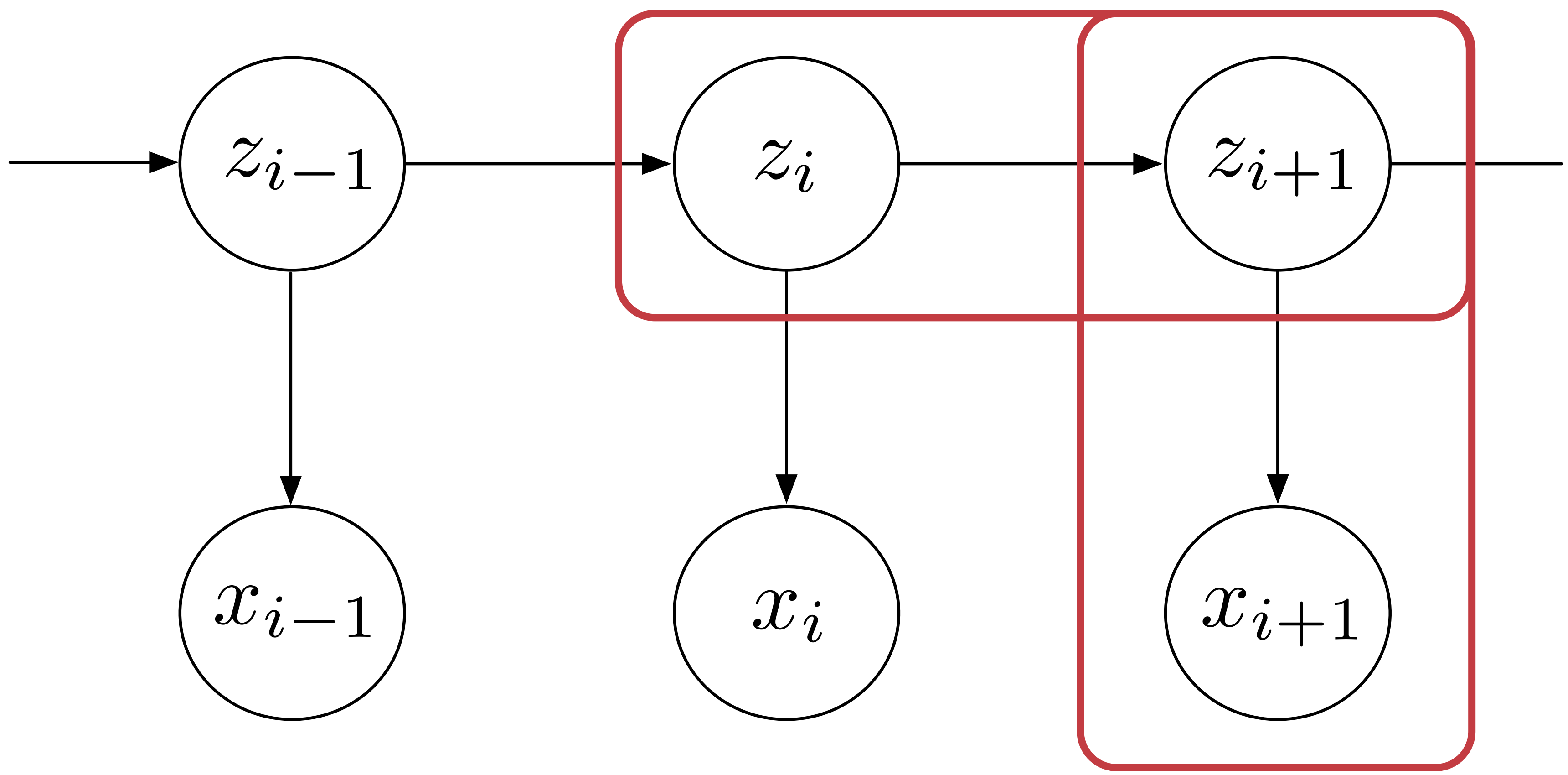
Basic HMM for POS Induction

Transition distribution:

$P(z'|z)$

Emission distribution:

$P(x|z)$



Parameterization

Key distribution: $P(x|\mathrm{NNP})$

$\mathbf{w}$:

+Cap	+2
+ing	-1

$\theta_{x\|\mathrm{NNP}}$	x	$\mathbf{f}$	$\mathrm{e}^{\mathbf{w}^\top \mathbf{f}}$
0.1	John	+Cap	0.3
0	Mary	+Cap	0.3
0.2	running	+ing	0.1
0	jumping	+ing	0.1

Parameterization

$$\mathbf{w} \dashrightarrow \theta$$

$$\theta_{x|z} = \frac{\exp(\mathbf{w}^{\top}\mathbf{f}(x,z))}{\sum_{x'}\exp(\mathbf{w}^{\top}\mathbf{f}(x',z))}$$

Unsupervised Learning with Features

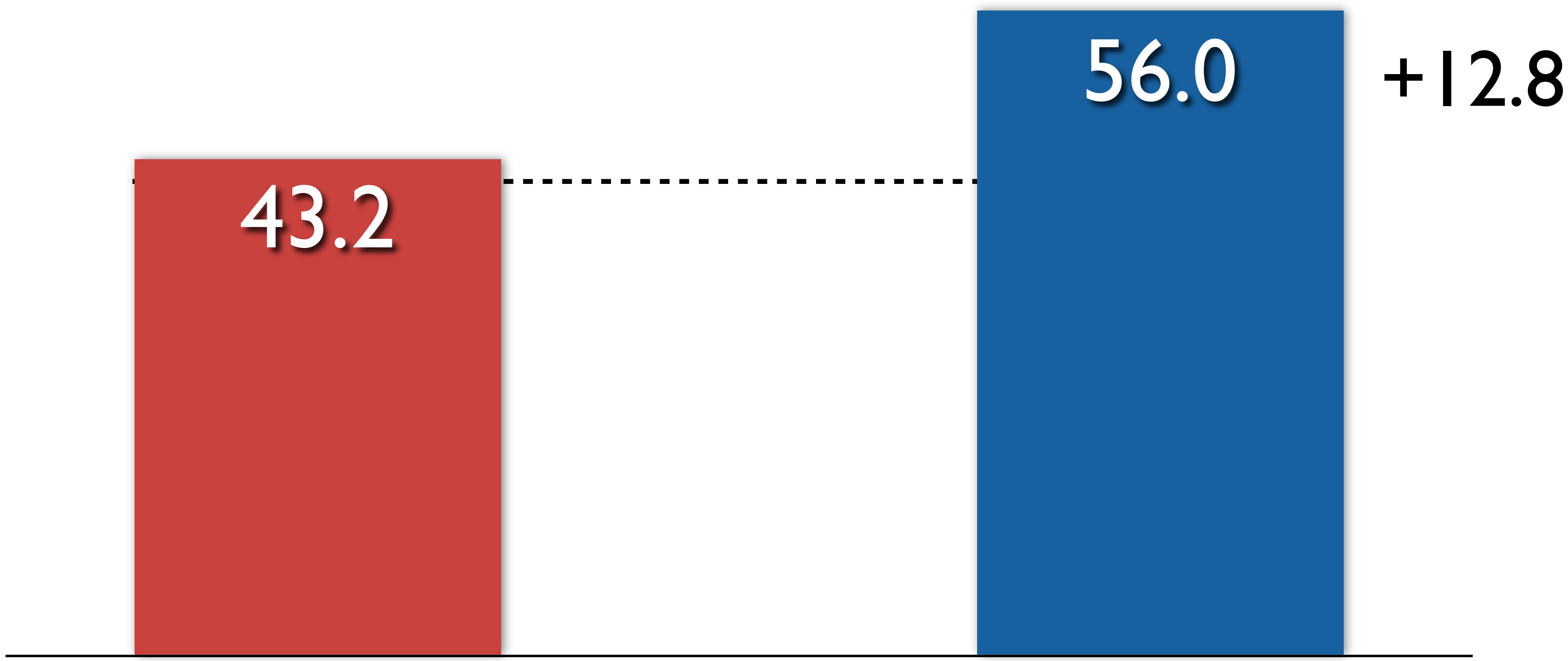
- Main idea:

  Local multinomials become maxents

- EM + Maxent M-Step =
  Unsupervised Learning w/ features

POS Induction Accuracy

1-to-1 Accuracy



Basic Multinomial:

$\text{John} \wedge \text{NNP}$

Rich Features:

$\text{John} \wedge \text{NNP}$

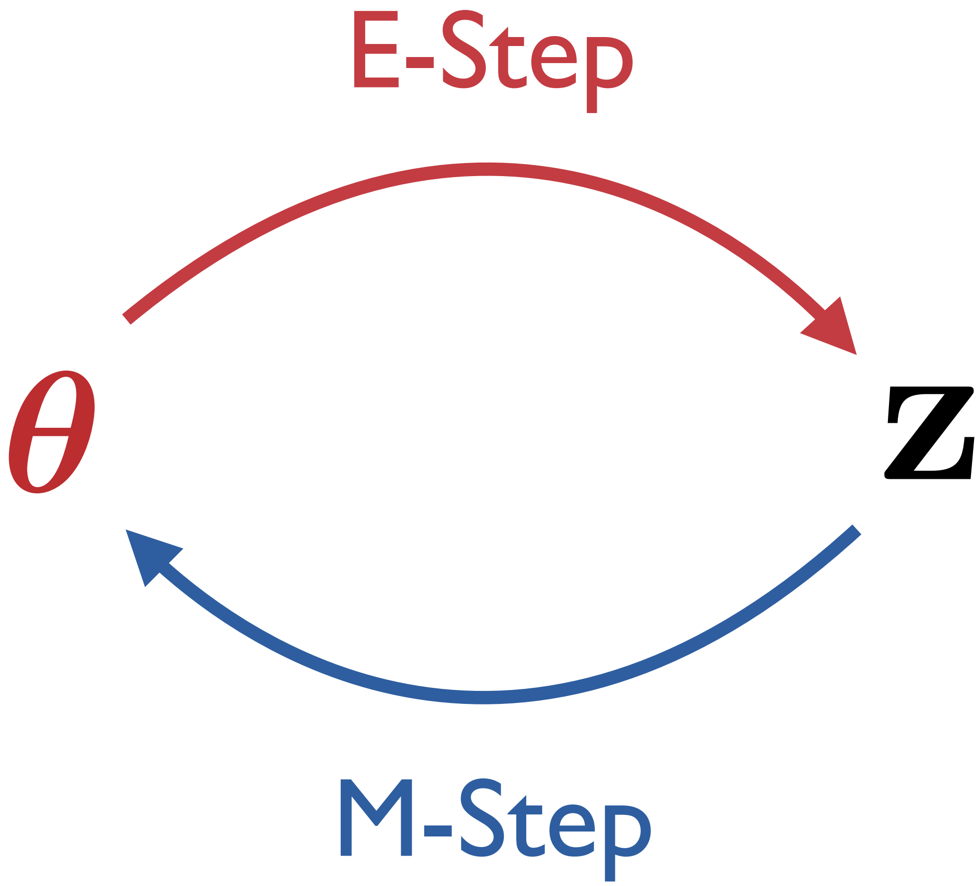
$+\text{Cap} \wedge \text{NNP}$

$+\text{Digit} \wedge \text{NNP}$

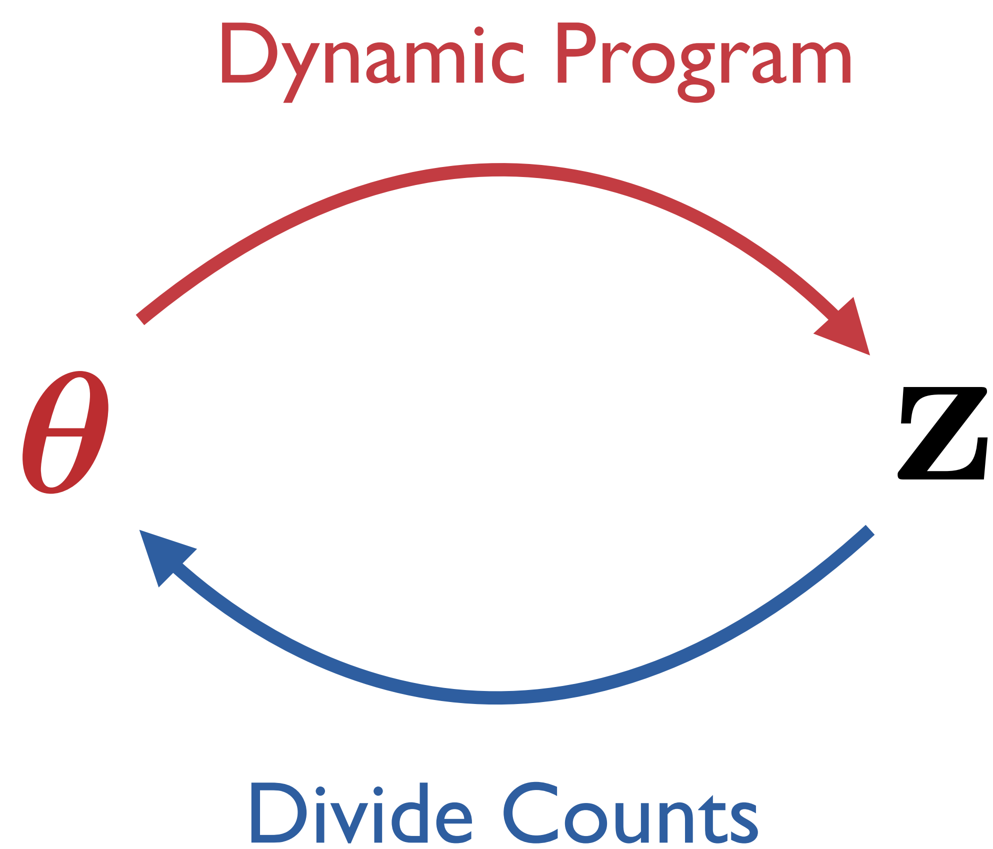
$+\text{Hyphen} \wedge \text{NNP}$

$+\text{ing} \wedge \text{NNP}$

Basic Hard EM



Basic Hard EM



E-Step: Dynamic Program

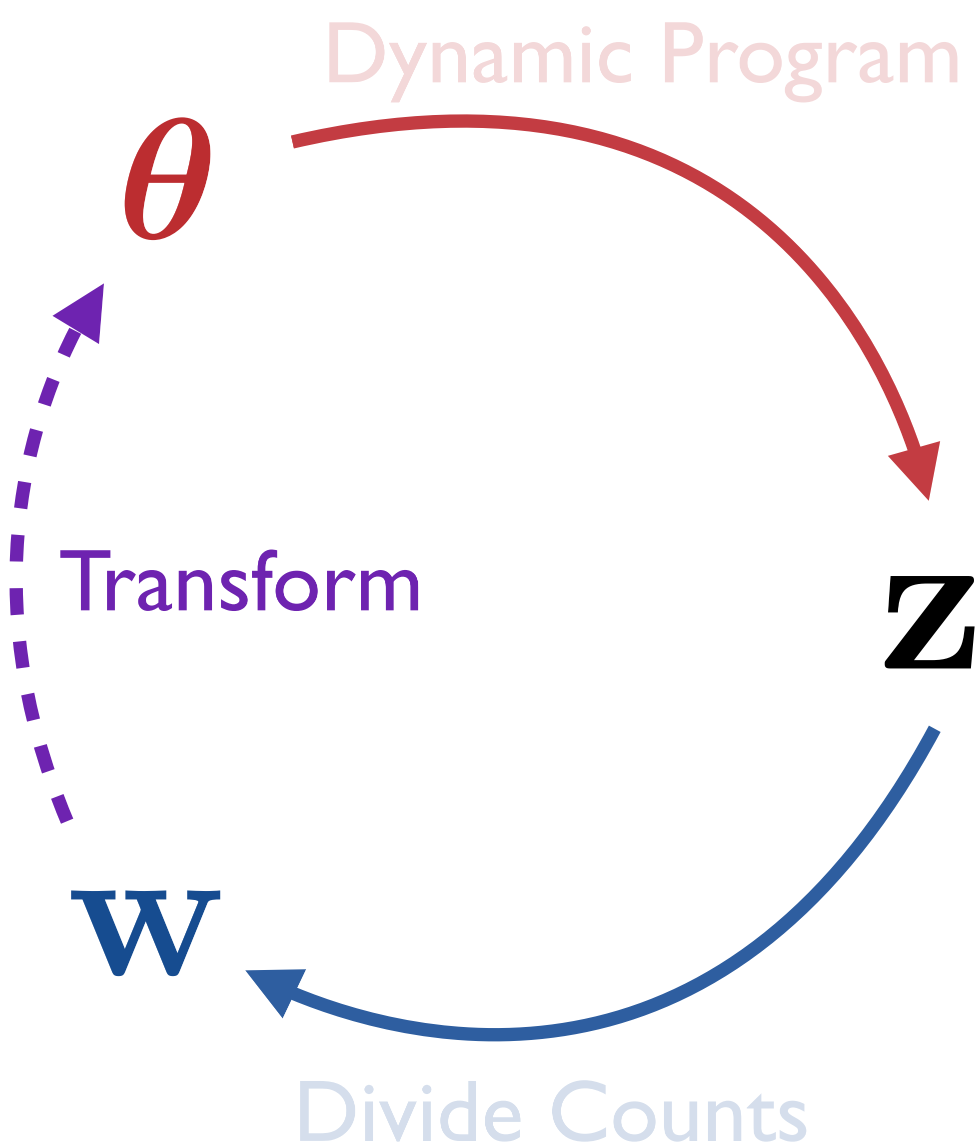
