Probabilistic Modeling and Expectation Maximization

CMSC 678
UMBC
Course Overview (so far)

Basics of Probability
- Requirements to be a distribution (“proportional to”, ∝)
- Definitions of conditional probability, joint probability, and independence
- Bayes rule, (probability) chain rule
- Expectation (of a random variable & function)

Empirical Risk Minimization
- Gradient Descent
- Loss Functions: what is it, what does it measure, and what are some computational difficulties with them?
- Regularization: what is it, how does it work, and why might you want it?

Tasks (High Level)
- Data set splits: training vs. dev vs. test
- Classification: Posterior decoding/MAP classifier
- Classification evaluations: accuracy, precision, recall, and F scores
- Regression (vs. classification)
- Comparing supervised vs. Unsupervised Learning and their tradeoffs: why might you want to use one vs. the other, and what are some potential issues?
- Clustering: high-level goal/task, K-means as an example
- Tradeoffs among clustering evaluations

Linear Models
- Basic form of a linear model (classification or regression)
- Perceptron (simple vs. other variants, like averaged or voted)
- When you should use perceptron (what are its assumptions?)
- Perceptron as SGD

Maximum Entropy Models
- Meanings of feature functions and weights
- How to learn the weights: gradient descent
- Meaning of the maxent gradient

Neural Networks
- Relation to linear models and maxent
- Types (feedforward, CNN, RNN)
- Learning representations (e.g., "feature maps")
- What is a convolution (e.g., 1D vs 2D, high-level notions of why you might want to change padding or the width)
- How to learn: gradient descent, backprop
- Common activation functions
- Neural network regularization

Dimensionality Reduction
- What is the basic task & goal in dimensionality reduction?
- Dimensionality reduction tradeoffs: why might you want to, and what are some potential issues?
- Linear Discriminant Analysis vs. Principal Component Analysis: what are they trying to do, how are they similar, how do they differ?

Kernel Methods & SVMs
- Feature expansion and kernels
- Two views: maximizing a separating hyperplane margin vs. loss optimization (norm minimization)
- Non-separability & slack
- Sub-grads
Remember from the first day: A Terminology Buffet

the **task**: what kind of problem are you solving?

the **data**: amount of human input/number of labeled examples

the **approach**: how any data are being used

- Classification
- Regression
- Clustering

Fully-supervised

Semi-supervised

Un-supervised

Probabilistic

Generative

Conditional

Memory-based

Exemplar

Spectral

Neural...
Remember from the first day: A Terminology Buffet

- **Classification**
- **Regression**
- **Clustering**

**Fully-supervised**

**Semi-supervised**

**Un-supervised**

**What we’ve currently sampled...**

**What we’ll be sampling next...**

- **Probabilistic**
- **Neural**
- **Generative**
- **Conditional**
- **Memory-based**
- **Exemplar**
- **Spectral**

**The task**: what kind of problem are you solving?

**The data**: amount of human input/number of labeled examples

**The approach**: how any data are being used
Outline

Latent and probabilistic modeling

**Generative Modeling**
Example 1: A Model of Rolling a Die
Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
Basic idea
Three coins example
Why EM works
What is (Generative) Probabilistic Modeling?

So far, we’ve (mostly)

had *labeled* data pairs \((x, y)\), and

built classifiers \(p(y \mid x)\)
What is (Generative) Probabilistic Modeling?

So far, we’ve (mostly)

had *labeled* data pairs \((x, y)\), and

built classifiers \(p(y \mid x)\)

What if we want to model *both* \(x\) and \(y\) together?

\[ p(x, y) \]
What is (Generative) Probabilistic Modeling?

So far, we’ve (mostly)

had *labeled* data pairs \((x, y)\), and

built classifiers \(p(y | x)\)

What if we want to model *both* \(x\) and \(y\) together?

\[ p(x, y) \]

Q: Where have we used \(p(x, y)\)?
What is (Generative) Probabilistic Modeling?

So far, we’ve (mostly)

had *labeled* data pairs \((x, y)\), and

built classifiers \(p(y \mid x)\)

What if we want to model *both* \(x\) and \(y\) together?

\[ p(x, y) \]

---

Q: Where have we used \(p(x,y)\)?

A: Linear Discriminant Analysis
What is (Generative) Probabilistic Modeling?

So far, we’ve (mostly)

had *labeled* data pairs \((x, y)\), and

built classifiers \(p(y \mid x)\)

What if we want to model *both* \(x\) and \(y\) together?

\[ p(x, y) \]

Or what if we only have data but no labels?

\[ p(x) \]

Q: Where have we used \(p(x,y)\)?

A: Linear Discriminant Analysis

- Like A3, Q1
- Piazza Q68
Generative Stories

“A useful way to develop probabilistic models is to tell a generative story. This is a *fictional* story that explains how you believe your training data came into existence.” --- CIML Ch 9.5
Generative Stories

“A useful way to develop probabilistic models is to tell a generative story. This is a fictional story that explains how you believe your training data came into existence.” --- CIML Ch 9.5

Generative stories are most often used with joint models $p(x, y)$.... but despite their name, generative stories are applicable to both generative and conditional models
p(x, y) vs. p(y | x): Models of our Data

p(x, y) is the joint distribution

Two main options for estimating:
1. Directly
2.
p(x, y) vs. p(y | x): Models of our Data

p(x, y) is the **joint** distribution

Two main options for estimating:
1. Directly
2. Using Bayes rule: \( p(x, y) = p(x \mid y)p(y) \)

Using Bayes rule *transparently* provides a **generative story** for how our data x and labels y are generated
\( p(x, y) \) vs. \( p(y \mid x) \): Models of our Data

\( p(x, y) \) is the **joint** distribution

\( p(y \mid x) \) is the **conditional** distribution

Two main options for estimating:

