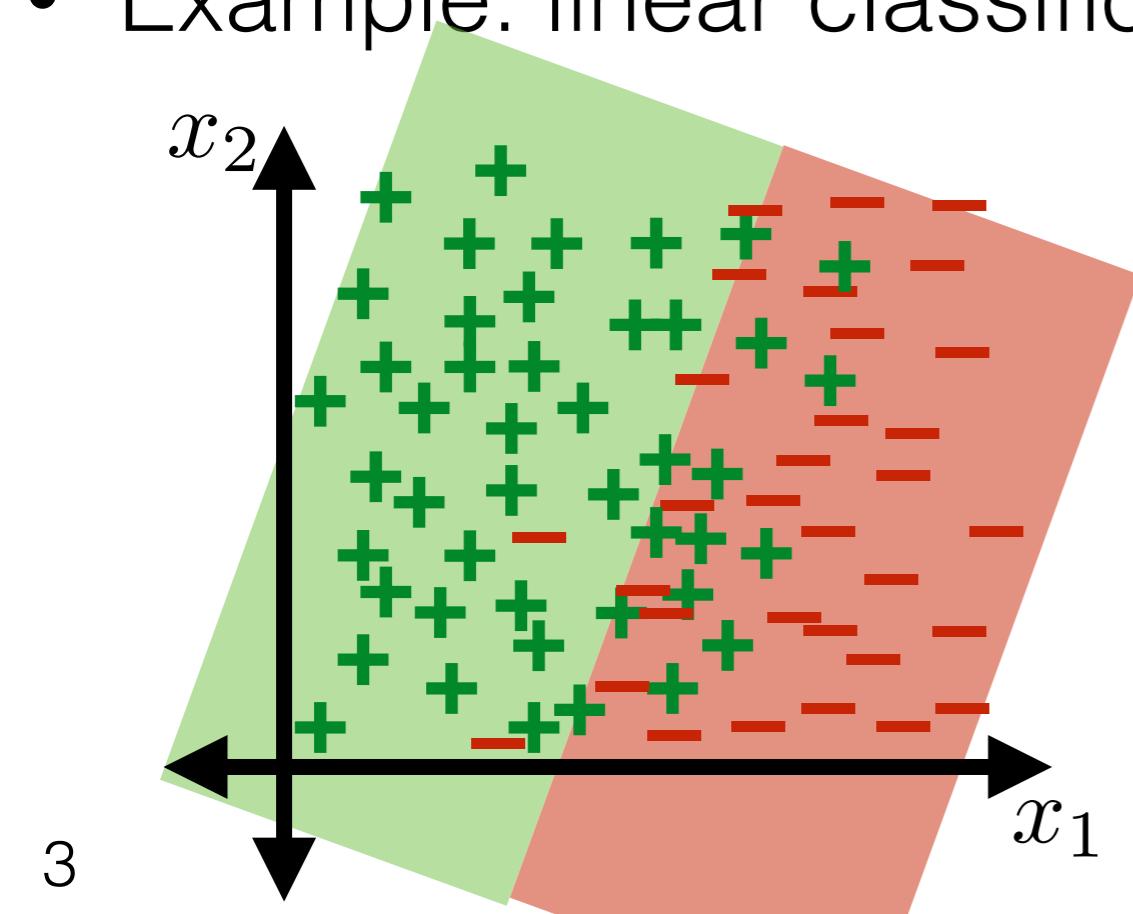


Recall

Classification

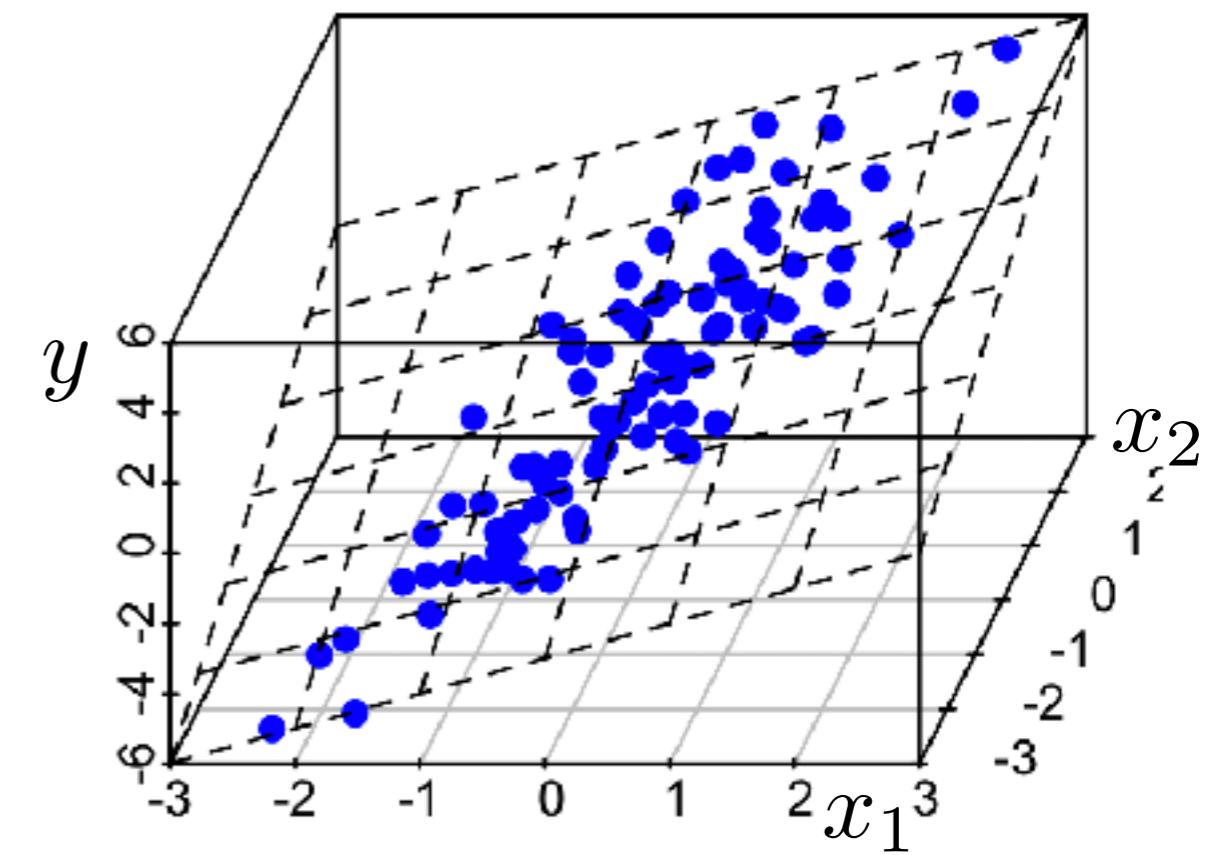
- Datum i : feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
- Label $y^{(i)} \in \{-1, +1\}$
- Hypothesis $h : \mathbb{R}^d \rightarrow \{-1, +1\}$
- Loss: 0-1, asymmetric, NLL
- Example: linear classification