$$\mathbf{z} \leftarrow \underset{\mathbf{z}}{\operatorname{argmax}} \; P(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta})$$

M-Step: Divide Counts

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; P(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$$

$$= \left[\frac{c(z \longrightarrow x)}{c(z \longrightarrow \cdot)}, \; \ldots \right]$$

Hard EM with Features



E-Step: Dynamic Program

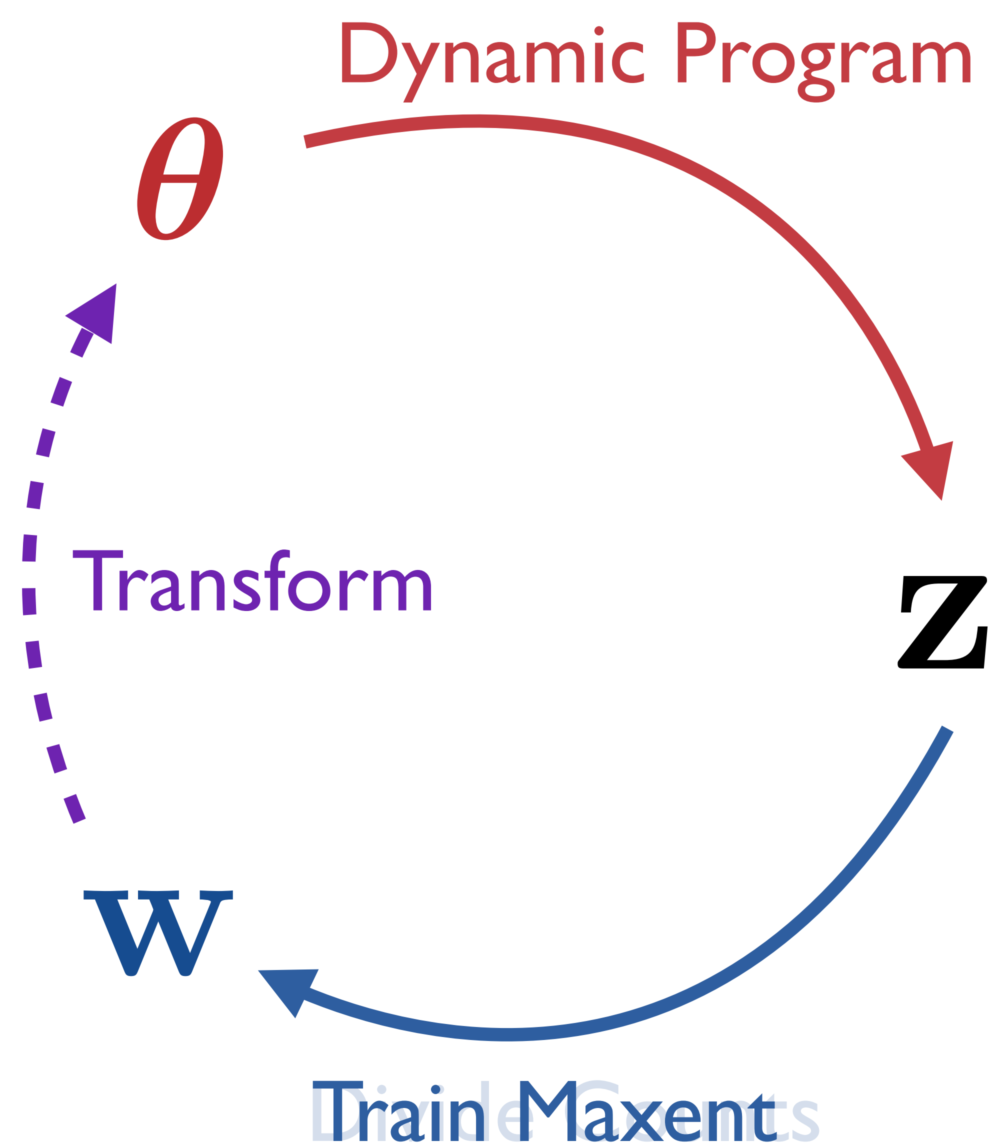
$$\mathbf{z} \leftarrow \operatorname*{argmax}_{\mathbf{z}} P(\mathbf{z}|\mathbf{x};\boldsymbol{\theta})$$