1. Directly
2. Using Bayes rule: 
   \[
   p(x, y) = p(x \mid y)p(y)
   \]

Using Bayes rule *transparently* provides a **generative story** for how our data \( x \) and labels \( y \) are generated

Two main options for estimating:

1. Directly: used when you *only* care about making the right prediction
   Examples: perceptron, logistic regression, neural networks (we’ve covered)
2.
p(x,y) vs. p(y | x): Models of our Data

p(x, y) is the **joint** distribution

Two main options for estimating:
1. Directly
2. Using Bayes rule: \( p(x, y) = p(x | y)p(y) \)

Using Bayes rule *transparently* provides a **generative story** for how our data x and labels y are generated

p(y | x) is the **conditional** distribution

Two main options for estimating:
1. Directly: used when you *only* care about making the right prediction
   Examples: perceptron, logistic regression, neural networks (we’ve covered)
2. Estimate the joint
Outline

Latent and probabilistic modeling
Generative Modeling
Example 1: A Model of Rolling a Die
Example 2: A Model of Conditional Die Roles

EM (Expectation Maximization)
Basic idea
Three coins example
Why EM works
Example: Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]
Example: Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_{i} p(w_i) \]

\( w_1 = 1 \)
\( w_2 = 5 \)
\( w_3 = 4 \)
\( \ldots \)
Generative Story for Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ w_1 = 1 \]
\[ w_2 = 5 \]
\[ w_3 = 4 \]
\[ \ldots \]

N different (independent) rolls

Generative Story

for roll \( i = 1 \) to \( N \):
Generative Story for Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

N different (independent) rolls

\[ w_1 = 1 \]

\[ w_2 = 5 \]

\[ w_3 = 4 \]

\[ \vdots \]

Generative Story

for roll \( i = 1 \) to \( N \):

\[ w_i \sim \text{Cat}(\theta) \]
Generative Story for Rolling a Die

\[ p(w_1, w_2, ..., w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

N different (independent) rolls

\[ w_1 = 1 \]
\[ w_2 = 5 \]
\[ w_3 = 4 \]
\[ ... \]

“for each” loop becomes a product

Generative Story

for roll \( i = 1 \) to \( N \):

\( w_i \sim \text{Cat} (\theta) \)

Calculate \( p(w_i) \) according to provided distribution
Generative Story for Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

N different (independent) rolls

\[ w_1 = 1 \]
\[ w_2 = 5 \]
\[ w_3 = 4 \]

\[ \cdots \]

for roll \( i = 1 \) to \( N \):
\[ w_i \sim \text{Cat}(\theta) \]

\[ \sum_{k=1}^{6} \theta_k = 1 \quad 0 \leq \theta_k \leq 1, \forall k \]
Learning Parameters for the Die Model

\[ p(w_1, w_2, ..., w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?
Learning Parameters for the Die Model

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

A: Develop a good model for what we observe
Learning Parameters for the Die Model

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

A: Develop a good model for what we observe

Q: (for discrete observations) What loss function do we minimize to maximize log-likelihood?
Learning Parameters for the Die Model

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?
A: Develop a good model for what we observe

Q: (for discrete observations) What loss function do we minimize to maximize log-likelihood?
A: Cross-entropy
Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls...

...what are “reasonable” estimates for \( p(w) \)?

\[
\begin{align*}
p(1) &= \? & p(2) &= \? \\
p(3) &= \? & p(4) &= \? \\
p(5) &= \? & p(6) &= \?
\end{align*}
\]
Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

\[ p(w_1, w_2, ..., w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls...

...what are “reasonable” estimates for \( p(w) \)?

\[
\begin{align*}
p(1) &= 2/9 \\
p(2) &= 1/9 \\
p(3) &= 1/9 \\
p(4) &= 3/9 \\
p(5) &= 1/9 \\
p(6) &= 1/9
\end{align*}
\]
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

N different (independent) rolls

\begin{align*}
  w_1 &= 1 \\
  w_2 &= 5 \\
  w_3 &= 4 \\
  \cdots
\end{align*}

Generative Story

for roll \( i = 1 \) to \( N \):
\( w_i \sim \text{Cat}(\theta) \)

Maximize Log-likelihood
\[ \mathcal{L}(\theta) = \sum_{i} \log p_{\theta}(w_i) \\
  = \sum_{i} \log \theta_{w_i} \]
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

Generative Story

for roll \( i = 1 \) to \( N \):

\( w_i \sim \text{Cat}(\theta) \)

Maximize Log-likelihood

\[ \mathcal{L}(\theta) = \sum_i \log \theta_{w_i} \]

Q: What’s an easy way to maximize this, as written exactly (even without calculus)?
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[
p(w_1, w_2, ..., w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i)
\]

Generative Story
for roll \( i = 1 \) to \( N \):
\( w_i \sim \text{Cat}(\theta) \)

Maximize Log-likelihood
\[
\mathcal{L}(\theta) = \sum_i \log \theta_{w_i}
\]

Q: What’s an easy way to maximize this, as written exactly (even without calculus)?

