Logistic Regression Models

Instructor: Lara J. Martin (she/they)

TA: Omkar Kulkarni (he)

https://laramartin.net/NLP-class/

Slides modified from Dr. Frank Ferraro

Learning Objectives

Model classification problems using logistic regression

Define appropriate features for a logistic regression problem

Review: F1 (or F-score)

Weighted (harmonic) average of **P**recision & **R**ecall

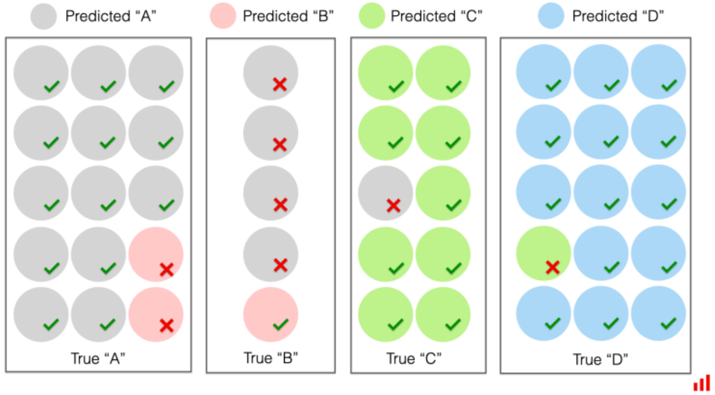
F1 measure: equal weighting between precision and recall

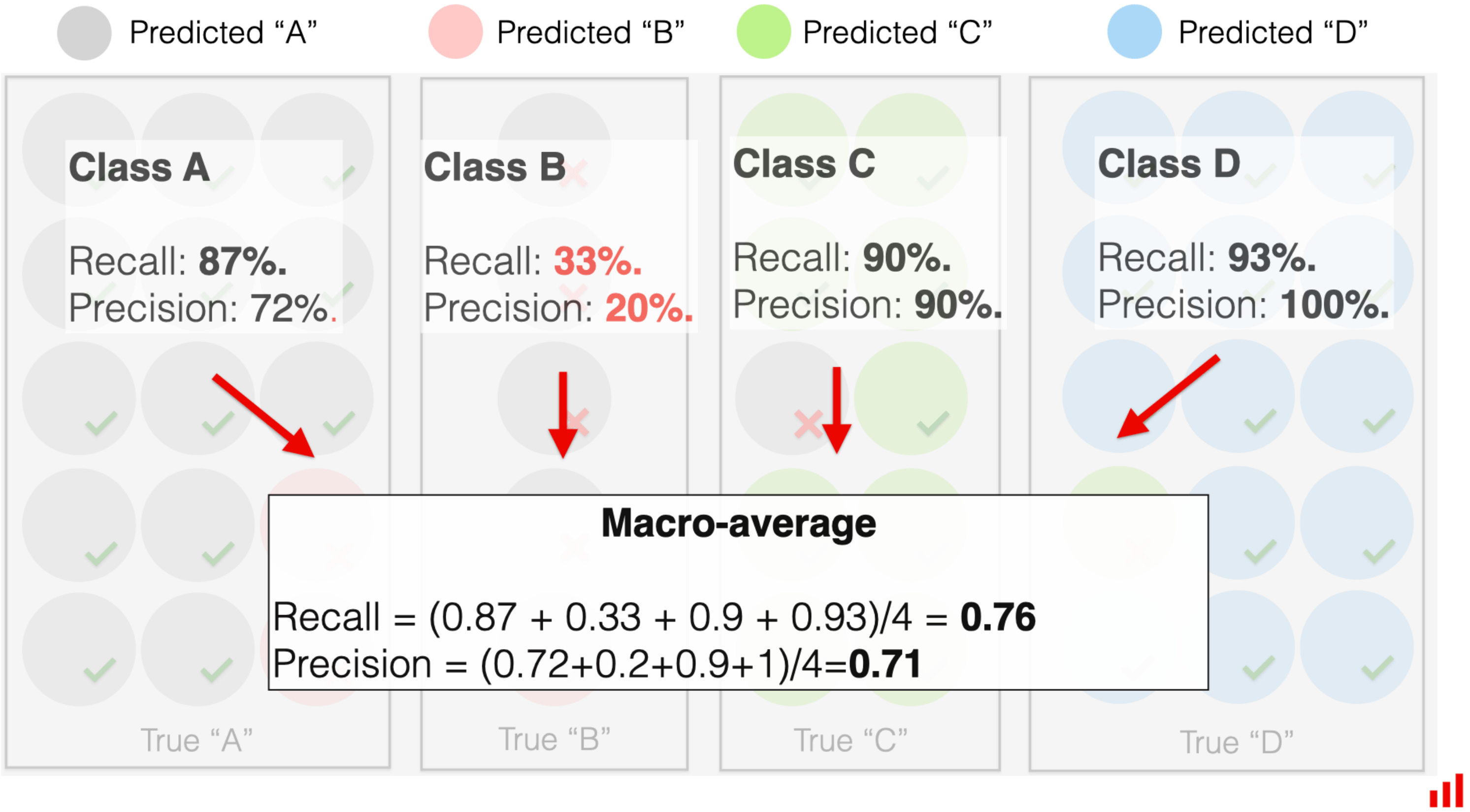
$$F_1 = \frac{2 * P * R}{P + R} = \frac{2 * TP}{2 * TP + FP + FN}$$

(useful when $P = R = 0$)

Each *class* has equal weight

Review: Macro-Average





Class A

Recall: **87%.**
Precision: 72%.

Class B

Recall: **33%.**
Precision: **20%.**

Class C

Recall: **90%.**
Precision: **90%.**

Class D

Recall: **93%.**
Precision: **100%.**

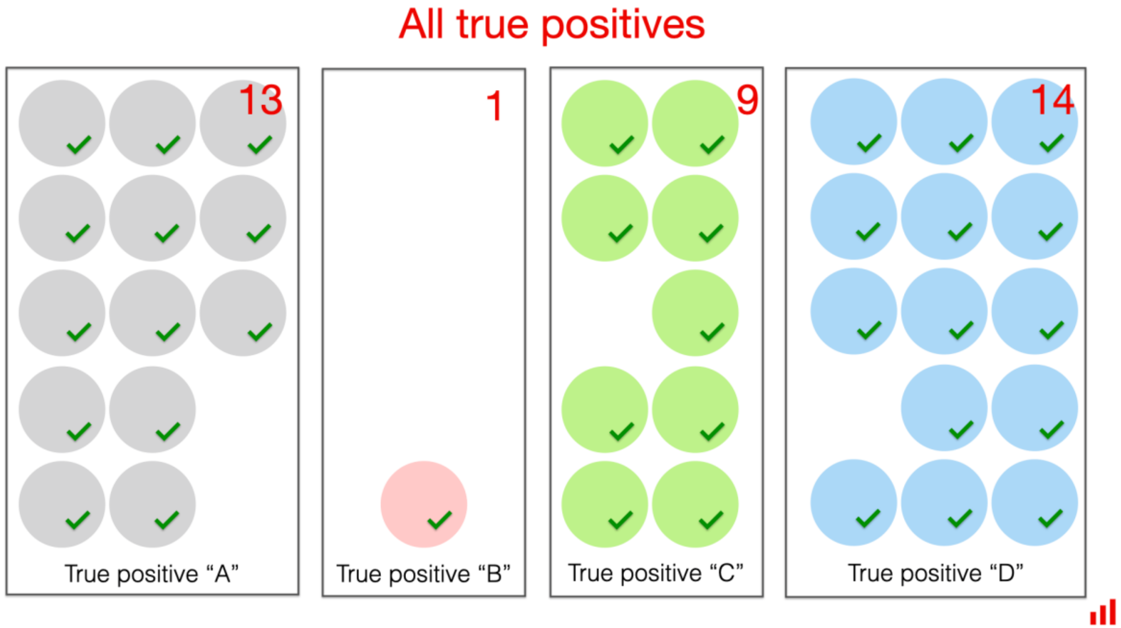
Macro-average

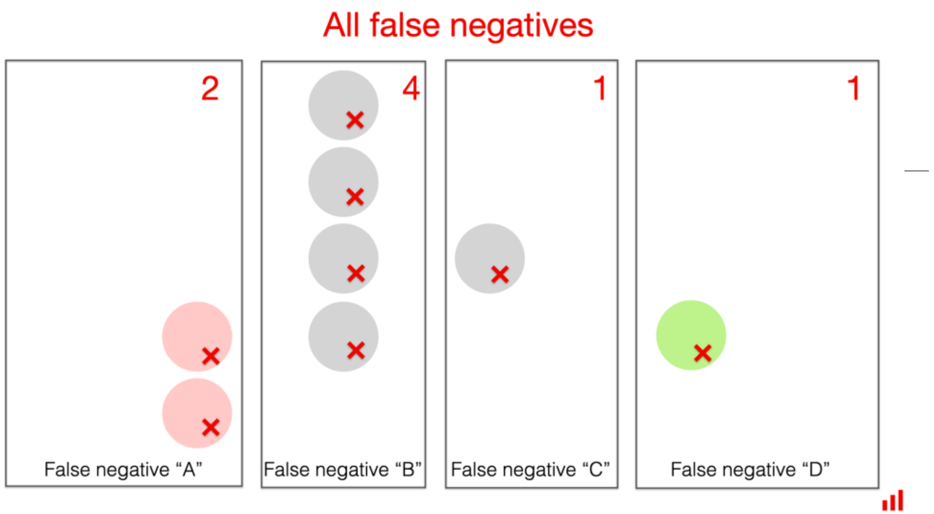
Recall = (0.87 + 0.33 + 0.9 + 0.93)/4 = **0.76**
Precision = (0.72+0.2+0.9+1)/4=**0.71**

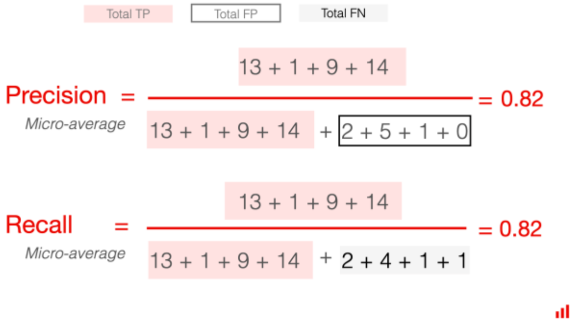
https://www.evidentlyai.com/classification-metrics/multi-class-metrics

Each *instance* has equal weight

Review: Micro-Average

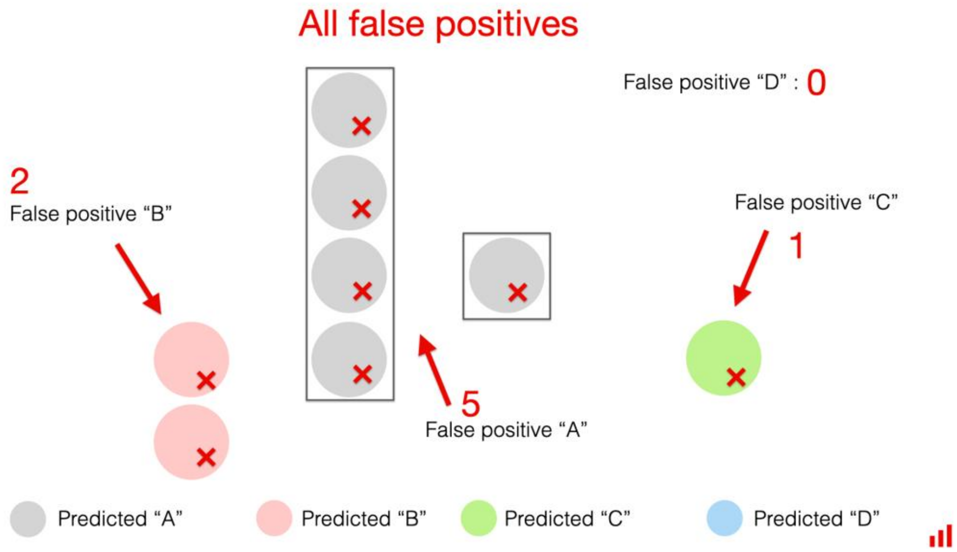






$$\text{Precision}_{\text{Micro-average}} = \frac{13+1+9+14}{13+1+9+14 + \boxed{2+5+1+0}} = 0.82$$

$$\text{Recall}_{\text{Micro-average}} = \frac{13+1+9+14}{13+1+9+14 + 2+4+1+1} = 0.82$$



https://www.evidentlyai.com/classification-metrics/multi-class-metrics

Types of Classification Metrics

AUC https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html

F-score https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Accuracy https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Confusion matrix https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

Precision (can specify macro/micro average) https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

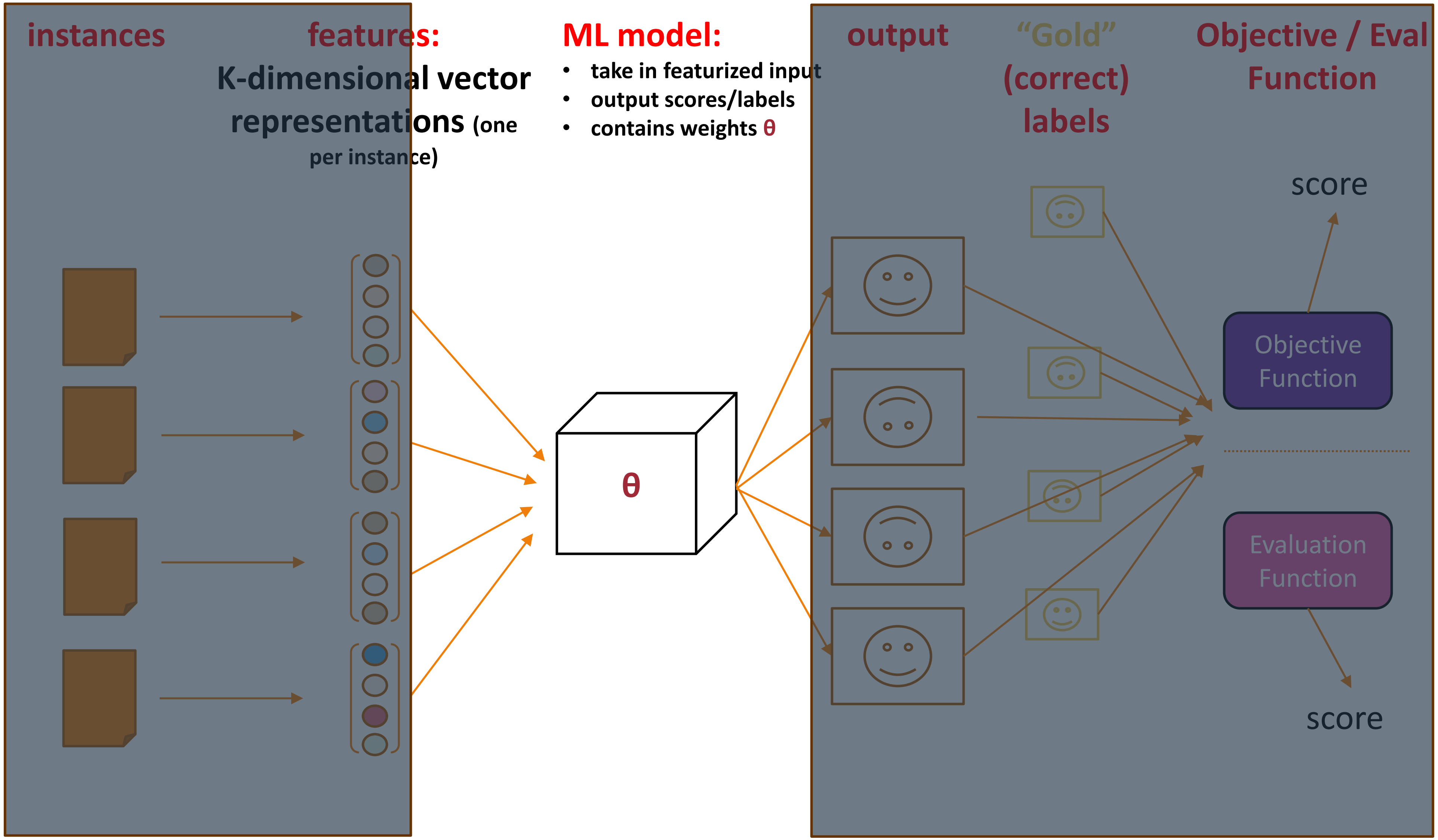
Recall (can specify macro/micro average) https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html

