Linear Regression

Simon Šuster, University of Groningen

Course *Learning from data*
December 16, 2013

References

- Peter Flach: Machine learning : the art and science of algorithms that make sense of data
- Kevin P. Murphy: Machine learning : a probabilistic perspective
- Hal Daumé III: A Course in Machine Learning `http://ciml.info` (regularization, gradient-based optimization)

Some slides/plots are adapted from Andrew Ng.

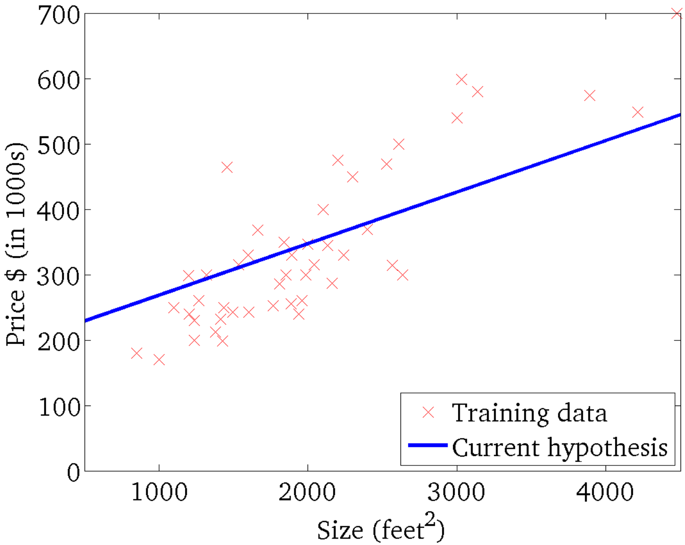
Regression vs. classification

- Predictions in regression are real-valued
  - prices
  - age
  - student success
  - vowel length...
- Supervised; ground-truth is now continuous

Modeling in LR

- Fitting a model to the training data that generalizes well to unseen data
- Hypothesis/model/function
- The model is a function that knows how to map x to y

One-feature example



Note about linearity

- *Linear* regression can model *non-linear* hypothesis!
- Linearity is about how the parameters are combined
- Complex functions can be represented by a linear combination of (expanded) features

Hypothesis

- Parameters (weights) represented by Theta, Θ
- With one feature only:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1$$

  - geometrical interpretation: intercept and slope

Hypothesis

- Parameters (weights) represented by Theta, Θ
- With one feature only:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1$$

  - geometrical interpretation: intercept and slope
- Multiple features:

$$\begin{aligned} h_\Theta(x) &= \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n \\ &\text{for convenience, } x_0 = 1 \\ h_\Theta(x) &= \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n \\ &= \mathbf{\Theta^T x} \end{aligned}$$

(Have we seen similar operations in a previous lecture?)

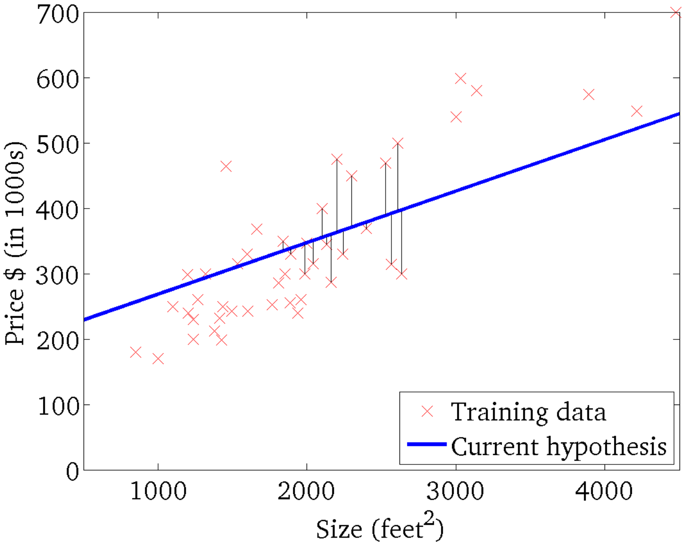
  - example interpretation:
    price $= \Theta_0 + \Theta_1$Size $+ \Theta_2$Age $+ \Theta_3$#Floors...

