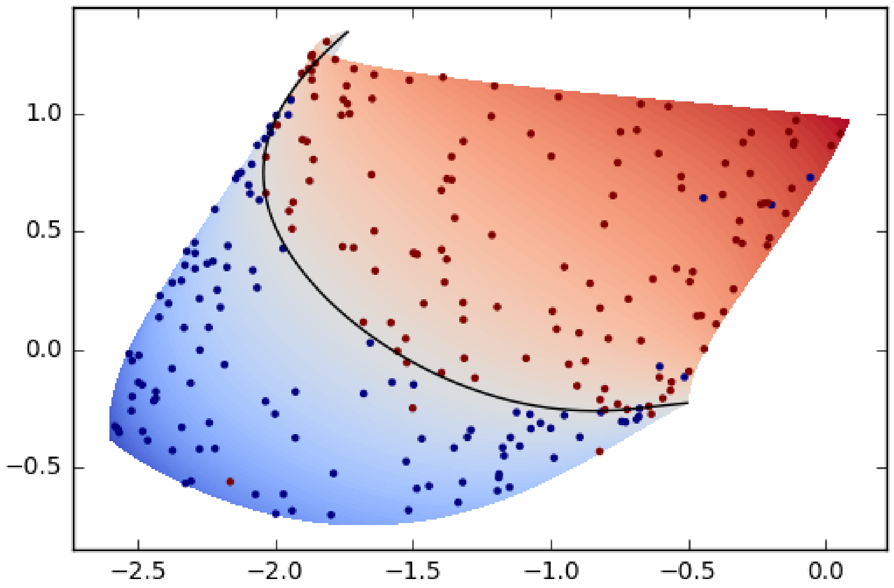


Linear Regression

PROF. SAMEER SINGH

FALL 2017

CS 273A: Machine Learning

Machine Learning

- Bayes Error Wrapup
- Gaussian Bayes Models
- Linear Regression: Definition and Cost
- Gradient Descent Algorithms

Measuring errors

Confusion matrix

Can extend to more classes

	Predict 0	Predict 1
Y=0	380	5
Y=1	338	3

tn, fp, fn, tp, True

$$\text{accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$\text{err} = 1 - \text{accuracy}$$

True positive rate: #(y=1 , ŷ=1) / #(y=1) -- “sensitivity”

False negative rate: #(y=1 , ŷ=0) / #(y=1)

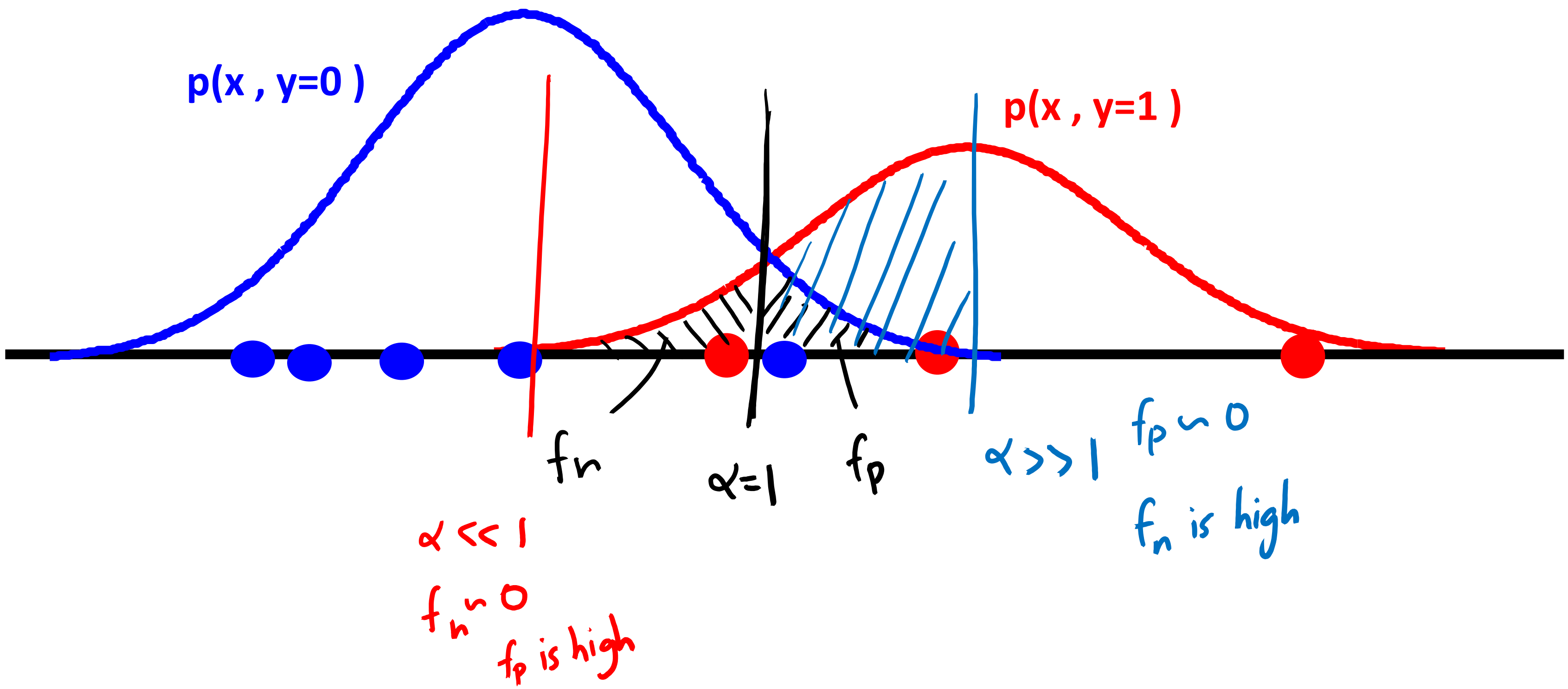
False positive rate: #(y=0 , ŷ=1) / #(y=0)

True negative rate: #(y=0 , ŷ=0) / #(y=0) -- “specificity”

Decision Surfaces

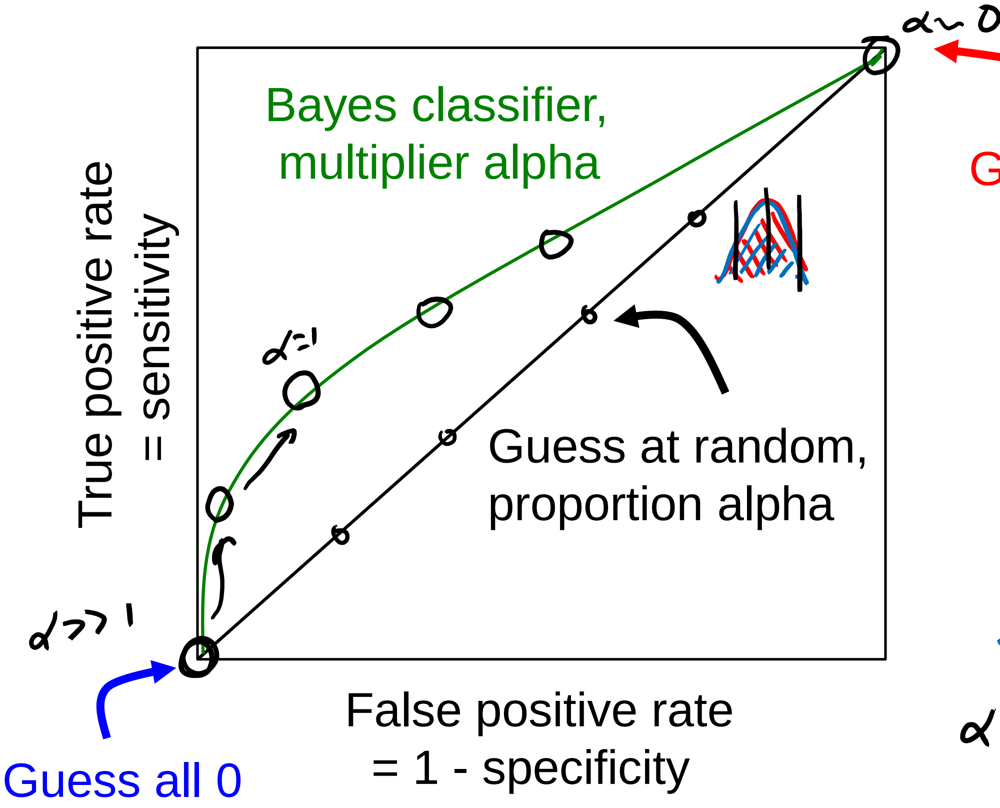
Add multiplier alpha:

$$\alpha \ p(y=0, x) \ \begin{matrix} < \\ > \end{matrix} \ p(y=1, x)$$

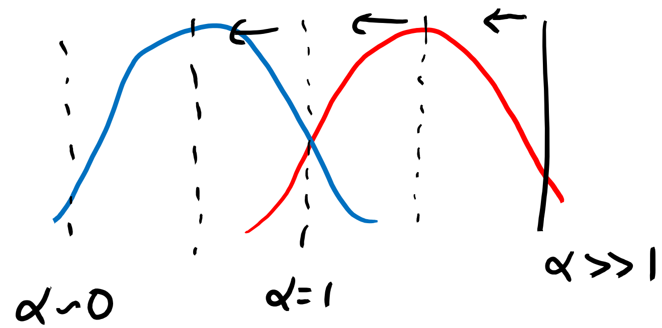


ROC Curves

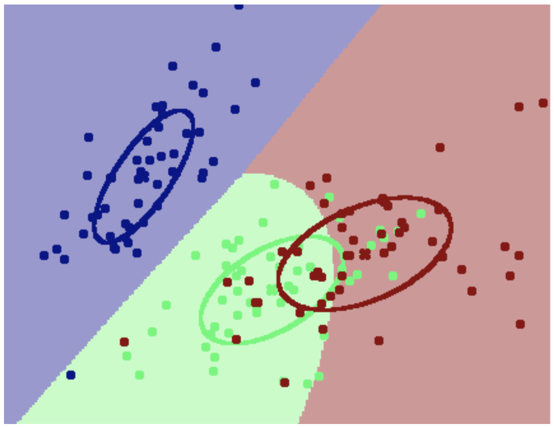
Characterize performance as we vary the decision threshold?



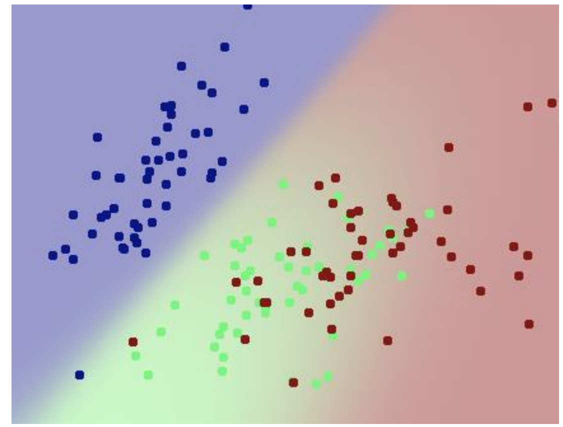
$$\alpha\ p(y=0,x) \ \substack{<\\>}\ p(y=1,x)$$



Probabilistic vs. Discriminative learning



"Discriminative" learning:
Output prediction $\hat{y}(x)$

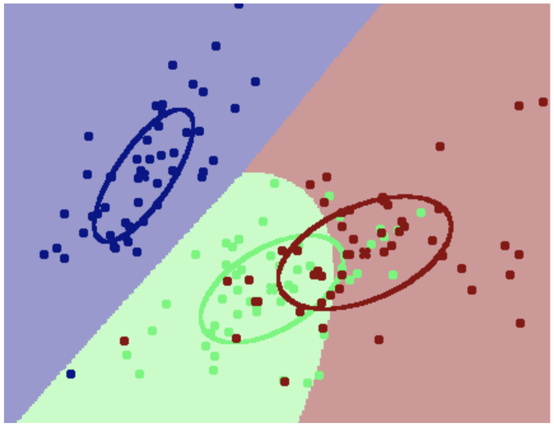


"Probabilistic" learning:
Output probability $p(y|x)$
(expresses confidence in outcomes)

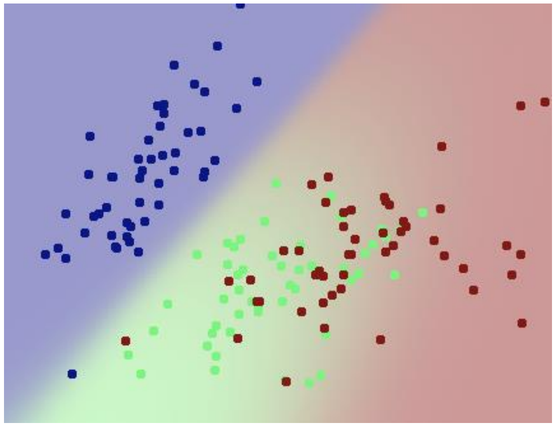
"Probabilistic" learning

- Conditional models just explain y: $p(y|x)$
- Generative models also explain x: $p(x,y)$
  - Often a component of unsupervised or semi-supervised learning
- Bayes and Naïve Bayes classifiers are generative models

Probabilistic vs. Discriminative learning



"Discriminative" learning:
Output prediction ŷ(x)



"Probabilistic" learning:
Output probability p(y|x)
(expresses confidence in outcomes)

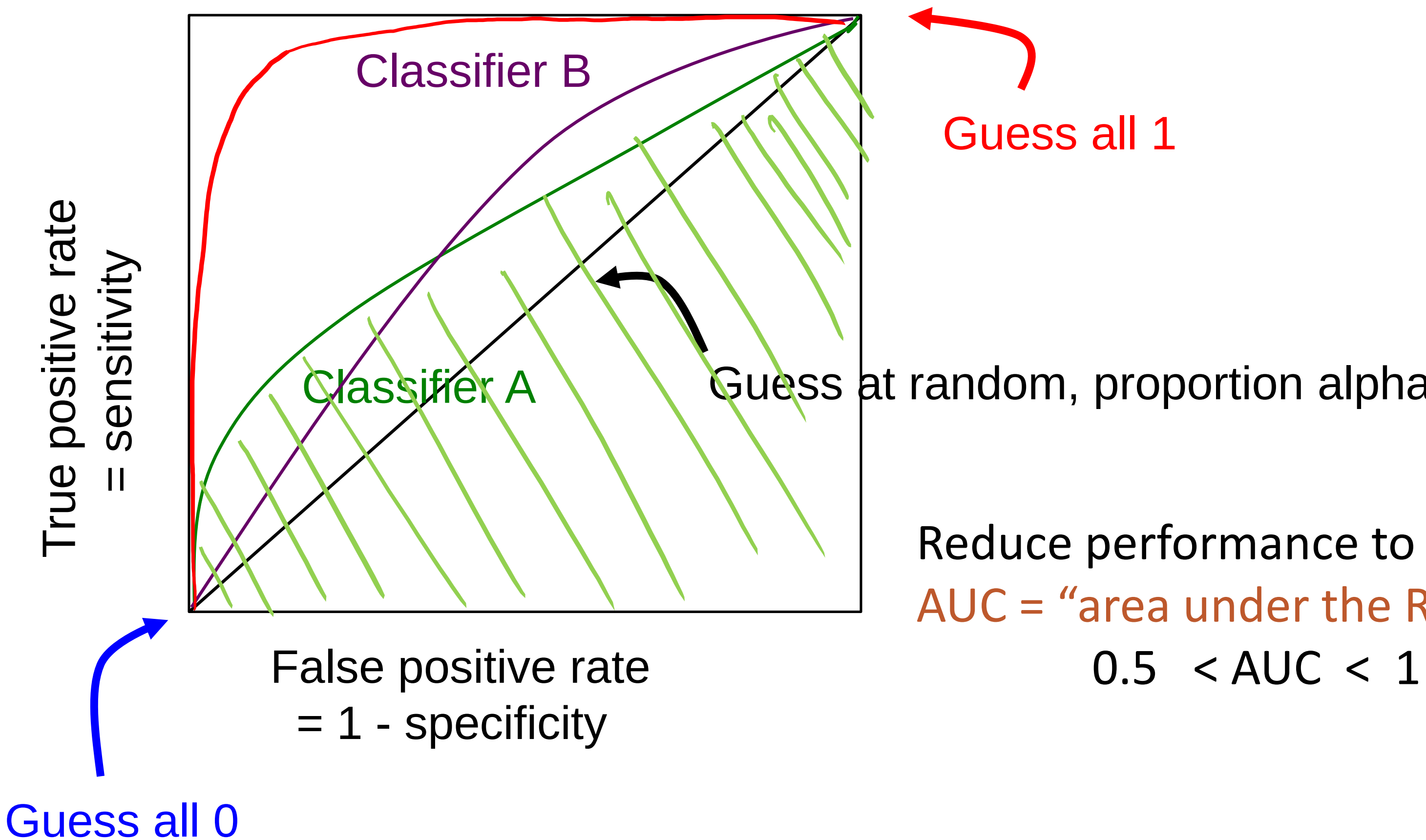
Can use ROC curves for discriminative models also:

- Some notion of confidence, but doesn't correspond to a probability
- In our code: "predictSoft" (vs. hard prediction, "predict")

```
>> learner = gaussianBayesClassify(X,Y);  % build a classifier
>> Ysoft = predictSoft(learner, X);       %  M x C matrix of confidences
>> plotSoftClassify2D(learner,X,Y);       %  shaded confidence plot
```

ROC Curves

Characterize performance as we vary our confidence threshold?