Outline

Maximum Entropy classifiers

- Defining the model
- Defining the objective
- Learning: Optimizing the objective

Defining the Model



instances

features:
K-dimensional vector representations (one per instance)

ML model:

- take in featurized input
- output scores/labels
- contains weights θ

θ

output

“Gold” (correct) labels

Objective / Eval Function

score

Objective Function

Evaluation Function

score

Maxent Models for Classification: Discriminatively or Generatively Trained

Directly model the posterior

$$p(Y \mid X) = \mathbf{maxent}(X; Y)$$

Discriminatively trained classifier

Model the posterior with Bayes rule

$$p(Y \mid X) \propto \mathbf{maxent}(X \mid Y)p(Y)$$

Generatively trained classifier with maxent-based language model

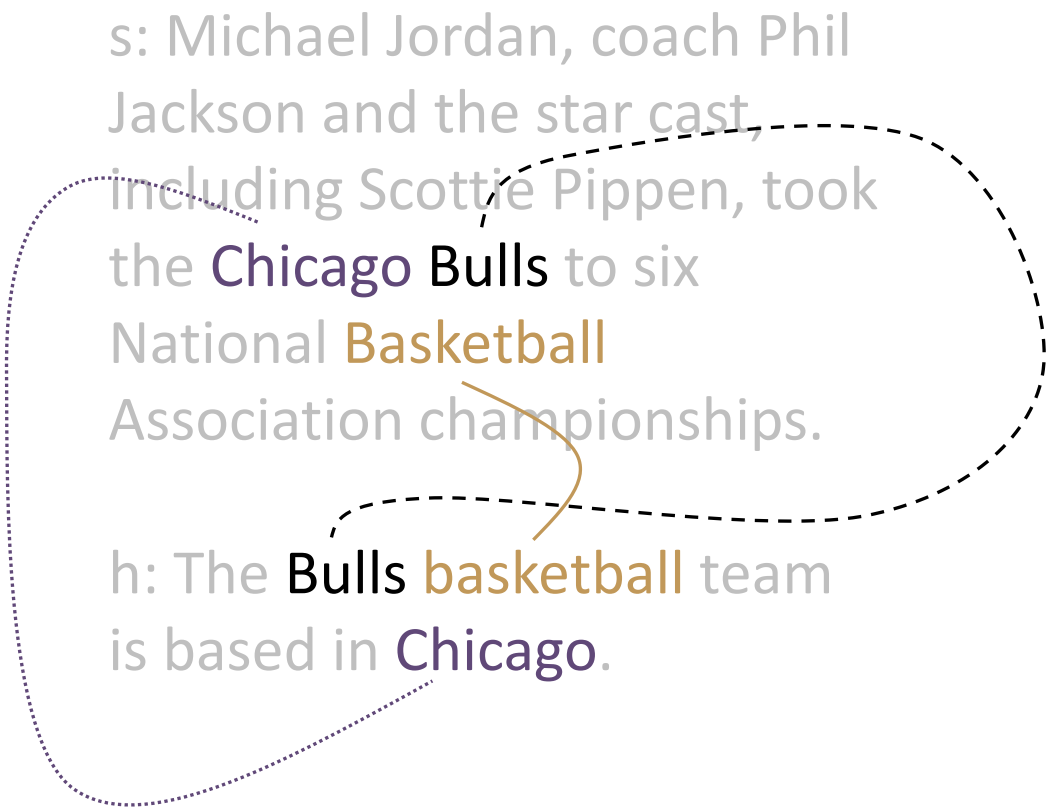
Review:
Discriminative Model using Document Classification Example

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

)

Review: Extracting Features



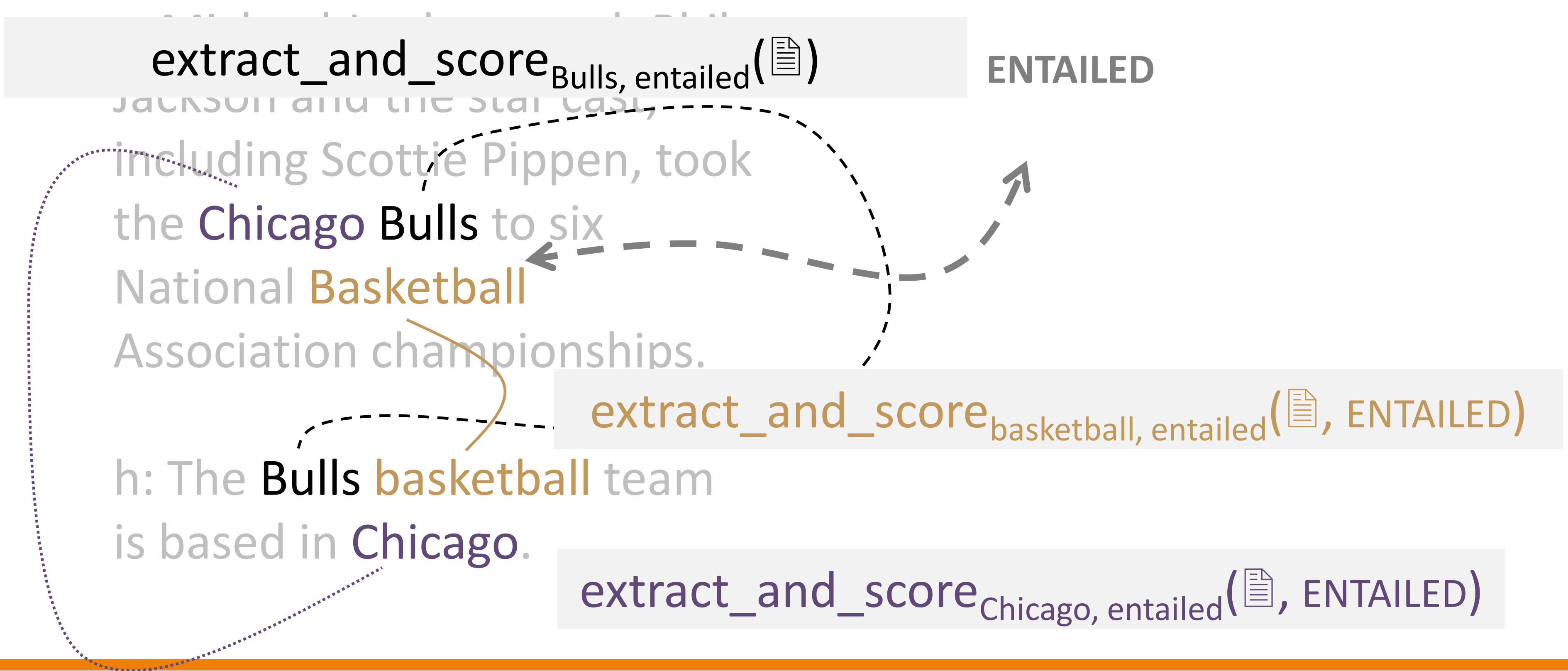
s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

ENTAILED

These extractions are all **features** that have **fired** (likely have some significance)

We need to *score* the different extracted clues.



Michael Jordan and Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

$\text{extract\_and\_score}_{\text{Bulls, entailed}}$(🗎)

ENTAILED

$\text{extract\_and\_score}_{\text{basketball, entailed}}$(🗎, ENTAILED)

$\text{extract\_and\_score}_{\text{Chicago, entailed}}$(🗎, ENTAILED)

Review: Scoring Our Clues

score(

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

, ENTAILED) =

(ignore the feature indexing for now)

$\text{score}_{1\text{ , Entailed}}$(🗎) ✚

$\text{score}_{2\text{ , Entailed}}$(🗎) ✚

$\text{score}_{3\text{ , Entailed}}$(🗎) ✚

…

Review: Turning Scores into Probabilities

score(s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago. , ENTAILED) > score(s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago. , NOT ENTAILED)

p(ENTAILED | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.) > p(NOT ENTAILED | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.)

KEY IDEA

Maxent Modeling

$$\underbrace{p\left(\text{ENTAILED} \;\middle|\; \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took}\\ \text{the Chicago Bulls to six}\\ \text{National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}}\right)}_{\textit{This must be a probability}} \propto$$

Proportional to

$$\overbrace{\color{brown}{G}}^{\textit{Convert through function G? What is this function?}}\left(\overbrace{\text{score}\left(\boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took the}\\ \text{Chicago Bulls to six National}\\ \text{Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}},\ \text{ENTAILED}\right)}^{\textit{This could be any real number}}\right)$$

What function G…

operates on any real number?

is never less than 0?

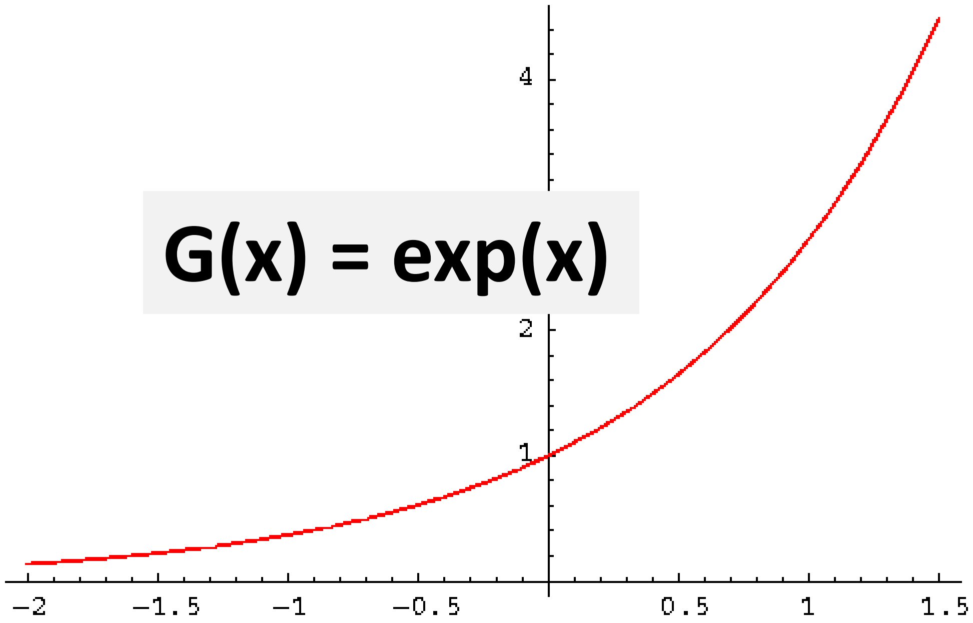
is monotonic? (if $a < b$, then $G(a) < G(b)$)

What function G…

operates on any real number?

is never less than 0?

is monotonic? (if a < b, then G(a) < G(b))



G(x) = exp(x)

Maxent Modeling

p(Entailed | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.) ∝

exp(score(s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago. , Entailed))

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

) ∝

exp(score$_{1\text{ , Entailed}}$(📄) +

score$_{2\text{ , Entailed}}$(📄) +

score$_{3\text{ , Entailed}}$(📄) +

…

))

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

$) \propto$

exp(

$\text{weight}_{1\text{ , Entailed}} * \text{applies}_1($🗎$)$ +

$\text{weight}_{2\text{ , Entailed}} * \text{applies}_2($🗎$)$ +

$\text{weight}_{3\text{ , Entailed}} * \text{applies}_3($🗎$)$ +

…

))

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

) $\propto$

exp(weight$_{1,\text{ Entailed}}$ $*$ applies$_1$(📄) ✚

weight$_{2,\text{ Entailed}}$ $*$ applies$_2$(📄) ✚

weight$_{3,\text{ Entailed}}$ $*$ applies$_3$(📄) ✚

…

))

K different weights…

for K different features

$$\begin{array}{cc} \theta & f(x) \\ \begin{bmatrix} .31 \\ -.5 \\ .1 \\ .002 \\ .522 \\ \dots \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ \dots \end{bmatrix} \end{array}$$

Maxent Modeling

p(ENTAILED |

> s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
> h: The Bulls basketball team is based in Chicago.

