Data Mining & Machine Learning

CS37300
Purdue University

Sep 27, 2023

Take-home Quiz

- Recall the previous example in the Decision Tree lecture

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Please construct a decision tree using information gain and full growth criteria.

Take-home Quiz

- Step 1. Compute the contingency table for each feature.

Age

	BC=yes	BC=no
<=30	2	3
31-40	4	0
>40	3	2

Income

	BC=yes	BC=no
High	2	2
Med	4	2
Low	3	1

Student

	BC=yes	BC=no
Yes	6	1
No	3	4

Credit

	BC=yes	BC=no
Fair	6	2
Excellent	3	3

Step 2. Compute the conditional probabilities $P(y|x_i=a)$.

Age

	BC=yes	BC=no
<=30	2/5	3/5
31-40	1	0
>40	3/5	2/5

Income

	BC=yes	BC=no
High	2/4	2/4
Med	4/6	2/6
Low	3/4	1/4

Student

	BC=yes	BC=no
Yes	6/7	1/7
No	3/7	4/7

Credit

	BC=yes	BC=no
Fair	6/8	2/8
Excellent	3/6	3/6

Take-home Quiz

- Compute the entropy and information gain for each feature at the root of the tree

$$Entropy(BC) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.9403$$

Yes: 9
No: 5

Age

	BC=yes	BC=no
<=30	2/5	3/5
31-40	1	0
>40	3/5	2/5

$$Entropy(BC, age <= 30) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$Entropy(BC, age = 31...40) = 0$$

$$Entropy(BC, age > 40) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$Gain(BC, age) = 0.940 - \frac{5}{14}\cdot 0.971 - \frac{5}{14}\cdot 0.971 = 0.177$$

Take-home Quiz

- Compute the entropy and information gain for each feature at the root of the tree

$$Entropy(BC) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.9403$$

Yes: 9
No: 5

Income

	BC=yes	BC=no
High	2/4	2/4
Med	4/6	2/6
Low	3/4	1/4

$$Entropy(BC, Income = high) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$$

$$Entropy(BC, Income = medium) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$$

$$Entropy(BC, Income = low) = -\frac{3}{4}\log_2\frac{1}{4} = 0.811$$

$$Gain(BC, Income) = 0.940 - (\frac{4}{14}\cdot 1 + \frac{6}{14}\cdot 0.918 + \frac{4}{14}\cdot 0.811) = 0.029$$

Take-home Quiz

- Compute the entropy and information gain for each feature at the root of the tree

$$Entropy(BC) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.9403$$

Yes: 9
No: 5

student

	BC=yes	BC=no
Yes	6/7	1/7
No	3/7	4/7

$$Entropy(BC, student = no) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$$

$$Entropy(BC, student = yes) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.592$$

$$Gain(BC, student) = 0.940 - (\frac{7}{14}\cdot 0.985 + \frac{7}{14}\cdot 0.592) = 0.151$$

Take-home Quiz

- Compute the entropy and information gain for each feature at the root of the tree

$$Entropy(BC) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.9403$$

Yes: 9
No: 5

Credit

	BC=yes	BC=no
Fair	6/8	2/8
Excellent	3/6	3/6

$$Entropy(BC, credit = fair) = -\frac{2}{8}\log_2\frac{2}{8} - \frac{6}{8}\log_2\frac{6}{8} = 0.811$$

$$Entropy(BC, credit = excellent) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$$

$$Gain(BC, credit) = 0.940 - (\frac{8}{14}\cdot 0.811 + \frac{6}{14}\cdot 1) = 0.048$$

Take-home Quiz

- Pick the feature with the largest information gain

$$Gain(BC, age) = 0.940 - \frac{5}{14} \cdot 0.971 - \frac{5}{14} \cdot 0.971 = 0.177$$

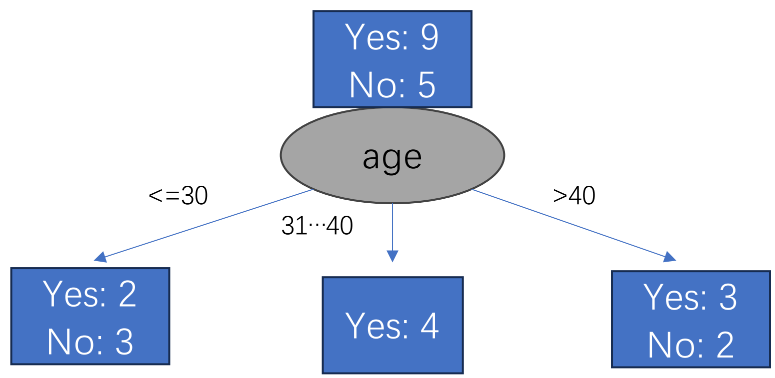
$$Gain(BC, Income) = 0.940 - (\frac{4}{14} \cdot 1 + \frac{6}{14} \cdot 0.918 + \frac{4}{14} \cdot 0.811) = 0.029$$