Reduce performance to one number?

AUC = "area under the ROC curve"

0.5 < AUC < 1

Machine Learning

- Bayes Error Wrapup
- **Gaussian Bayes Models**
- Linear Regression: Definition and Cost
- Gradient Descent Algorithms

Gaussian models

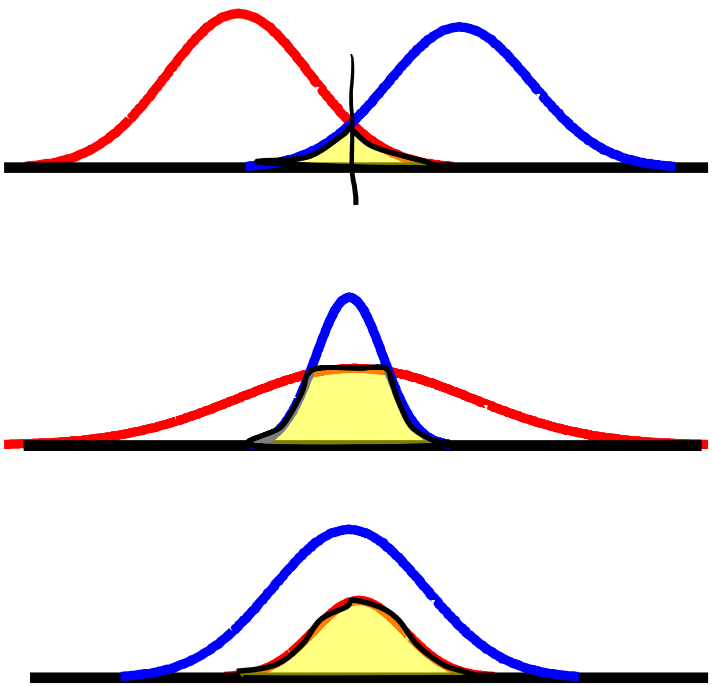
"Bayes optimal" decision

- Choose most likely class

Decision boundary

- Places where probabilities equal

What shape is the boundary?



Gaussian models

Bayes optimal decision boundary

- $p(y=0 \mid x) = p(y=1 \mid x)$
- Transition point between $p(y=0|x)$ >/< $p(y=1|x)$

$p(x|y=1)$

$$\mathcal{N}(\underline{x}\,;\,\underline{\mu}_1, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu}_1)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_1) \right\}$$

$$0 \begin{matrix} < \\ > \end{matrix} \log \frac{p(x|y=0)}{p(x|y=1)} \frac{p(y=0)}{p(y=1)} = \log \frac{p(y=0)}{p(y=1)} - \frac{1}{2}\left(x \Sigma^{-1} x - 2\mu_0 \Sigma^{-1} x + \mu_0^T \Sigma^{-1} \mu_0\right)$$

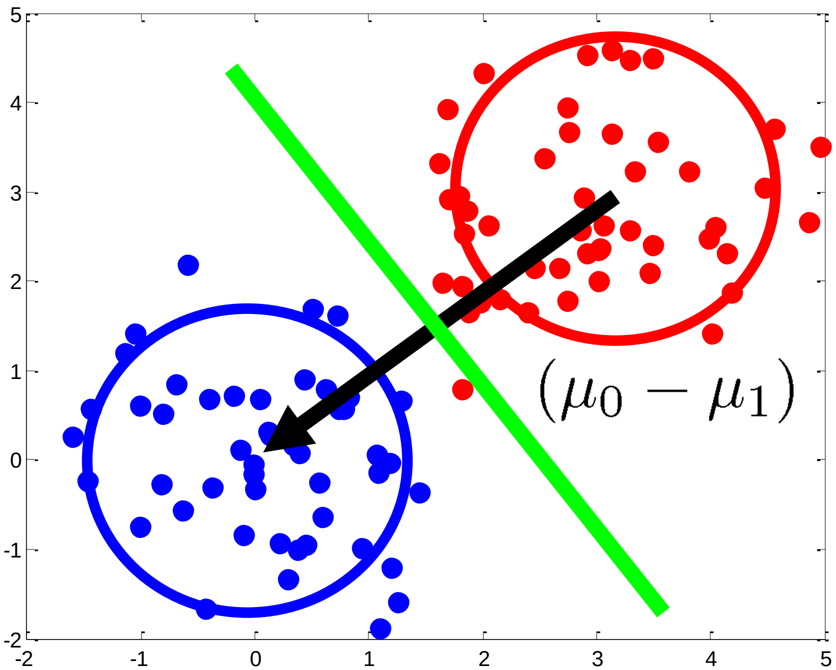
$$+ \frac{1}{2}\left(x \Sigma^{-1} x - 2\mu_1 \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1\right)$$

$$= (\mu_0 - \mu_1)^T \Sigma^{-1} x + \text{constants}$$

Gaussian example

Spherical covariance: $\Sigma = \sigma^2 I$ $\quad = (\mu_0 - \mu_1)^T \Sigma^{-1} x + constants$

Decision rule $\quad (\mu_0 - \mu_1)^T x \begin{matrix} < \\ > \end{matrix} C$



$$C = .5(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) - \log \frac{p(y=0)}{p(y=1)}$$

Non-spherical Gaussian distributions

Equal covariances => still linear decision rule

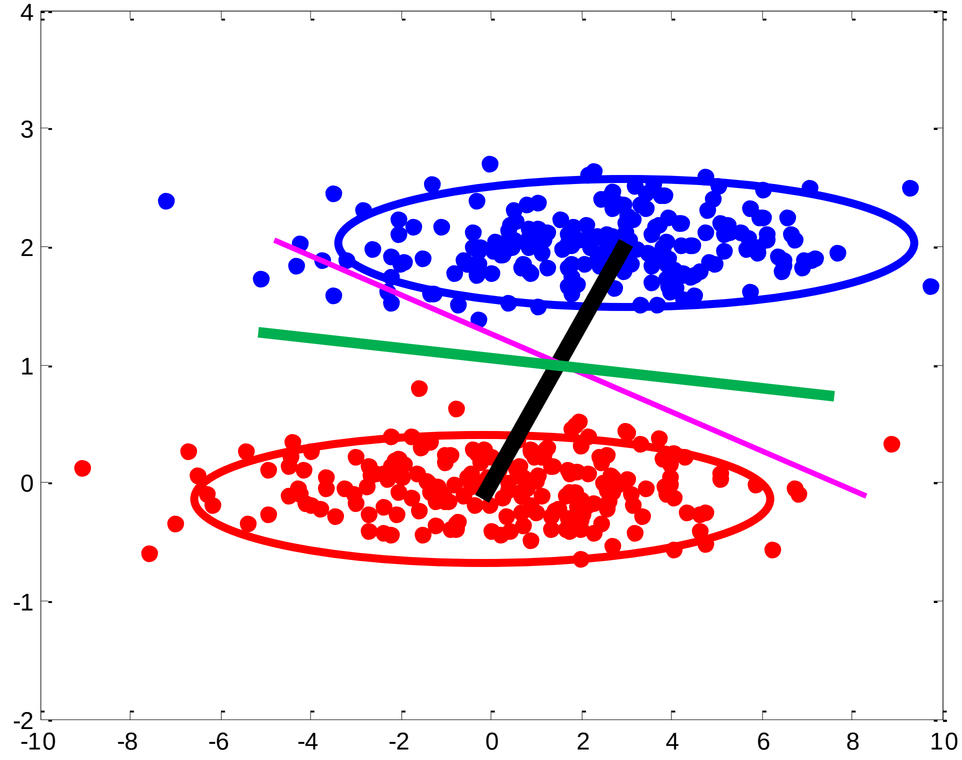
- May be "modulated" by variance direction
- Scales; rotates (if correlated)