Cost function J

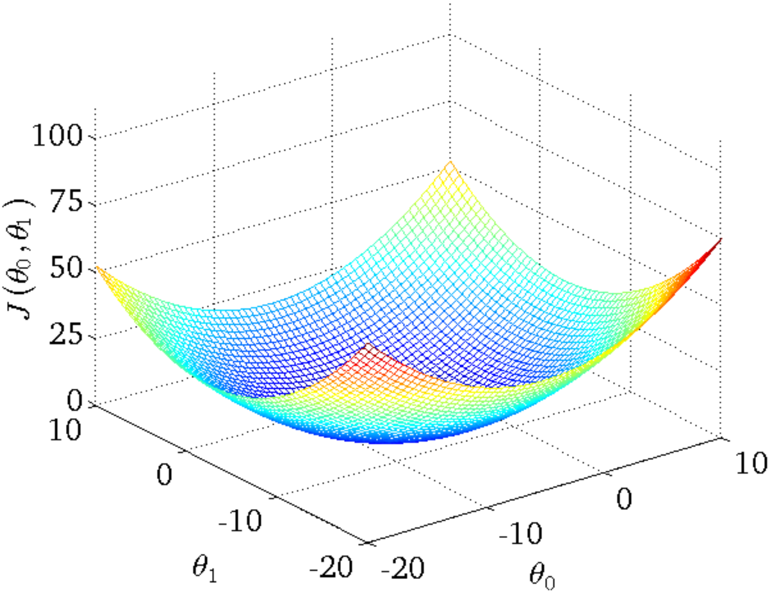
How to fit the best possible model to our training data?

- Find Θs that minimize the cost
- Cost is squared error $\Rightarrow$
- Minimizing squared difference between predicted output and true output $(h_\Theta(x) - y)^2$
- Complete form of J is one half of the average of squared differences over all training instances

Error as vertical lines



Two parameter surface plot for cost function



Minimizing J I

Best parameters are the ones leading to smallest J.

Gradient descent

- Finds (local) minimum of a function
- Cost function J is bowl-shaped (convex)
- GD thus finds the global minimum
- Way of knowing at which point on the function we are/how to get to the minimum?
- Calculate derivative/gradient at that point $\Rightarrow$ slope
- When slope is 0, we've reached the minimum

Minimizing J II

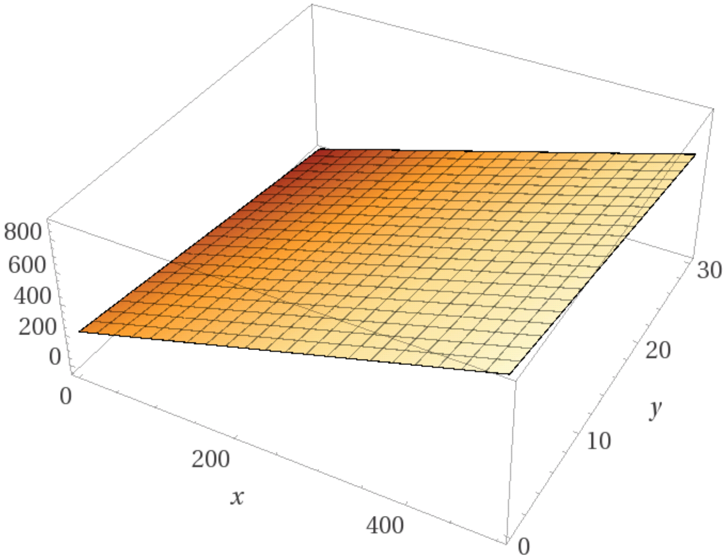
- Start with some parameters Θ
- Repeat until converged*
  - Update parameters with derivatives (gradient) of J for current Θ
    - Must get to the minimum, so *subtract* the derivatives
      (Parameters now better, closer to the minimum)

Example on blackboard

- Suppose we only have Θ_1
- Compute derivative of $J(\Theta_1)$
  $\Rightarrow \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)}) x^{(i)}$
- Update Θ_1