M-Step: Divide Counts

$$\boldsymbol{\theta} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} P(\mathbf{x},\mathbf{z};\boldsymbol{\theta})$$

$$= \left[\frac{c(z \rightarrow x)}{c(z \rightarrow \cdot)}, \; \ldots\right]$$

Hard EM with Features



E-Step: Dynamic Program

$$\mathbf{z} \leftarrow \underset{\mathbf{z}}{\operatorname{argmax}} \; P(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta})$$

M-Step: Train Maxent

$$\mathbf{w} \leftarrow \underset{\mathbf{w}}{\operatorname{argmax}} \log P(\mathbf{x}, \mathbf{z}; \mathbf{w})$$

M-Step: Divide Counts

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \; P(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$$

$$= \left[\frac{c(z \rightarrow x)}{c(z \rightarrow \cdot)}, \; \ldots \right]$$

Hard EM with Features

$$\log P(\mathbf{x}, \mathbf{z}; \mathbf{w})$$

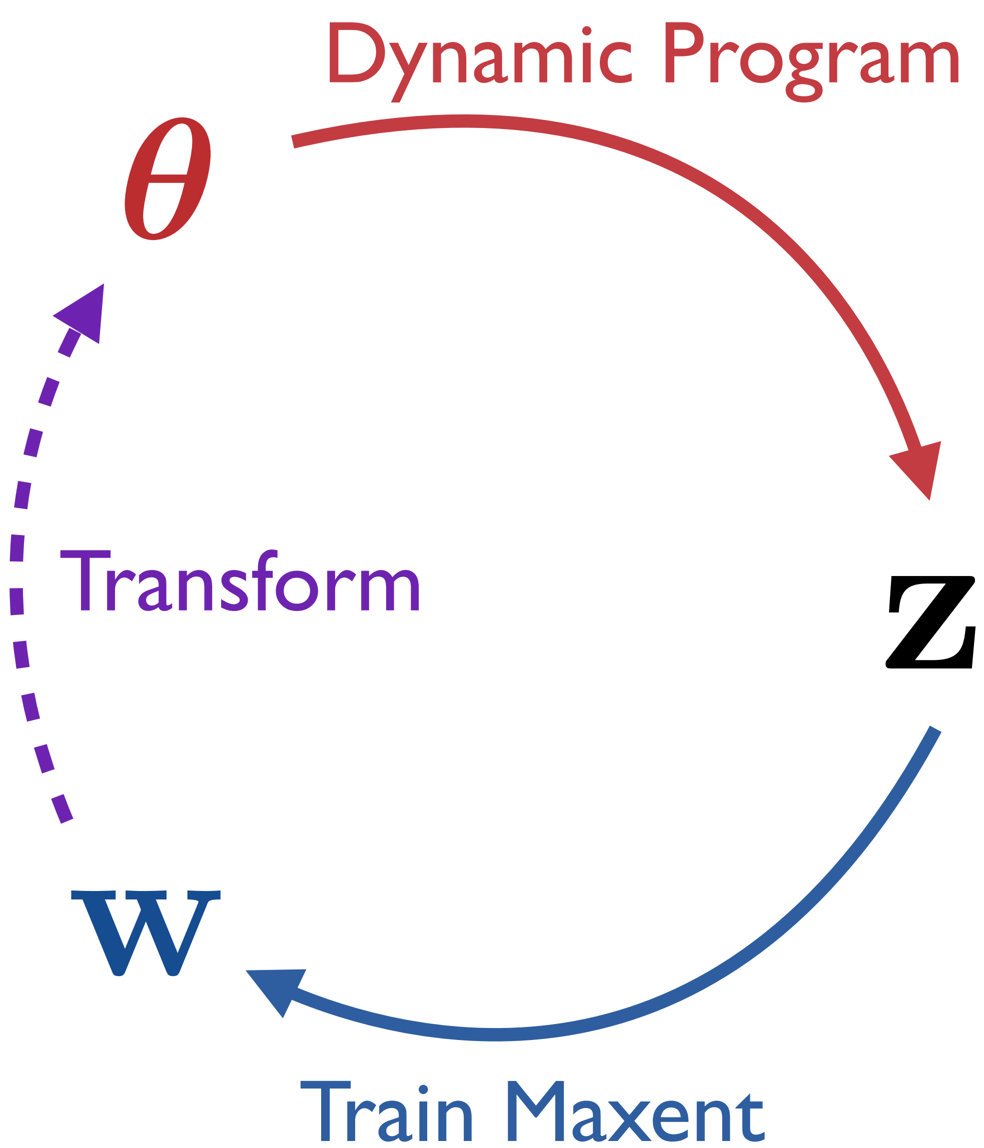
$$= \sum_i \log P(x_i | z_i; \mathbf{w}) + \ldots$$

Maxent training example

$$= \sum_{z,x} c(z \to x) \log P(x | z; \mathbf{w}) + \ldots$$

Multiplicity

Hard EM with Features



E-Step: Dynamic Program

$$\mathbf{z} \leftarrow \operatorname*{argmax}_{\mathbf{z}} P(\mathbf{z} | \mathbf{x}; \boldsymbol{\theta})$$

