covered by the form (e.g. technical issues in hybrid lectures).

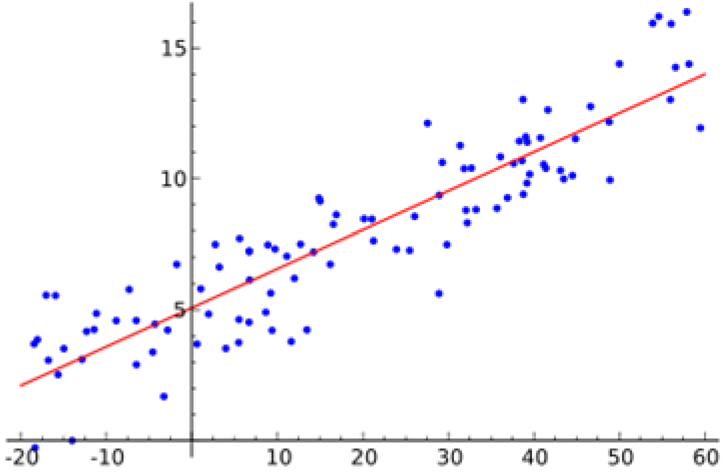
Please try to <u>provide improvement suggestions</u> for all issues raised!

Semester Schedule

CW 42	14. Oct	Lecture	1	Orga & Intro	1-26
CW 43	21. / 23. Oct	Lecture + Exercises	2	Probability, Statistics & Correlation	27-56
CW 44	28. Oct	Lecture	3	Data Munging, Cleaning & Bias	57-94 / “Invisible Women”
CW 45	04. / 06. Nov	Lecture + Exercises	4	Scores & Rankings	95-120
CW 46	11. Nov	Lecture	5	Statistical Distributions & Significance	121-154
CW 47	18. / 20. Nov	Lecture + Exercises	6	Building & Evaluating Models	201-236
CW 48	25. Nov	Guest Lecture	7	Data Visualization	155-200
CW 49	02. / 04. Dec	Lecture + Exercises	8	Intro to Machine Learning	351-390
CW 50	09. Dec	Lecture	9	Linear Algebra	237-266
CW 51	16. / 18. Dec	Lecture + Exercises	10	Linear Regression & Gradient Descent	267-288
CW 02	06. Jan	Lecture	11	Logistic Regression & Classification	289-302
CW 03	13. / 15. Jan	Lecture + Exercises	12	Nearest Neighbor Methods & Clustering	303-350
CW 04	20. Jan	Lecture	13	Data Science in the Wild	391-426
CW 05	27. / 29. Jan	Lecture + Exercises	14	Q&A / Feedback	
CW 06	03. / 04. Feb	Oral Exams (Block 1)	Preparation in our last session („Oral Exam Briefing“)		
CW 13	24. / 25. Mar	Oral Exams (Block 2)			

Linear Regression

Given a collection of *n* points, find the line which best approximates or fits the points.



Why Linear Functions?

Linear relationships are easy to understand, and *grossly* appropriate as a default model:

- Income grows linearly with time worked.
- Housing prices grow linearly with area.
- Weight increases linearly with food eaten.

Statistician's rule: If you really want a function to be linear, measure it at only two points.

Linear Regression and Duality

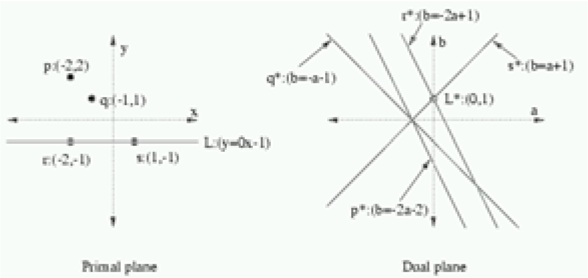
In solving linear systems, given *n* lines we seek the point that lies on all the lines.

In regression, we seek the line that lies on “all” *n* points.

point linear equation

By the duality transformation (s,t) <-> y = (s)x-t

lines are equivalent to points in another space.