Compare

Regression

- Datum i : feature vector $x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})^\top \in \mathbb{R}^d$
- Label $y^{(i)} \in \mathbb{R}$
- Hypothesis $h : \mathbb{R}^d \rightarrow \mathbb{R}$
- Loss: $L(g, a) = (g - a)^2$
- Example: linear regression



5

* Comparison between Regression & Classification (Slide - 2 se hont)

- In regression, the labels aren't binary or ternary or anything like that. We map the features to a continuous entity like for example the Real line.
- Generally, it's \mathbb{R} maps to \mathbb{R} (features & labels $\in \mathbb{R}$)
- Many types of losses are used in regression analysis. But the 'squared error loss' is of focus in this study.

Linear regression

- Hypotheses for linear regression: $h(x; \theta, \theta_0) = \theta^\top x + \theta_0$
- Training error (using squared error loss)

$$J(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^n L(h(x^{(i)}; \theta, \theta_0), y^{(i)}) = \frac{1}{n} \sum_{i=1}^n (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2$$

- Training error if we augment features with feature “1”

$$\begin{aligned} J(\theta) &= \frac{1}{n} \sum_{i=1}^n (\theta^\top x^{(i)} - y^{(i)})^2 = \frac{1}{n} \sum_{i=1}^n ((\underbrace{x^{(i)}}_{1 \times d, dx1})^\top \underbrace{\theta}_{1 \times n} - y^{(i)})^2 \\ &= \frac{1}{n} \|\tilde{X}\theta - \tilde{Y}\|^2 = \frac{1}{n} (\underbrace{\tilde{X}\theta - \tilde{Y}}_{1 \times n})^\top (\underbrace{\tilde{X}\theta - \tilde{Y}}_{n \times 1}) \end{aligned}$$

Define $\tilde{X} = \begin{bmatrix} x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix}$

$\tilde{Y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

* Linear Regression (Slide - 4 - 10)

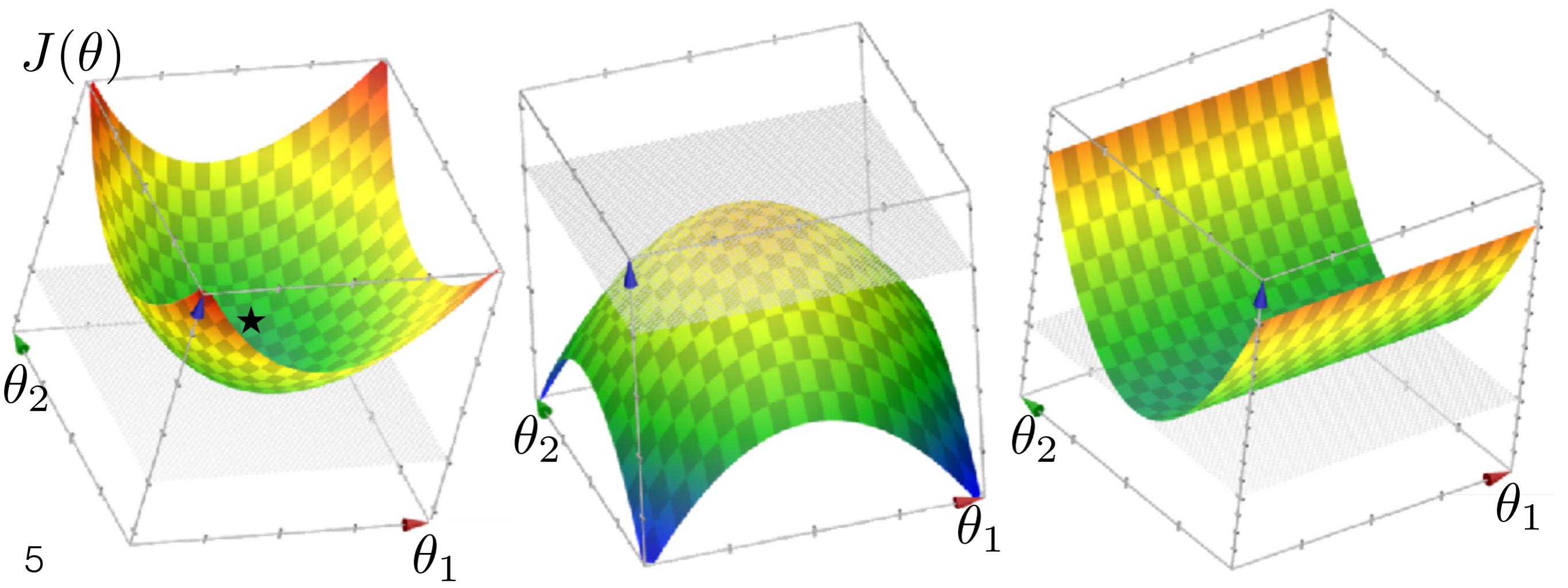
- Here instead of classifying data based on value of $\theta^T x + \theta_0$, we actually use it's value as our label.
- And we measure our loss by taking difference between the guess & actual value, & averaging their squares.
- In the loss function, for ease of functionality, we expand the feature vector by 1 & add value '1' to