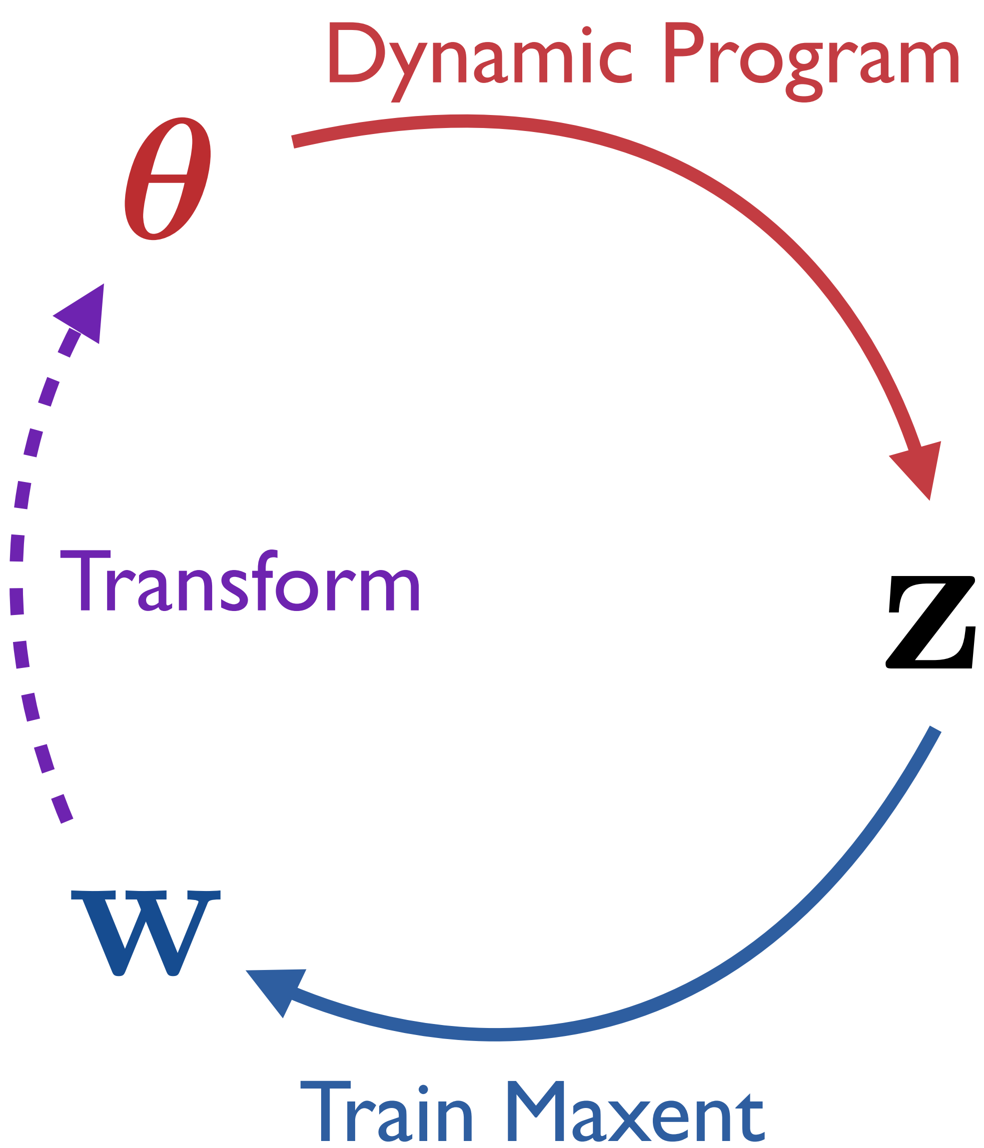
M-Step: Train Maxent

$$\mathbf{w} \leftarrow \operatorname*{argmax}_{\mathbf{w}} \log P(\mathbf{x}, \mathbf{z}; \mathbf{w})$$

Transform parameters

$$\theta_{x|z} \leftarrow \frac{\exp(\mathbf{w}^T \mathbf{f}(x, z))}{\sum_{x'} \exp(\mathbf{w}^T \mathbf{f}(x', z))}$$

EM with Features



<u>E-Step: Dynamic Program</u>

$$e(z \to x) \leftarrow \mathbb{E}\big[c(z \to x)\big]$$

<u>M-Step: Train Maxent</u>

$$\mathbf{w} \leftarrow \underset{\mathbf{w}}{\operatorname{argmax}}\ \mathbb{E}\big[\log P(\mathbf{x}, \mathbf{z}; \mathbf{w})\big]$$

<u>Transform parameters</u>

$$\theta_{x|z} \leftarrow \frac{\exp(\mathbf{w}^T \mathbf{f}(x, z))}{\sum_{x'} \exp(\mathbf{w}^T \mathbf{f}(x', z))}$$