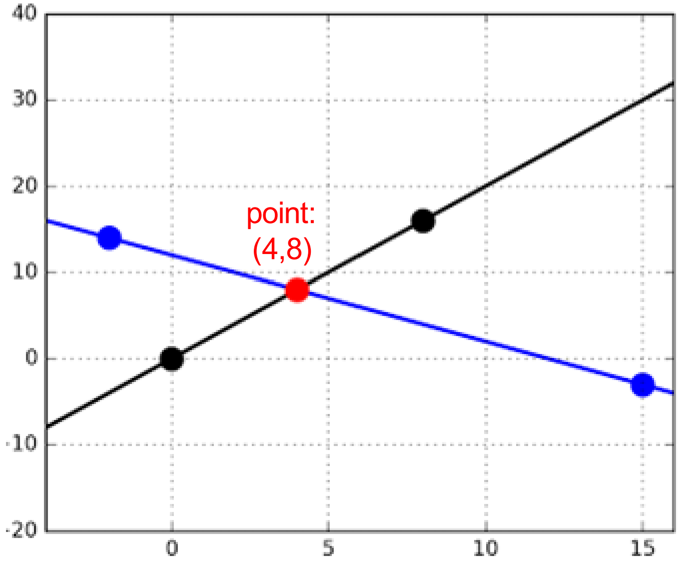


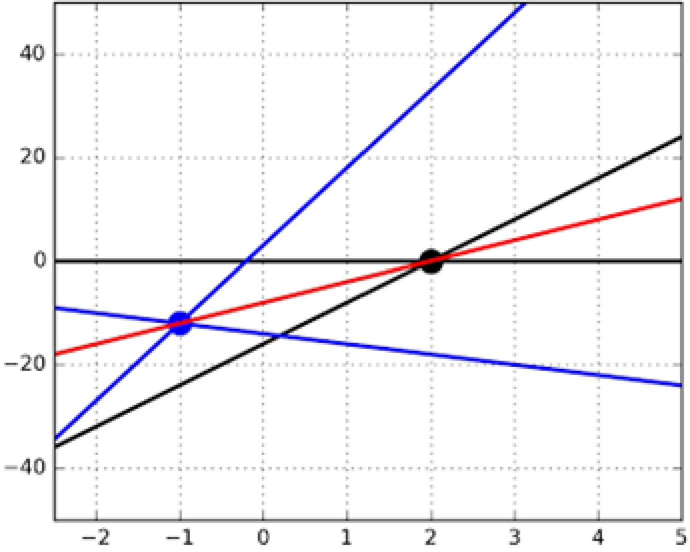
Duality Example

points

linear equations

$$(s,t) \longleftrightarrow y = sx - t$$



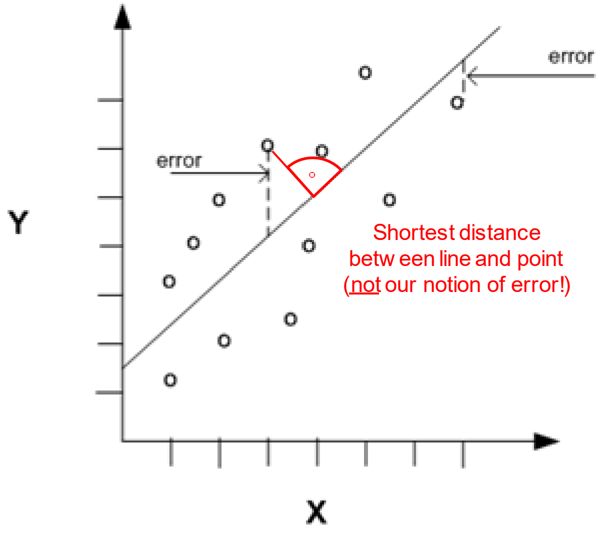


linear equation:
y = 4x – 8

Error in Linear Regression

The residual error is the difference between the predicted and actual values: $r_i = y_i - f(x_i)$

Least squares regression minimizes the sum of the squares of the residuals of all points.
This metric is chosen because (1) it has a nice closed form and (2) it ignores the sign of the errors.



Solving Linear Regression

Consider the $n \times m$ system $Aw=b$. The vector w of coefficients for the best fitting line is given by:

$$w = (A^T A)^{-1} A^T b$$

Product of $((m \times n) \times (n \times m)) \times (m \times n)\ (n \times 1)$ is $m \times 1$

Thus least squares optimization reduces to inversion and multiplying matrices.