all data points' last feature. (The extra feature) & add 0. to the last feature (extra feature).

- We formulate those equations for all data points into matrices. So now, we can say that we are just taking the magnitude squared of the resultant vector for our loss.

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]



* Linear Regression (slide - 5 -) (A direct solution)

- In this case, where we have to minimize the loss function, we can directly minimize it without using numerical methods or gradient descents or stuff because if you ~~look~~ look carefully & expand the loss function, it's a simple paraboloid. So, it has only 1 global minimum (which is unique).
- Now, if you invert it (multiply loss function by -1), you would see that there is no global minimum.
- Another example would be the 3rd one given in the slide where there exists a global minimum, it's just not unique. A line defines the global minimum instead of a single point in this case.
- In linear algebra, if the matrix of second derivatives is positive definite, then the function has unique minima & it "curves up".

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]
- Gradient $\nabla_{\theta} J(\theta) = \frac{2}{n} \tilde{X}^\top (\tilde{X}\theta - \tilde{Y})$ set 0

Exercise:
check the
vector
elements

$$\frac{2}{n} \tilde{X}^\top (\tilde{X}\theta - \tilde{Y})$$

$n \times n$ $n \times d$, $d \times 1$ $n \times 1$

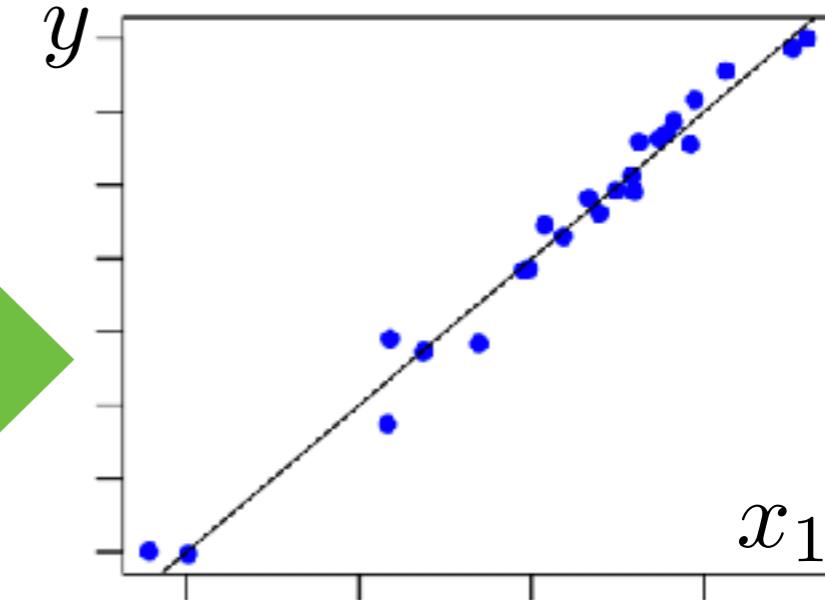
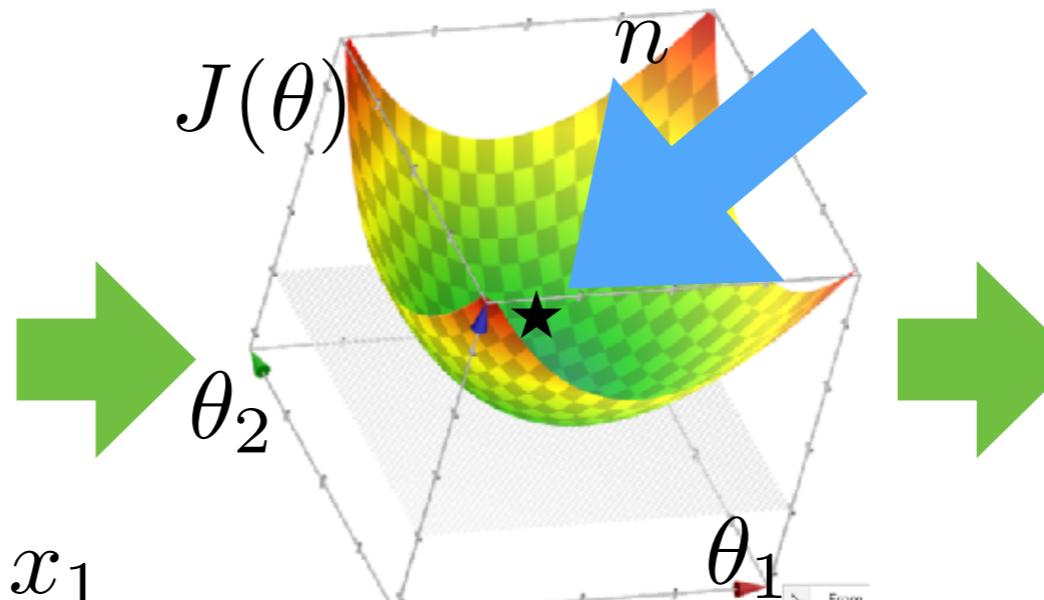
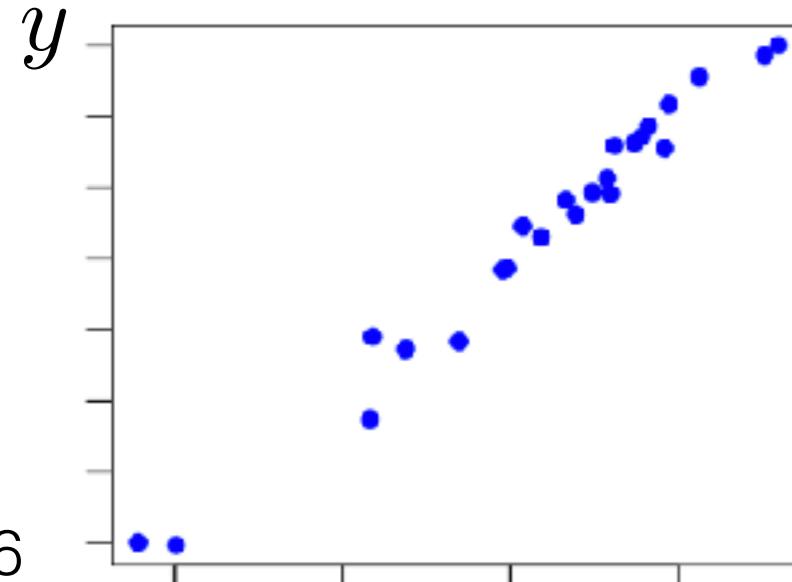
$$\tilde{X}^\top \tilde{X}\theta - \tilde{X}^\top \tilde{Y} = 0$$

$$(\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{X}\theta = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{Y}$$

$$\theta = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{Y}$$

- Matrix of second derivatives

$$\frac{2}{n} \tilde{X}^\top \tilde{X}$$



Exercise:
check $n, d=1$

* Linear Regression (slide = 6-10-94)

(A direct Solution)

- See matrix derivatives to understand how we get $\nabla_{\theta} J(\theta)$
- Function "curves up" \rightarrow Matrix of 2nd derivatives is invertible

- Here, we have only considered a feature vector of only 1 feature. But, loss function $J(\theta)$ is plotted in 3-D because we have assumed θ_0 to be the 2nd dimension (though θ_0 is already included in the θ vector)

Linear regression: A Direct Solution

- Goal: minimize $J(\theta) = \frac{1}{n}(\tilde{X}\theta - \tilde{Y})^\top(\tilde{X}\theta - \tilde{Y})$
- Uniquely minimized at a point if gradient at that point is zero and function “curves up” [see linear algebra]
- Gradient $\nabla_{\theta} J(\theta) = \frac{2}{n} \tilde{X}^\top (\tilde{X}\theta - \tilde{Y})$ set $= 0$