Basic EM

Initialize probabilities $\boldsymbol{\theta}$

repeat

- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

until convergence

EM

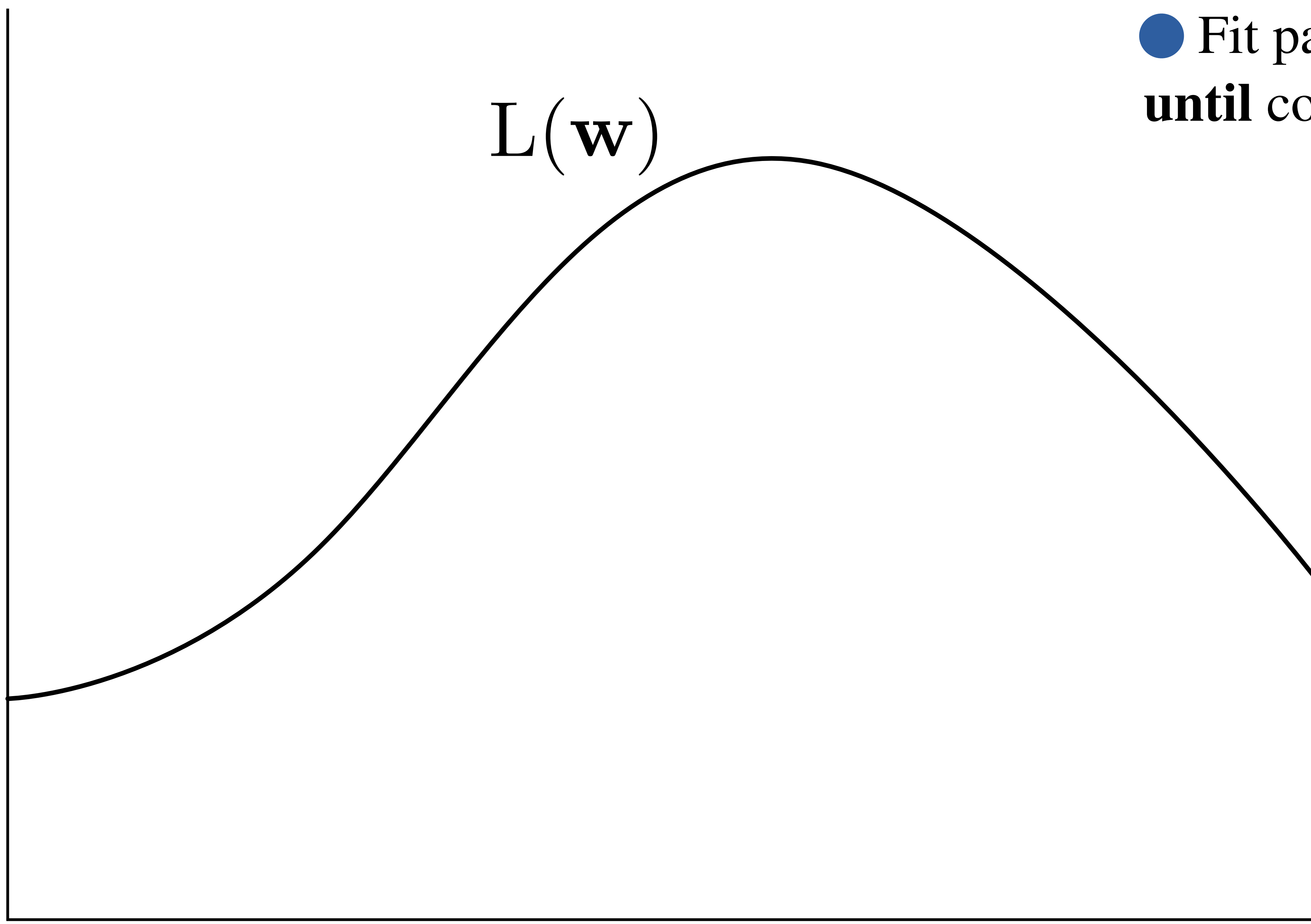
Basic EM

Initialize probabilities $\boldsymbol{\theta}$

repeat

- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

until convergence



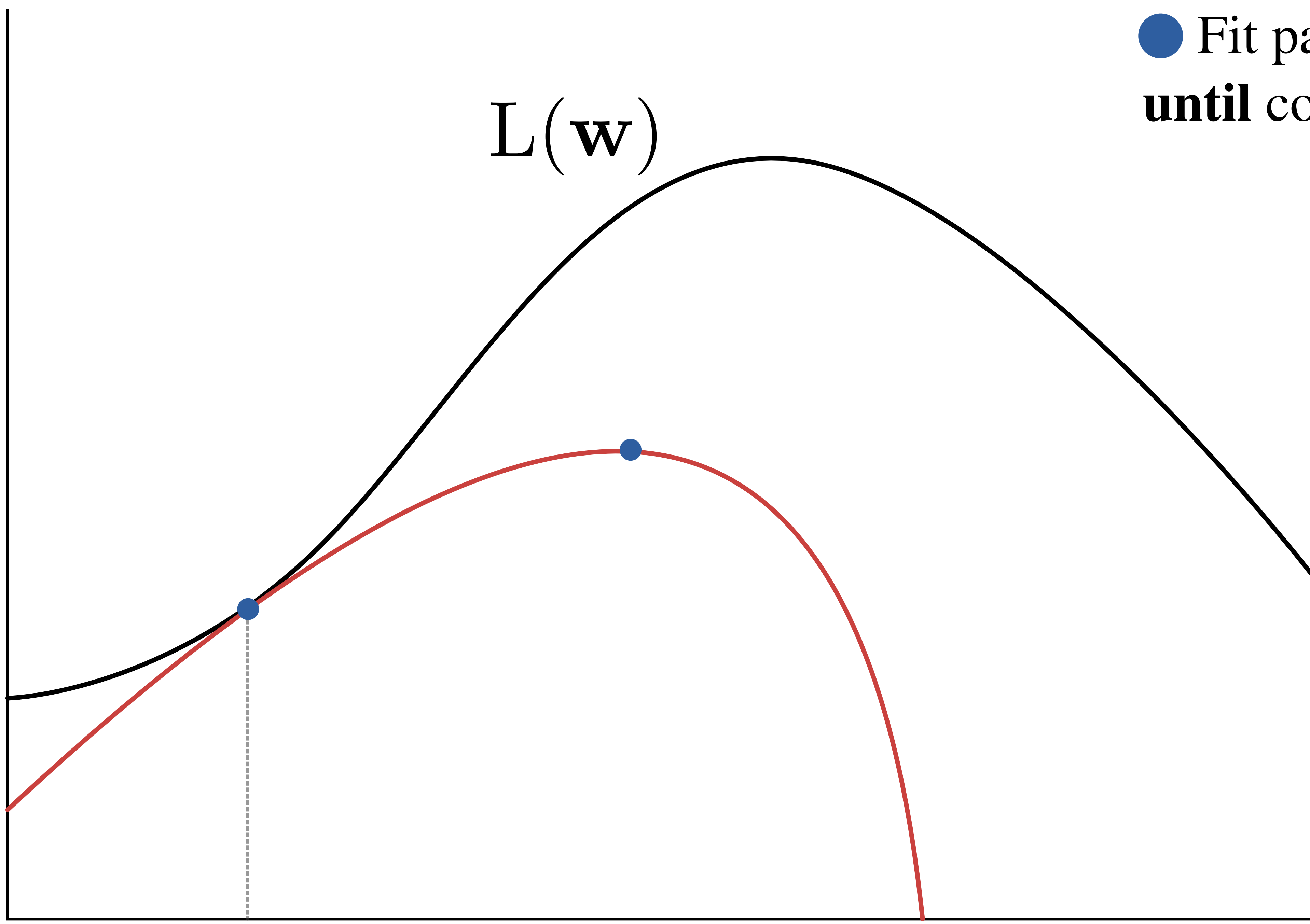
Basic EM

Initialize probabilities $\boldsymbol{\theta}$

repeat

- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

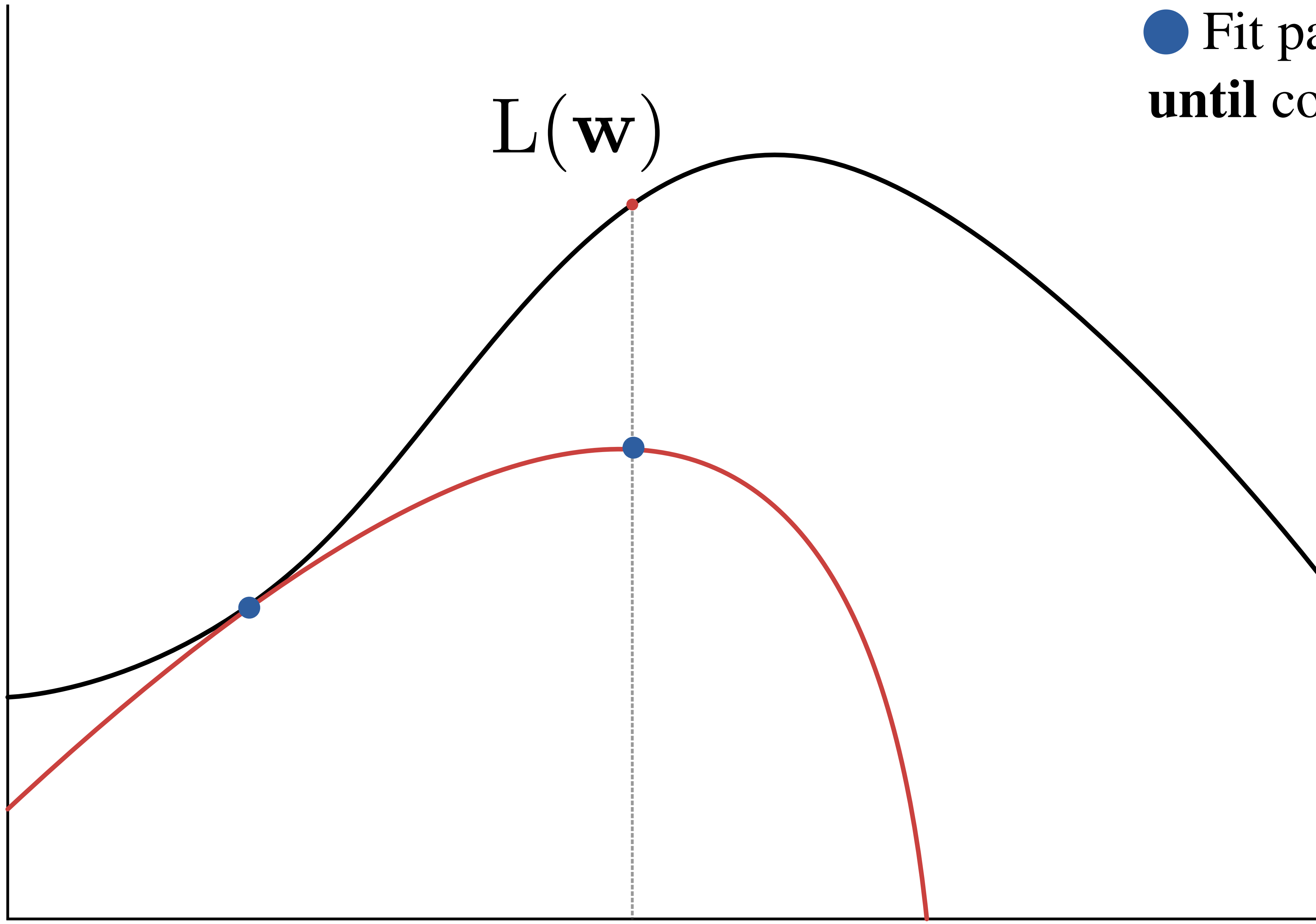
until convergence



Basic EM

Initialize probabilities $\boldsymbol{\theta}$
repeat
- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