$$Gain(BC, student) = 0.940 - (\frac{7}{14} \cdot 0.985 + \frac{7}{14} \cdot 0.592) = 0.151$$

$$Gain(BC, credit) = 0.940 - (\frac{8}{14} \cdot 0.811 + \frac{6}{14} \cdot 1) = 0.048$$

Take-home Quiz

- Partition the initial dataset based on age and grow the tree by one level



Take-home Quiz

- The first subset is not a “pure” (i.e. entropy = 0) set, continue to partition it.
- Repeat the previous steps on the first subset

Income

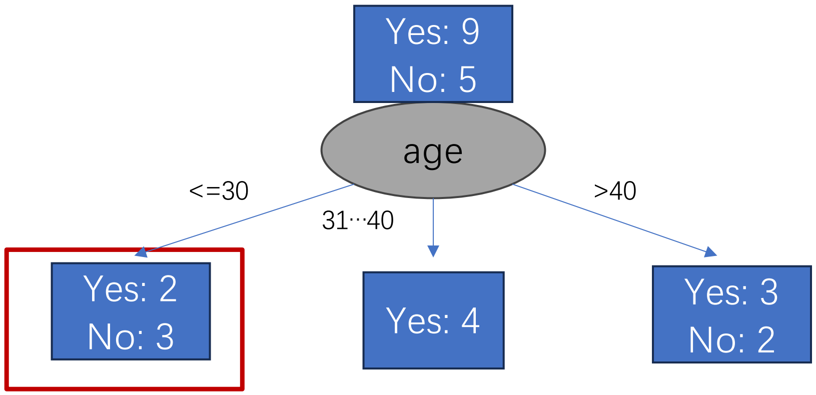
	BC=yes	BC=no
High	0	2
Med	1	1
Low	1	0

Student

	BC=yes	BC=no
Yes	2	0
No	0	3

Credit

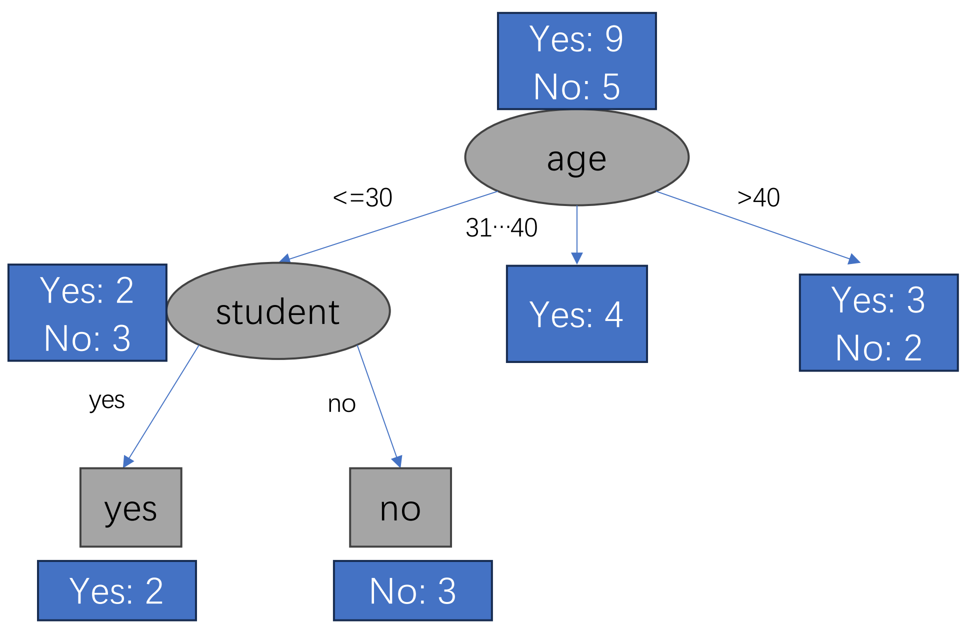
	BC=yes	BC=no
Fair	1	2
Excellent	1	1



No need to further calculate. We already see a feature that splits the first subset to two pure sets.