Example:

Variance

[3   0   ]

[0 .25]



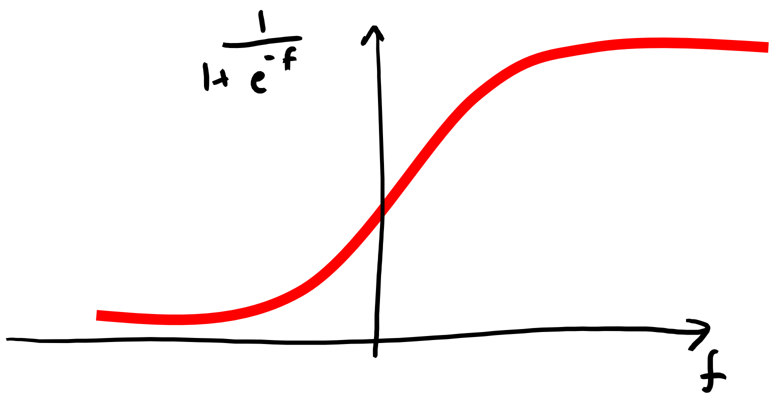
Class posterior probabilities

Consider comparing two classes

- p(x | y=0) * p(y=0)   vs   p(x | y=1) * p(y=1)
- Write probability of each class as
- p(y=0 | x) = p(y=0, x) / p(x)
- = p(y=0, x) / (p(y=0,x) + p(y=1,x)) $= \dfrac{1}{1 + \dfrac{p(y=1,x)}{p(y=0,x)}}$
- = 1 / (1 + exp(- *f*))

- *f* = log [p(y=0, x) / p(y=1, x)]

the logistic function, or logistic sigmoid



Gaussian models

$$\mathcal{N}(\underline{x} \; ; \; \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$

$$0 \; \substack{<\\>} \; \log \frac{p(x|y=0)}{p(x|y=1)} \frac{p(y=0)}{p(y=1)} = \quad (\mu_0 - \mu_1)^T \Sigma^{-1} x + constants$$

f

Now we also know that the probability of each class is given by:

$p(y=0 \mid x) = \text{Logistic}(\, f \,) = \text{Logistic}(\; a^T x + b \,)$

We'll see this form again soon…

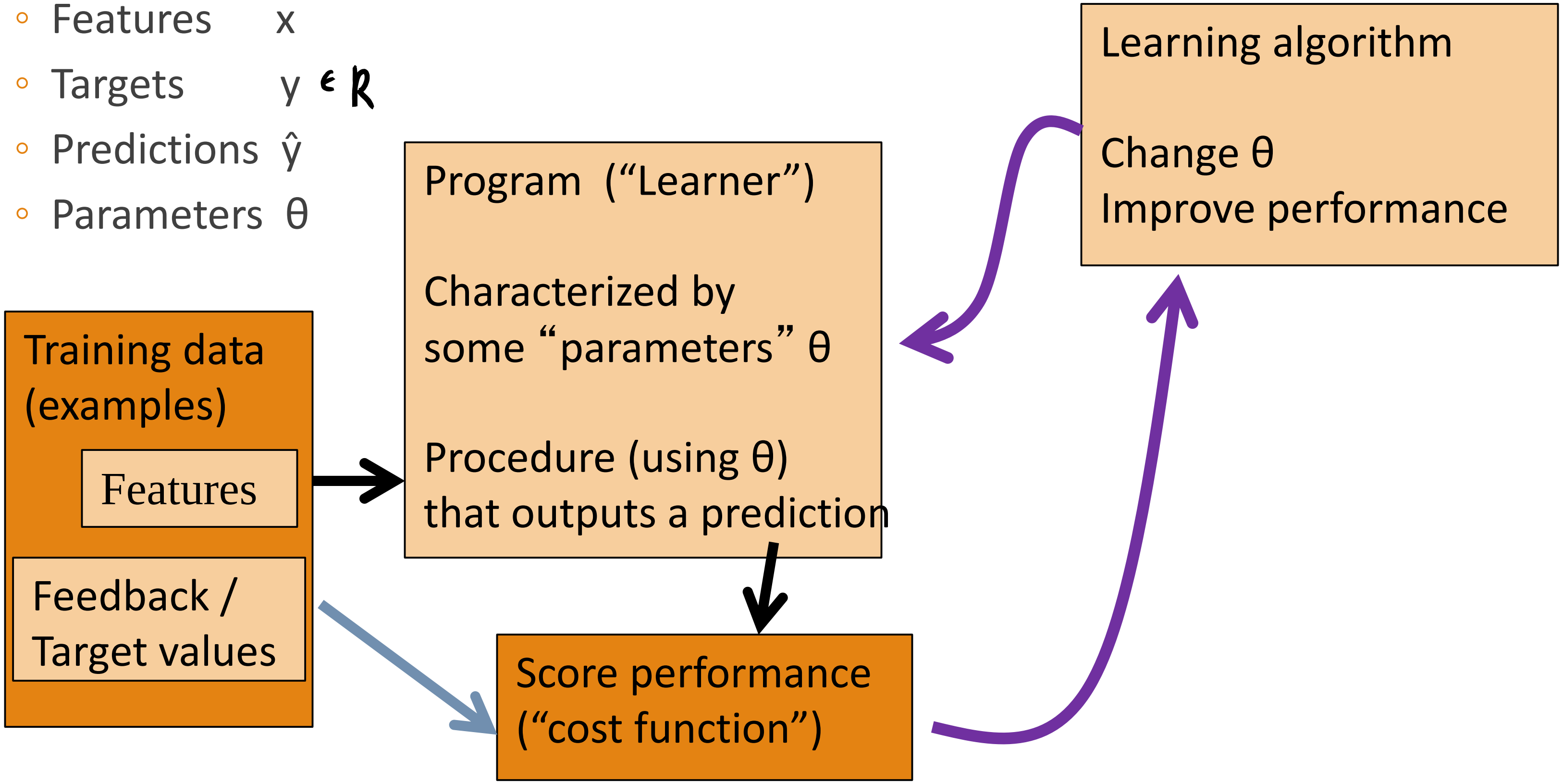
Machine Learning

- Bayes Error Wrapup
- Gaussian Bayes Models
- **Linear Regression: Definition and Cost**
- Gradient Descent Algorithms

Supervised learning

Notation

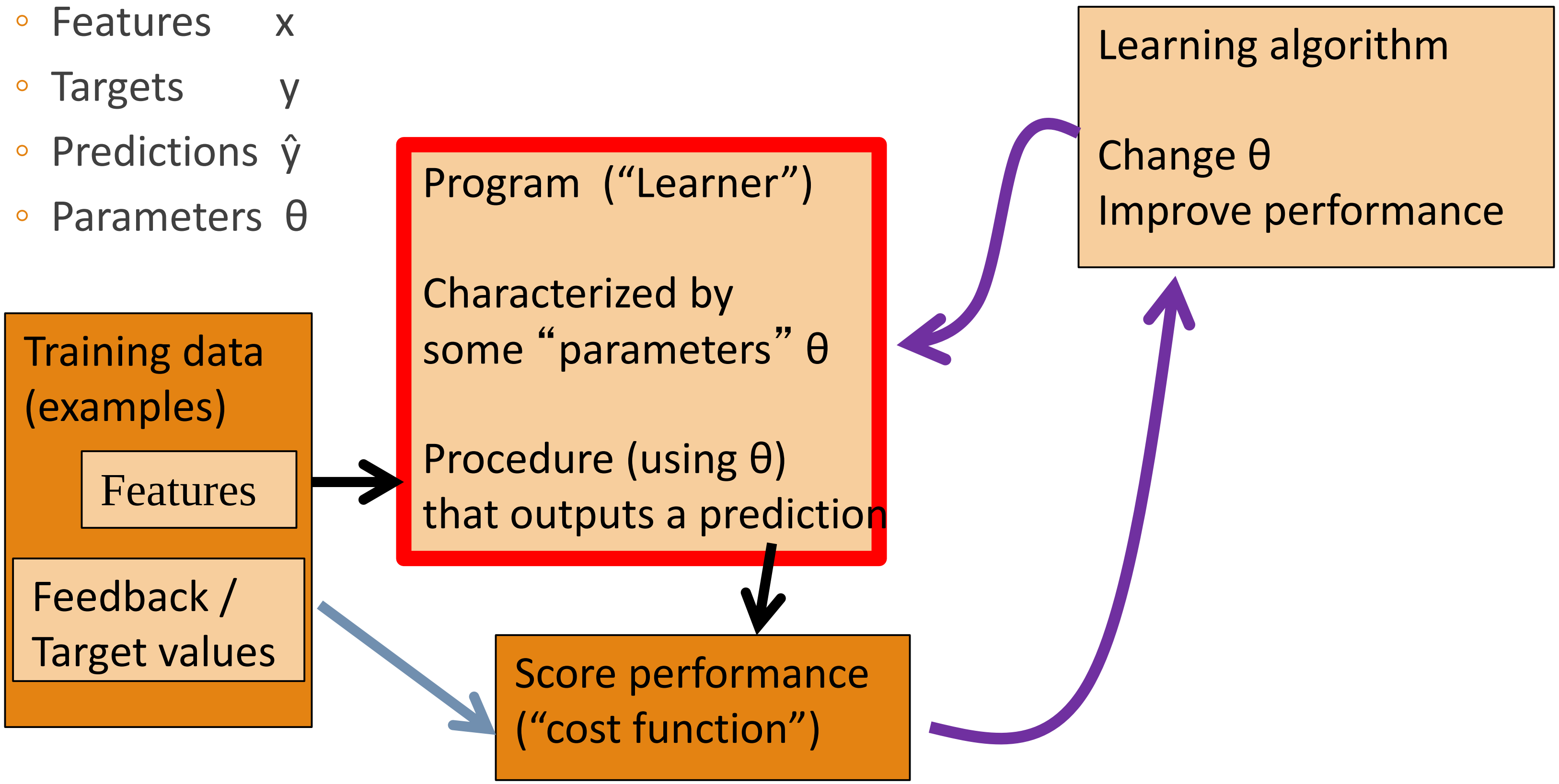
- Features x
- Targets y ∈ R
- Predictions ŷ
- Parameters θ