A: Just keep increasing \( \theta_k \) (we know \( \theta \) must be a distribution, but it’s not specified)
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

Maximize Log-likelihood (with distribution constraints)

\[ \mathcal{L}(\theta) = \sum_i \log \theta_w \quad \text{s.t. } \sum_{k=1}^{6} \theta_k = 1 \]

solve using Lagrange multipliers

N different (independent) rolls
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

Maximize Log-likelihood (with distribution constraints)

\[ \mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda \left( \sum_{k=1}^{6} \theta_k - 1 \right) \]

\[ \frac{\partial \mathcal{F}(\theta)}{\partial \theta_k} = \sum_{i:w_i=k} \frac{1}{\theta_{w_i}} - \lambda \]

\[ \frac{\partial \mathcal{F}(\theta)}{\partial \lambda} = - \sum_{k=1}^{6} \theta_k + 1 \]

N different (independent) rolls

(we can include the inequality constraints \(0 \leq \theta_k\), but it complicates the problem and, right now, is not needed)
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_{i} p(w_i) \]

Maximize Log-likelihood (with distribution constraints)

\[ F(\theta) = \sum_{i} \log \theta_{w_i} - \lambda \left( \sum_{k=1}^{6} \theta_k - 1 \right) \]

\[ \theta_k = \frac{\sum_{i:w_i=k} 1}{\lambda} \]

optimal \( \lambda \) when \( \sum_{k=1}^{6} \theta_k = 1 \)
Learning Parameters for the Die Model: Maximum Likelihood (Math)

\[ p(w_1, w_2, ..., w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

Maximize Log-likelihood (with distribution constraints)

\[ \mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda \left( \sum_{k=1}^{6} \theta_k - 1 \right) \]

\[ \theta_k = \frac{\sum_{i:w_i=k} 1}{\sum_k \sum_{i:w_i=k} 1} = \frac{N_k}{N} \]

optimal \( \lambda \) when \( \sum_{k=1}^{6} \theta_k = 1 \)

(we can include the inequality constraints \( 0 \leq \theta_k \), but it complicates the problem and, right now, is not needed)
Outline

Latent and probabilistic modeling
Generative Modeling
Example 1: A Model of Rolling a Die
Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
Basic idea
Three coins example
Why EM works
Example: Conditionally Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) \]
\[ = \prod_i p(w_i|z_i) p(z_i) \]

Add complexity to better explain what we see.
Example: Conditionally Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) = \prod_i p(w_i|z_i)p(z_i) \]

First flip a coin...

\[ z_1 = T \]
\[ z_2 = H \]

...
Example: Conditionally Rolling a Die

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) \]

\[ = \prod_i p(w_i|z_i)p(z_i) \]

First flip a coin... ...then roll a different die depending on the coin flip

\[ z_1 = T \quad w_1 = 1 \]

\[ z_2 = H \quad w_2 = 5 \]

...
Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) \]
\[ = \prod_i p(w_i|z_i)p(z_i) \]

If you observe the \( z_i \) values, this is easy!
Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

\[ p(z_1, w_1, z_2, w_2, ..., z_N, w_N) = \prod_i p(w_i | z_i) p(z_i) \]

If you observe the \( z_i \) values, this is easy!

First: Write the Generative Story

\( \lambda = \) distribution over coin \((z)\)
\( \gamma^{(H)} = \) distribution for die when coin comes up heads
\( \gamma^{(T)} = \) distribution for die when coin comes up tails

for item \( i = 1 \) to \( N \):
\( z_i \sim \text{Bernoulli}(\lambda) \)
\( w_i \sim \text{Cat}(\gamma^{(z_i)}) \)
Learning in Conditional Die Roll Model: Maximize \((\log-)\)Likelihood

\[
p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_{i} p(w_i|z_i) p(z_i)
\]

If you observe the \(z_i\) values, this is easy!

First: Write the Generative Story

\[
\lambda = \text{distribution over coin (z)}
\]

\[
\gamma^{(H)} = \text{distribution for H die}
\]

\[
\gamma^{(T)} = \text{distribution for T die}
\]

for item \(i = 1\) to \(N\):

\[
\begin{align*}
z_i &\sim \text{Bernoulli}(\lambda) \\
w_i &\sim \text{Cat}(\gamma^{(z_i)})
\end{align*}
\]

Second: Generative Story \(\rightarrow\) Objective

\[
\mathcal{F}(\theta) = \sum_{i} \left( \log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)} \right)
\]

Lagrange multiplier constraints
Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

\[
p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i) p(z_i)
\]

If you observe the \( z_i \) values, this is easy!

First: Write the Generative Story

\[ \lambda = \text{distribution over coin (z)} \]
\[ \gamma^{(H)} = \text{distribution for H die} \]
\[ \gamma^{(T)} = \text{distribution for T die} \]

for item \( i = 1 \) to \( N \):

\[ z_i \sim \text{Bernoulli(} \lambda \text{)} \]
\[ w_i \sim \text{Cat(} \gamma^{(z_i)} \text{)} \]

Second: Generative Story → Objective

\[
F(\theta) = \sum_i^n (\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)})
\]

\[
-\eta \left( \sum_{k=1}^2 \lambda_k - 1 \right) - \sum_{k=1}^2 \delta_k \left( \sum_{j=1}^6 \gamma_j^{(k)} - 1 \right)
\]
Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

\[ p(z_1, w_1, z_2, w_2, ..., z_N, w_N) = \prod_i p(w_i | z_i) p(z_i) \]

If you observe the \( z_i \) values, this is easy!

But if you don’t observe the \( z_i \) values, this is not easy!

First: Write the Generative Story

\( \lambda = \text{distribution over coin (z)} \)

\( \gamma^{(H)} = \text{distribution for H die} \)

\( \gamma^{(T)} = \text{distribution for T die} \)

for item \( i = 1 \) to \( N \):

\( z_i \sim \text{Bernoulli(} \lambda \text{)} \)

\( w_i \sim \text{Cat(} \gamma^{(z_i)} \text{)} \)

Second: Generative Story \( \rightarrow \) Objective

\[
\mathcal{F}(\theta) = \sum_i \left( \log \lambda_{z_i} + \log \gamma^{(z_i)}_{w_i} \right)
- \eta \left( \sum_{k=1}^{2} \lambda_k - 1 \right)
- \sum_{k=1}^{2} \delta_k \left( \sum_{j=1}^{6} \gamma^{(k)}_j - 1 \right)
\]
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i) p(z_i) \]

goal: maximize (log-)likelihood

we don’t actually observe these \( z \) values
we just see the items \( w \)

if we \textit{did} observe \( z \), estimating the
probability parameters would be easy...
but we don’t! :(
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, ..., z_N, w_N) = \prod_{i} p(w_i|z_i) p(z_i) \]

goal: maximize (log-)likelihood

we don’t actually observe these \( z \) values
we just see the items \( w \)

if we did observe \( z \), estimating the probability parameters would be easy...
but we don’t! :( 

if we knew the probability parameters then we could estimate \( z \) and evaluate likelihood... but we don’t! :( 

Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_{i} p(w_i|z_i) p(z_i) \]

we don’t actually observe these \( z \) values

goal: maximize \textit{marginalized} (log-)likelihood
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i) p(z_i) \]

we don’t actually observe these \( z \) values

goal: maximize \textit{marginalized} (log-)likelihood
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_{i} p(w_i | z_i) p(z_i) \]

*we don’t actually observe these \( z \) values*

goal: maximize *marginalized* (log-)likelihood
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i) p(z_i) \]

we don’t actually observe these \( z \) values

Goal: maximize \textit{marginalized} (log-)likelihood

\[ p(w_1, w_2, \ldots, w_N) = \left( \sum_{z_1} p(z_1, w) \right) \left( \sum_{z_2} p(z_2, w) \right) \cdots \left( \sum_{z_N} p(z_N, w) \right) \]
Example: Conditionally Rolling a Die

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) \]

goal: maximize \textit{marginalized} (log-)likelihood

\[ p(w_1, w_2, \ldots, w_N) = \left( \sum_{z_1} p(z_1, w) \right) \left( \sum_{z_2} p(z_2, w) \right) \cdots \left( \sum_{z_N} p(z_N, w) \right) \]

if \textit{we did} observe \( z \), estimating the probability parameters would be easy... but we don’t! :(  

if \textit{we knew} the probability parameters then we could estimate \( z \) and evaluate likelihood... but we don’t! :(
if we did observe $z$, estimating the probability parameters would be easy...

but we don’t! :(
if we did observe $z$, estimating the probability parameters would be easy...
but we don’t! :(

if we knew the probability parameters then we could estimate $z$ and evaluate likelihood... but we don’t! :(
if we knew the probability parameters then we could estimate $z$ and evaluate likelihood... but we don’t! :(

if we did observe $z$, estimating the probability parameters would be easy...
but we don’t! :

Expectation Maximization:
give you model estimation the needed “spark”
Outline

Latent and probabilistic modeling
Generative Modeling
Example 1: A Model of Rolling a Die
Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
Basic idea
Three coins example
Why EM works
Expectation Maximization (EM)

0. Assume *some* value for your parameters

Two step, iterative algorithm

1. E-step: count under uncertainty (compute expectations)

2. M-step: maximize log-likelihood, assuming these uncertain counts
Expectation Maximization (EM): E-step

0. Assume *some* value for your parameters

Two step, iterative algorithm

1. E-step: count under uncertainty, assuming these parameters
   \[ p(z_i) \] \[ \rightarrow \] \[ \text{count}(z_i, w_i) \]

2. M-step: maximize log-likelihood, assuming these uncertain counts
Expectation Maximization (EM): E-step

0. Assume *some* value for your parameters

Two step, iterative algorithm

1. E-step: count under uncertainty, assuming these parameters

\[ p(z_i) \quad \rightarrow \quad \text{count}(z_i, w_i) \]

2. M-step: maximize log-likelihood, assuming these uncertain counts

We’ve already seen this type of counting, when computing the gradient in maxent models.
Expectation Maximization (EM): M-step

0. Assume *some* value for your parameters

Two step, iterative algorithm

1. E-step: count under uncertainty, assuming these parameters

2. M-step: maximize log-likelihood, assuming these uncertain counts

\[ p^{(t+1)}(z) \]
\[
\max \text{ the average log-likelihood of our complete data } (z, w), \text{ averaged across all } z \text{ and according to how likely our } current \text{ model thinks } z \text{ is}
\]
maximize the average log-likelihood of our complete data \((z, w)\), averaged across all \(z\) and according to how likely our \textit{current} model thinks \(z\) is

\[
\max_{\theta} \mathbb{E}_{z \sim p_{\theta}(t)(\cdot | w)} \left[ \log p_{\theta}(z, w) \right]
\]
maximize the average log-likelihood of our complete data \((z, w)\), averaged across all \(z\) and according to how likely our current model thinks \(z\) is

\[
\max_{\theta} \mathbb{E}_{z \sim p_{\theta}(t) (\cdot | w)} \left[ \log p_{\theta}(z, w) \right]
\]
maximize the average log-likelihood of our complete data \((z, w)\), averaged across all \(z\) and according to how likely our \textit{current} model thinks \(z\) is

\[
\max_{\theta} \mathbb{E}_{z \sim p_{\theta}(t) (\cdot | w)} \left[ \log p_{\theta}(z, w) \right]
\]
maximize the average log-likelihood of our complete data \((z, w)\), averaged across all \(z\) and according to how likely our \emph{current} model thinks \(z\) is
maximize the average log-likelihood of our complete data \((z, w)\), averaged across all \(z\) and according to how likely our \textit{current} model thinks \(z\) is

\[
\max_{\theta} \mathbb{E}_z \sim p_{\theta(t)}(\cdot | w) \left[ \log p_{\theta}(z, w) \right]
\]

\textit{E-step: count under uncertainty}

\textit{M-step: maximize log-likelihood}
Why EM? Un-Supervised Learning

NO labeled data:
• human annotated
• relatively small/few examples

unlabeled data:
• raw; not annotated
• plentiful

EM/generative models in this case can be seen as a type of clustering
Why EM? Semi-Supervised Learning

labeled data:
• human annotated
• relatively small/few examples

unlabeled data:
• raw; not annotated
• plentiful
Why EM? Semi-Supervised Learning

labeled data:
- human annotated
- relatively small/few examples

unlabeled data:
- raw; not annotated
- plentiful
Why EM? Semi-Supervised Learning

labeled data:
- human annotated
- relatively small/few examples

unlabeled data:
- raw; not annotated
- plentiful
Why EM? Semi-Supervised Learning