) ∝

exp(
weight$_{1, \text{Entailed}}$ * applies$_1$(🗎) +
weight$_{2, \text{Entailed}}$ * applies$_2$(🗎) +
weight$_{3, \text{Entailed}}$ * applies$_3$(🗎) +
…
))

K different weights…

for K different features

multiplied and then summed

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

) ∝

exp(Dot_product of Entailed weight_vec feature_vec(🗎))

K different weights…

for K different features

multiplied and then summed

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

$) \propto$

$$\begin{bmatrix} .31 & -.5 & .1 & .002 & .522 & \ldots \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ \ldots \end{bmatrix}$$

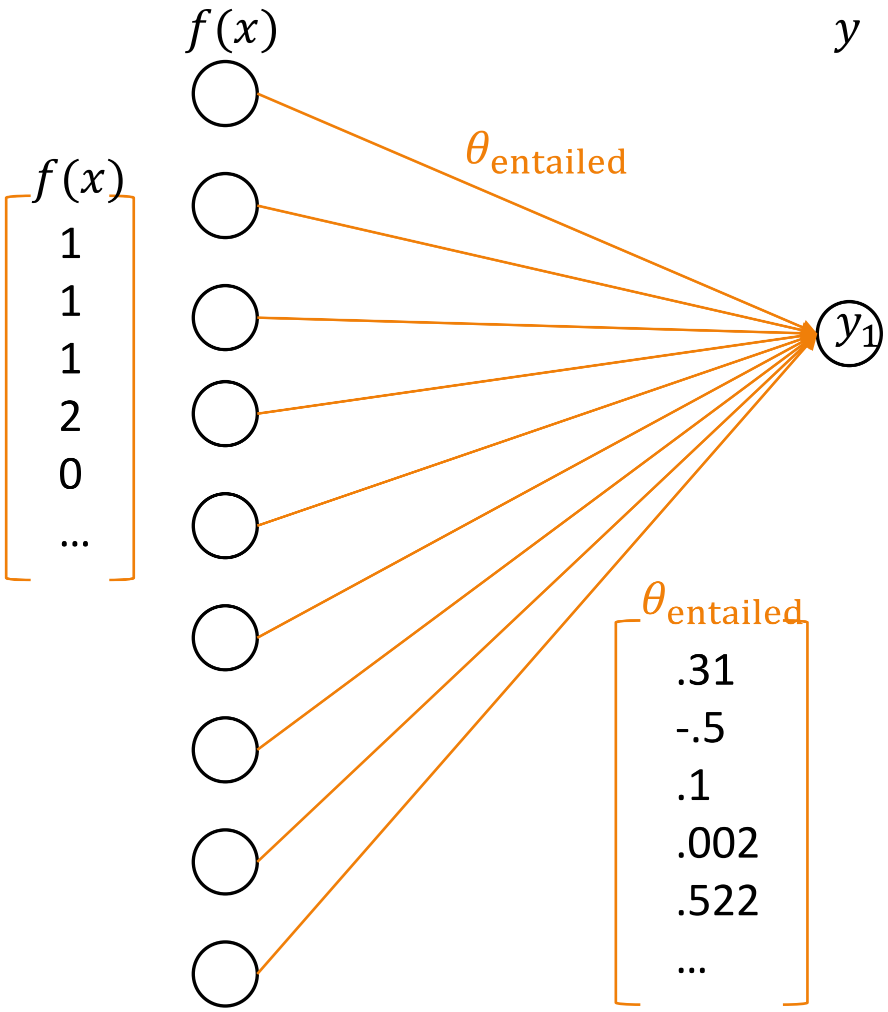
$$\exp\left(\theta^{T}_{\text{ENTAILED}} f(📄) \right)$$

K different weights…

for K different features

multiplied and then summed

Maxent Classifier, schematically



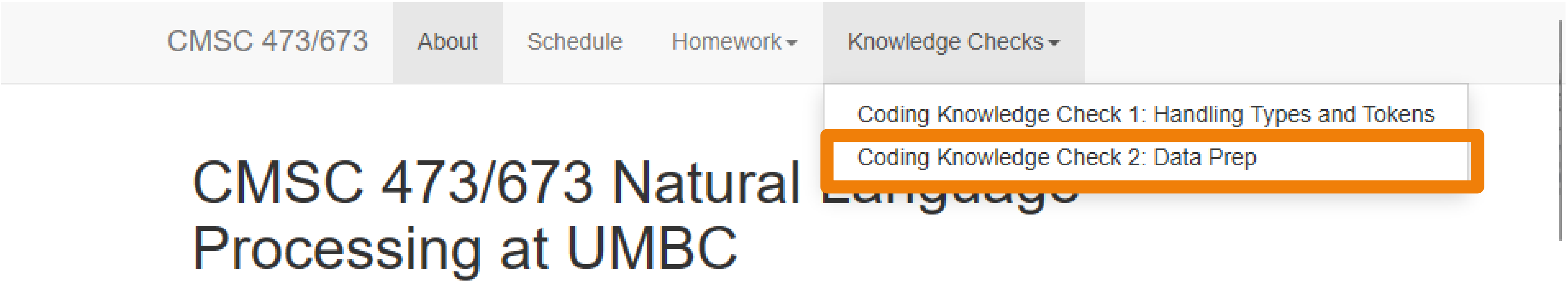
$$f(x) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ \dots \end{bmatrix}$$

$$\theta_{\text{entailed}} = \begin{bmatrix} .31 \\ -.5 \\ .1 \\ .002 \\ .522 \\ \dots \end{bmatrix}$$

$$p(y = \text{entailed}|x) \propto \exp(\theta_{\text{entailed}} f(x))$$

Knowledge Check: Data Prep

https://colab.research.google.com/drive/19yg0EUXQtHozBiSuO6cKOBhoSPzQHgug?usp=sharing


CMSC 473/673
About
Schedule
Homework
Knowledge Checks
Coding Knowledge Check 1: Handling Types and Tokens
Coding Knowledge Check 2: Data Prep

CMSC 473/673 Natural Language Processing at UMBC

Maxent Modeling

$$p\left(\textsc{Entailed} \;\middle|\; \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took}\\ \text{the Chicago Bulls to six}\\ \text{National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}}\right) \propto$$

$$\frac{1}{Z}\exp\left(\theta_{\text{ENTAILED}}^{T} f(\text{🗎})\right)$$

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

$) =$

How do we define Z?

$$\frac{1}{Z}\exp\left(\theta^{T}_{\text{ENTAILED}} f(📄)\right)$$

Normalization for Classification

$$Z =$$

$$\sum_{\text{label } j} \exp(\theta_j^T f(\text{🗎}))$$

$$p(y \mid x) \propto \exp(\theta_y^T f(x))$$

classify doc x with label y in one go

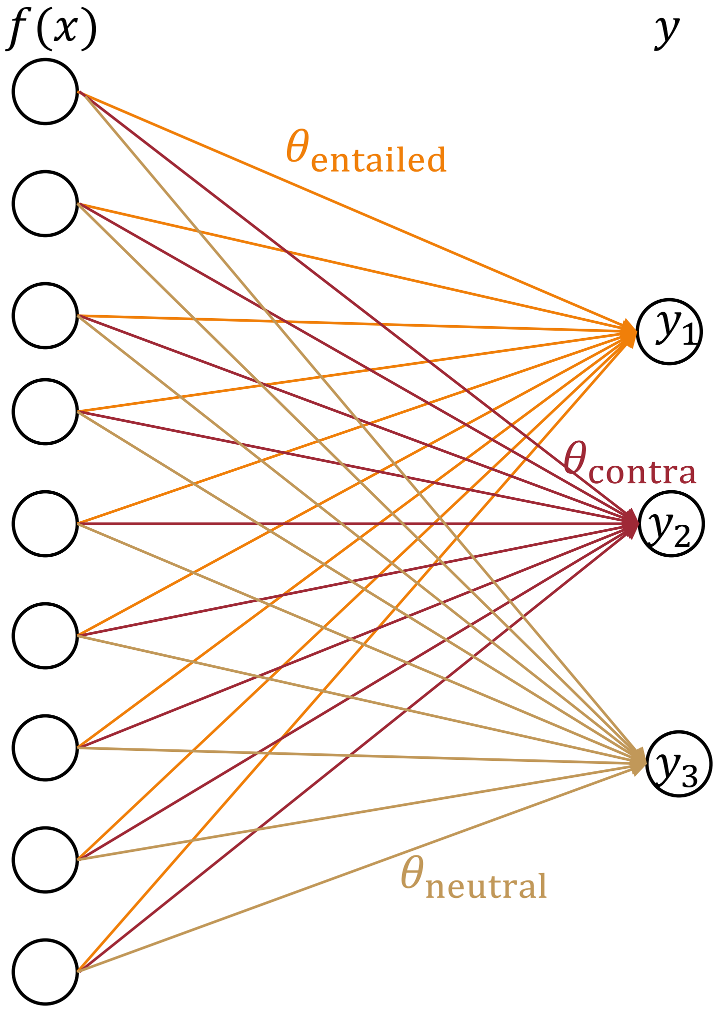
Normalization for Classification (long form)

$$Z = \sum_{\text{label } j} \exp\Big(\text{weight}_{1,j} * \text{applies}_1(\text{🗎}) + \text{weight}_{2,j} * \text{applies}_2(\text{🗎}) + \text{weight}_{3,j} * \text{applies}_3(\text{🗎}) + \ldots \Big)$$

$p(y \mid x) \propto \exp(\theta_y^T f(x))$ *classify doc x with label y in one go*

Multiclass Maxent Classifier, schematically

Why would we want to normalize the weights?



$p(y = \text{entailed}|x) \propto \exp(\theta_{\text{entailed}} f(x))$

$p(y = \text{contra}|x) \propto \exp(\theta_{\text{contra}} f(x))$

$p(y = \text{neutral}|x) \propto \exp(\theta_{\text{neutral}} f(x))$

output:
i = argmax score$_i$
class *i*

Final Equation for Logistic Regression

features $f(x)$ from x that are meaningful;

weights θ (at least one per feature, often one per feature/label combination) to say how important each feature is; and

a way to **form probabilities** from f and θ

$$p(y \mid x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

Different Notation, Same Meaning

$$p(Y = y\,|x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

$$p(Y = y \mid x) \propto \exp(\theta_y^T f(x))$$

$$p(Y \mid x) = \mathrm{softmax}(\theta f(x))$$

Defining Appropriate Features in a Maxent Model

Feature functions help extract useful features (characteristics) of the data

They turn *data* into *numbers*

Features that are not 0 are said to have fired

Generally *templated*

Binary-valued (0 or 1) or real-valued

Representing a Linguistic “Blob”

User-defined

Integer representation/one-hot encoding

Assign each word to some index i, where $0 \leq i < V$

Represent each word w with a V-dimensional **binary** vector e_w, where $e_{w,i} = 1$ and 0 otherwise

Model-produced

Dense embedding

Let E be some *embedding size* (often 100, 200, 300, etc.)

Represent each word w with an E-dimensional **real-valued** vector e_w

Featurization is Similar but…

Vocab types (V) / embedding dimension (E) ➔ number of features (number of "clues")

"Linguistic blob" ➔ Instances to represent

Features are extracted on each instance

Review: Bag-of-words as a Function

Based on some tokenization, turn an input document into an array (or dictionary or set) of its unique vocab items

Think of getting a BOW rep. as a function f

input: Document

output: Container of size E, indexable by each vocab type v

Some Bag-of-words Functions

Kind	Type of f_v	Interpretation
Binary	0, 1	Did *v* appear in the document?
Count-based	Natural number (int >= 0)	How often did *v* occur in the document?
Averaged	Real number (>=0, <= 1)	How often did *v* occur in the document, normalized by doc length?
TF-IDF (term frequency, inverse document frequency)	Real number (>= 0)	How frequent is a word, tempered by how prevalent it is across the corpus (to be covered later!)
…		

Q: Is this a reasonable representation?

Q: What are some tradeoffs (benefits vs. costs)?

Useful Terminology: n-gram

Within a larger string (e.g., sentence),
a contiguous sequence of n items (e.g., words)

Colorless green ideas sleep furiously

n	Commonly called	History Size (Markov order)	Example n-gram ending in "furiously"
1	unigram	0	furiously
2	bigram	1	sleep furiously
3	trigram (3-gram)	2	ideas sleep furiously
4	4-gram	3	green ideas sleep furiously
n	n-gram	n-1	$w_{i-n+1} \ldots w_{i-1}\ w_i$

Templated Features

Define a feature fclue(🗎) for each clue you want to consider

The fclue fires if the clue applies to/can be found in 🗎

Clue is often a target phrase (an n-gram)

Maxent Modeling: Templated Binary Feature Functions

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

) ∝

exp(

$\text{weight}_{1\text{, Entailed}}$ * applies_1(🗎) +

$\text{weight}_{1\text{, Entailed}}$ * applies_2(🗎) +

$\text{weight}_{1\text{, Entailed}}$ * applies_3(🗎) +

…

))

$$\text{applies}_{\text{target}}(🗎) = \begin{cases} 1, \text{target } matches \text{ 🗎} \\ 0, \quad\quad \text{otherwise} \end{cases}$$

binary

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(\text{🗎}) = \begin{cases} 1, \text{target } \mathit{matches} \text{ 🗎} \\ 0, \qquad \text{otherwise} \end{cases}$$

$$\text{applies}_{\text{ball}}\ (\text{🗎}) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of 🗎} \\ 0, \qquad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

2. How many features are defined if bigram targets are used (w/ each label)?

3. How many features are defined if unigram and bigram targets are used (w/ each label)?

Outline

Maximum Entropy classifiers

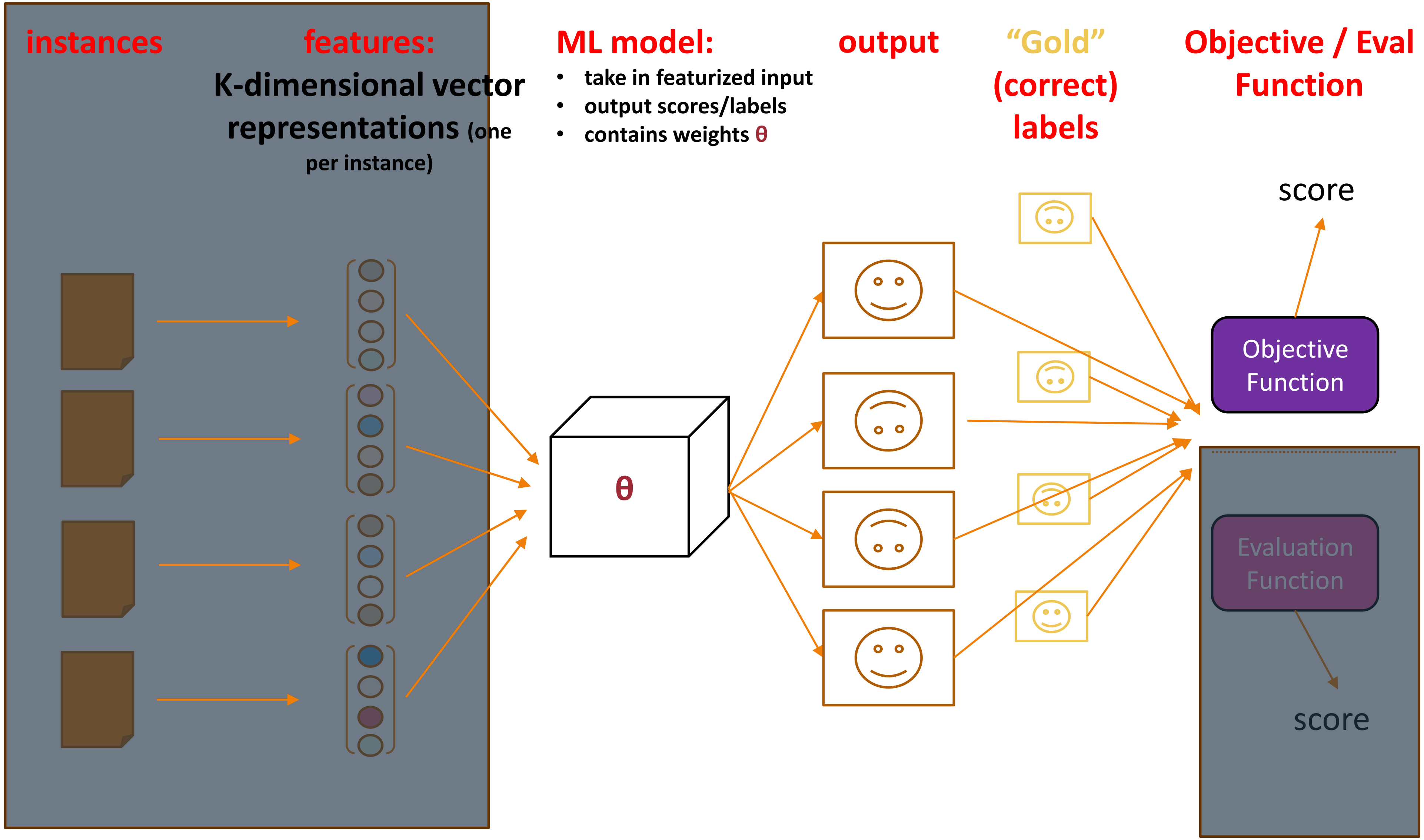
- Defining the model
- Defining the objective
- Learning: Optimizing the objective

$$p_\theta(y \mid x)$$ probabilistic model

$$\Big\Downarrow$$

$$F(\theta; x, y)$$ **objective**

Defining the Objective



instances

features: K-dimensional vector representations (one per instance)