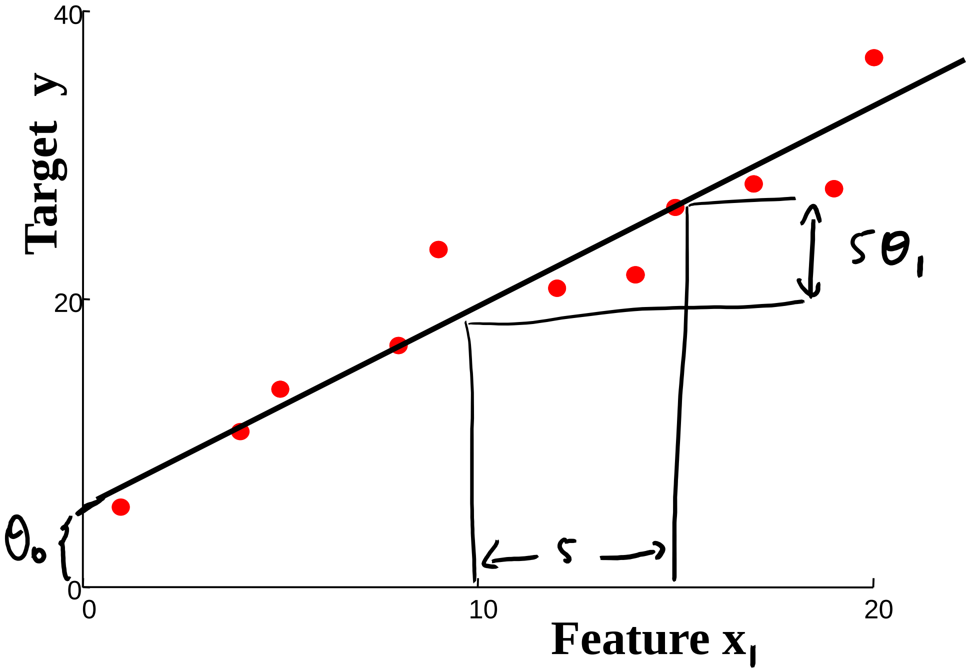
Supervised learning

Notation

- Features x
- Targets y
- Predictions ŷ
- Parameters θ



Linear regression



"Predictor":
Evaluate line:

$$r = \theta_0 + \theta_1 x_1$$

return r

Define form of function f(x) explicitly

Find a good f(x) within that family

Notation

$$\hat{y}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Define feature $x_0 = 1$ (constant)