Exercise:
check the
vector
elements

$$\frac{2}{n} \tilde{X}^\top (\tilde{X}\theta - \tilde{Y})$$

n dxn n xd, dx1 n x1

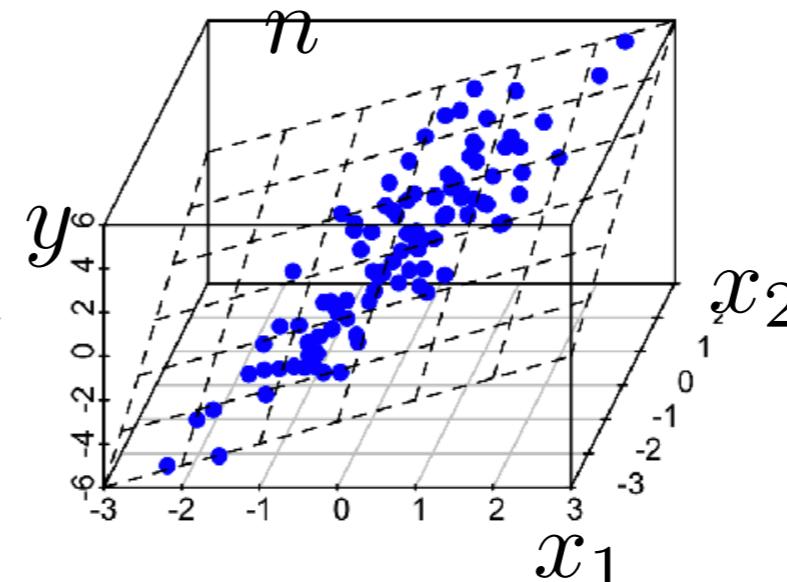
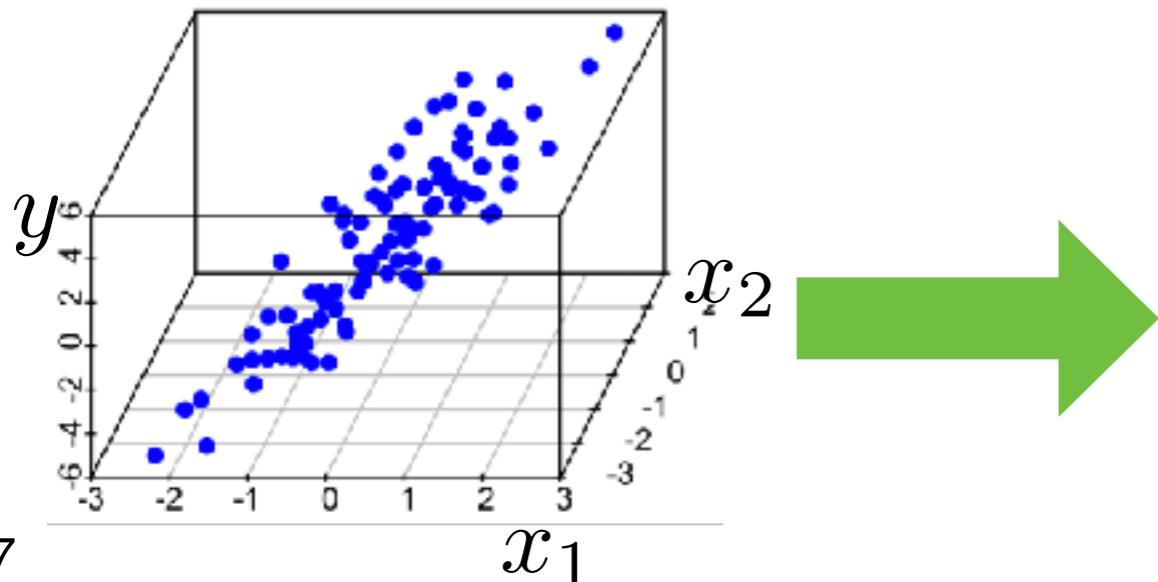
$$\tilde{X}^\top \tilde{X}\theta - \tilde{X}^\top \tilde{Y} = 0$$

$$(\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{X}\theta = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{Y}$$

$$\theta = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{Y}$$

$$\frac{2}{n} \tilde{X}^\top \tilde{X}$$

- Matrix of second derivatives



Exercise:
check $n, d=1$

Note:
hypothesis is
a hyperplane!

* Linear Regression (Slide - 7-199) (A direct solution)

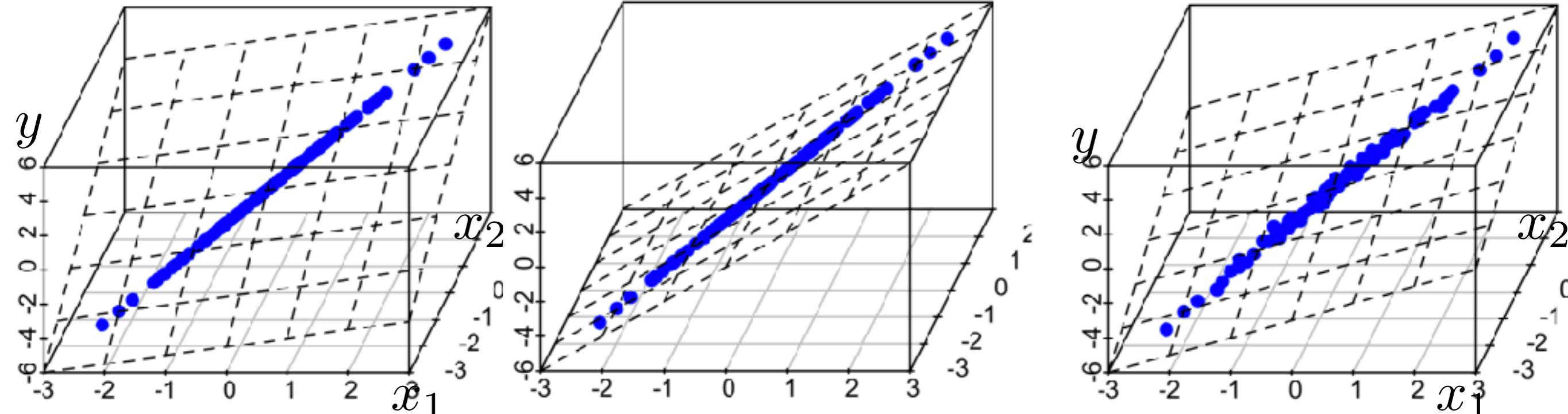
- Now, if we consider a 2 dimensional feature vector (excluding θ_0), we would get a 4-D graph of $J(\theta)$ which we can't draw.