*derivative small; J not changing sufficiently

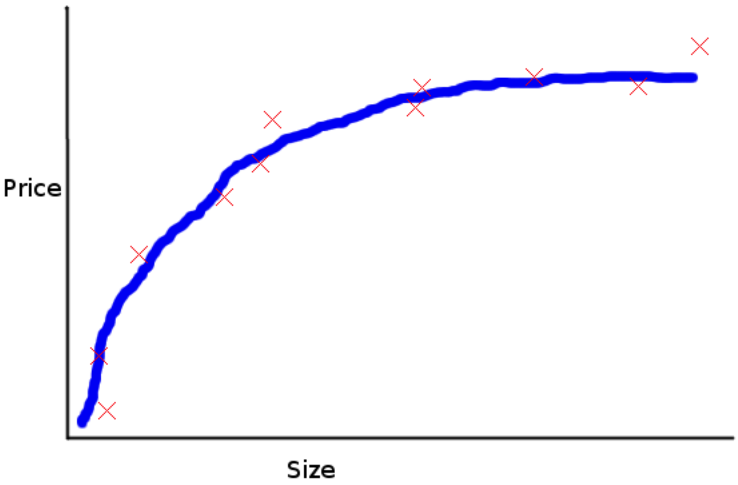
Blackboard example prediction



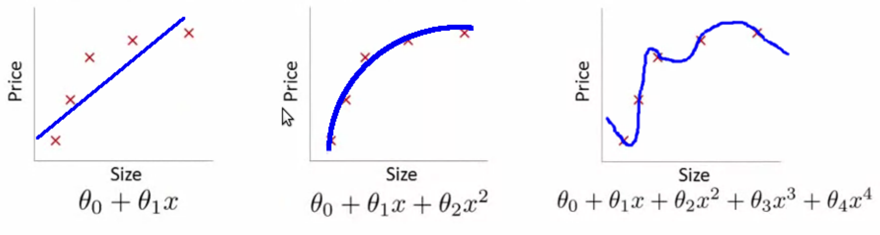
plot 204 + 1.4x − 8.4y, x=0..500, y=0..30 | Computed by Wolfram|Alpha

Expanded features

- Allow modeling of non-linear relationships
- Including polynomial terms (e.g. x^2, x^3)



Fit and complexity of hypothesis



Under-/Overfitting I

- With many features, can't select the order of feature complexity by visualizing
- Having a lot of features, not so much training data

Overfitting

- trying to fit the training data to closely
- solution not general enough to be applied successfully to unseen data

Solutions

- Remove some features $\Rightarrow$ Potentially harmful
- Better keep features but reduce values of parameters

Under-/Overfitting II

- Reduced values mean smoother, simpler functions
- Less overfitting
- Can think of it as penalizing solutions we want to discourage

- One type of regularization:
  - add *sum of squared parameters* to cost J
    - control how much to add (penalize) by a value λ
    - when λ is very small/zero $\Rightarrow$ no regularization (more likely to overfit)
    - high $\lambda \Rightarrow$ prefer simpler models (more likely to underfit)
  - Called "ridge regression"

Conclusion

- This was a brief introduction
- Many ways of optimization exist
- Closed form solutions to find parameters
- Different regularization techniques

- Weka includes ridge regression

Nominal features (and Weka)

Suggestion:

- Convert nominal feature with *n* levels to *n* binary features
- Example in housing price prediction: “neighborhood” as 1 feature needs converting to features for each neighborhood
- Weka: use *un*supervised NominalToBinary filter
- (Weka can also do automatic supervised NominalToBinary conversion which is less intuitive to interpret)

Final project

- Predicting opening-weekend revenue for movies from critic reviews
- Meta-information and reviews
- Dataset: `www.ark.cs.cmu.edu/movie$-data/`
  - Allowed to use train+dev
  - Reviews in `/net/shared/simsuster/movies-data-v1.0/7domains-train-dev.tl/` are or will be:
    - segmented with Splitta
    - POS-tagged with Citar
    - Dependency parsed with MSTParser
- Concentrate on predicting overall revenue, not per screen
- Joshi et al.: "Movie Reviews and Revenues: An Experiment in Text Regression"