Linear Regression in One Variable

We seek the best fitting line l with $y = w_0 + w_1 x$

The slope of this line is:

$$w_1 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = r_{xy} \frac{\sigma_y}{\sigma_x}$$

The intercept follows since l goes through the x-mean and y-mean.

Connections with Correlation

- If x is uncorrelated with y, w_1 should be zero.
- If x,y are perfectly correlated, the slope should depend upon the magnitudes of x,y, as given by their standard deviations.
- The formula $w = (A^T A)^{-1} A^T b$ includes correlation-related terms (covariance matrix of variables, and variables against target)

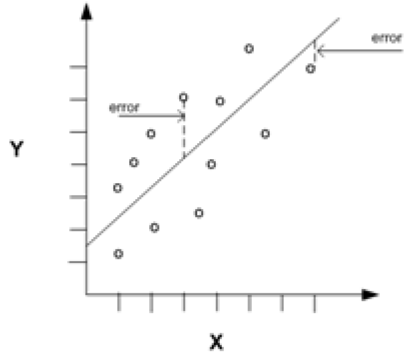
Where Does This Come From?

The error vector *(b-Aw)* must be orthogonal to the vector for each variable, or we could improve the fit by adjusting *w*.

These zero dot products mean $A^T(b - Aw) = 0$

Simple algebra then gives

$$w = (A^T A)^{-1} A^T b$$



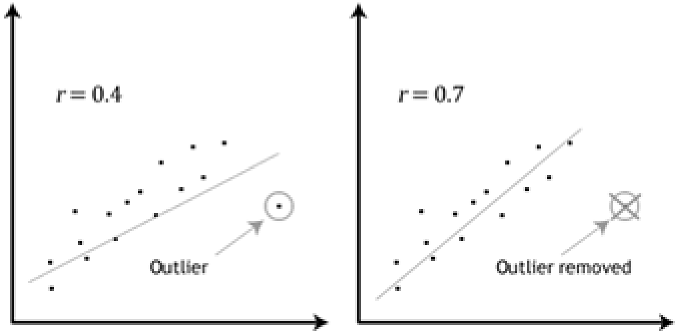
Better Regression Models

Proper treatment of variables yields better models:

- Removing outliers
- Fitting nonlinear functions
- Feature/target scaling
- Collapsing highly correlated variables

Outliers and Linear Regression

Because of the quadratic weight of residuals, outlying points can greatly affect the fit.



Identifying outlying points and removing them in a principled way can yield a more robust fit.

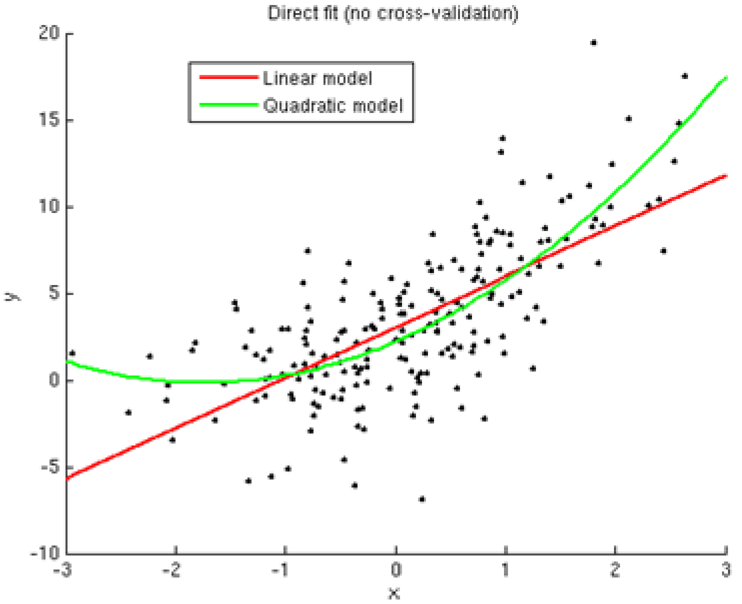
Fitting Non-Linear Functions

Linear regression fits lines, not high-order curves!

But we can fit quadratics by creating another variable with the value x^2 to our data matrix.