ML model:

- take in featurized input
- output scores/labels
- contains weights θ

output

"Gold" (correct) labels

Objective / Eval Function

Primary Objective: Likelihood

Goal: *maximize* the score your model gives to the training data it observes

This is called the **likelihood of your data**

In classification, this is p(label | 🖹)

For language modeling, this is p(word | history of words)

Objective = Full Likelihood? (Classification)

$$\prod_i \overbrace{p_\theta(y_i|x_i)}^{\text{Our goal probability}} \propto \prod_i \overbrace{\exp(\theta_{y_i}^T f(x_i))}^{\text{Our maxent equation}}$$

These values can have very small magnitude ➔ underflow

Differentiating this product could be a pain

Logarithms

(0, 1] ➔ (-∞, 0]

Products ➔ Sums

log(ab) = log(a) + log(b)

log(a/b) = log(a) – log(b)

Inverse of exp

log(exp(x)) = x

How might you find the log of this?

$$\prod_i p_\theta(y_i|x_i)$$

Log-Likelihood (Classification)

Wide range of (negative) numbers
Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

Products ➔ Sums

log(ab) = log(a) + log(b)

log(a/b) = log(a) – log(b)

Maximize Log-Likelihood (Classification)

Original maxent equation

$$\frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

Differentiating this becomes nicer (even though Z depends on θ)

Inverse of exp

log(exp(x)) = x

$$= \sum_i \overbrace{\theta_{y_i}^T f(x_i) - \log Z(x_i)}$$

Log-Likelihood (Classification)

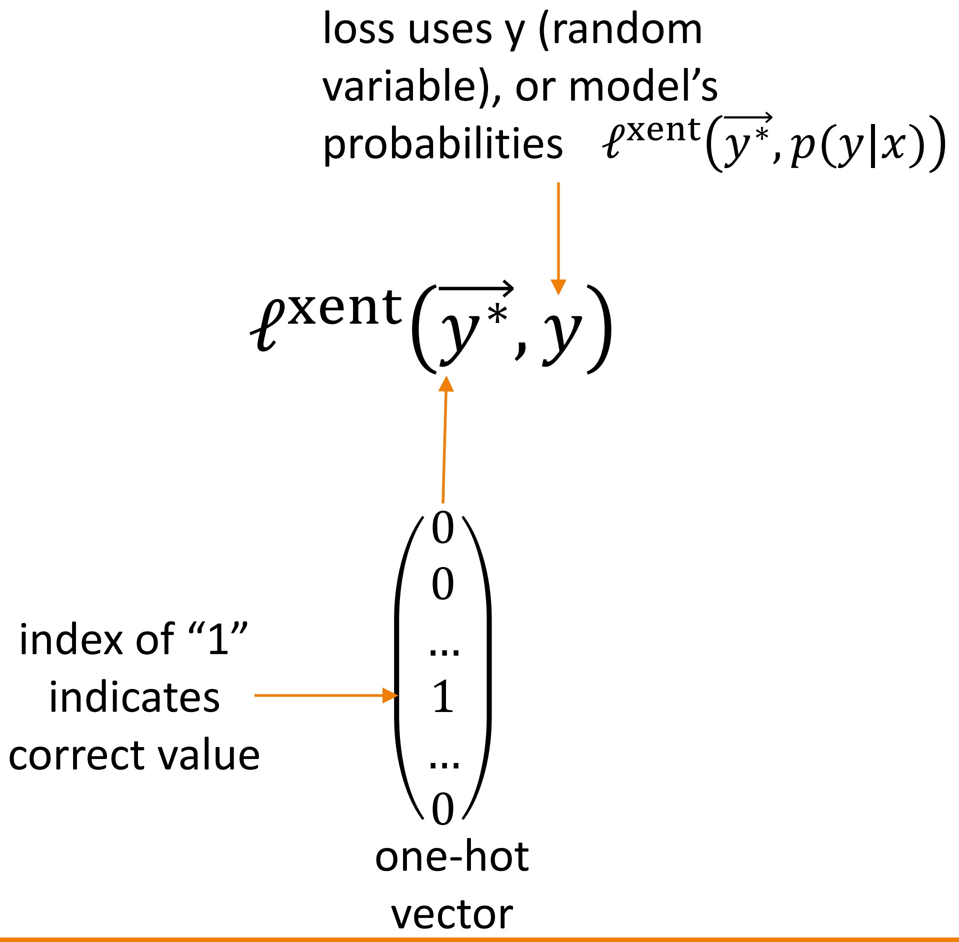
Wide range of (negative) numbers
Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

$$= \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

$$= F(\theta)$$

Equivalent Version 2: *Minimize* Cross Entropy Loss



loss uses y (random variable), or model's probabilities $\ell^{\text{xent}}\left(\overrightarrow{y^*}, p(y|x)\right)$

$$\ell^{\text{xent}}\left(\overrightarrow{y^*}, y\right)$$

$$\begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$$

one-hot vector

index of "1" indicates correct value

Cross entropy:
How much $\hat{y}$ differs from the true y

objective is convex (when f(x) is not learned)

Equivalent Version 2:
Minimize Cross Entropy Loss

loss uses y (random variable), or model's probabilities $\ell^{\text{xent}}\left(\overrightarrow{y^*}, p(y|x)\right)$

$$\ell^{\text{xent}}\left(\overrightarrow{y^*}, y\right) = -\sum_k \overrightarrow{y^*}[k] \log p(y = k|x)$$

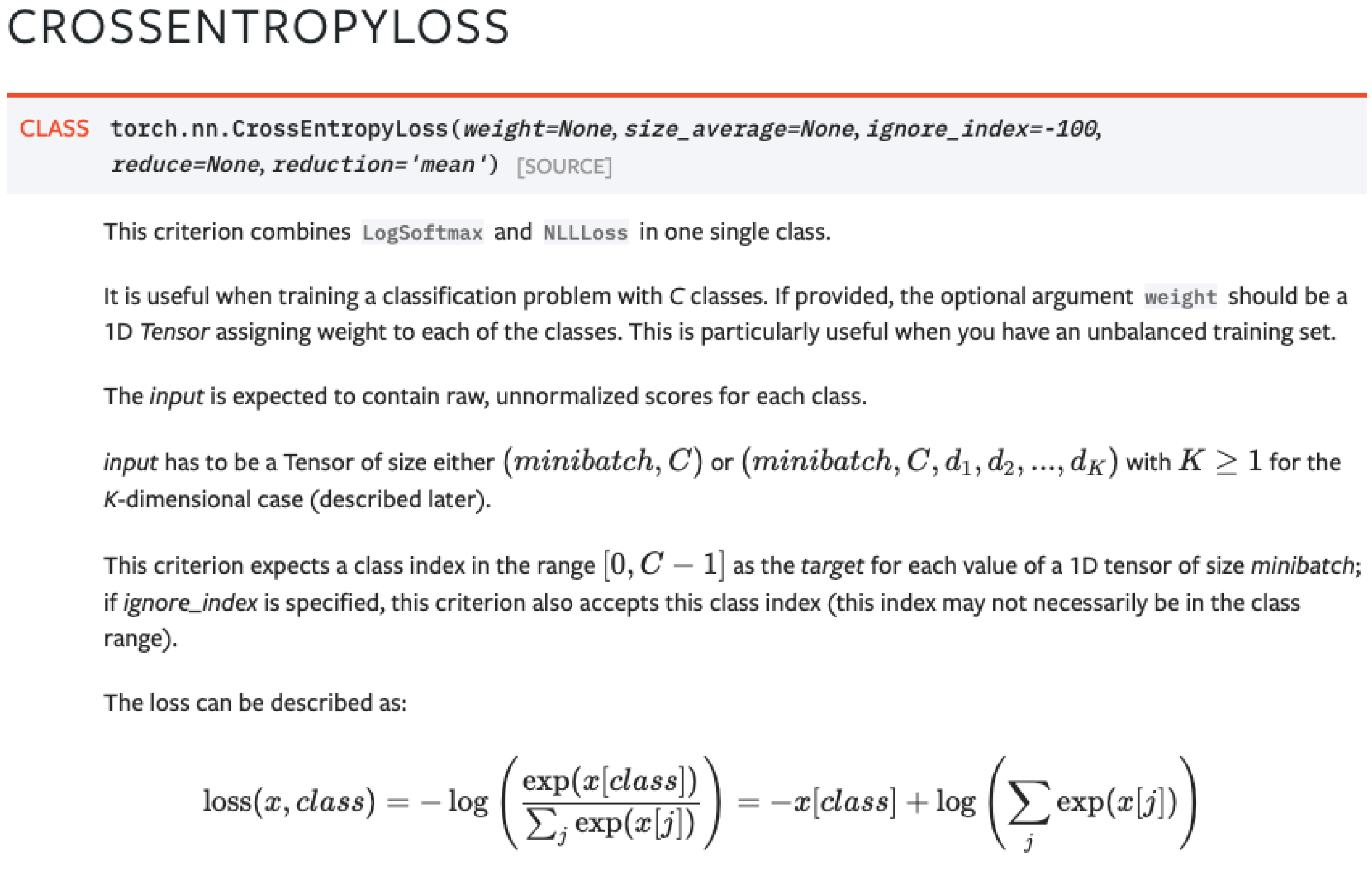
$$\begin{pmatrix} 0 \\ 0 \\ \ldots \\ 1 \\ \ldots \\ 0 \end{pmatrix}$$

index of “1” indicates correct value

one-hot vector

Classification Log-likelihood ≅ Cross Entropy Loss

$$F(\theta) = \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$



CROSSENTROPYLOSS

CLASS `torch.nn.CrossEntropyLoss(weight=None, size_average=None, ignore_index=-100, reduce=None, reduction='mean')` [SOURCE]

This criterion combines `LogSoftmax` and `NLLLoss` in one single class.

It is useful when training a classification problem with C classes. If provided, the optional argument `weight` should be a 1D *Tensor* assigning weight to each of the classes. This is particularly useful when you have an unbalanced training set.

The *input* is expected to contain raw, unnormalized scores for each class.

input has to be a Tensor of size either $(minibatch, C)$ or $(minibatch, C, d_1, d_2, ..., d_K)$ with $K \geq 1$ for the *K*-dimensional case (described later).

This criterion expects a class index in the range $[0, C-1]$ as the *target* for each value of a 1D tensor of size *minibatch*; if *ignore_index* is specified, this criterion also accepts this class index (this index may not necessarily be in the class range).

The loss can be described as:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

Preventing Extreme Values

Likelihood on its own can lead to overfitting and/or extreme values in the probability computation

$$F(\theta) = \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

Learn the parameters based on some (fixed) data/examples

Regularization: Preventing Extreme Values

$$F(\theta) = \sum_{i} \theta_{y_i}^{T} f(x_i) - \log Z(x_i)$$

With fixed/predefined features, the values of θ determine how “good” or “bad” our objective learning is

Regularization: Preventing Extreme Values

$$F(\theta) = \left(\sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i) \right) - R(\theta)$$

With fixed/predefined features, the values of θ determine how “good” or “bad” our objective learning is

- Augment the objective with a **regularizer**
  - This regularizer places an inductive bias (or, prior) on the general “shape” and values of θ

(Squared) L2 Regularization

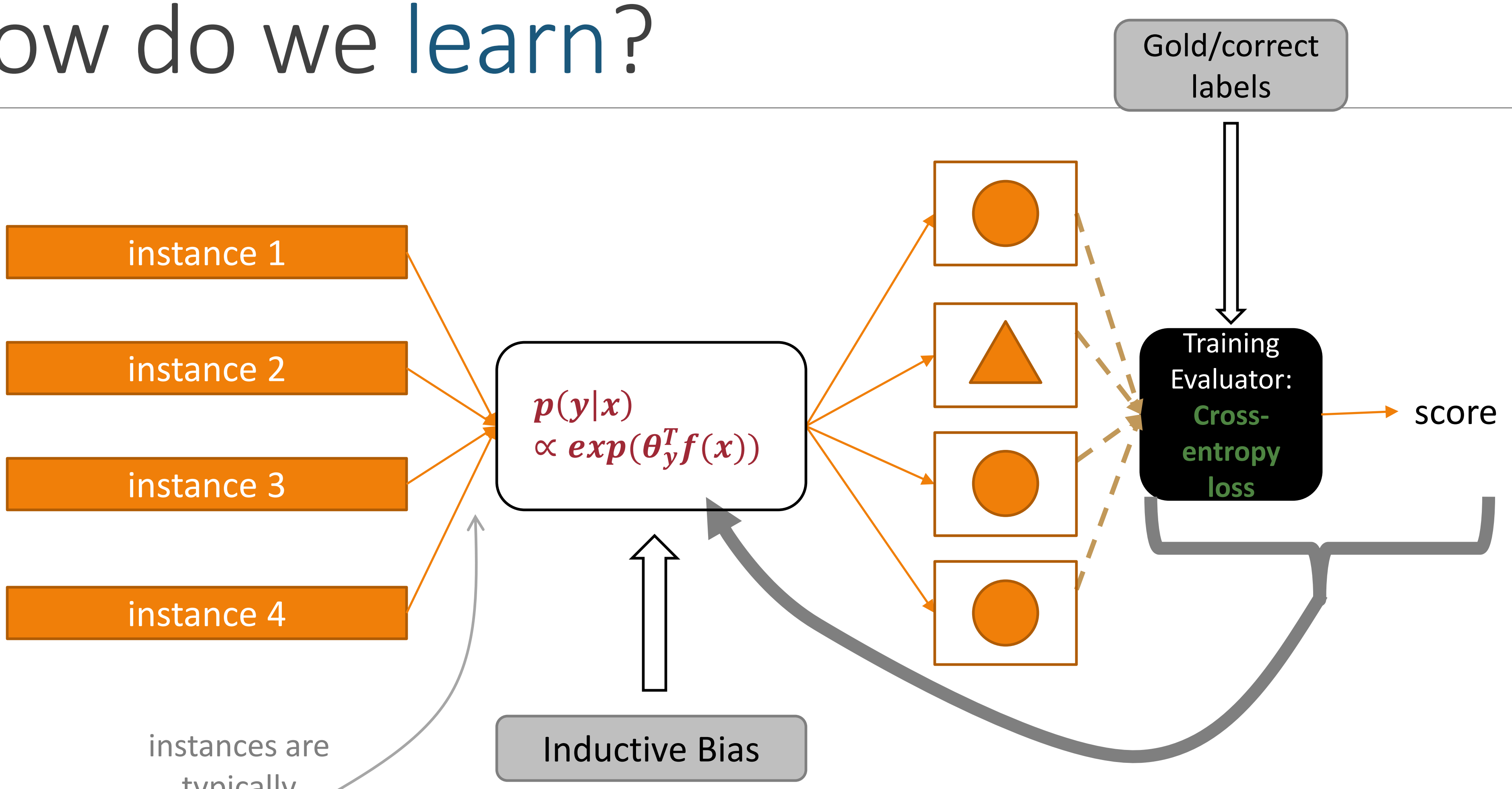
$$R(\theta) = \|\theta\|_2^2 = \sum_k \theta_k^2$$