$$\theta = [\theta_0, \theta_1, \ldots \theta_n] \quad 1 \times (n+1)$$

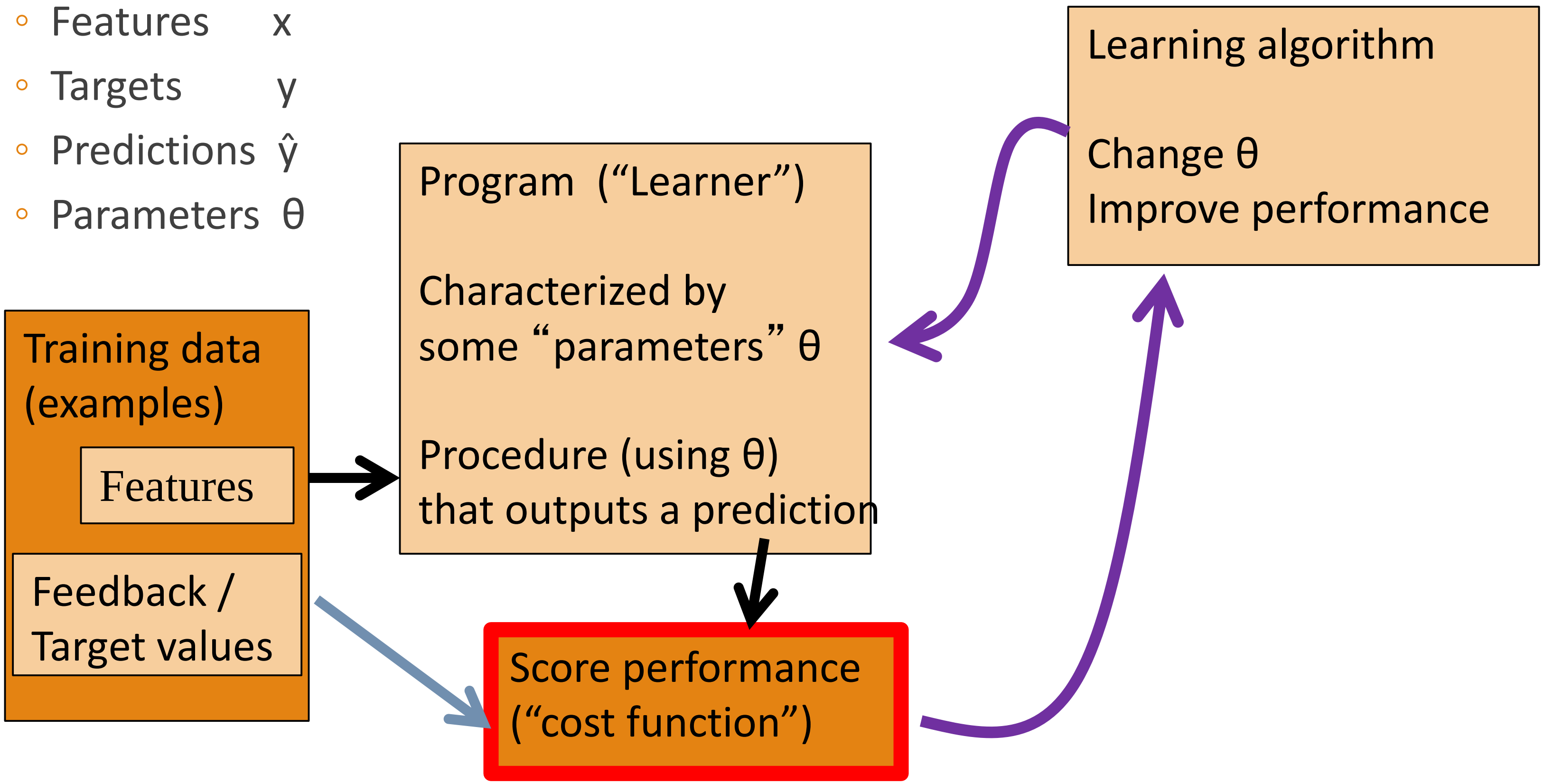
$$\hat{y}(x) = \theta x^T$$

$$x = [1, x_1, \ldots x_n] \quad 1 \times (n+1)$$

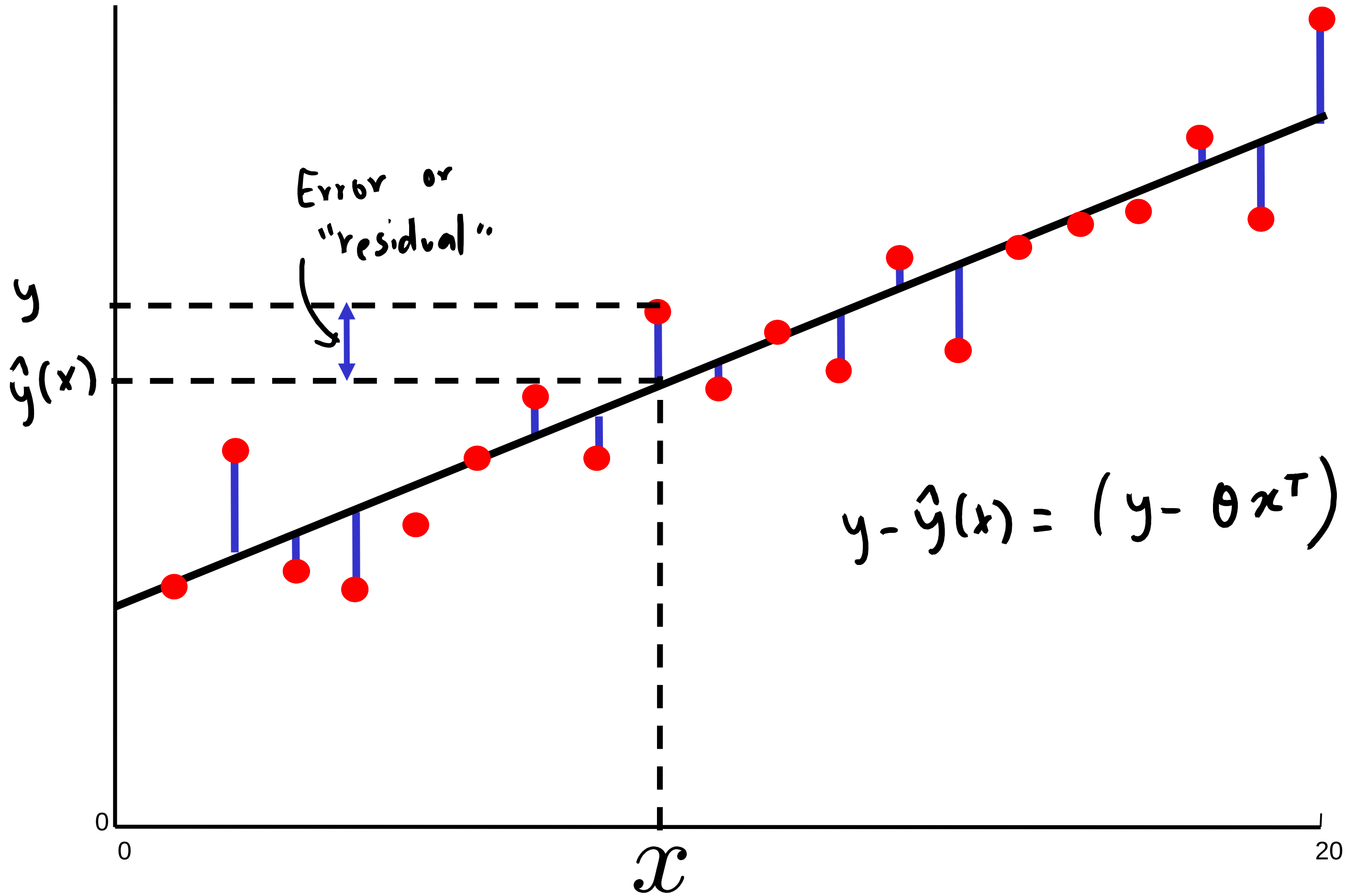
Supervised learning

Notation

- Features x
- Targets y
- Predictions ŷ
- Parameters θ



Measuring error



Mean squared error

How can we quantify the error?

$$\text{MSE},\quad J(\theta) = \frac{1}{m}\sum_j \left(y^{(j)} - \hat{y}(x^{(j)}) \right)^2$$

$$= \frac{1}{m}\sum_j \left(y^{(j)} - \theta \cdot x^{(j)T} \right)^2$$

Could choose something else, of course…

- Computationally convenient (more later)
- Measures the variance of the residuals
- Corresponds to likelihood under Gaussian model of "noise"

$$\mathcal{N}(y\,;\,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

MSE cost function

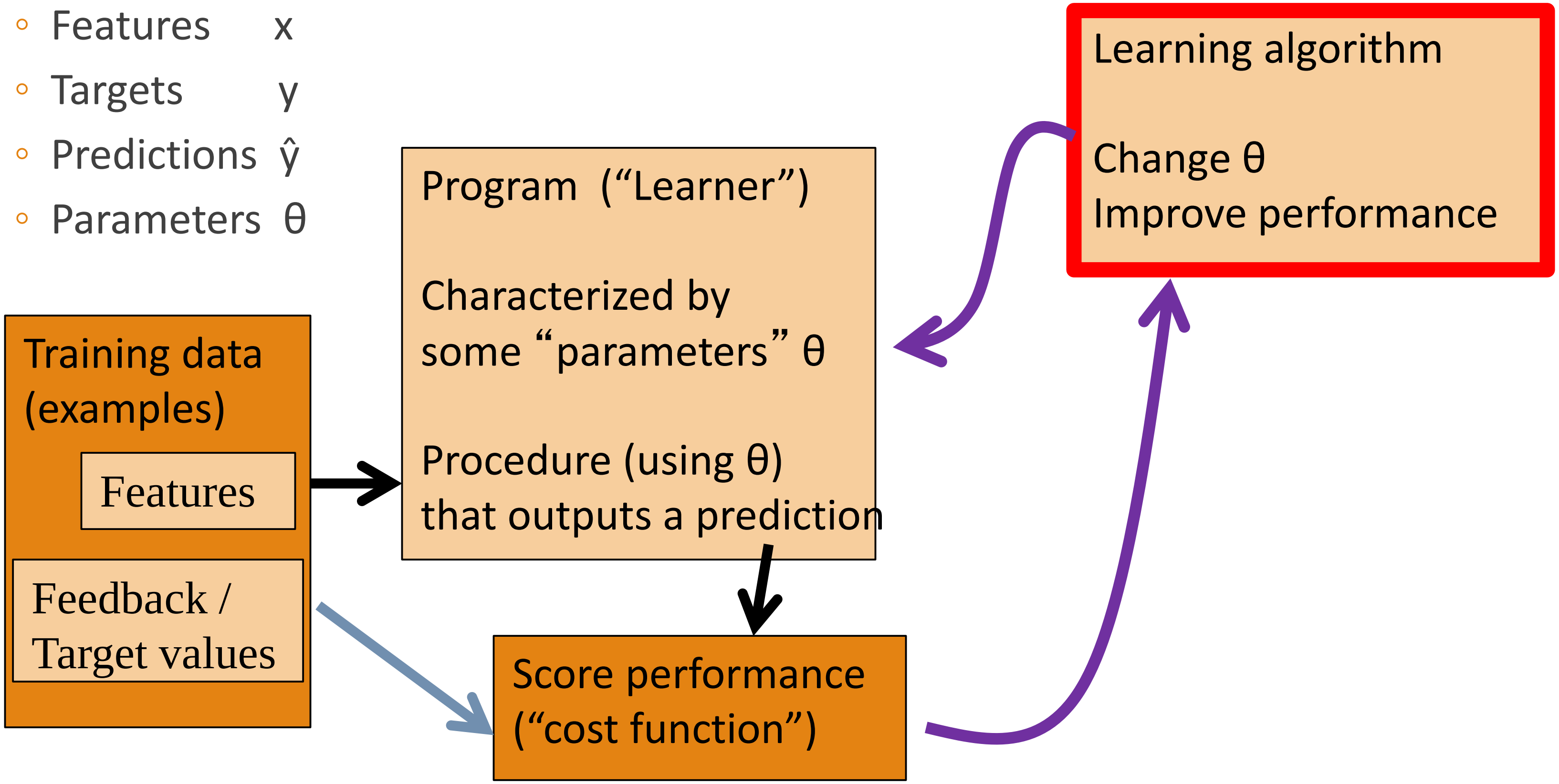
$$\text{MSE}, \; J(\theta) = \frac{1}{m} \sum_{j} \left(y^{(j)} - \theta \cdot x^{(j)T}\right)^2 \qquad y = \left[y^{(1)}, y^{(2)} \dots y^{(m)} \right]^T$$

$$J(\theta) = \frac{1}{m} \left(y^T - \theta x^T \right) \left(y^T - \theta x^T \right)^T \qquad X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_n^{(1)} \\ \vdots & \vdots & & \vdots \\ x_0^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix}$$