Note that the hypothesis is a hyperplane. That is always linear for linear regression & classification.

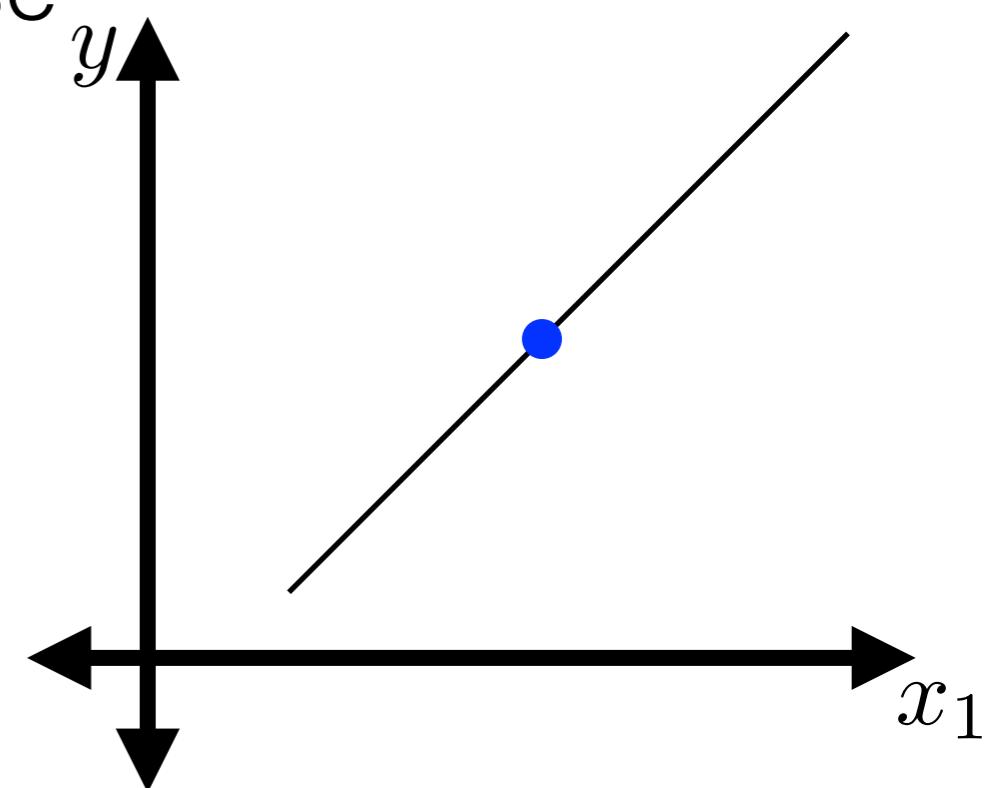
- Note:- θ is always perpendicular to the hyperplane. It's the normal vector of the hyperplane.

What can go wrong in practice?

- Sometimes there isn't a unique best hyperplane



- Sometimes there's technically a unique best hyperplane, but just because of noise
- Feature encodings (cf. one-hot); real-life features often correlated; many feature dimensions
- How to choose among planes? Preference for θ components being near zero



* What can go wrong in practice? (Slide - 8 - 119)

- Sometimes there isn't a unique hyperplane:- Imagine a case where you have 2-D Feature vectors. So, if we

Plot $\theta^T x + \theta_0$ against (θ_1, θ_2) , we will get a 3-D plot. Now, ideally for 2-D feature vectors, there might be a best fitting hyperplane for which all the data points are as co-planar as possible. That will be our required hyperplane.

But, imagine that the data points fit a line in 3-D instead of a plane. Then you can have infinitely many hyperplanes that contain that fitting line.

- Sometimes, there's technically a unique best hyperplane but just because of noise:-

Consider the above example, where the data fits a line in 3-D instead of a plane. Thus, there might be infinite many planes that satisfy our requirements / model.

But consider this: our data is very slightly noisy & this noise is spread out relatively more in one particular plane. Thus, we will get a hyperplane which is unique, even though the data seems to fit a line in 3-D.

- Feature encodings (one-hot); real life features often co-related; many features dimensions :-

It can happen that one or multiple of your features are functions of other features. i.e. they are correlated, hence there is redundancy & there probably won't be a unique hyperplane.

Another thing that can happen is you have equal no. of features and data points. For example, a 1-D feature with only one data point will have infinite lines (hyperplanes) passing through it. A $\neq 2$ -D feature with only two data points will have infinite planes (hyper-

(planes) passing through it. (Because you need at least 3 points to define a plane). Now, this doesn't happen for lower dimension feature vectors. But imagine this: you have higher no. of data points than raw features (schema). But like you saw with the example of one-hot encoding, you could get many features (extra dimensions) from just one type of feature. Hence, the features (extra dimensions) can blow up. So, if you get no. of features = no. of data points, you won't get unique hyperplanes.

How to choose among planes? - Preference for θ components

Imagine a situation like mentioned in above points where you have multiple hyperplanes as solutions & you have to choose from one of them. Every one has the same loss. Or in case of rather noisy data, some hyperplanes are just slightly better than the others (But you shouldn't discard others because the slightly better planes are result of noise & not actual good reasons).