- √
- √
- √
- √
- ×
- ×
- ×
- ×

+  

=  

EM
Outline

Latent and probabilistic modeling
  Generative Modeling
  Example 1: A Model of Rolling a Die
  Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
  Basic idea
  Three coins example
  Why EM works
Imagine three coins

Flip 1\textsuperscript{st} coin (\textit{penny})

If heads: flip 2\textsuperscript{nd} coin (\textit{dollar coin})

If tails: flip 3\textsuperscript{rd} coin (\textit{dime})
Imagine three coins

Flip 1\textsuperscript{st} coin (penny)

If heads: flip 2\textsuperscript{nd} coin (dollar coin)

If tails: flip 3\textsuperscript{rd} coin (dime)

- only observe these (record heads vs. tails outcome)
- don’t observe this
Imagine three coins

Flip 1\textsuperscript{st} coin (\textit{penny})

If heads: flip 2\textsuperscript{nd} coin (\textit{dollar coin})

If tails: flip 3\textsuperscript{rd} coin (\textit{dime})

unobserved: \textit{part of speech? genre?}

observed: \textit{a, b, e, etc.}

We \textit{run} the code, vs. The \textit{run} failed
Imagine three coins

Flip 1\textsuperscript{st} coin (penny)\hfill

\[ p(\text{heads}) = \lambda \]
\[ p(\text{tails}) = 1 - \lambda \]

If heads: flip 2\textsuperscript{nd} coin (dollar coin)\hfill

\[ p(\text{heads}) = \gamma \]
\[ p(\text{tails}) = 1 - \gamma \]

If tails: flip 3\textsuperscript{rd} coin (dime)\hfill

\[ p(\text{heads}) = \psi \]
\[ p(\text{tails}) = 1 - \psi \]
Imagine three coins

\[ p(\text{heads}) = \lambda \]
\[ p(\text{tails}) = 1 - \lambda \]

\[ p(\text{heads}) = \gamma \]
\[ p(\text{tails}) = 1 - \gamma \]

\[ p(\text{heads}) = \psi \]
\[ p(\text{tails}) = 1 - \psi \]

Three parameters to estimate: \( \lambda \), \( \gamma \), and \( \psi \)
**Generative Story for Three Coins**

\[ p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2) \cdots p(w_N) = \prod_i p(w_i) \]

\[ p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1) \cdots p(z_N)p(w_N|z_N) = \prod_i p(w_i|z_i)p(z_i) \]

\[ p(\text{heads}) = \lambda \]
\[ p(\text{tails}) = 1 - \lambda \]

\[ p(\text{heads}) = \gamma \]
\[ p(\text{tails}) = 1 - \gamma \]

**Generative Story**

\[ \lambda = \text{distribution over penny} \]
\[ \gamma = \text{distribution for dollar coin} \]
\[ \psi = \text{distribution over dime} \]

for item \( i = 1 \) to \( N \):

\[ z_i \sim \text{Bernoulli}(\lambda) \]
\[ \text{if } z_i = H: w_i \sim \text{Bernoulli}(\gamma) \]
\[ \text{else: } w_i \sim \text{Bernoulli}(\psi) \]
Three Coins Example

If all flips were observed

\[ p(\text{heads}) = \lambda \quad p(\text{heads}) = \gamma \quad p(\text{heads}) = \psi \]
\[ p(\text{tails}) = 1 - \lambda \quad p(\text{tails}) = 1 - \gamma \quad p(\text{tails}) = 1 - \psi \]
Three Coins Example

If all flips were observed

\[ p(\text{heads}) = \lambda \quad p(\text{heads}) = \gamma \quad p(\text{heads}) = \psi \]
\[ p(\text{tails}) = 1 - \lambda \quad p(\text{tails}) = 1 - \gamma \quad p(\text{tails}) = 1 - \psi \]

\[ p(\text{heads}) = \frac{4}{6} \quad p(\text{heads}) = \frac{1}{4} \quad p(\text{heads}) = \frac{1}{2} \]
\[ p(\text{tails}) = \frac{2}{6} \quad p(\text{tails}) = \frac{3}{4} \quad p(\text{tails}) = \frac{1}{2} \]
Three Coins Example

But not all flips are observed \(\rightarrow\) set parameter values

\[
\begin{align*}
    p(\text{heads}) &= \lambda = 0.6 &
    p(\text{heads}) &= 0.8 &
    p(\text{heads}) &= 0.6 \\
    p(\text{tails}) &= 0.4 &
    p(\text{tails}) &= 0.2 &
    p(\text{tails}) &= 0.4
\end{align*}
\]
Three Coins Example

But not all flips are observed → set parameter values

\[ p(\text{heads}) = \lambda = 0.6 \quad p(\text{heads}) = 0.8 \quad p(\text{heads}) = 0.6 \]
\[ p(\text{tails}) = 0.4 \quad p(\text{tails}) = 0.2 \quad p(\text{tails}) = 0.4 \]

Use these values to compute posteriors

\[ p(\text{heads} \mid \text{observed item H}) = \frac{p(\text{heads} \& H)}{p(H)} \]
\[ p(\text{heads} \mid \text{observed item T}) = \frac{p(\text{heads} \& T)}{p(T)} \]
Three Coins Example

But not all flips are observed $\rightarrow$ set parameter values

$p(\text{heads}) = \lambda = .6$ \quad $p(\text{heads}) = .8$ \quad $p(\text{heads}) = .6$