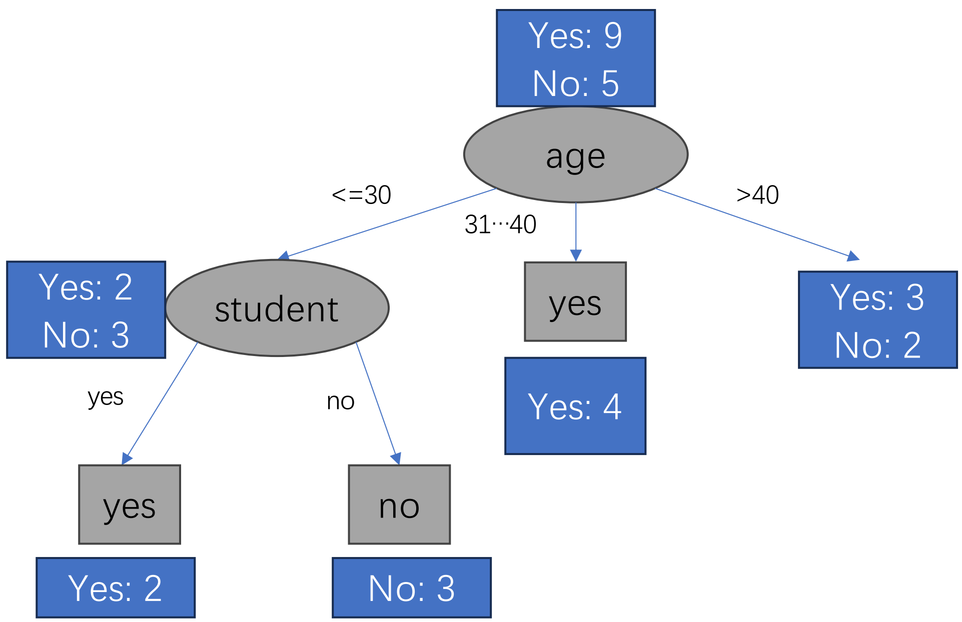
Take-home Quiz

- Spit the first subset based on the student feature



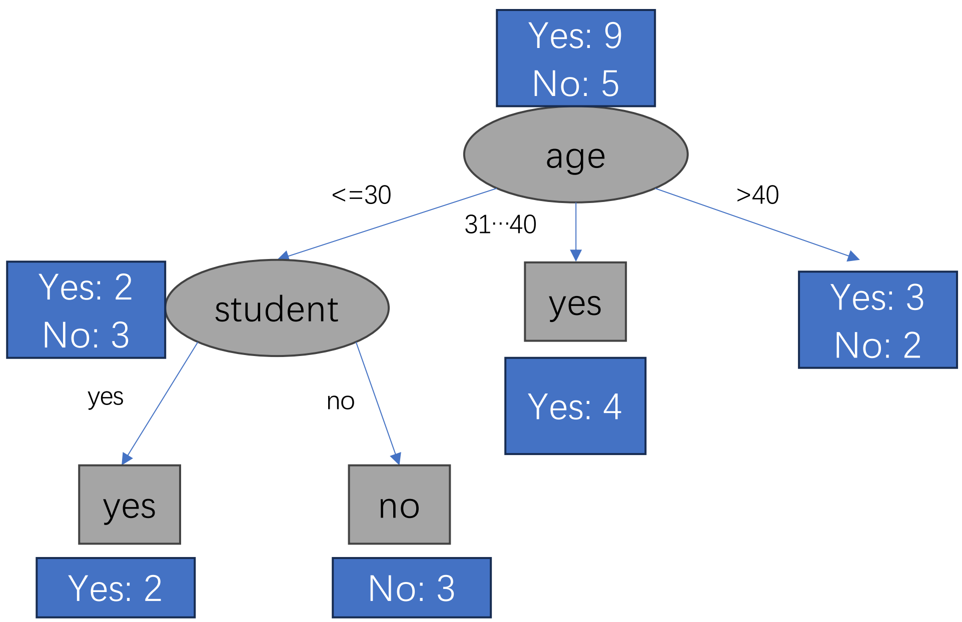
Take-home Quiz

- The second subset at Depth 1 is already a “pure” set
- No need to further partition



Take-home Quiz

- The third subset at Depth 1 is not a “pure” set
- Need to further partition. Repeat previous steps.