We can fit arbitrary polynomials (including square roots) and exponentials/logarithms by explicitly including the component variables in our data matrix: $\sqrt{x}, \log x, x^3, \frac{1}{x}$.

However, explicit inclusion of all possible non-linear terms quickly becomes intractable.



Feature Scaling: Z-scores

Features over wide numerical ranges (say national population vs. fractions) require coefficients over wide scales to bring together.

$$V = c_1 \cdot 300{,}000{,}000 + c_2 \cdot 0.02$$

Fixed learning rates (step size) will over/under shoot over such a range, in gradient descent.

Scale the features in your matrix to Z-scores!

Dominance of Power Law Features

Consider a linear model for years of education, which ranges from 0 to 13+5+5=23.

$$Y = c_1 \cdot income + c_2$$

No such model can gives sensible answers for your parents' kids and Bill Gates' kids.

Z-scores of such power law variables don't help because they are just a linear transformation.

Feature Scaling: Sublinear Functions

An enormous gap between the largest/smallest and median values means no coefficient can use the feature without blowup on big values.

The key is to replace/augment such features *x* with sublinear functions like log(x) and sqrt(x).

Z-scores of these variables will prove much more meaningful.

Small Coefficients Need Small Targets

Trying to predict income from Z-scored variables will *need* large coefficients: how can you get to \$100,000 from functions of -3 to +3?

If your features are normally distributed, you can only do a good job regressing to a similarly distributed target.

Taking logs of big targets can give better models.

Avoid Highly Correlated Features

Suppose you have two perfectly-correlated features (e.g. height in feet, height in meters).

This is confusing (how should weight be distributed between them?) but worse…

The rows in the covariance matrix are dependent ($r_1 = c \cdot r_2$) so $w = (A^T A)^{-1} A^T b$ requires inverting a singular matrix!

Punting Highly Correlated Features

Perfectly correlated features provide no additional information for modeling.

Identify them by computing the covariance matrix: either one can go with little loss.

This motivates the problem of dimension reduction: e.g singular value decomposition, principal component analysis.

Issues with Closed Form Solution

This closed form for linear regression is concise and elegant, but issues include:

- Inversion slow for large systems
- Formulation is brittle: the linear algebra magic is hard to extend to other formulations

This motivates the gradient descent approach to solving regression.

Regression as Parameter Fitting

We seek coefficients that minimize the sum of squared error of the points over all possible coefficients:

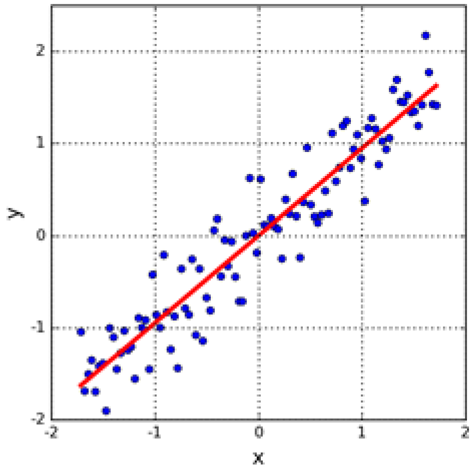
$$J(w_0, w_1) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Here the regression line is:

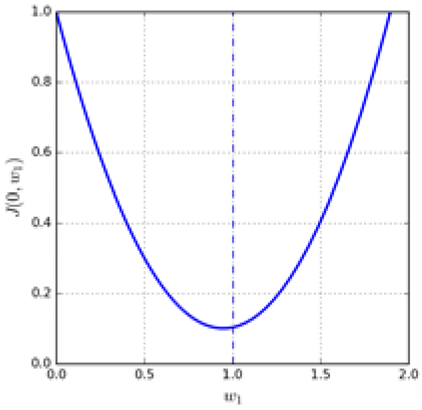
$$f(x) = w_0 + w_1 x$$

Lines in Parameter Space

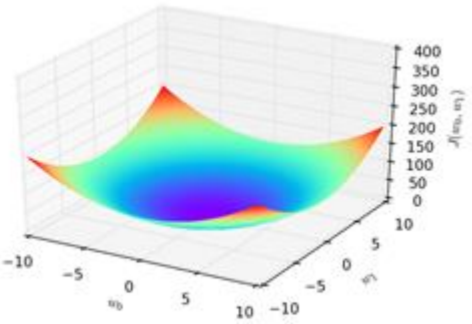
The error function $J(w_0, w_1)$ is convex, making it easy to find the single local/global minimum.