$p(\text{tails}) = .4$ \quad $p(\text{tails}) = .2$ \quad $p(\text{tails}) = .4$

Use these values to compute posteriors

$$p(\text{heads} \mid \text{observed item } H) = \frac{p(H | \text{heads})p(\text{heads})}{p(H)}$$

rewrite joint using Bayes rule

marginal likelihood
Three Coins Example

But not all flips are observed → set parameter values

\[ p(\text{heads}) = \lambda = 0.6 \quad p(\text{heads}) = 0.8 \quad p(\text{heads}) = 0.6 \]
\[ p(\text{tails}) = 0.4 \quad p(\text{tails}) = 0.2 \quad p(\text{tails}) = 0.4 \]

Use these values to compute posteriors

\[ p(\text{heads} \mid \text{observed item } H) = \frac{p(H \mid \text{heads}) p(\text{heads})}{p(H)} \]

\[ p(H \mid \text{heads}) = 0.8 \quad p(T \mid \text{heads}) = 0.2 \]
Three Coins Example

But not all flips are observed \( \rightarrow \) set parameter values

\[
\begin{align*}
    p(\text{heads}) &= \lambda = .6 & p(\text{heads}) &= .8 & p(\text{heads}) &= .6 \\
    p(\text{tails}) &= .4 & p(\text{tails}) &= .2 & p(\text{tails}) &= .4
\end{align*}
\]

Use these values to compute posteriors

\[
    p(\text{heads} \mid \text{observed item } H) = \frac{p(H \mid \text{heads})p(\text{heads})}{p(H)}
\]

\[
    p(H \mid \text{heads}) = .8 & \quad p(T \mid \text{heads}) = .2
\]

\[
    p(H) = p(H \mid \text{heads}) \ast p(\text{heads}) + p(H \mid \text{tails}) \ast p(\text{tails})
\]

\[
    = .8 \ast .6 + .6 \ast .4
\]
Three Coins Example

\[ H \ H \ T \ H \ T \ H \]
\[ H \ T \ H \ T \ T \ T \]

Use posteriors to update parameters

\[
p(\text{heads} | \text{obs. } H) = \frac{p(H | \text{heads})p(\text{heads})}{p(H)} = \frac{.8 \times .6}{.8 \times .6 + .6 \times .4} \approx 0.667
\]

\[
p(\text{heads} | \text{obs. } T) = \frac{p(T | \text{heads})p(\text{heads})}{p(T)} = \frac{.2 \times .6}{.2 \times .6 + .6 \times .4} \approx 0.334
\]

Q: Is \( p(\text{heads} | \text{obs. } H) + p(\text{heads} | \text{obs. } T) = 1? \)
Three Coins Example

Use posteriors to update parameters

\[
p(\text{heads} \mid \text{obs. } H) = \frac{p(H \mid \text{heads})p(\text{heads})}{p(H)} = \frac{.8 \times .6}{.8 \times .6 + .6 \times .4} \approx 0.667\]

\[
p(\text{heads} \mid \text{obs. } T) = \frac{p(T \mid \text{heads})p(\text{heads})}{p(T)} = \frac{.2 \times .6}{.2 \times .6 + .6 \times .4} \approx 0.334\]

Q: Is \( p(\text{heads} \mid \text{obs. } H) + p(\text{heads} \mid \text{obs. } T) = 1? \)

A: No.
Use posteriors to update parameters

\[
p(\text{heads} | \text{obs. } H) = \frac{p(H|\text{heads})p(\text{heads})}{p(H)}
= \frac{.8 \cdot .6}{.8 \cdot .6 + .6 \cdot .4} \approx 0.667
\]

\[
p(\text{heads} | \text{obs. } T) = \frac{p(T|\text{heads})p(\text{heads})}{p(T)}
= \frac{.2 \cdot .6}{.2 \cdot .6 + .6 \cdot .4} \approx 0.334
\]

(in general, \(p(\text{heads} | \text{obs. } H)\) and \(p(\text{heads} | \text{obs. } T)\) do NOT sum to 1)

\[
p(\text{heads}) = \frac{\# \text{ heads from penny}}{\# \text{ total flips of penny}}
\]

fully observed setting

our setting: partially-observed
Three Coins Example

Use posteriors to update parameters

\[
p(\text{heads} \mid \text{obs. H}) = \frac{p(H \mid \text{heads})p(\text{heads})}{p(H)} = \frac{.8 \times .6}{.8 \times .6 + .6 \times .4} \approx 0.667
\]

\[
p(\text{heads} \mid \text{obs. T}) = \frac{p(T \mid \text{heads})p(\text{heads})}{p(T)} = \frac{.2 \times .6}{.2 \times .6 + .6 \times .4} \approx 0.334
\]

our setting: partially-observed

\[
p^{(t+1)}(\text{heads}) = \frac{\# \text{ expected heads from penny}}{\# \text{ total flips of penny}} = \mathbb{E}_{p^{(t)}}[\# \text{ expected heads from penny}] / \# \text{ total flips of penny}
\]
Three Coins Example

Use posteriors to update parameters

\[ p(\text{heads} \mid \text{obs. H}) = \frac{p(H \mid \text{heads})p(\text{heads})}{p(H)} \]
\[ = \frac{.8 \times .6}{.8 \times .6 + .6 \times .4} \approx 0.667 \]

\[ p(\text{heads} \mid \text{obs. T}) = \frac{p(T \mid \text{heads})p(\text{heads})}{p(T)} \]
\[ = \frac{.2 \times .6}{.2 \times .6 + .6 \times .4} \approx 0.334 \]

Our setting: partially observed

\[ p^{(t+1)}(\text{heads}) = \frac{\# \text{expected heads from penny}}{\# \text{total flips of penny}} \]
\[ = \frac{\mathbb{E}_{p(v)}[\# \text{expected heads from penny}]}{\# \text{total flips of penny}} \]
\[ = \frac{2 \times p(\text{heads} \mid \text{obs. H}) + 4 \times p(\text{heads} \mid \text{obs. T})}{6} \]
\[ \approx 0.444 \]
Expectation Maximization (EM)

0. Assume *some* value for your parameters

Two step, iterative algorithm:

1. E-step: count under uncertainty (compute expectations)

2. M-step: maximize log-likelihood, assuming these uncertain counts
Outline

Latent and probabilistic modeling
Generative Modeling
Example 1: A Model of Rolling a Die
Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
Basic idea
Three coins example
Why EM works
Why does EM work?