Take-home Quiz

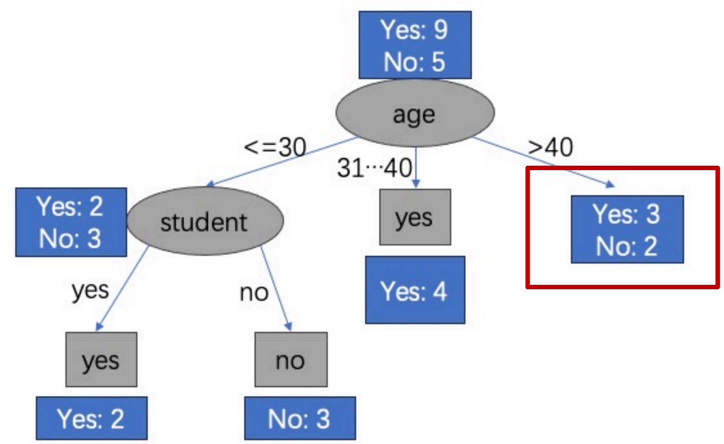
- The third subset at Depth 1 is not a pure set
- Need to further partition. Repeat previous steps.

Income

	BC=yes	BC=no
High	0	0
Med	2	1
Low	1	1

Student

	BC=yes	BC=no
Yes	2	1
No	1	1



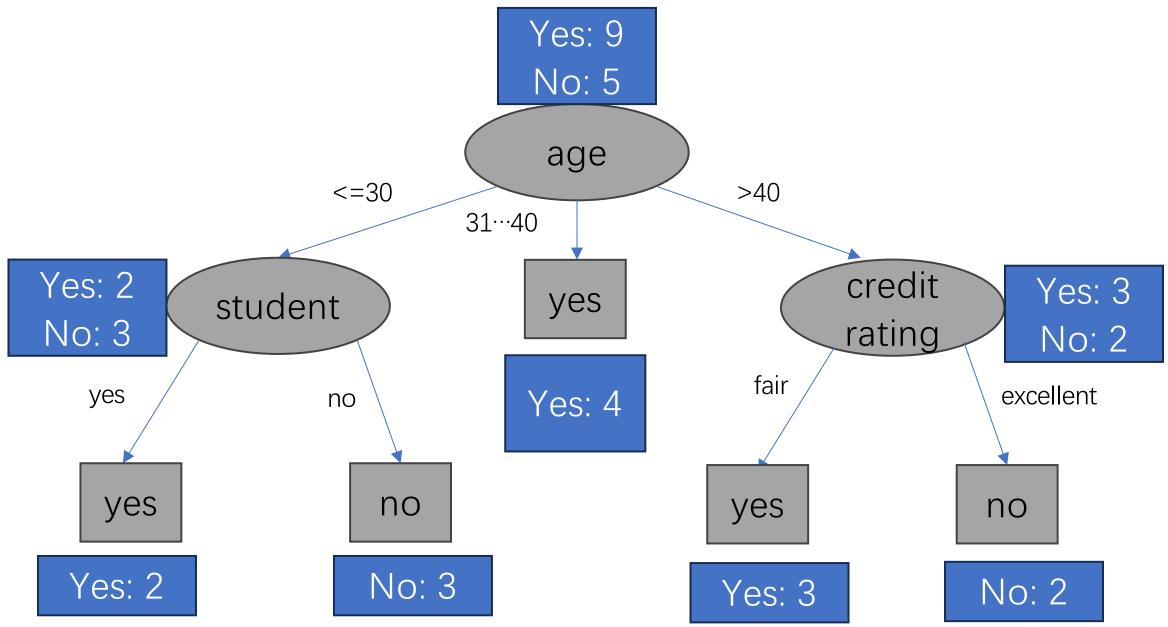
Credit

	BC=yes	BC=no
Fair	3	0
Excellent	0	2

No need to further calculate. We already see a feature that splits the first subset to two pure sets.

Take-home Quiz

- Spit the third subset based on the credit feature



Today’s topics

- Gradient descent (contd)
- Linear regression

Gradient descent

Convex optimization problems

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

- *x* is the optimization variable (*e.g., model parameters*)
  f (*e.g., score function*) is a **convex function**
  C is a **convex set** (*e.g., constraints on model parameters*)

- For convex optimization problems, all locally optimal points are globally optimal

Solve convex optimization problem

- Minimize a convex function without any constraints on the variables
  - If f'(x)=0 then x is a stationary point of f
  - If f'(x)=0 and f''(x) is not negative then x is a local minimum of f (for convex function, this is also a global minimum)
  - If f is a strictly convex function, any stationary point of ***f*** is the unique global minimum of f

Gradient descent

- For some convex functions, we may be able to take the derivative, but it may be difficult to directly solve for parameter values
- Solution:
  - Start at some value of the parameters
  - Take derivative and use it to move the parameters in the direction of the negative gradient
  - Repeat until stopping criteria is met (e.g., gradient close to 0)