Now, we have earlier studied that higher θ values imply higher confidence. But in this case, higher θ values also imply that some features may have higher influence than others. But even though some θ s have high value & some have low value, their loss is same or almost the same.

So, ideally you should choose θ with smaller values, so that you don't show ~~bias~~ illogical bias between features and their values.

High $\theta \Rightarrow$ high confidence (But we don't have any reason for this high confidence/bias).

Hence, we opt for lower θ values (hyperplanes with
lower θ values)

Regularizing linear regression

- Linear regression with square penalty: ridge regression

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^n (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda \|\theta\|^2 \quad (\lambda > 0)$$

- Special case: ridge regression with no offset

$$\begin{aligned} J_{\text{ridge}}(\theta) &= \frac{1}{n} (\tilde{X}\theta - \tilde{Y})^\top (\tilde{X}\theta - \tilde{Y}) + \lambda \|\theta\|^2 \\ &= \frac{1}{n} \|\tilde{X}\theta - \tilde{Y}\|^2 + \lambda \|\theta\|^2 \end{aligned}$$

What happens if $\lambda < 0$?

- Min at: $\nabla_\theta J_{\text{ridge}}(\theta) = 0 \Rightarrow \theta = (\tilde{X}^\top \tilde{X} + n\lambda I)^{-1} \tilde{X}^\top \tilde{Y}$
 - Matrix of second derivatives: $\tilde{X}^\top \tilde{X} + n\lambda I$ (always “curves up” & invertible when $\lambda > 0$)
- Can also solve for minimizing parameters in case with offset; just a bit more math

* Regularizing Linear Regression (slide- 9- 138)

- One way we can regularize linear regression is adding a penalty for higher θ s
- When we use $\|\theta\|^2$ penalty or L-2 regularizer, it's called ridge regression.

Some notes on features

- Linear regression with square penalty: ridge regression
- $$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^n (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda \|\theta\|^2 \quad (\lambda > 0)$$
- Assumption: features on same scale (cf. standardization)
 - Featurization still matters! (ridge or “ordinary” regression)

The screenshot shows a news article from The New York Times and a sidebar from CEPR. The main headline reads "A Bitter Election. Accusations of Fraud. And Now Second Thoughts". Below it, a snippet discusses Bolivian election data. The CEPR sidebar has a menu, a search icon, and a link to "Major Coding Error Reveals Another Fatal Flaw in OAS Analysis of Bolivia's 2019 Elections".

A close look at Bolivian election data suggests by the O.A.S. that raised questions of vote-rig force out a president — was flawed.

- Can never take a data set blindly
- Share code/data

“time stamps were sorted alphanumerically, instead of chronologically”

[\[https://cepr.net/press-release/major-coding-error-reveals-another-fatal-flaw-in-oas-analysis-of-boliviass-2019-elections/\]](https://cepr.net/press-release/major-coding-error-reveals-another-fatal-flaw-in-oas-analysis-of-boliviass-2019-elections/)

