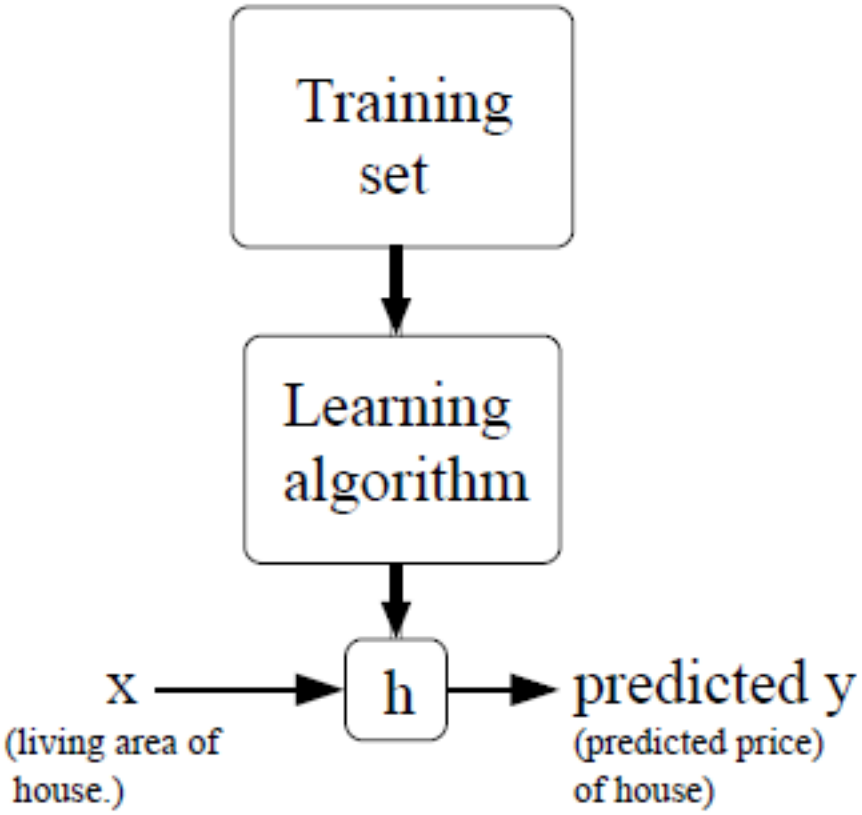
Linear Regression

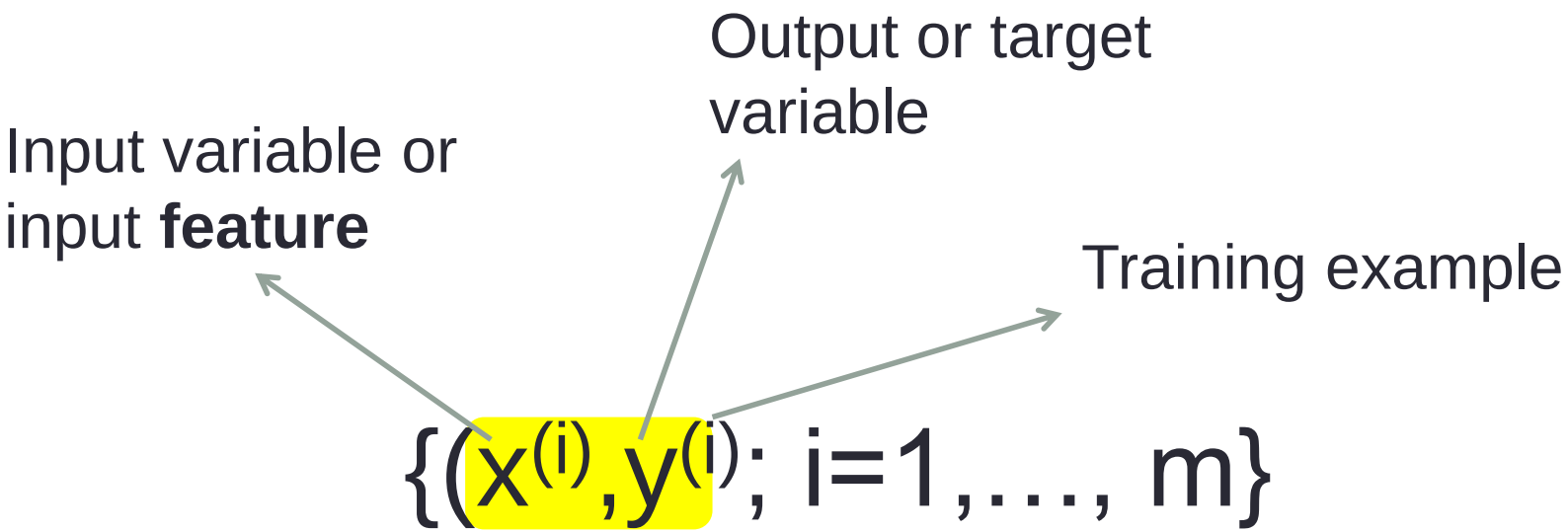
Supervised Learning

Supervised Learning

- To learn a function, *h* so that h(x) is a good predictor of the corresponding value of y
- **Regression**: Target variable is **continuous**



Training Set



- Training set: m training examples
- Superscript (i) is an index. Not exponentiation.