data space



slope space



error surface

Gradient Descent Search

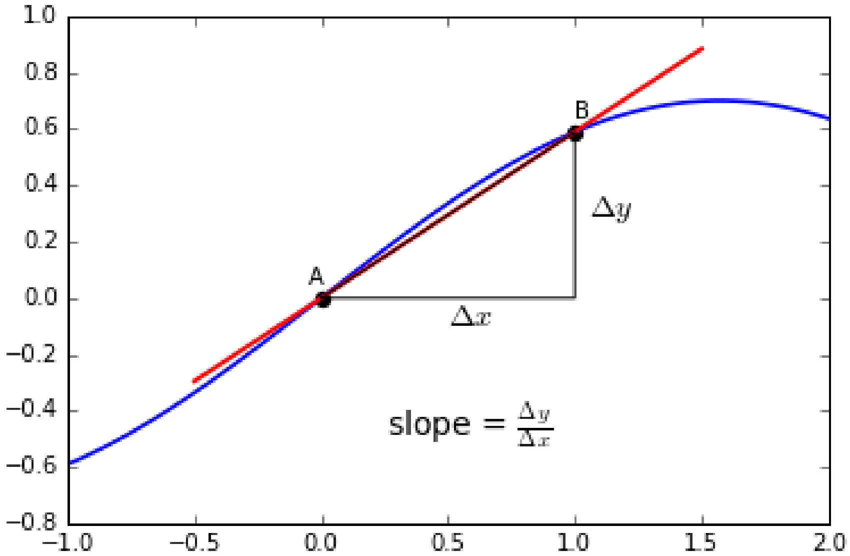
A space with only one local/global minimum is called convex.

When a search space is convex, it is easy to find the minimum: just keep walking down.

The fastest direction down is defined by the slope or tangent at the current point.

The Fastest Way Down

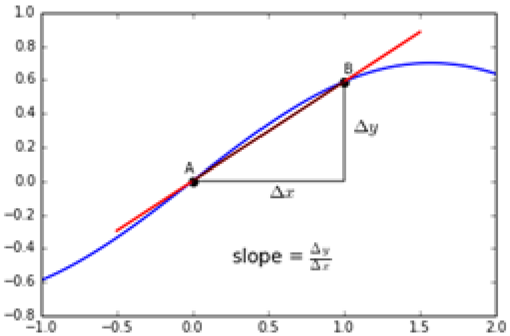
The direction down at a point is given by its derivative, which specified by its tangent line:



The Fastest Way Down

The direction down at a point is given by its derivative, which specified by its tangent line:

This *could* be approximately computed by finding the point $(x + \Delta x, y(x + \Delta x))$ and fitting the line with $(x, y(x))$



Gradient Descent for Regression

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\text{(for } j = 1 \text{ and } j = 0)$$

}

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Which Functions are Convex?

Remember your calculus!

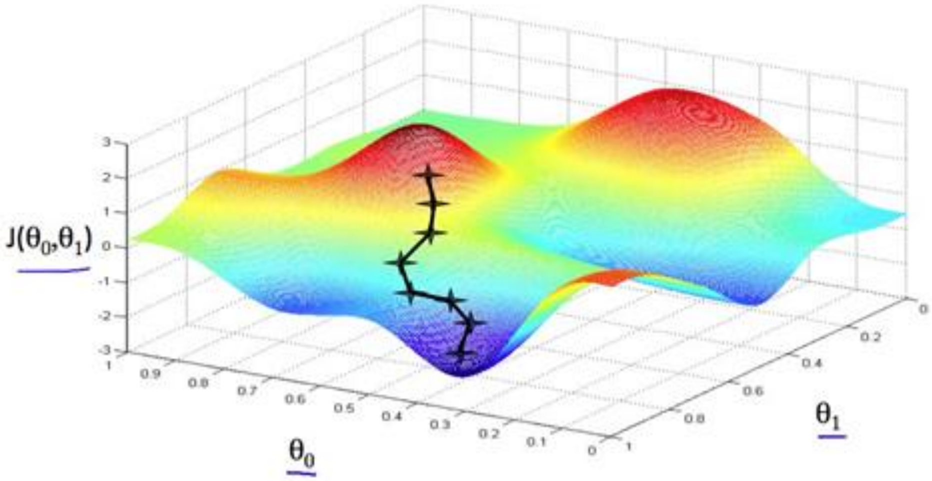
Whenever the first derivative is zero, you get a maximum or minimum.