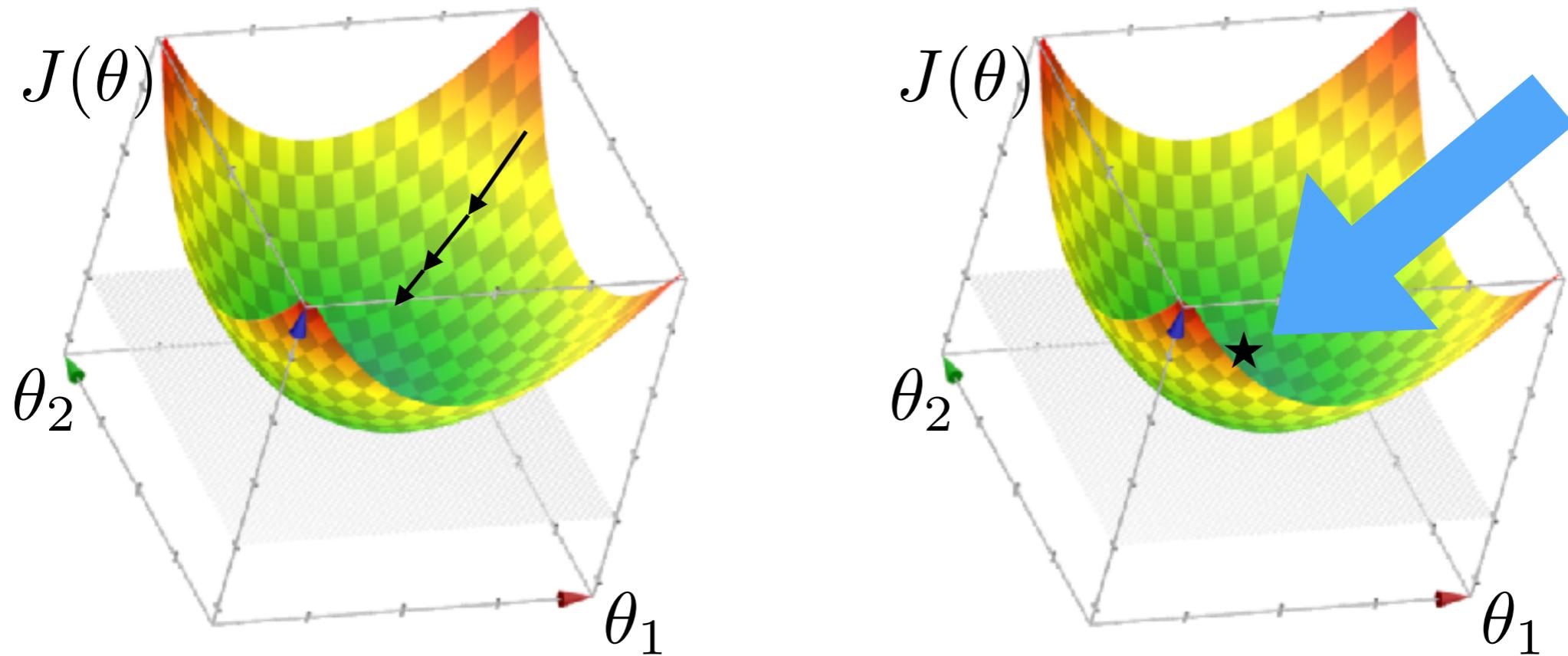
* Some notes on features

(Slide- 10- Last)

- Note that when we use a ~~linear~~ regularizer in the loss function, we assume that all the features in the feature vector are all on the same scale.
- If they are not, the features won't be equally penalized. We need equal penalization.

Optimizing linear regression

- Gradient descent vs. analytical/closed-form/direct solution



- Accuracy doesn't mean anything without running time
- Running time doesn't mean anything without accuracy
- Need to measure accuracy for the running time we have

$$\theta = \underbrace{(\tilde{X}^\top \tilde{X} + n\lambda I)^{-1}}_{d \times d} \tilde{X}^\top \tilde{Y}$$

Matrix inversion: $O(d^3)$

* Optimizing Linear regression (slide - 11 - Last)

- Note that the direct solution was an analytical solution. For complex problems, we can't use or rather can't get an analytical solution. Hence, we go for numerical solutions.
- Note that irrespective of which solution we choose, we also have to consider the running time & computational power required. For example, we can always reduce the

step size parameter in gradient descent to as small as we want, but that would require more computational power & more no. of steps. This will result in very high running times which may be impractical. So, you have to compromise between accuracy & running time.

- Now, consider the above example where we have an analytical solution. We just have to calculate the inverse & multiply it by the remaining matrices in the equation. But (and it's a big but), the time complexity for computing matrix inverse of $d \times d$ dimensions is $O(d^3)$ which is really bad. So, we may prefer gradient descent when there are very high no. of dimensions.

Gradient descent for linear regression

LinearRegression-Gradient-Descent ($\theta_{\text{init}}, \theta_{0,\text{init}}, \eta, T$)

Initialize $\theta^{(0)} = \theta_{\text{init}}$

Initialize $\theta_0^{(0)} = \theta_{0,\text{init}}$

for $t = 1$ **to** T

Exactly gradient descent
with f given by linear
regression objective

$$\theta^{(t)} = \theta^{(t-1)} - \eta \left\{ \frac{2}{n} \sum_{i=1}^n [\theta^{(t-1)\top} x^{(i)} + \theta_0^{(t-1)} - y^{(i)}] x^{(i)} + 2\lambda \theta^{(t-1)} \right\}$$

$$\theta_0^{(t)} = \theta_0^{(t-1)} - \eta \left\{ \frac{2}{n} \sum_{i=1}^n [\theta^{(t-1)\top} x^{(i)} + \theta_0^{(t-1)} - y^{(i)}] \right\}$$

Return $\theta^{(t)}, \theta_0^{(t)}$

Stochastic gradient descent

- Linear regression objective with $\lambda = 0$:

$$J_{\text{linreg}}(\Theta) = J_{\text{linreg}}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^n (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2$$

- A common machine learning objective:

$$f(\Theta) = \frac{1}{n} \sum_{i=1}^n f_i(\Theta)$$

- Also recall logistic regression with $\lambda = 0$

Stochastic-Gradient-Descent ($\Theta_{\text{init}}, \eta, T$)

Initialize $\Theta^{(0)} = \Theta_{\text{init}}$

for $t = 1$ **to** T

randomly select i from $\{1, \dots, n\}$ (with equal probability)

$$\Theta^{(t)} = \Theta^{(t-1)} - \eta(t) \nabla_{\Theta} f_i(\Theta^{(t-1)})$$

Return $\Theta^{(t)}$

Lecture 5* continued:-

* Stochastic Gradient Descent (Slide - 13 - last) (& G.D. vs S.G.D.)

- In gradient descent, what you do is first select the initial parameters θ & ϕ . Then, you calculate the gradient loss on the entire dataset with given values of θ & ϕ . Then you do gradient & multiply by step size & use it to update the θ & ϕ . Basically, before you update θ & ϕ , you calculate the loss on the entire dataset. This becomes time costly for large no. of iterations.
- In stochastic gradient descent, you calculate the loss on a random data point from the dataset & then you update the θ & ϕ values.

- Thus, you only use 1 single data point instead of the entire dataset before updation of θ & ϕ .
- This becomes very noisy & relatively inaccurate than regular gradient descent & you would need much more no. of iterations to obtain good θ & ϕ .
- But, obviously, S.G.D. becomes extremely cheap in terms of time & computational power required.
- Hence, again it comes to the question of the right compromise & you should choose between G.D & S.G.D. based on time, power, necessity, etc.