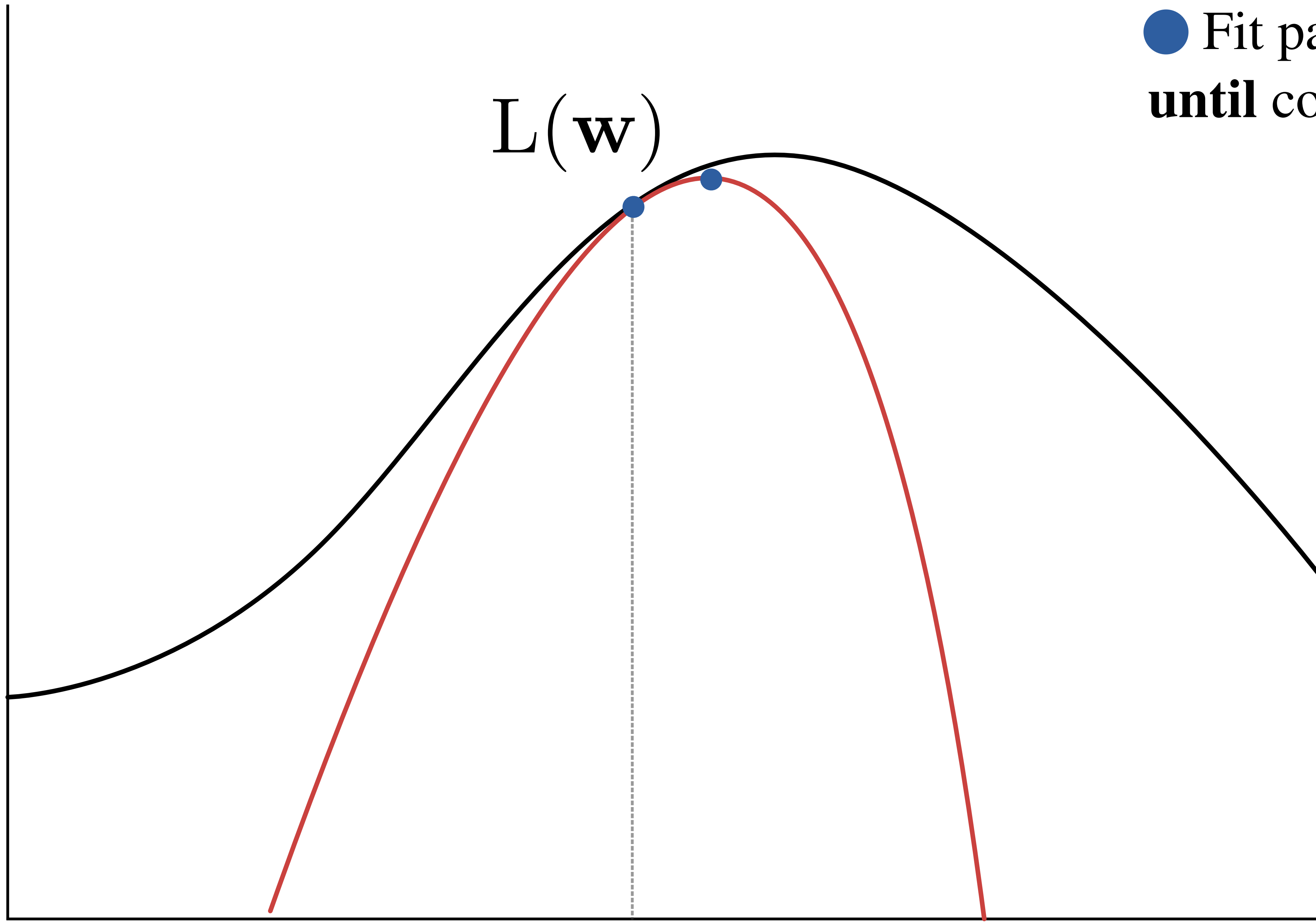
until convergence



Basic EM

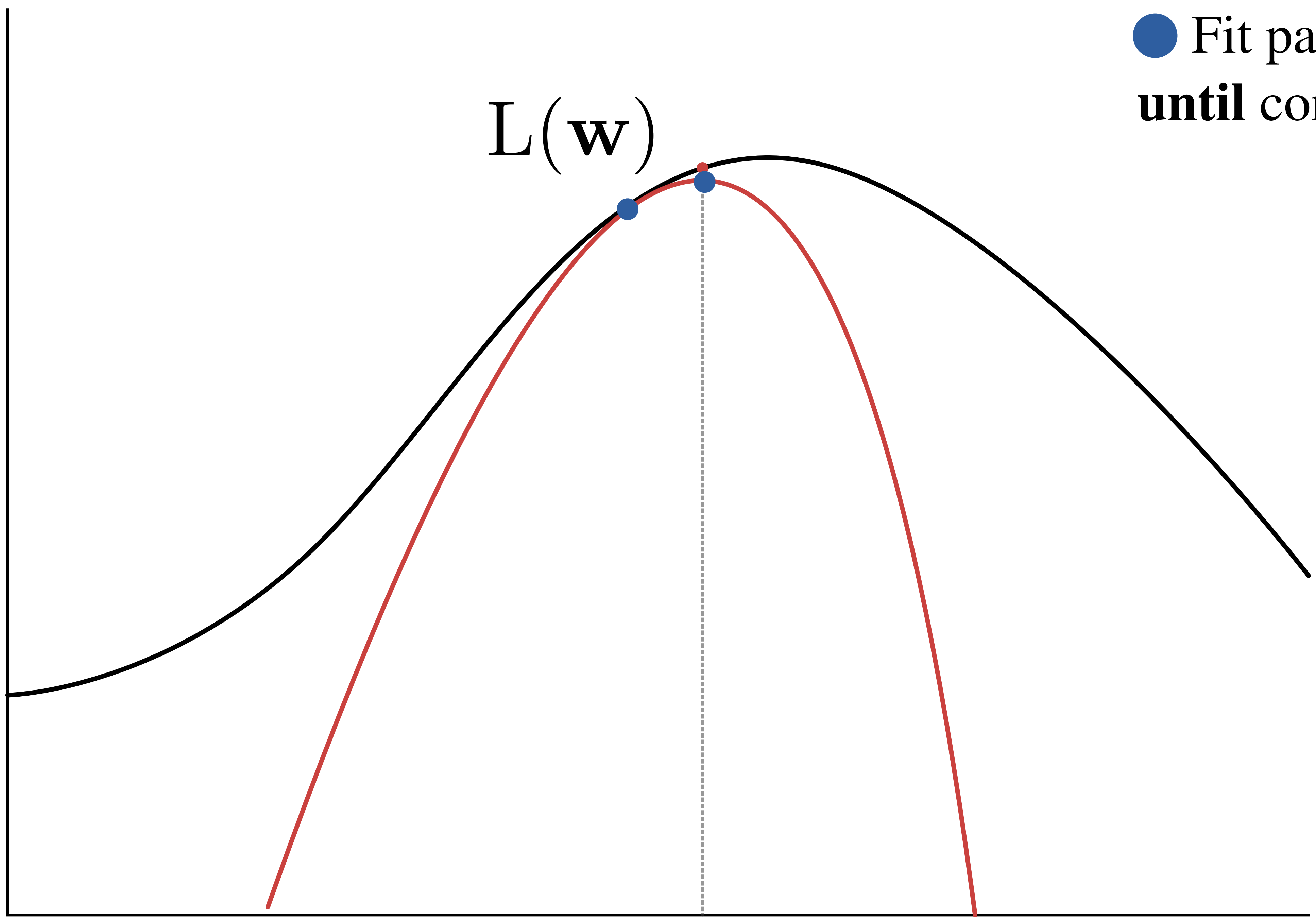
Initialize probabilities $\boldsymbol{\theta}$
repeat
- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

until convergence



Basic EM

Initialize probabilities $\boldsymbol{\theta}$
repeat
● Compute expected counts $\mathbf{e}$
● Fit parameters $\boldsymbol{\theta}$
until convergence



Basic EM

EM

Initialize probabilities $\boldsymbol{\theta}$

repeat

- Compute expected counts $\mathbf{e}$
- Fit parameters $\boldsymbol{\theta}$

until convergence

EM with Features

Initialize weights $\mathbf{w}$

EM

repeat

- Compute expected counts $\mathbf{e}$