Thus analysis of such derivatives can tell which functions are and are not convex.

Gradient descent search can get trapped in local minima only for non-convex functions.

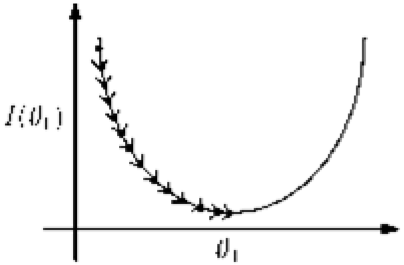
Getting Trapped in Local Optima

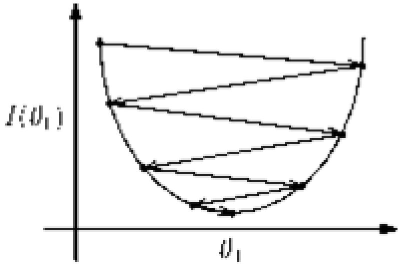
Always going upward does not reach the ski slope from a two story cabin in the valley.



Effect of Learning Rate / Step Size

- Taking too small steps results in slow convergence to the optima.
- But too large a step overshoots the goal.





What is the Right Learning Rate?

Monitor the value of the loss function J(…) over the course of optimization.

If progress is too slow, increase by a multiplicative factor (say 3) or accept.

If J gets larger, the step size is too large, decrease by a multiplicative factor (say ⅓).

Library functions should use algorithms for this.

Stochastic Gradient Descent

Evaluating the partial derivative takes time linear in the number of examples for each step!

A good heuristic is to use only a few examples to estimate the derivative, and hope it is down.

Optimizing the learning rate and the batch size for gradient descent leads to very fast optimization for convex functions.

Too Many Features?

Providing a rich set of features to regression is good, but remember Occam's Razor:

"The simplest explanation is best."

Ideally our regression would select the most important variables and fit them, but our objective function only tries to minimize sum of squares error.

Regularization

The trick is to add terms to the objective function seeking to keep coefficients small:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

We pay a penalty proportional to the sum of squares of the coefficients, thus ignoring sign.

This rewards us for setting coefficients to zero.

Interpreting / Penalizing Coefficients

When variables have mean zero, its coefficient magnitude is a measure of value to the objective function.

Penalizing the sum of squared coefficients is *ridge regression* or *Tikhonov regularization*.

Penalizing the absolute value of the coefficients (L_1 metric vs. L_2) is *LASSO regularization*.

What is the right Lambda?

How do we set the constant lambda:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

Big-enough lambda emphasizes small parameters, i.e. set to all zeros.

Small-enough lambda freely uses all parameters to minimize training error.

We seek balance between under- / overfitting.

Tradeoffs Between Fit / Complexity

A good fit to the training data with few parameters is more robust than a slightly better fit with many parameters.

Metrics to help with model selection include:

- Akaike Information Criteria: $AIC = 2k - 2\ln(L)$
- Baysian Info Critera: $-2 \cdot \ln p(x|M) \approx \mathrm{BIC} = -2 \cdot \ln \hat{L} + k \cdot (\ln(n) - \ln(2\pi)).$

k is a parameter count and L an error metric.

Normal Form with Regularization

The normal form equation can be generalized to deal with regularization…

If $\lambda > 0$,

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

Or we can just use gradient descent with the proper loss function and derivatives.

Wrapup:

Linear Regr. & Gradient Descent

- There are closed form solutions for regression, but they are often intractable in practice
- Gradient descent is optimal for convex problems and still useful for non-convex ones
- Performance can be improved, e.g. by outlier removal, feature scaling or collapsing variables
- Remember: simpler models are better!