Linear Regression

- Given dataset on living areas and prices of 47 houses from Portland, Oregon, how can we learn to predict the prices of other houses in Portland as a function of size of their living areas?

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

Represent function *h*

- We may decide to approximate y, represented as *h,* as a linear function of x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Or simply, when $x_0 = 1$ (intercept term)

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

$\theta_{(i)}$ = parameters (or weights)

How to learn parameters θ?

- Define the cost function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

- Least Mean Squares Algorithm
  - Start with an initial guess for θ
  - Repeatedly change θ to make J(θ) smaller, until converge to a value of θ that minimizes J(θ)
  - E.g. Gradient descent algorithm

Gradient Descent Algorithm

- Algorithm takes repeated steps in the direction of steepest decrease in J

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

α = learning rate

- For a single training example,

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}.$$

Error term

- This is called the LMS update rule, also known as Widrow-Hoff learning rule

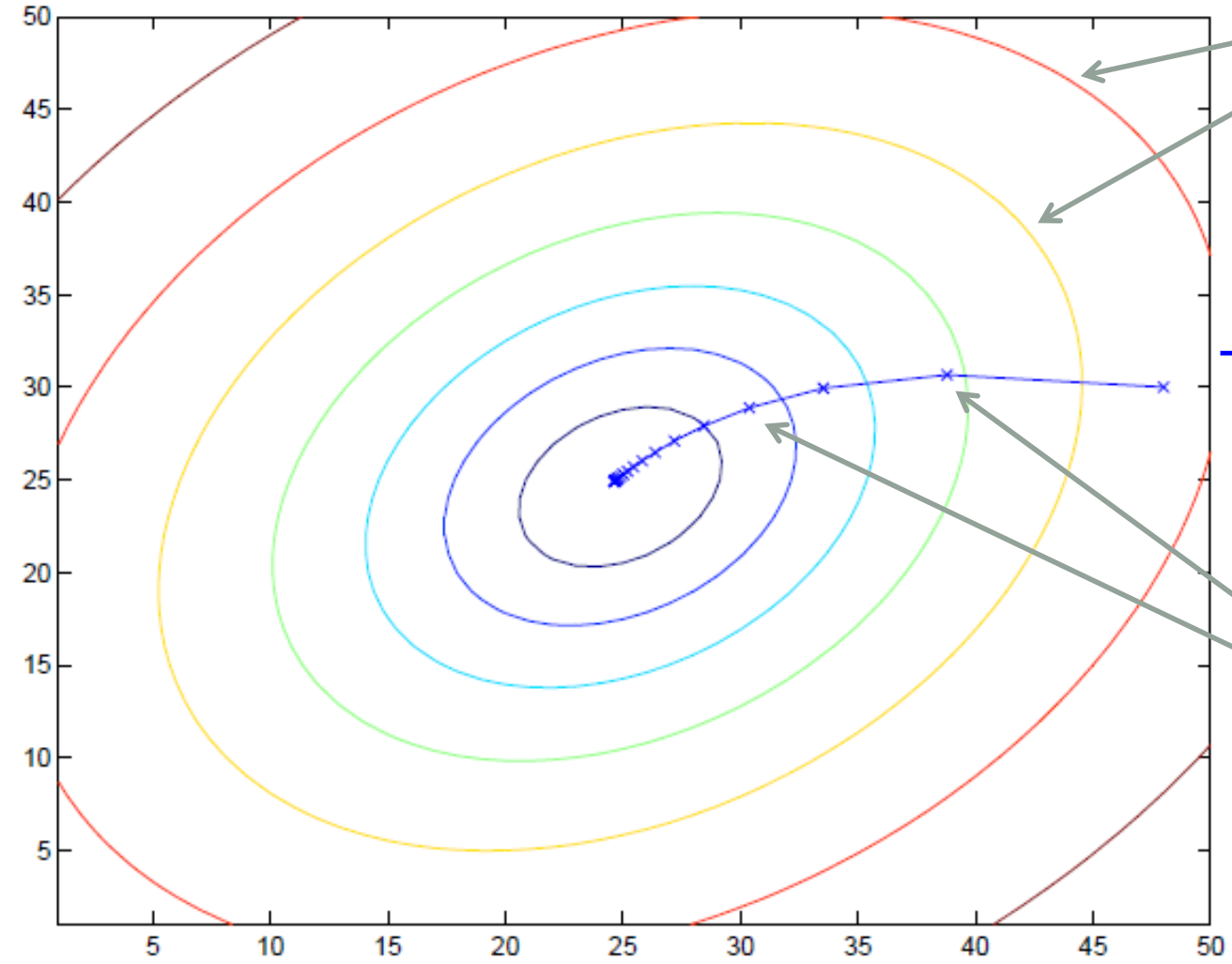
Batch Gradient Descent

- We modify the LMS rule, for a training set of more than one example:

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad (\text{for every } j).$$

- Looks at every example in the entire training set on every step

Gradient Descent

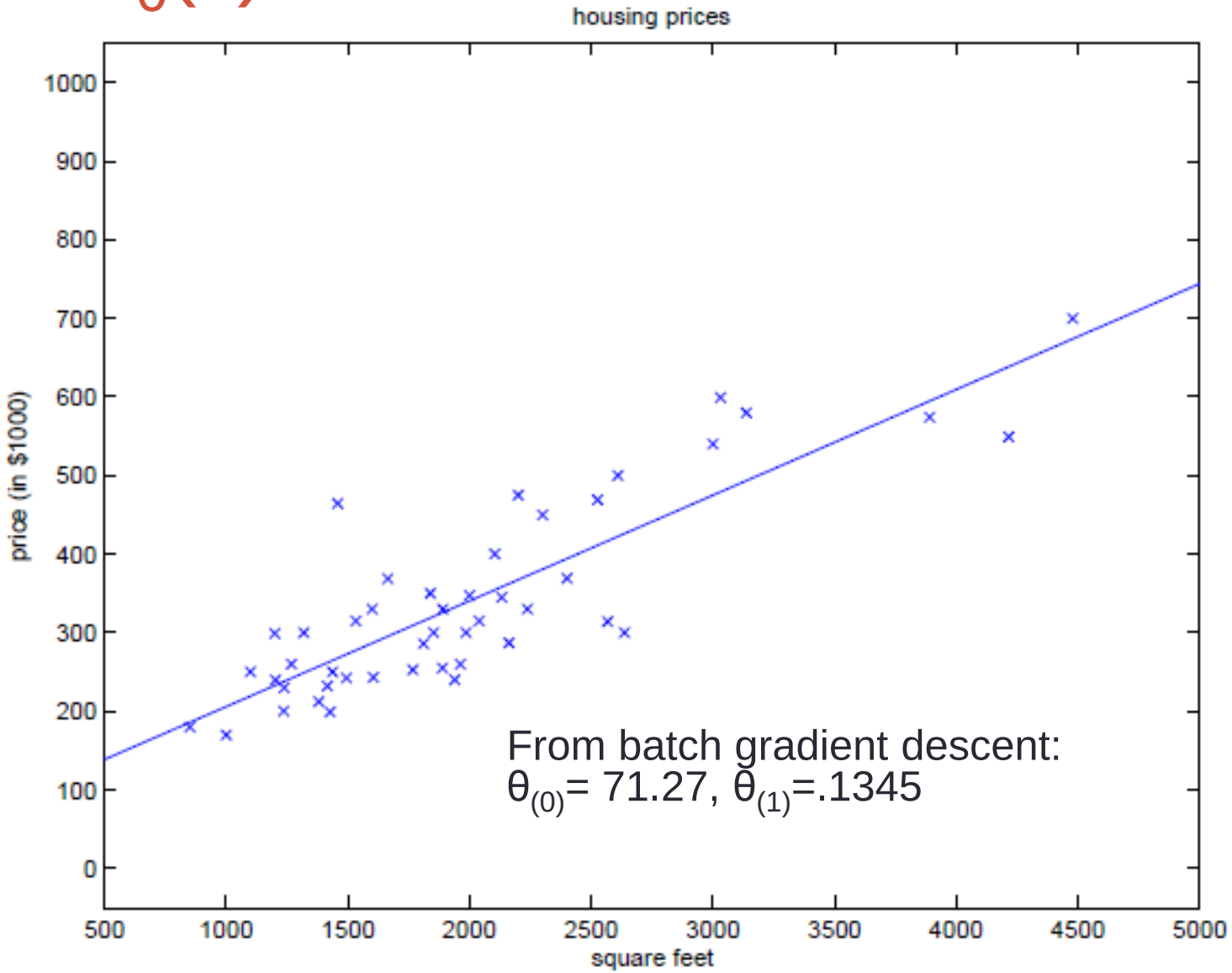


Contours of quadratic function

Trajectory taken by gradient descent

Successive values of θ

Plot $h_\theta(x)$ as a function of x



Other methods

- Stochastic gradient descent (incremental gradient descent)
  - More efficient for large training sets
  - θ oscillates around minimum of J(θ) rather than converging to minimum
- The Normal Equations
  - Minimize J by explicitly deriving J with respect to the θ's and setting them to 0
  - No iterative algorithm

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

Why LS-cost function?