Outline

Maximum Entropy classifiers

- Defining the model
- Defining the objective
- Learning: Optimizing the objective

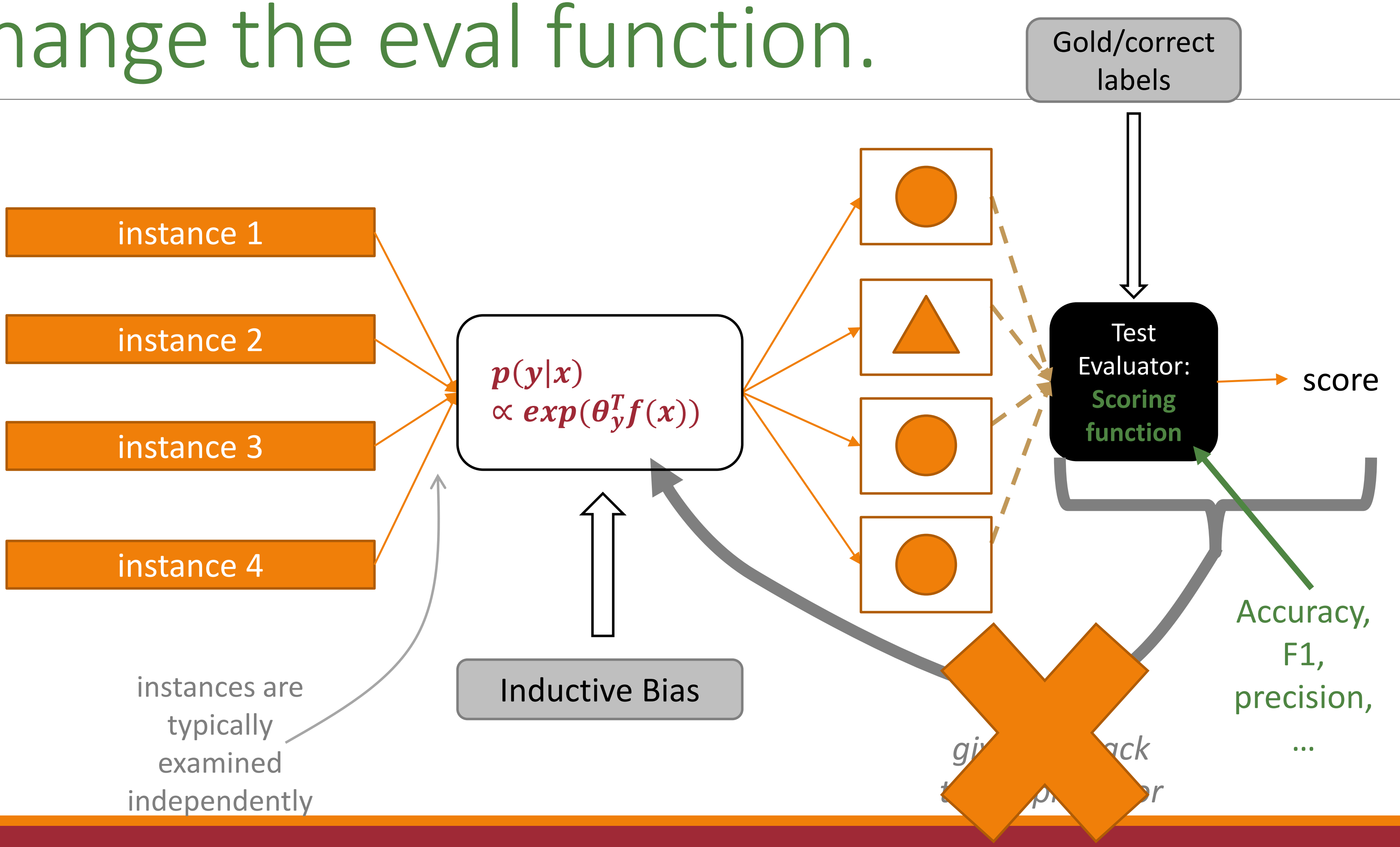
How do we learn?



instances are typically examined independently

give feedback to the predictor

How do we evaluate (or use)?
Change the eval function.



Gold/correct labels

instance 1

instance 2

instance 3

instance 4

$p(y|x)$
$\propto exp(\theta_y^T f(x))$

Test Evaluator: Scoring function

score

Inductive Bias

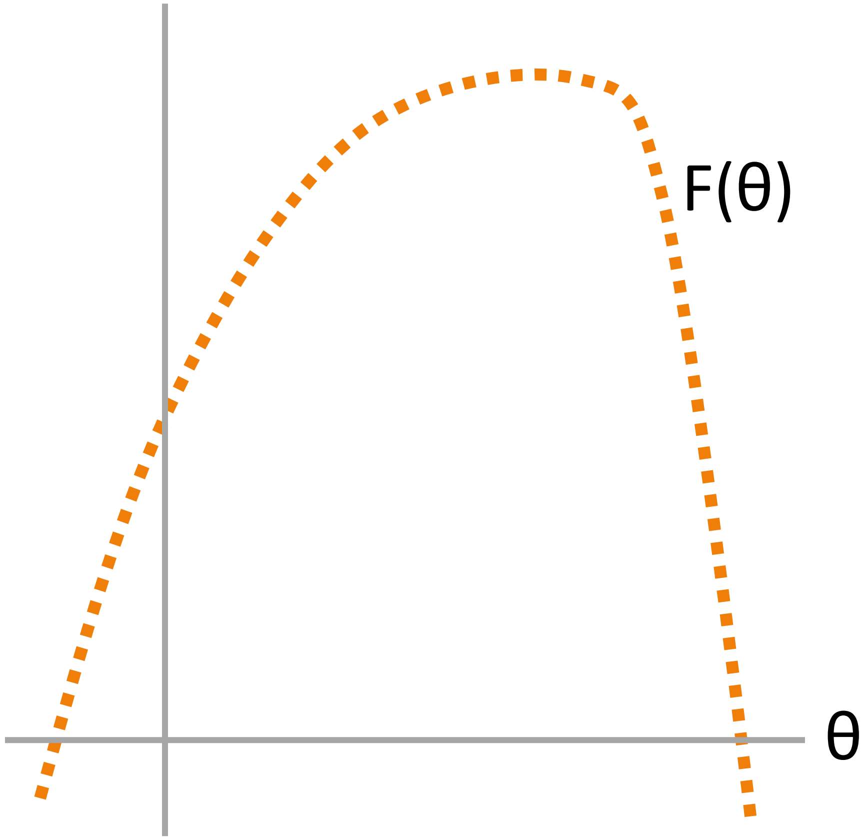
instances are typically examined independently

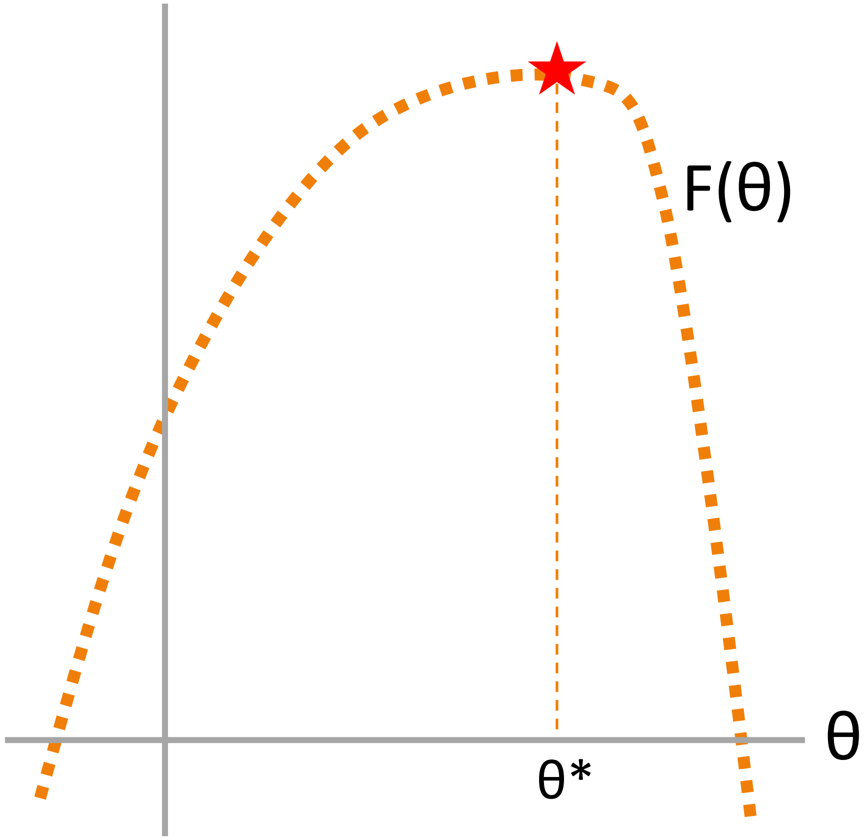
gi ack
t pr or

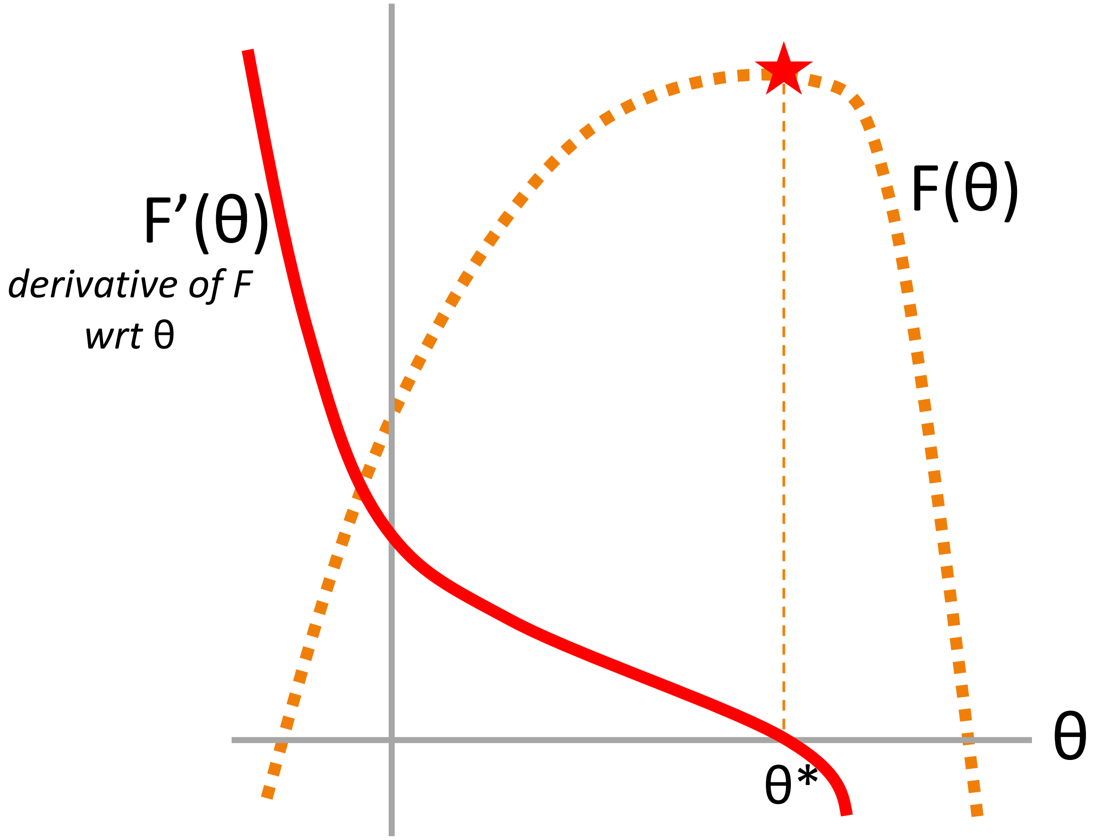
Accuracy, F1, precision, …

How will we optimize F(θ)?

Calculus.


F(θ)
θ

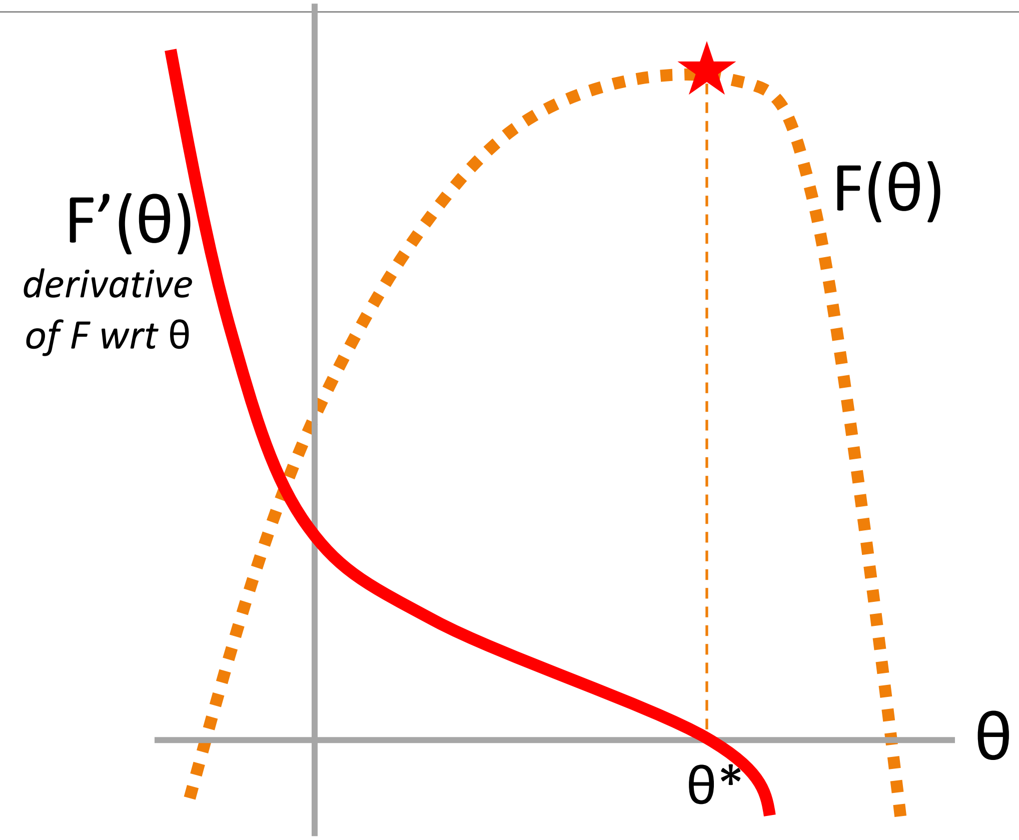

F(θ)
θ
θ*


F'(θ)
derivative of F
wrt θ
F(θ)
θ
θ*

Example
(Best case, solve for roots of the derivative)

$$\begin{tikzcd} F(x) = -(x-2)^2 \arrow[d, "\textit{differentiate}"] \\ F'(x) = -2x + 4 \arrow[d, "\textit{Solve } F'(x) = 0"] \\ x = 2 \end{tikzcd}$$