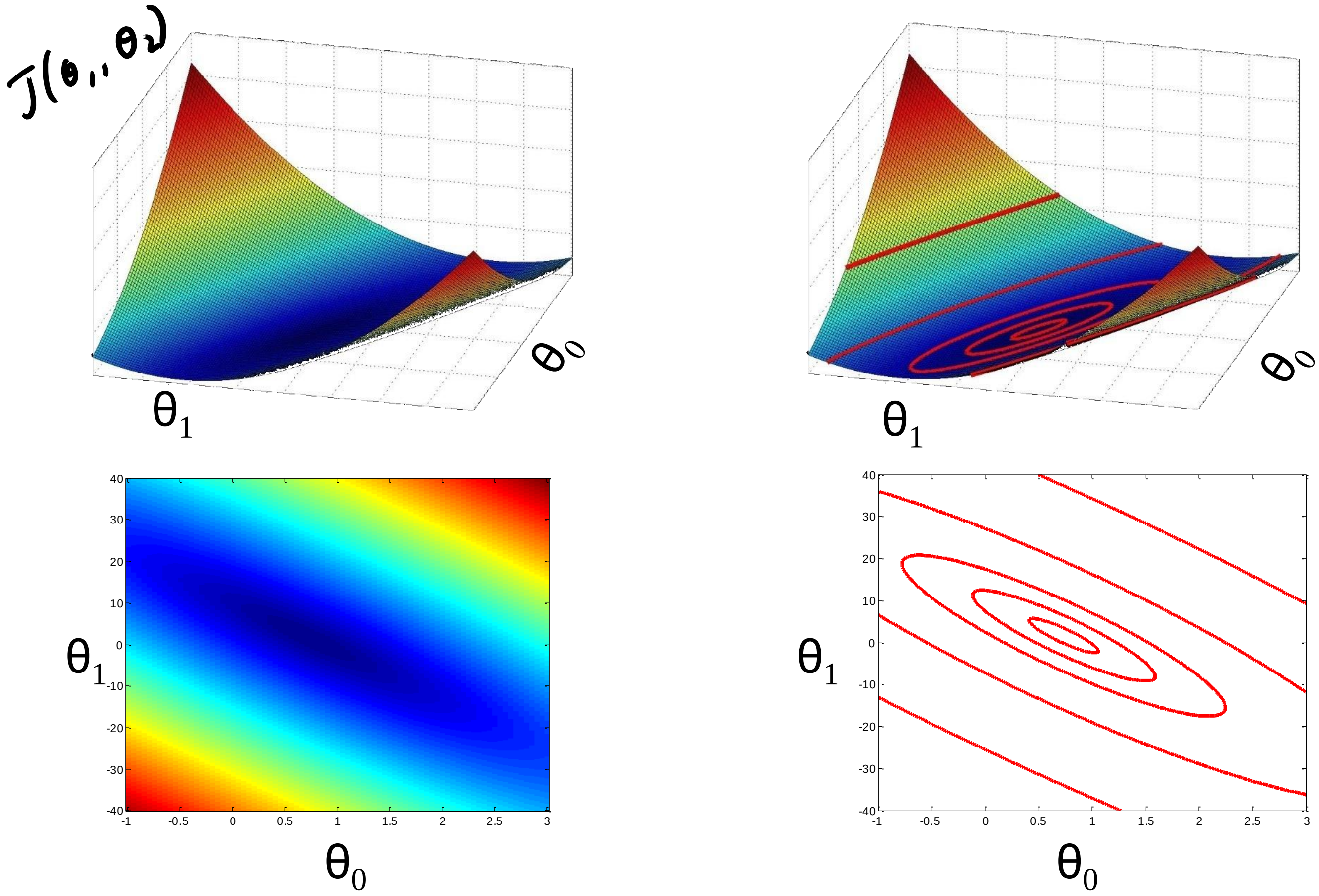
```
# Python / NumPy:
e = Y - X.dot( theta.T );
J = e.T.dot( e ) / m  # = np.mean( e ** 2 )
```