- LS regression corresponds to finding the maximum likelihood estimate of θ

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

$\varepsilon^{(i)}$ = unmodeled effects or random noise

- Probabilistic assumptions:
  - $\varepsilon^{(i)}$ are independently and identically distributed Gaussian distributions
  - $\varepsilon^{(i)} \sim N(0, \sigma^2)$

Why LS-cost function?

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right).$$

$$\mathbf{y^{(i)}|x^{(i)};\ \theta \sim N(\theta^T x^{(i)},\ \sigma^2)}$$

- Likelihood function:

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X;\theta).$$

Why LS-cost function?

- Independence assumption of $\varepsilon^{(i)}$'s, hence

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2} \right).
\end{aligned}
$$

- Principal of Maximum Likelihood: choose θ to maximize L(θ)

Why LS-cost function?

- Maximize the log likelihood

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2.
\end{aligned}
$$

- Same answer as minimizing $\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2$

Why LS-cost function?

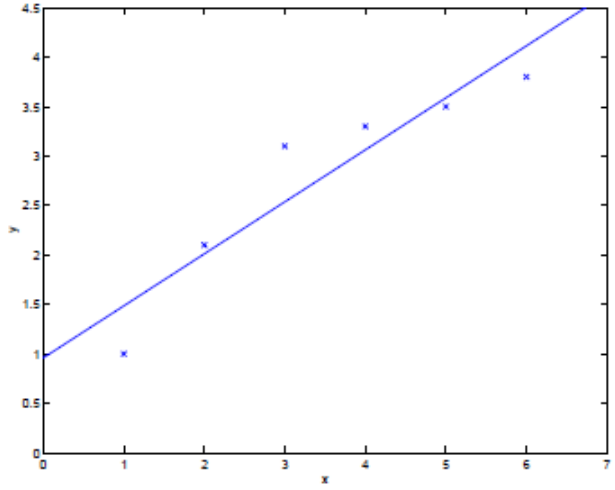
Minimizing $\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2$

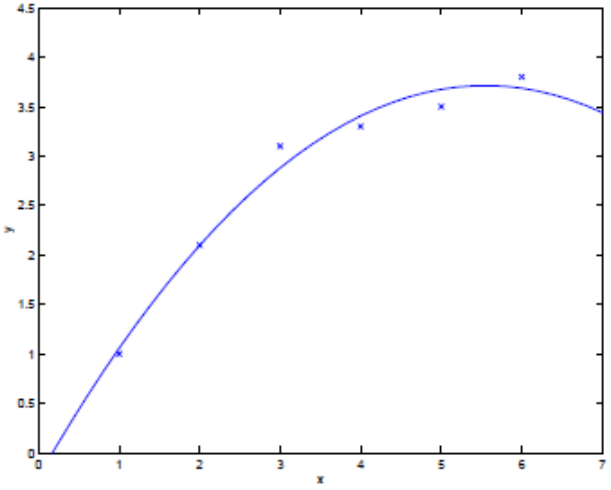
…

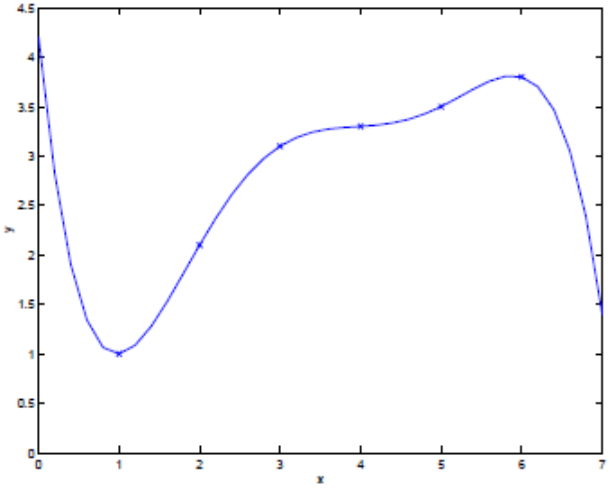
…is J(θ), the original least-squares cost function !

To summarize: LS regression can be justified because it is just doing maximum likelihood estimation

Which features to include?







Locally Weighted Linear Regression:

- Non-parametric algorithm
- Makes choice of which features are less critical

Confidence Intervals

- 95% CI: There is a 95% probability that the true best-fit line for the population lies within the confidence interval.
- Confidence interval ≠ prediction interval