What if you can’t find the roots? Follow the derivative



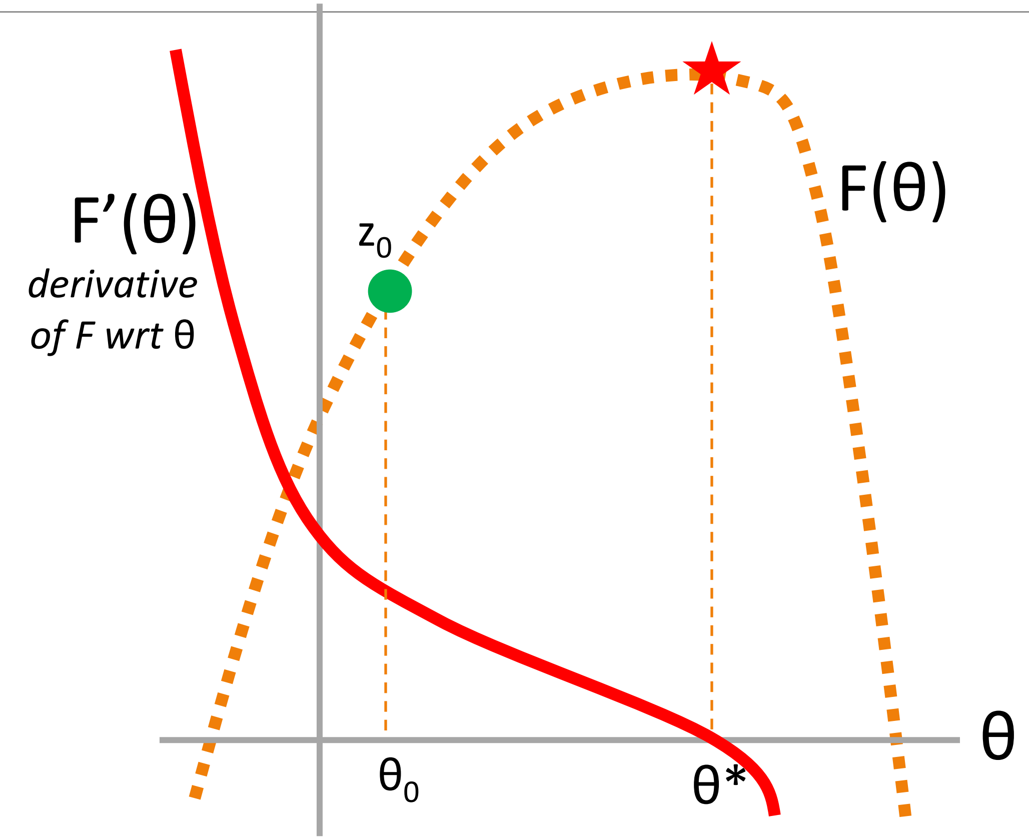
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

Until converged:

1. Get value $z_t = F(\theta_t)$



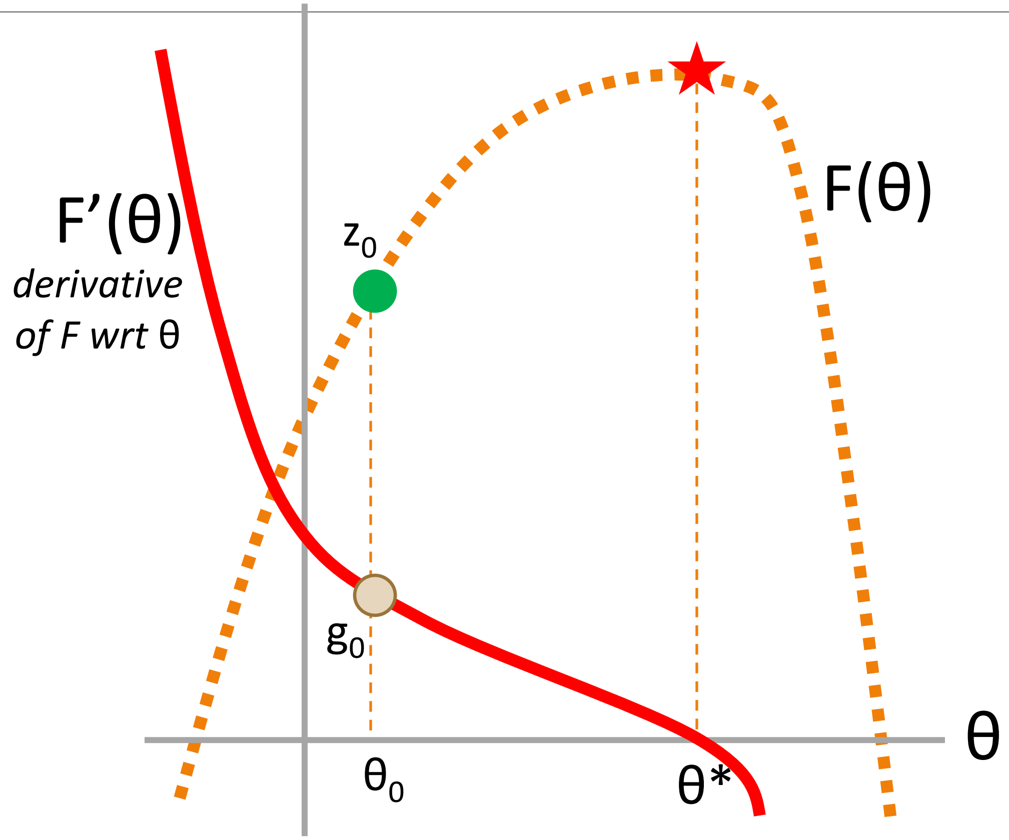
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

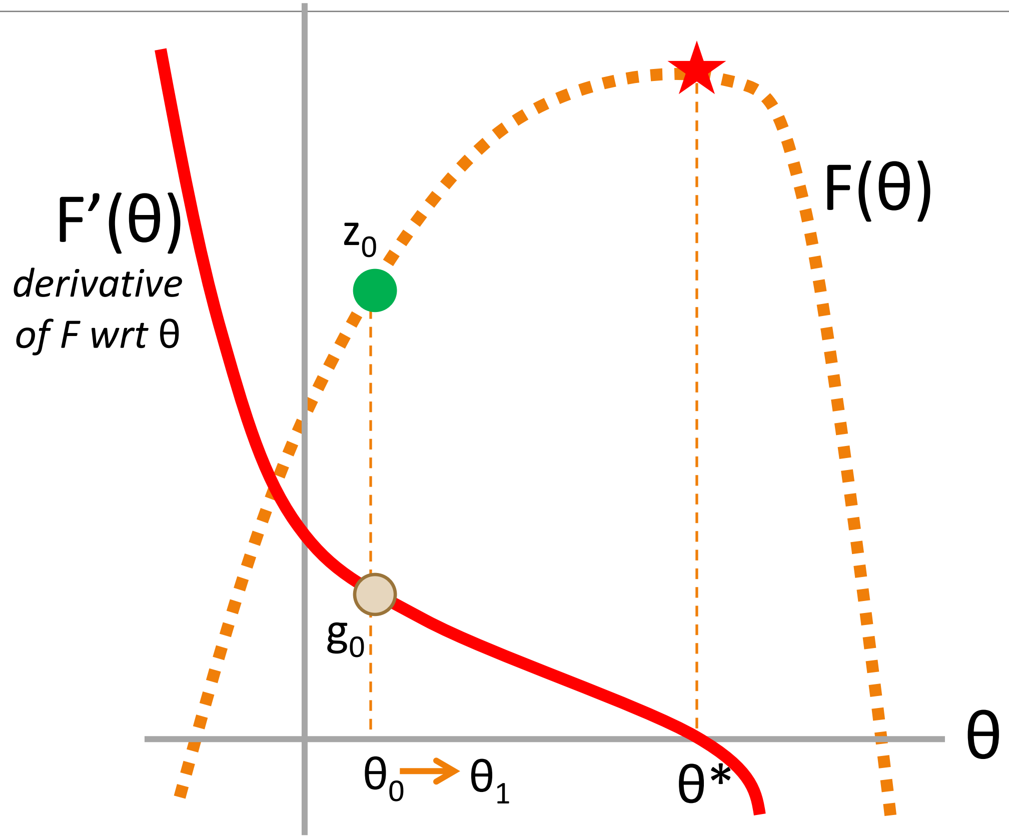
Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$



What if you can’t find the roots? Follow the derivative

```
Set t = 0
Pick a starting value $\theta_t$
Until converged:
1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get scaling factor $\rho_t$
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1
```



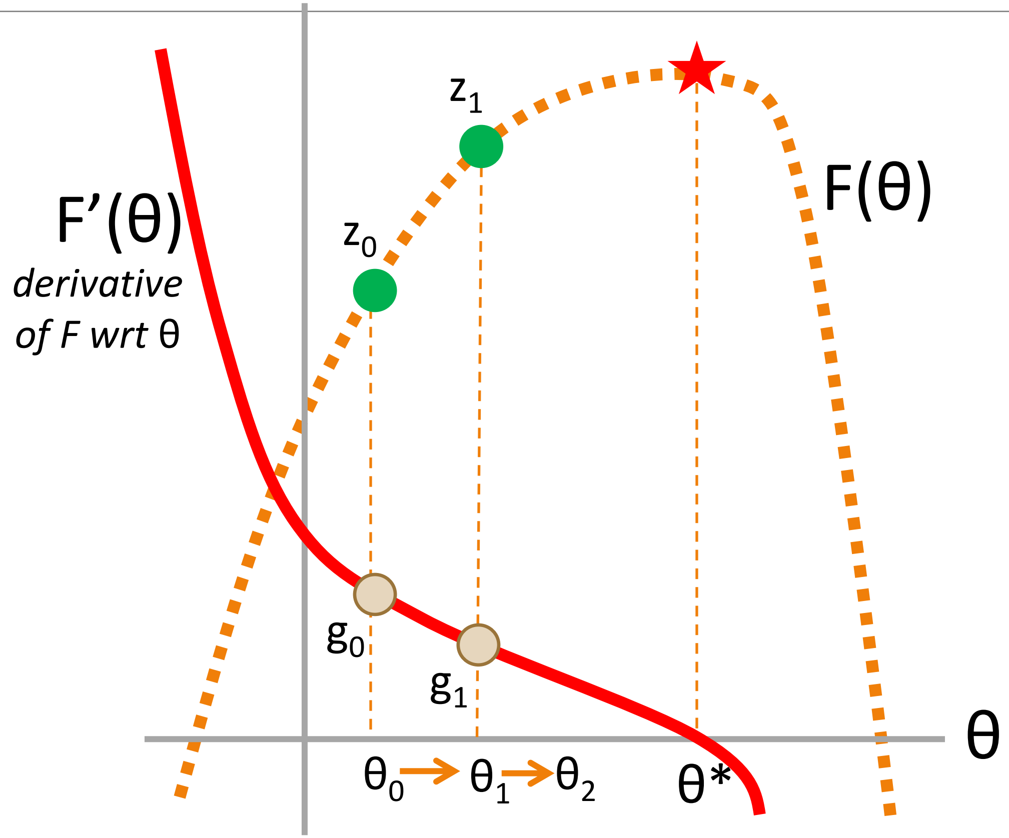
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

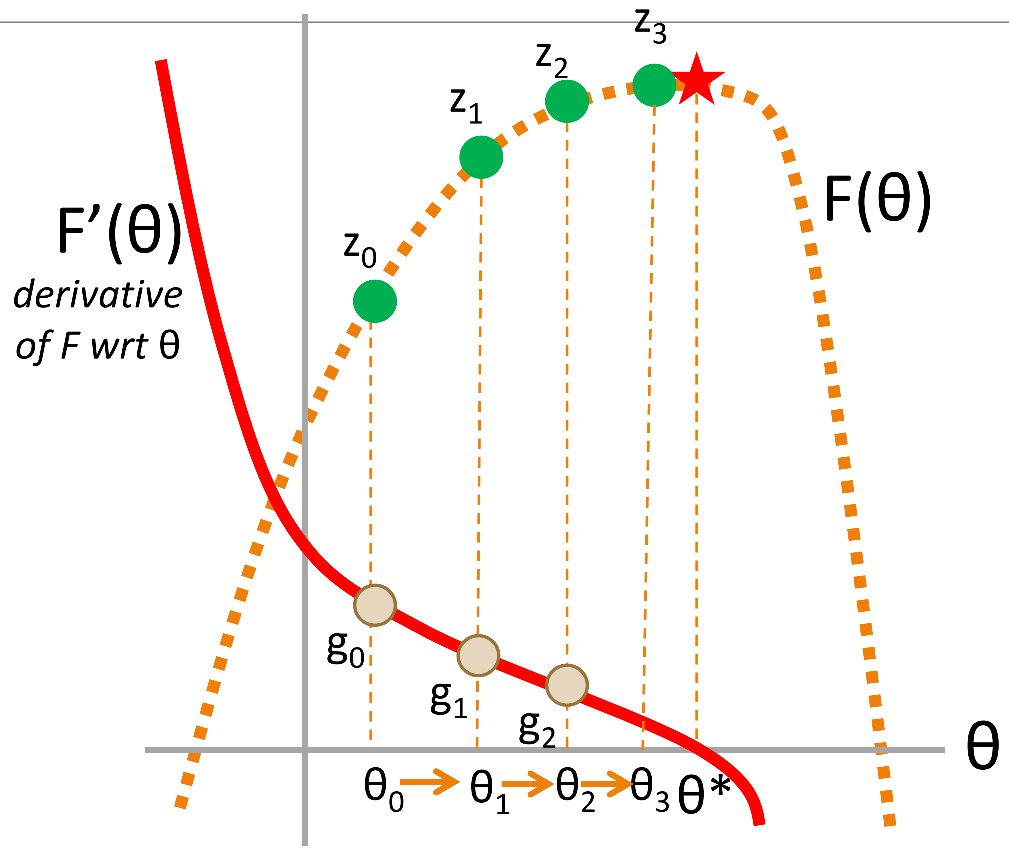
Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get scaling factor ρ_t
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1



What if you can’t find the roots? Follow the derivative

```
Set $t = 0$
Pick a starting value $\theta_t$
Until converged:
    1. Get value $z_t = F(\theta_t)$
    2. Get derivative $g_t = F'(\theta_t)$
    3. Get scaling factor $\rho_t$
    4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
    5. Set $t$ += 1
```