Supervised learning

Notation
- Features x
- Targets y
- Predictions ŷ
- Parameters θ



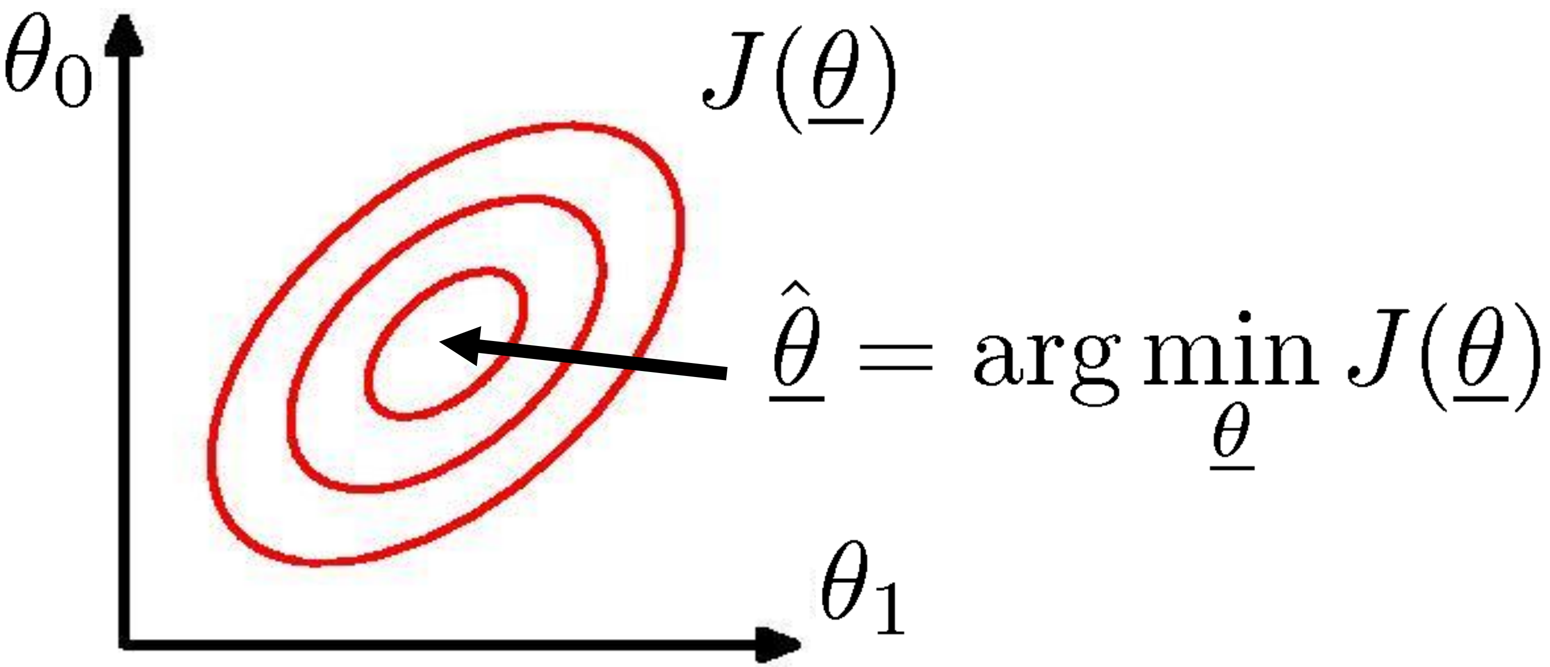
Visualizing the cost function



Finding good parameters

Want to find parameters which minimize our error…

Think of a cost "surface": error residual for that θ…



Machine Learning

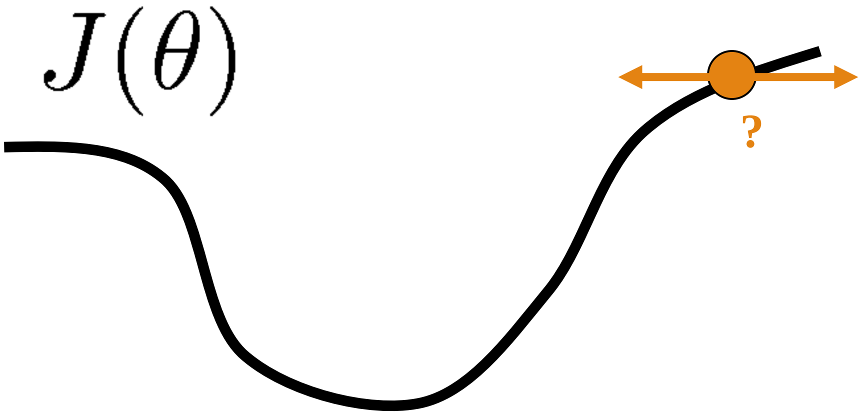
Bayes Error Wrapup

Gaussian Bayes Models

Linear Regression: Definition and Cost

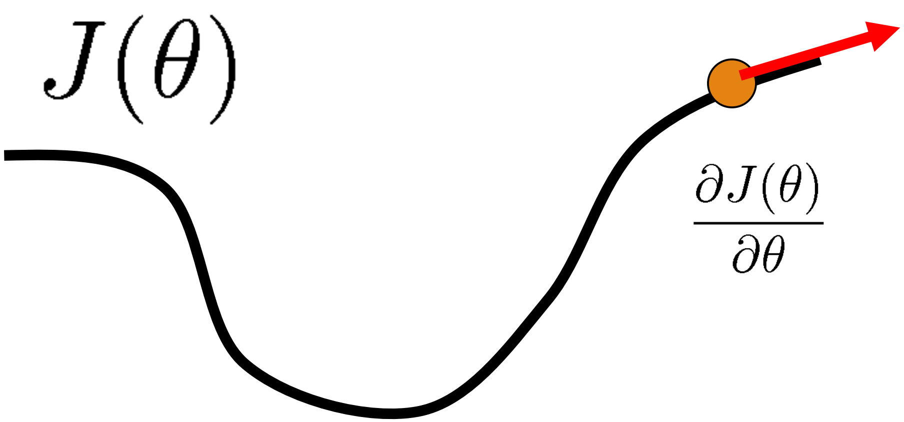
Gradient Descent Algorithms

Gradient descent



- How to change θ to improve J(θ)?
- Choose a direction in which J(θ) is decreasing

Gradient descent

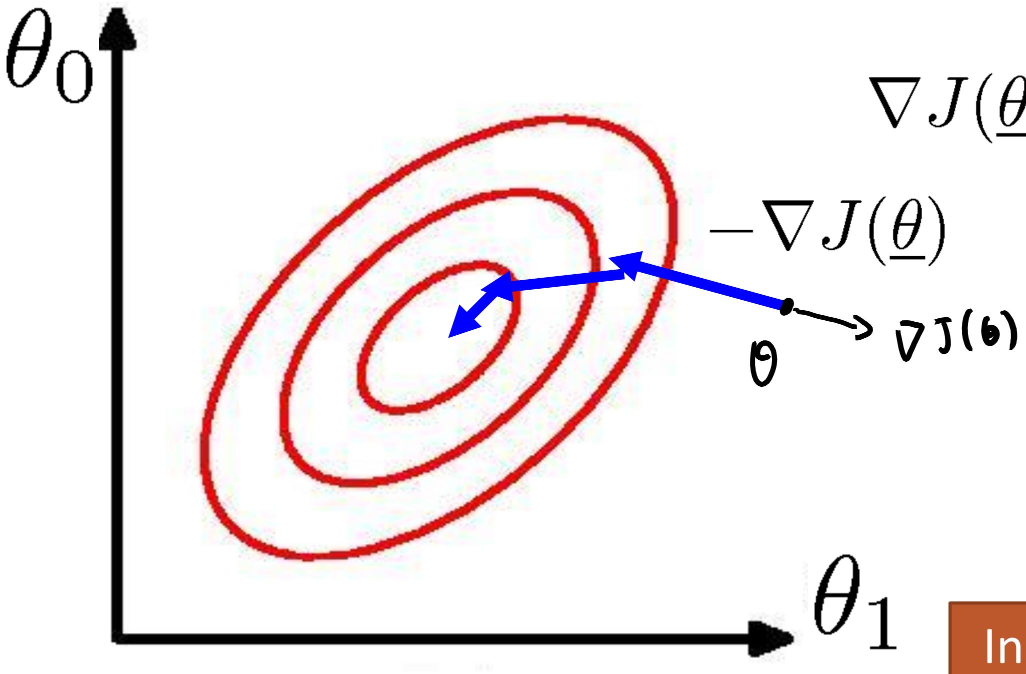


- How to change θ to improve J(θ)?
- Choose a direction in which J(θ) is decreasing
- Derivative $\frac{\partial J(\theta)}{\partial \theta}$
- Positive => increasing
- Negative => decreasing

Gradient descent in >2 dimensions

- Gradient vector

$$\nabla J(\underline{\theta}) = \left[\frac{\partial J(\underline{\theta})}{\partial \theta_0} \quad \frac{\partial J(\underline{\theta})}{\partial \theta_1} \quad \ldots \right]$$



Indicates direction of steepest ascent
(negative = steepest descent)

Gradient descent

Initialization

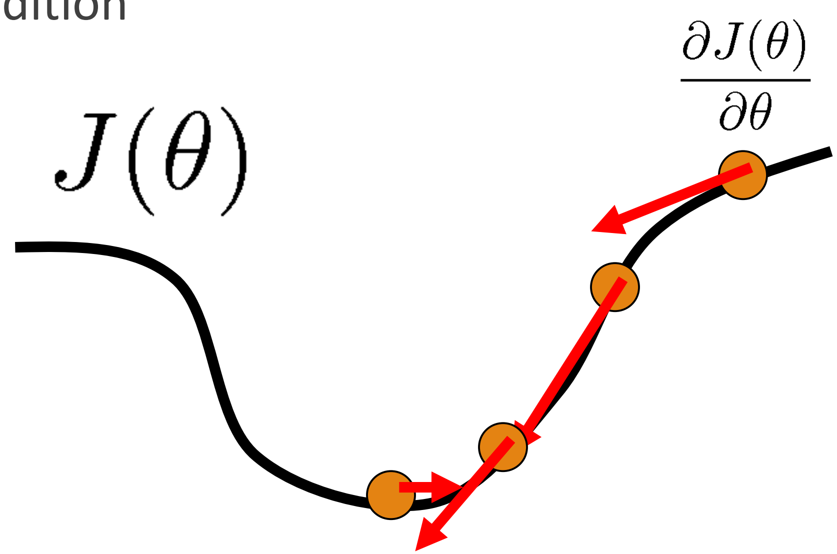
Step size

- Can change as a function of iteration

Gradient direction

Stopping condition

```
Initialize θ   randomly (small)
               0s
Do{
 θ ← θ - α∇θJ(θ)
} while (α||∇θJ|| > ε)
```



Gradient for the MSE

$$J(\theta) = \frac{1}{m} \sum_j \left(\underbrace{y^{(j)} - \theta_0 x_0^{(j)} - \theta_1 x_1^{(j)} \cdots - \theta_n x_n^{(j)}}_{e_j(\theta)} \right)^2$$

$e_j(\theta)$ (error on j)

$$\nabla J(\theta) = \left[\frac{\partial J(\theta)}{\partial \theta_0} \quad \frac{\partial J(\theta)}{\partial \theta_1} \quad \cdots \quad \frac{\partial J(\theta)}{\partial \theta_n} \right]$$

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m} \frac{\partial}{\partial \theta_0} \sum_j \left(e_j(\theta)\right)^2$$

$$= \frac{1}{m} \sum_j 2\, e_j(\theta) \frac{\partial}{\partial \theta_0} e_j(\theta)$$

$$\frac{\partial e_j(\theta)}{\partial \theta_0} = \frac{\partial y^{(j)}}{\partial \theta_0} - \frac{\partial}{\partial \theta_0} \theta_0 x_0^{(j)} - \frac{\partial}{\partial \theta_0} \theta_1 x_1^{(j)} + \cdots$$

$$= -x_0^{(j)}$$

Upcoming...

Misc.

- Lot of activity on Piazza
- You have been added to Gradescope

Homework

- Homework 1 due tonight
- Homework 2 released tonight
- HW2 Due: October 19, 2017