<u>Gradient Descent Rule:</u>

$$\underline{\mathbf{w}}_{\mathbf{new}} = \underline{\mathbf{w}}_{\mathbf{old}} - \eta \ \Delta\,(\underline{\mathbf{w}})$$

where

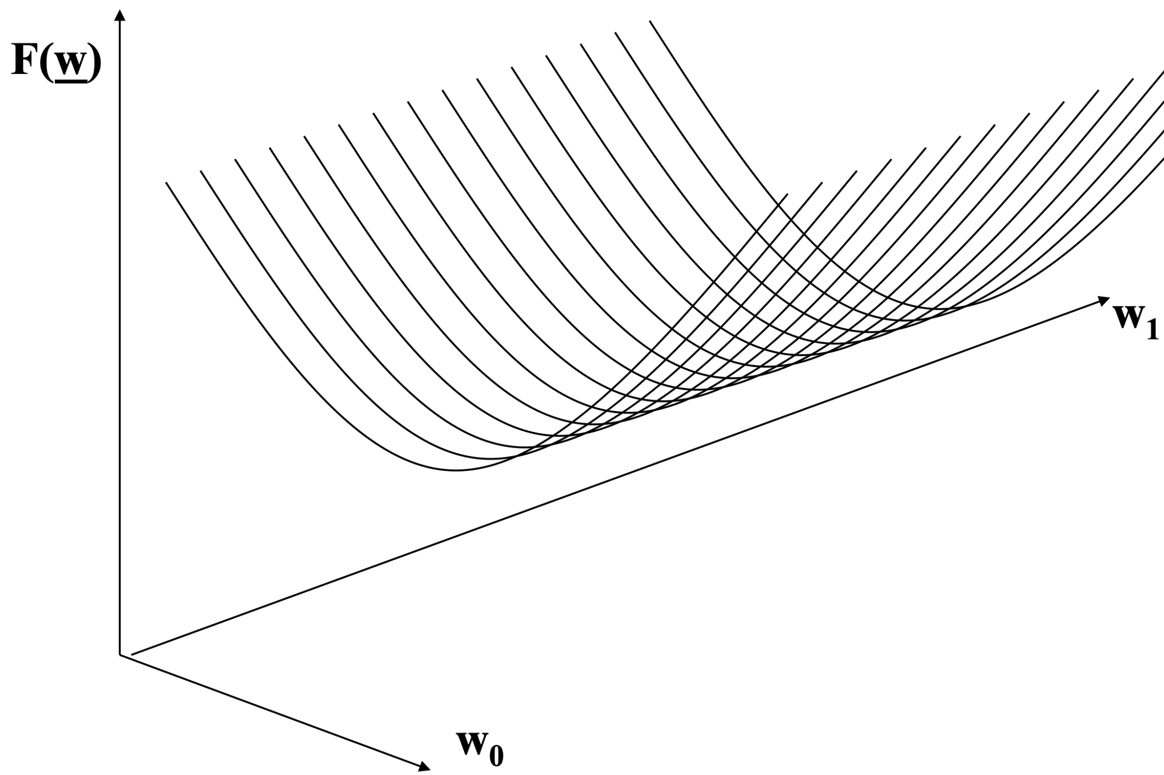
$\Delta\,(\underline{w})$ is the gradient and

η is the learning rate (small, positive)

Notes:

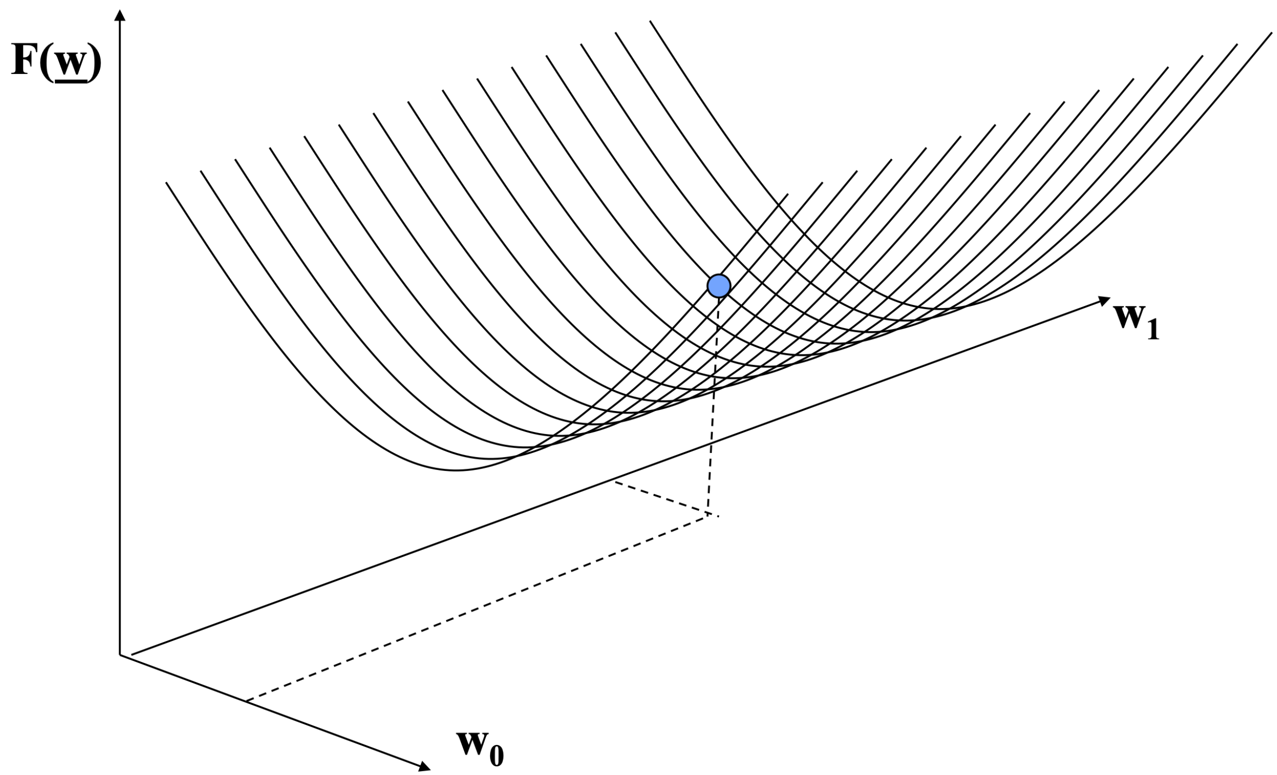
1. This moves us downhill in direction $\Delta\,(\underline{\mathbf{w}})$ (steepest downhill direction)
2. How far we go is determined by the value of η

Illustration of gradient descent



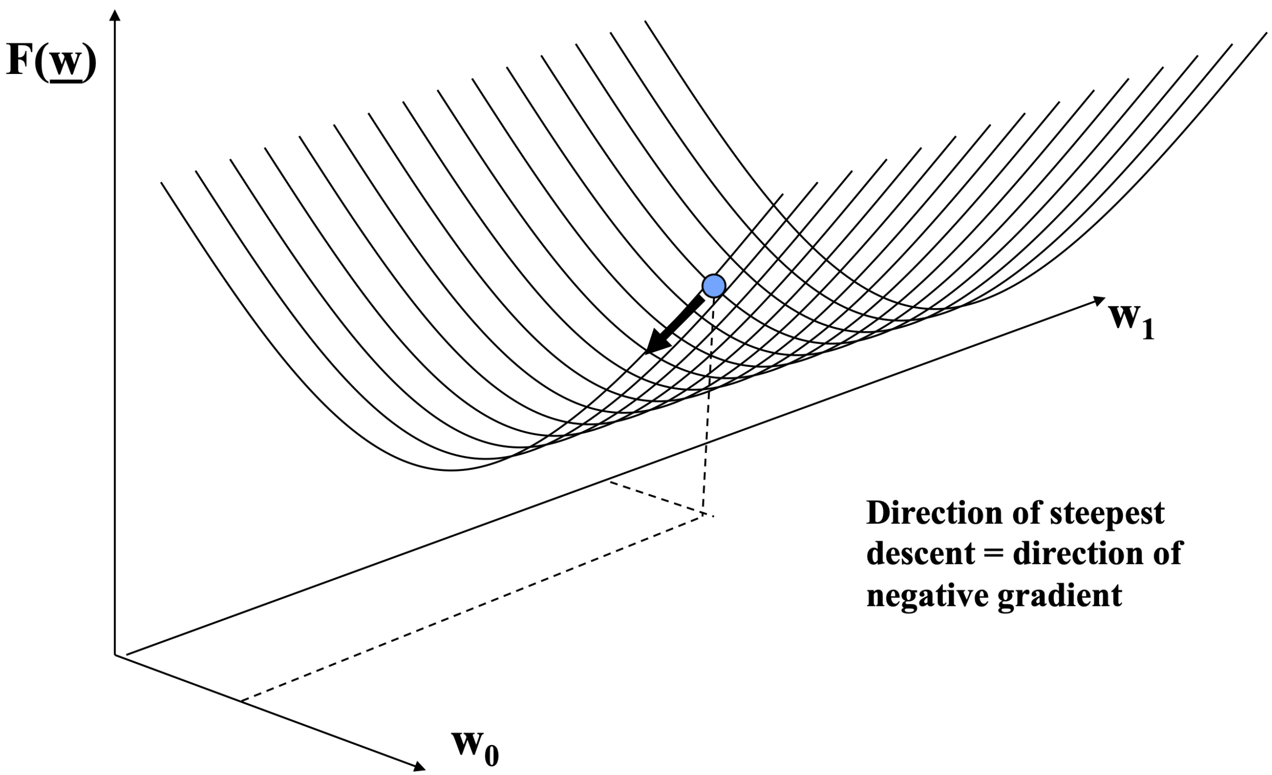
Slides adapted from CS175, UCIrvine, Padhraic Smyth