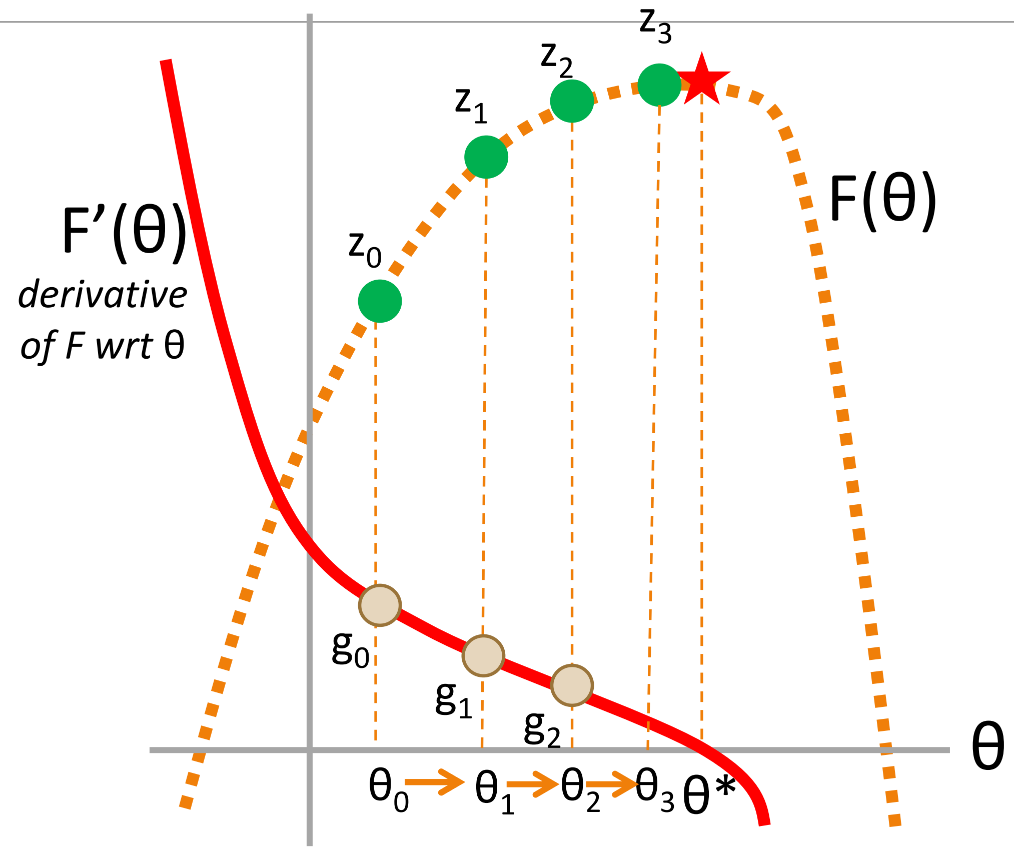
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

Until **converged**:

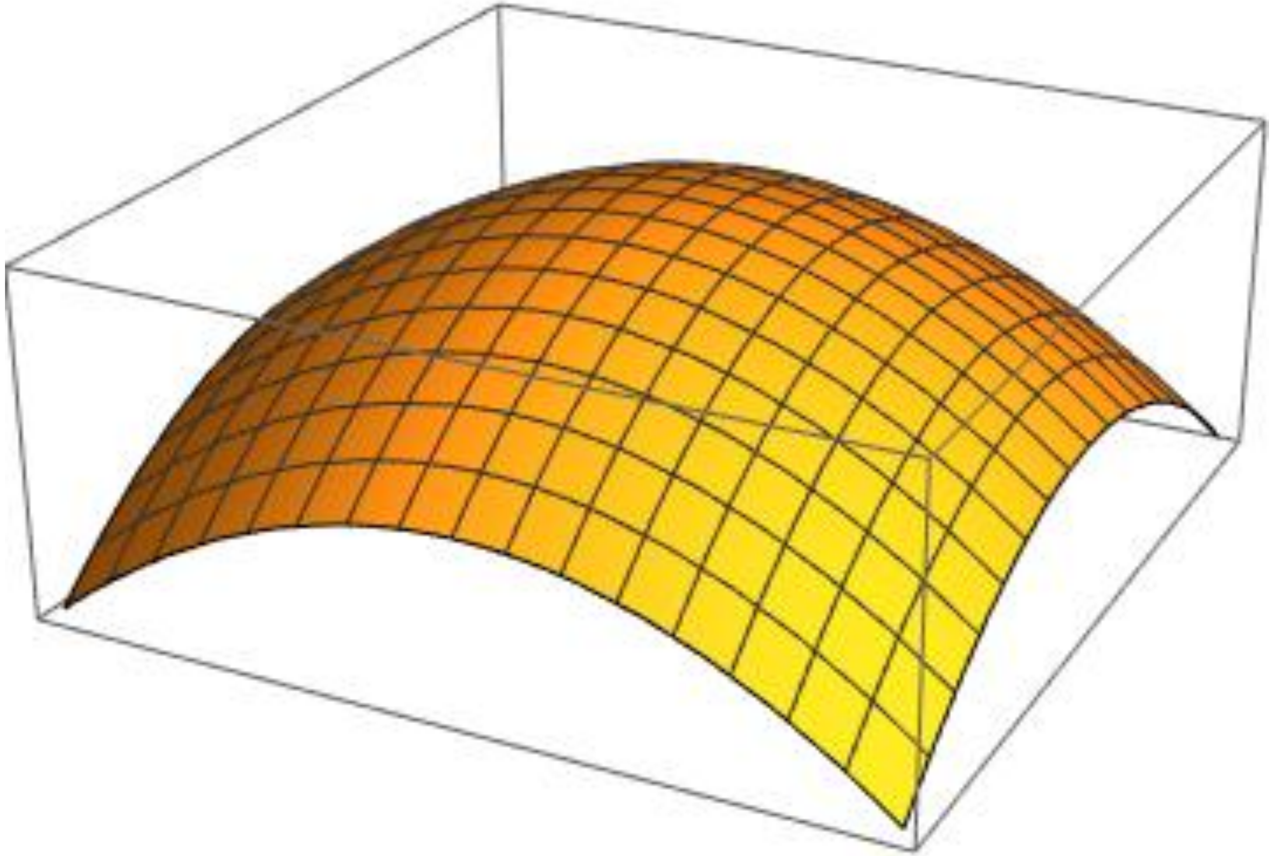
1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get **scaling factor ρ_t**
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1


z_3
z_2
z_1
z_0
F’(θ)
derivative of F wrt θ
F(θ)
g_0
g_1
g_2
θ
$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3$ θ*

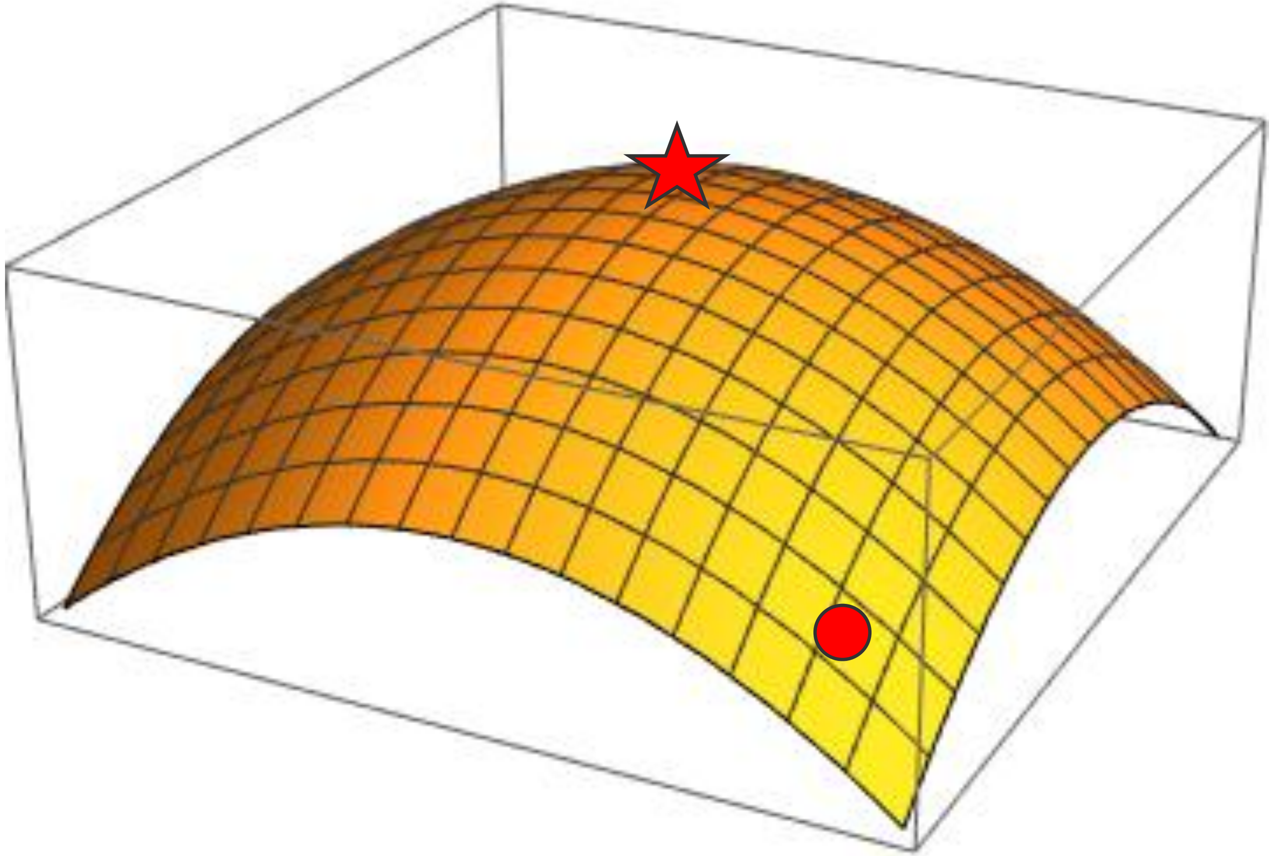
Gradient = Multi-variable derivative

$$\nabla_\theta F\left(\overset{\substack{\text{K-dimensional input}\\ \big\downarrow}}{\theta}\right) = \left(\underbrace{\frac{\partial F}{\partial \theta_1}, \frac{\partial F}{\partial \theta_2}, \ldots, \frac{\partial F}{\partial \theta_K}}_{\text{K-dimensional output}}\right)$$