- \( X\): observed data
- \( Y\): unobserved data
- \( \mathcal{C}(\theta) = \) log-likelihood of complete data \((X,Y)\)
- \( \mathcal{M}(\theta) = \) marginal log-likelihood of observed data \(X\)
- \( \mathcal{P}(\theta) = \) posterior log-likelihood of incomplete data \(Y\)

what do \( \mathcal{C}, \mathcal{M}, \mathcal{P}\) look like?
Why does EM work?

\(X: \text{observed data}\)
\(Y: \text{unobserved data}\)

\(\mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X\)

\(C(\theta) = \text{log-likelihood of complete data } (X,Y)\)

\(\mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y\)

\[C(\theta) = \sum_i \log p(x_i, y_i)\]
Why does EM work?

\[ X: \text{observed data} \quad Y: \text{unobserved data} \]

\[ C(\theta) = \text{log-likelihood of complete data (X,Y)} \]

\[ M(\theta) = \text{marginal log-likelihood of observed data } X \]

\[ P(\theta) = \text{posterior log-likelihood of incomplete data } Y \]

\[ C(\theta) = \sum_i \log p(x_i, y_i) \]

\[ M(\theta) = \sum_i \log p(x_i) = \sum_i \log \sum_k p(x_i, y = k) \]
### Why does EM work?

- **X**: observed data
- **Y**: unobserved data

**\( \mathcal{M}(\theta) \)** = marginal log-likelihood of observed data X

**\( \mathcal{C}(\theta) \)** = log-likelihood of complete data (X,Y)

**\( \mathcal{P}(\theta) \)** = posterior log-likelihood of incomplete data Y

\[
\mathcal{C}(\theta) = \sum_i \log p(x_i, y_i)
\]

\[
\mathcal{M}(\theta) = \sum_i \log p(x_i) = \sum_i \log \sum_k p(x_i, y = k)
\]

\[
\mathcal{P}(\theta) = \sum_i \log p(y_i|x_i)
\]
Why does EM work?

\( X: \) observed data \quad \text{\( Y: \) unobserved data}

\( \mathcal{M}(\theta) = \) marginal log-likelihood of observed data \( X \)

\( \mathcal{C}(\theta) = \) log-likelihood of complete data \( (X,Y) \)

\( \mathcal{P}(\theta) = \) posterior log-likelihood of incomplete data \( Y \)

\[
p_{\theta}(Y \mid X) = \frac{p_{\theta}(X, Y)}{p_{\theta}(X)} \quad \text{definition of conditional probability}
\]

\[
p_{\theta}(X) = \frac{p_{\theta}(X, Y)}{p_{\theta}(Y \mid X)} \quad \text{algebra}
\]
Why does EM work?

\[ X: \text{observed data} \quad Y: \text{unobserved data} \]

\[ \mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X \]

\[ \mathcal{C}(\theta) = \text{log-likelihood of complete data } (X,Y) \]

\[ \mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y \]

\[ p_\theta(Y \mid X) = \frac{p_\theta(X,Y)}{p_\theta(X)} \quad \Rightarrow \quad p_\theta(X) = \frac{p_\theta(X,Y)}{p_\theta(Y \mid X)} \]

\[ \mathcal{C}(\theta) = \sum_i \log p(x_i, y_i) \]

\[ \mathcal{M}(\theta) = \sum_i \log p(x_i) = \sum_i \log \sum_k p(x_i, y = k) \]

\[ \mathcal{P}(\theta) = \sum_i \log p(y_i \mid x_i) \]

\[ \mathcal{M}(\theta) = \mathcal{C}(\theta) - \mathcal{P}(\theta) \]
Why does EM work?

\[ X: \text{observed data} \quad Y: \text{unobserved data} \]

\[ \mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X \]

\[ \mathcal{C}(\theta) = \text{log-likelihood of complete data } (X,Y) \]

\[ \mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y \]

\[ p_\theta(Y \mid X) = \frac{p_\theta(X,Y)}{p_\theta(X)} \quad \rightarrow \quad p_\theta(X) = \frac{p_\theta(X,Y)}{p_\theta(Y \mid X)} \]

\[ \mathcal{M}(\theta) = \mathcal{C}(\theta) - \mathcal{P}(\theta) \]

\[ \mathbb{E}_{Y \sim \theta(t)}[\mathcal{M}(\theta) \mid X] = \mathbb{E}_{Y \sim \theta(t)}[\mathcal{C}(\theta) \mid X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta) \mid X] \]

*take a conditional expectation
(why? we’ll cover this more in variational inference)*
Why does EM work?