Illustration of gradient descent



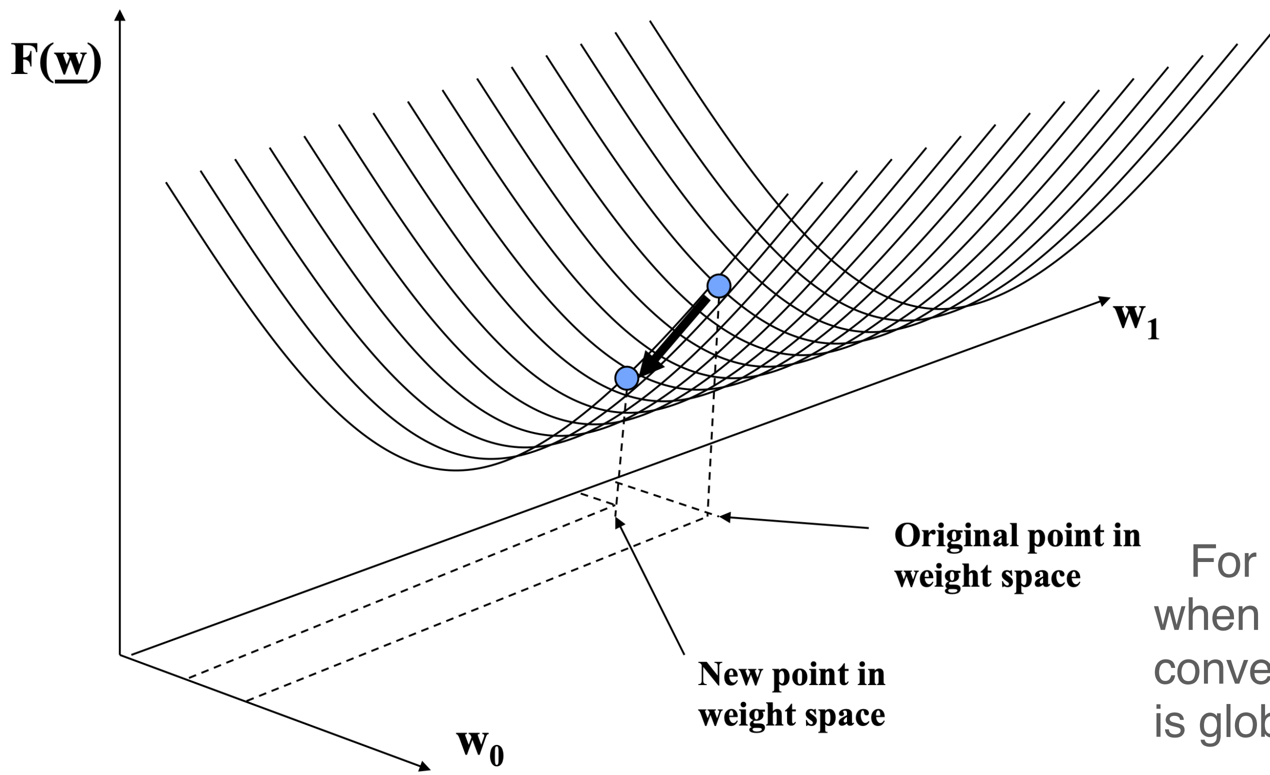
Slides adapted from CS175, UCIrvine, Padhraic Smyth

Illustration of gradient descent



Slides adapted from CS175, UCIrvine, Padhraic Smyth

Illustration of gradient descent



For convex functions, when gradient descent converges, the solution is global minimum.