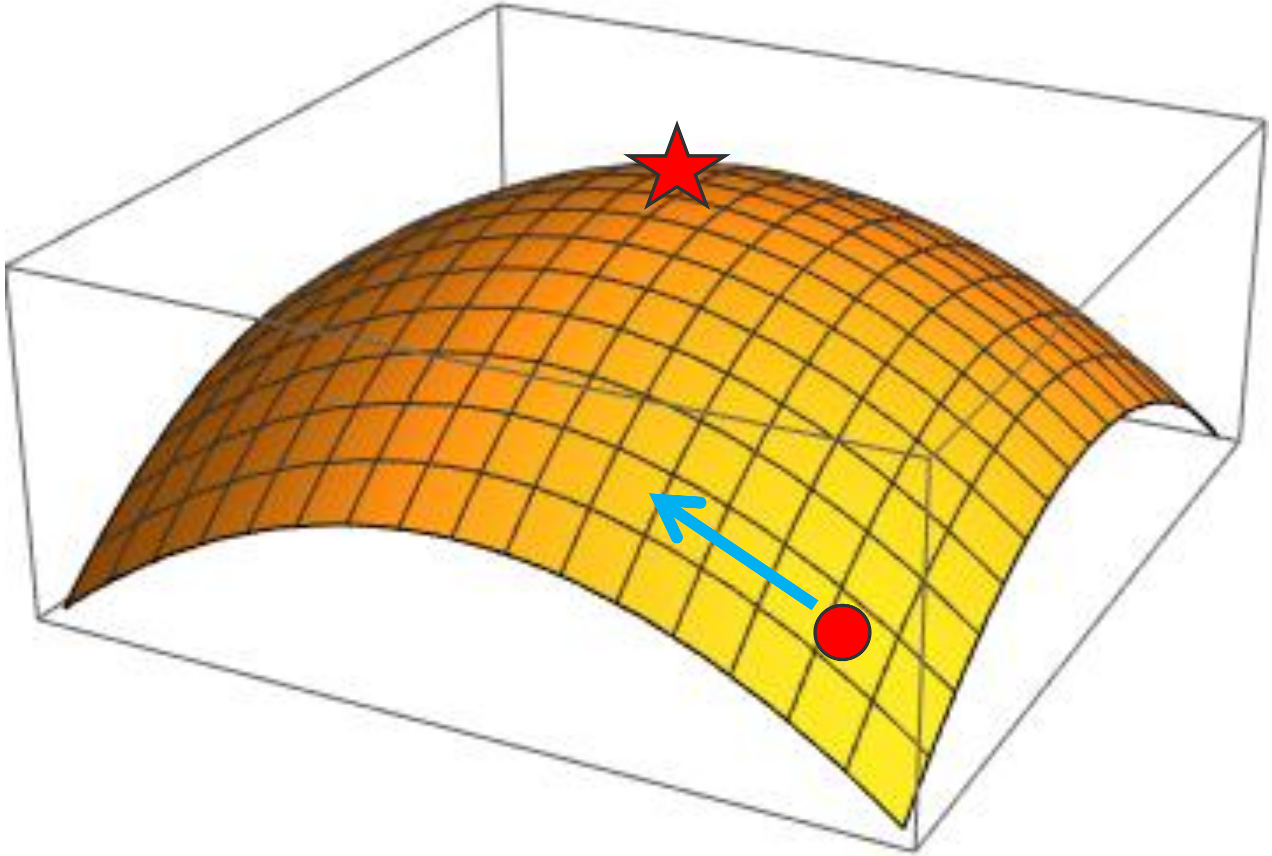
Gradient Ascent



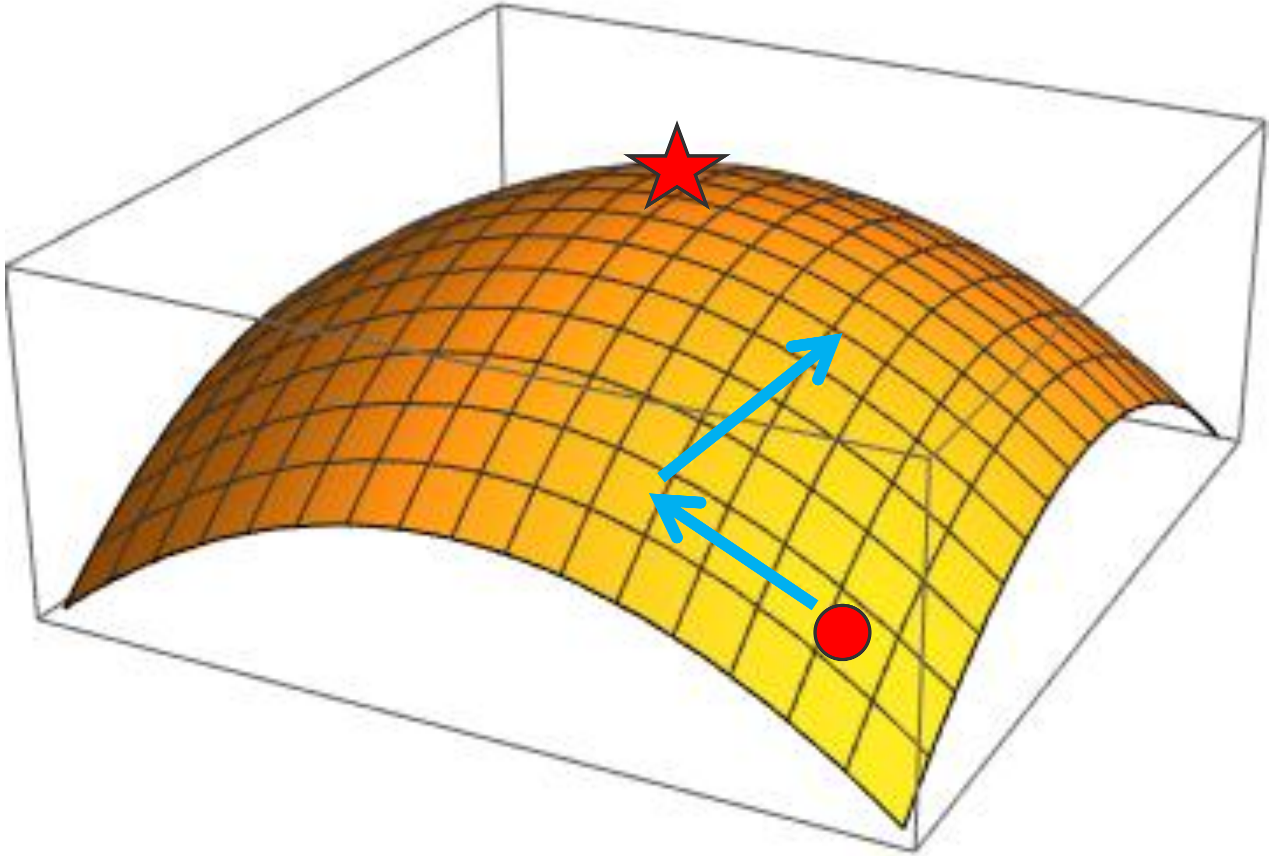
Gradient Ascent



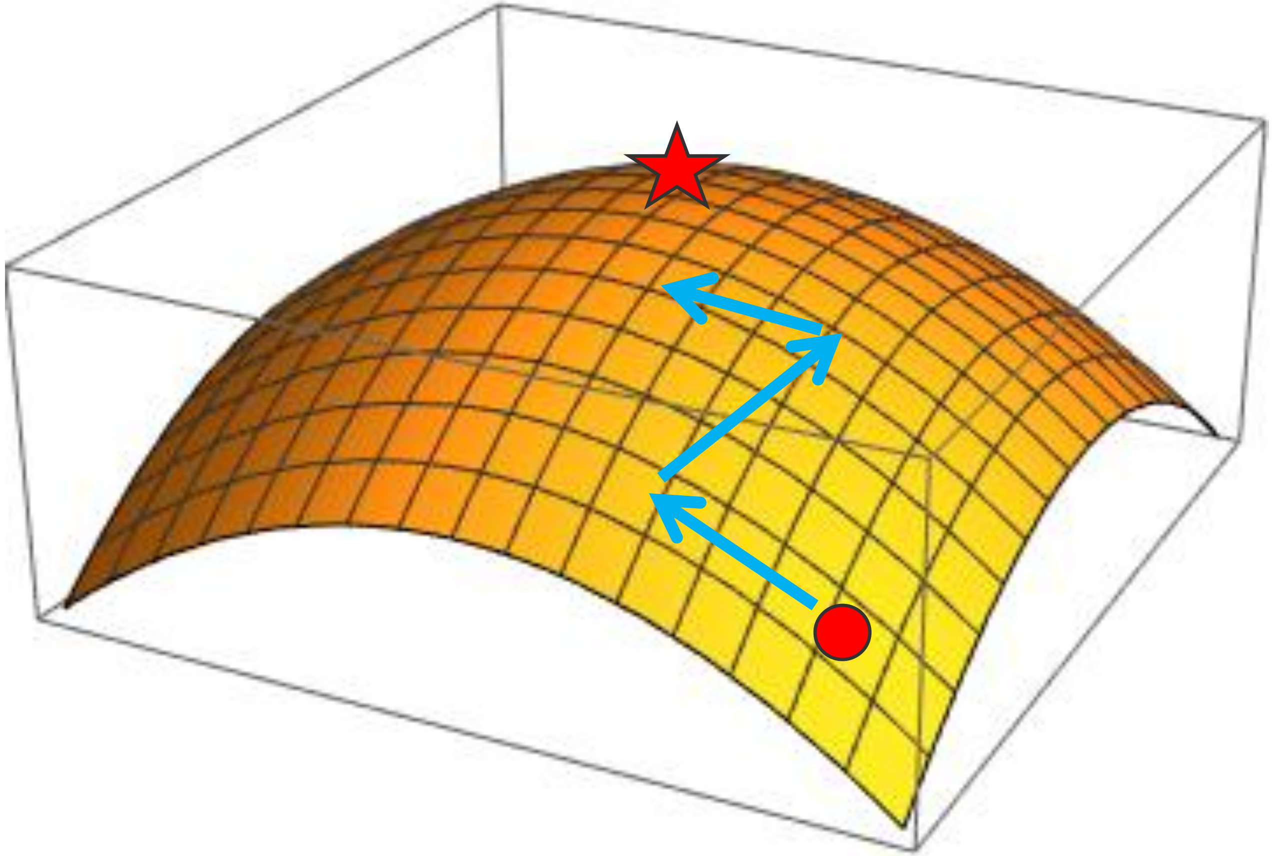
Gradient Ascent



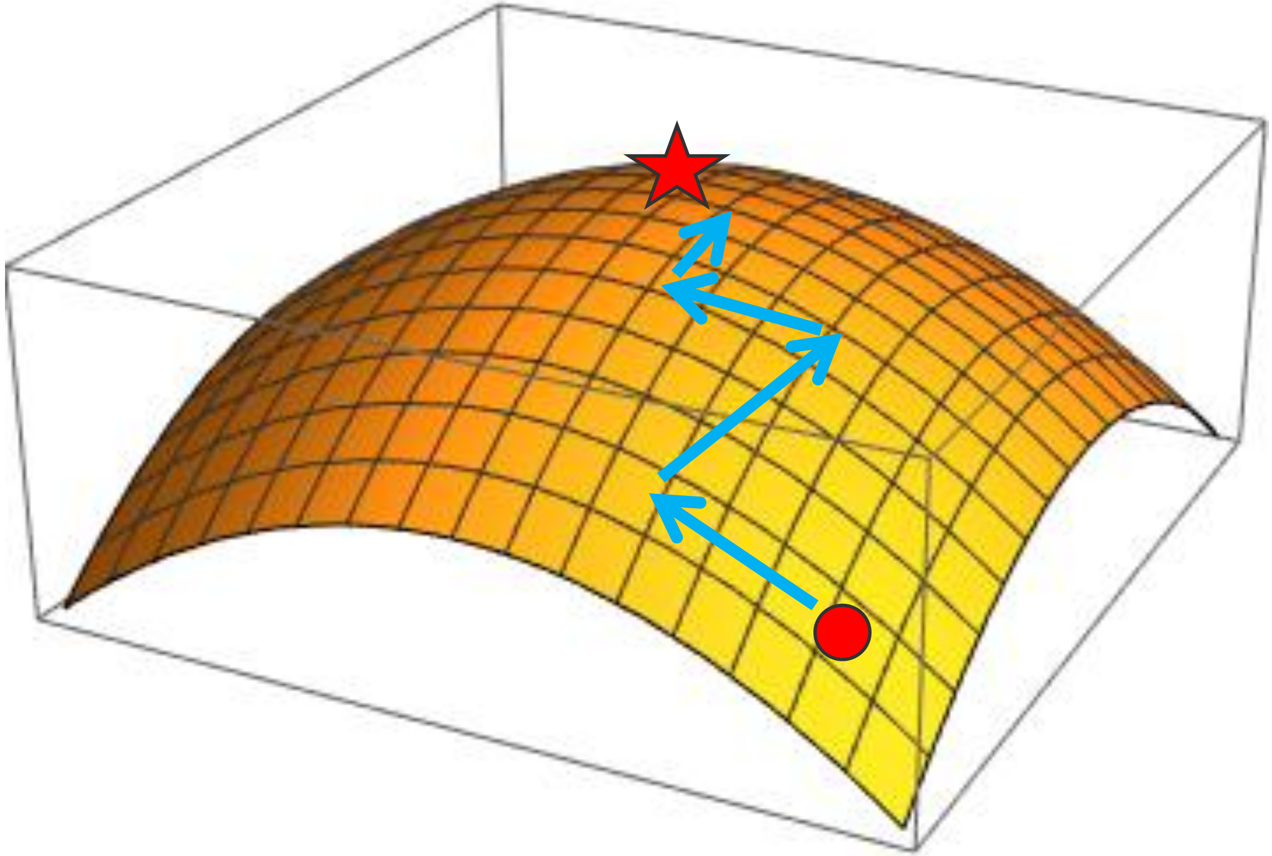
Gradient Ascent



Gradient Ascent



Gradient Ascent



What if you can't find the roots? Follow the **gradient**

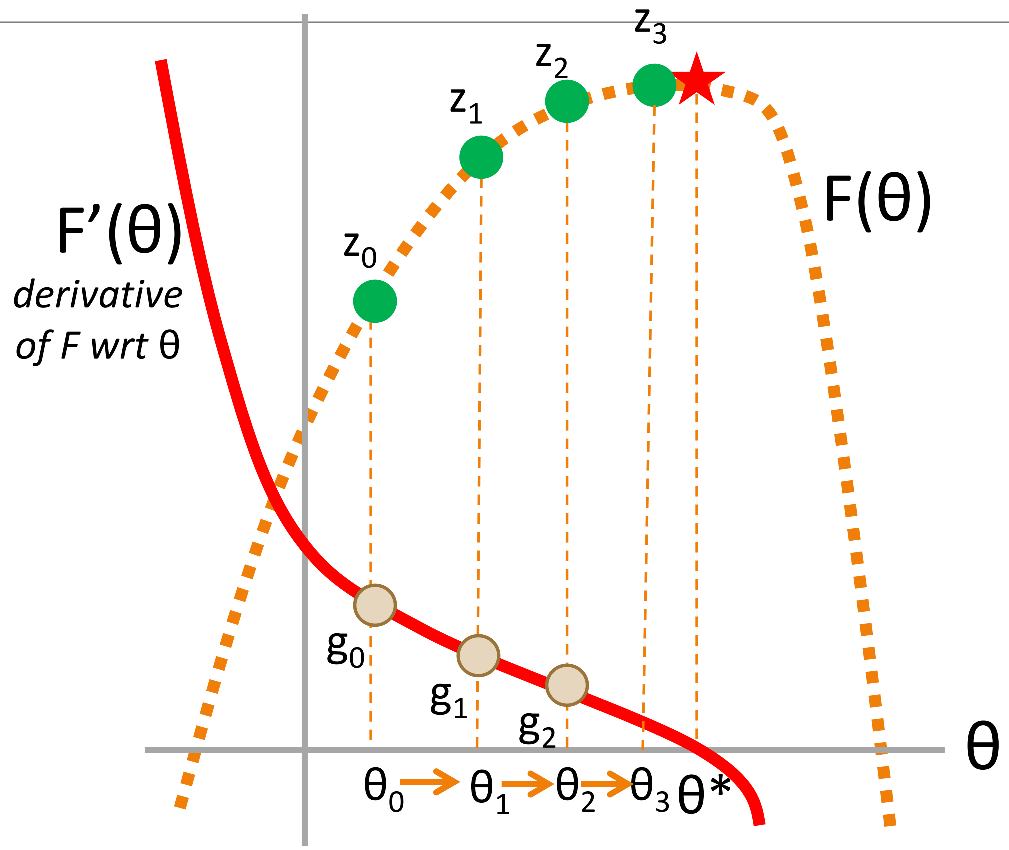
Set t = 0

Pick a starting value θ_t

Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get **gradient** $g_t = F'(\theta_t)$
3. Get scaling factor ρ_t
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1

K-dimensional vectors



F'(θ)
derivative of F wrt θ
F(θ)
z_0
z_1
z_2
z_3
g_0
g_1
g_2
θ
$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3$ θ*

Maxent Models for Classification: Discriminatively or Generatively Trained

Directly model the posterior

$$p(Y \mid X) = \textbf{maxent}(X; Y)$$

Discriminatively trained classifier

Model the posterior with Bayes rule

$$p(Y \mid X) \propto \textbf{maxent}(X \mid Y) p(Y)$$

Generatively trained classifier with maxent-based language model

Bayes' Rule

$$\underbrace{P(Y|X)}_{\text{Posterior}} = \frac{\overbrace{P(X|Y)}^{\text{Likelihood}} \cdot \overbrace{P(Y)}^{\text{Prior}}}{P(X)}$$

It's harder to model P(Y|X) directly since it might be that we only see that set of features once!

Bayes’ Rule

$$P(c|d) = \frac{P(d|c)\cdot P(c)}{P(d)}$$

$$P\left(\textsc{Entailed} \,\middle|\, \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil Jackson and the star}\\ \text{cast, including Scottie Pippen, took the Chicago}\\ \text{Bulls to six National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is based in Chicago.}\end{array}}\right)$$

$$= \frac{P\left(\boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil Jackson and the star}\\ \text{cast, including Scottie Pippen, took the Chicago}\\ \text{Bulls to six National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is based in Chicago.}\end{array}} \,\middle|\, \textsc{Entailed}\right)\cdot P\left(\textsc{Entailed}\right)}{P\left(\boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil Jackson and the star}\\ \text{cast, including Scottie Pippen, took the Chicago}\\ \text{Bulls to six National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is based in Chicago.}\end{array}}\right)}$$

Bayes’ Rule → Naïve Bayes Assumption

Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c) \cdot P(c)}{P(d)}$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c) \cdot P(c)}{\cancel{P(d)}}$$

We can make this assumption because P(d) stays the same regardless of the class!

Naïve Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) \approx \underset{c \in C}{\operatorname{argmax}} P(d|c) \cdot P(c)$$

Bayes' Rule → Naïve Bayes Assumption

Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c) \cdot P(c)}{P(d)}$$

Naïve Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) \approx \underset{c \in C}{\operatorname{argmax}} P(d|c) \cdot P(c)$$

Naïve bayes is **generative** because we are sort of assuming this is how the data point is generated: pick a class *c* and then generate the words by sampling from P(d|c)

SLP 4.1