<table>
<thead>
<tr>
<th>$X$: observed data</th>
<th>$Y$: unobserved data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X$</td>
<td>$C(\theta) = \text{log-likelihood of complete data (X,Y)}$</td>
</tr>
<tr>
<td>$\mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y$</td>
<td></td>
</tr>
</tbody>
</table>

$$p_\theta(Y \mid X) = \frac{p_\theta(X, Y)}{p_\theta(X)} \quad \Rightarrow \quad p_\theta(X) = \frac{p_\theta(X, Y)}{p_\theta(Y \mid X)}$$

$$\mathcal{M}(\theta) = C(\theta) - \mathcal{P}(\theta)$$

$$\mathbb{E}_{Y \sim \theta(t)}[\mathcal{M}(\theta) \mid X] = \mathbb{E}_{Y \sim \theta(t)}[C(\theta) \mid X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta) \mid X]$$

$$\mathcal{M}(\theta) = \mathbb{E}_{Y \sim \theta(t)}[C(\theta) \mid X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta) \mid X]$$

$\mathcal{M}$ already sums over $Y$
**Why does EM work?**

<table>
<thead>
<tr>
<th>$X$: observed data</th>
<th>$Y$: unobserved data</th>
</tr>
</thead>
</table>

\[ \mathcal{M}(\theta) = \text{marginal log-likelihood of observed data $X$} \]

\[ \mathcal{C}(\theta) = \text{log-likelihood of complete data (X,Y)} \]

\[ \mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data Y} \]

\[
\mathcal{M}(\theta) = \mathbb{E}_{Y \sim \theta(t)}[\mathcal{C}(\theta)|X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta)|X]
\]

\[\mathbb{E}_{Y \sim \theta(t)}[\mathcal{C}(\theta)|X] = \sum_{i} \sum_{k} p_{\theta(t)}(y = k | x_i) \log p(x_i, y = k)\]
Why does EM work?

<table>
<thead>
<tr>
<th>X: observed data</th>
<th>Y: unobserved data</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(\theta) = ) marginal log-likelihood of observed data ( X )</td>
<td>( C(\theta) = ) log-likelihood of complete data ( (X, Y) )</td>
</tr>
<tr>
<td>( P(\theta) = ) posterior log-likelihood of incomplete data ( Y )</td>
<td></td>
</tr>
</tbody>
</table>

\[
M(\theta) = \mathbb{E}_{Y \sim \theta(t)}[C(\theta)|X] - \mathbb{E}_{Y \sim \theta(t)}[P(\theta)|X]
\]

Let \( \theta^* \) be the value that maximizes \( Q(\theta, \theta(t)) \)
Why does EM work?

\(X: \text{observed data}
\]
\(Y: \text{unobserved data}
\]
\(\mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X\)
\(\mathcal{C}(\theta) = \text{log-likelihood of complete data } (X,Y)\)
\(\mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y\)

\[
\mathcal{M}(\theta) = \mathbb{E}_{Y \sim \theta(t)}[\mathcal{C}(\theta)|X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta)|X]
\]
\(Q(\theta, \theta^{(t)})\)
\(R(\theta, \theta^{(t)})\)

Let \(\theta^*\) be the value that maximizes \(Q(\theta, \theta^{(t)})\)

\[
\mathcal{M}(\theta^*) - \mathcal{M}(\theta^{(t)}) = (Q(\theta^*, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})) - (R(\theta^*, \theta^{(t)}) - R(\theta^{(t)}, \theta^{(t)}))
\]
Why does EM work?

$X$: observed data  
$Y$: unobserved data

$\mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X$

$\mathcal{C}(\theta) = \text{log-likelihood of complete data } (X,Y)$

$\mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y$

$\mathcal{M}(\theta) = \mathbb{E}_{Y \sim \theta^{(t)}}[\mathcal{C}(\theta)|X] - \mathbb{E}_{Y \sim \theta^{(t)}}[\mathcal{P}(\theta)|X]$

$Q(\theta, \theta^{(t)})$

$R(\theta, \theta^{(t)})$

Let $\theta^*$ be the value that maximizes $Q(\theta, \theta^{(t)})$

$\mathcal{M}(\theta^*) - \mathcal{M}(\theta^{(t)}) = (Q(\theta^*, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})) - (R(\theta^*, \theta^{(t)}) - R(\theta^{(t)}, \theta^{(t)})) \geq 0$

$\leq 0$ (we’ll see why with Jensen’s inequality, in variational inference)
**Why does EM work?**

<table>
<thead>
<tr>
<th>$X$: observed data</th>
<th>$Y$: unobserved data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}(\theta) = \text{marginal log-likelihood of observed data } X$</td>
<td>$\mathcal{C}(\theta) = \text{log-likelihood of complete data } (X,Y)$</td>
</tr>
</tbody>
</table>

| $\mathcal{P}(\theta) = \text{posterior log-likelihood of incomplete data } Y$ |

\[
\mathcal{M}(\theta) = \mathbb{E}_{Y \sim \theta(t)}[\mathcal{C}(\theta)|X] - \mathbb{E}_{Y \sim \theta(t)}[\mathcal{P}(\theta)|X] \\
\quad = Q(\theta, \theta(t)) - R(\theta, \theta(t)) \\
\]

Let $\theta^*$ be the value that maximizes $Q(\theta, \theta(t))$

\[
\mathcal{M}(\theta^*) - \mathcal{M}(\theta(t)) = (Q(\theta^*, \theta(t)) - Q(\theta(t), \theta(t))) - (R(\theta^*, \theta(t)) - R(\theta(t), \theta(t))) \\
\]

\[
\mathcal{M}(\theta^*) - \mathcal{M}(\theta(t)) \geq 0 \\
\text{EM does not decrease the marginal log-likelihood}
\]
Generalized EM

Partial M step: find a $\theta$ that simply increases, rather than maximizes, $Q$

Partial E step: only consider some of the variables (an online learning algorithm)
EM has its pitfalls

Objective is not convex $\rightarrow$ converge to a bad local optimum

Computing expectations can be hard: the E-step could require clever algorithms

How well does log-likelihood correlate with an end task?
A Maximization-Maximization Procedure

\[ F(\theta, q) = \mathbb{E}[C(\theta)] - \mathbb{E}[\log q(Z)] \]

any distribution over \( Z \)

we’ll see this again with variational inference
Outline

Latent and probabilistic modeling
  Generative Modeling
    Example 1: A Model of Rolling a Die
    Example 2: A Model of Conditional Die Rolls

EM (Expectation Maximization)
  Basic idea
  Three coins example
  Why EM works