Slides adapted from CS175, UCIrvine, Padhraic Smyth

Stopping criteria

- Ideally, f'(x)=0…
- In practice…
  - $\| \nabla f(x) \| < \varepsilon$
  - $|f(x_{k+1}) - f(x_k)| < \varepsilon$
  - $\| x_{k+1} - x_k \| < \varepsilon$
  - Maximum number of iterations has been reached

Extensions

- “Higher” order methods
    - Approximate higher order methods
- Constrained optimization
- Accelerated optimization
- Stochastic optimization
- Online setting

Stochastic Gradient descent

- Sample data points (say uniformly at random)
- Use only the gradients on the sampled “batch”
- Lower per-iteration cost
- Usually higher number of iterations
- Is NOT a ”descent” algorithm
- Can possibly help escape local minima (better for non-convex functions)

Analysis

- What is space and time complexity of a single iteration of gradient descent for a *separable* loss function (separable on data points, NOT in context of margin here)?
    - Assume cost of calculating the gradient is g, total number of data points is n
- How many iterations?
    - Second order vs first order
    - Stochastic vs Deterministic
- Total cost associated with the algorithm is sum of the above costs.
- Except for constrained optimization, which may have a “projection” cost associated

Linear Regression

Setup

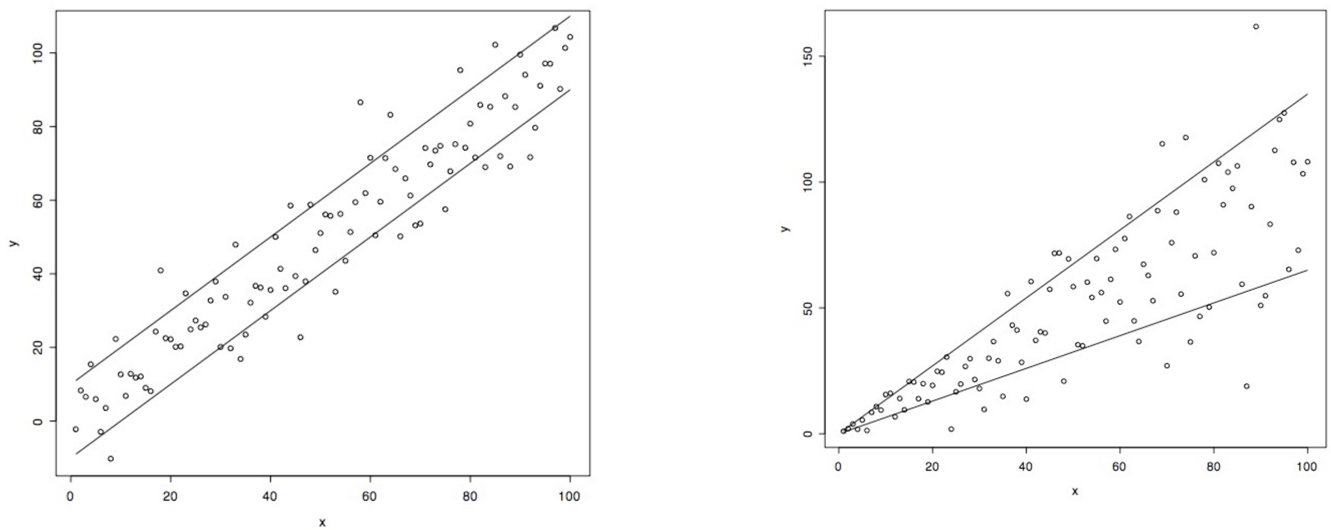
- Data $\{x_i, y_i\}$ Total *n* data points, $x_i \in R^d, y \in R$.
- $y \sim w^\top x$
- Cost function:
  - $\min_w \sum_i \quad |y_i \, - w^\top x_i|$^2

Closed form solution

- $w^{\star} = (X^T X)^{-1} X^T y$
- Prediction: $w^{\star T} x_{test}$
- What is the time and space complexity ?
- What if $X^{\top} X$ is not invertible? When does this happen?

Probabilistic interpretation

- $y = x^{\top}w + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$
- MLE in this model is equivalent to linear regression
- Clearly delineates the underlying "implicit" assumptions in the least squared loss e.g. Homoskedasticity.



- Probabilistic interpretation also quantifies uncertainty under the same assumptions

Generalized Linear Models (GLMs)

- Y is a RV from a distribution in the exponential family e.g. Gaussian, Bernoulli, Poisson etc
- For some function $g()$
  - $g(E[Y]) = (x^T w)$
- Allows for certain algorithms to be applied widely to all GLMs.