- Fit parameters $\mathbf{w}$
- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

EM with Features

EM

Fit Params

Initialize weights $\mathbf{w}$

repeat

- Compute expected counts $\mathbf{e}$

  repeat

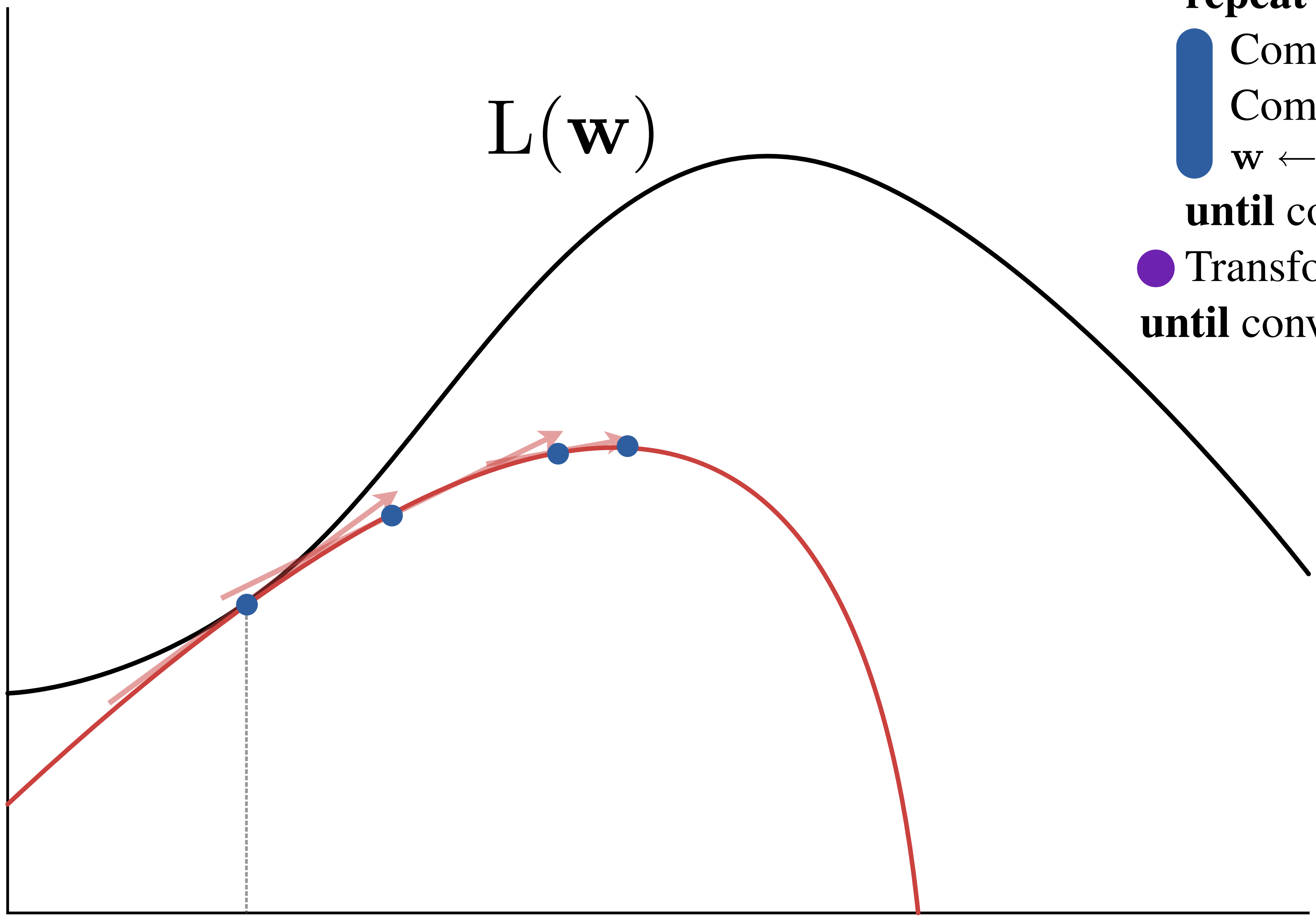
  - Compute $\ell(\mathbf{w}, \mathbf{e})$
  - Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$
  - $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

  until convergence

- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

EM with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

repeat

Compute $\ell(\mathbf{w}, \mathbf{e})$

Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$

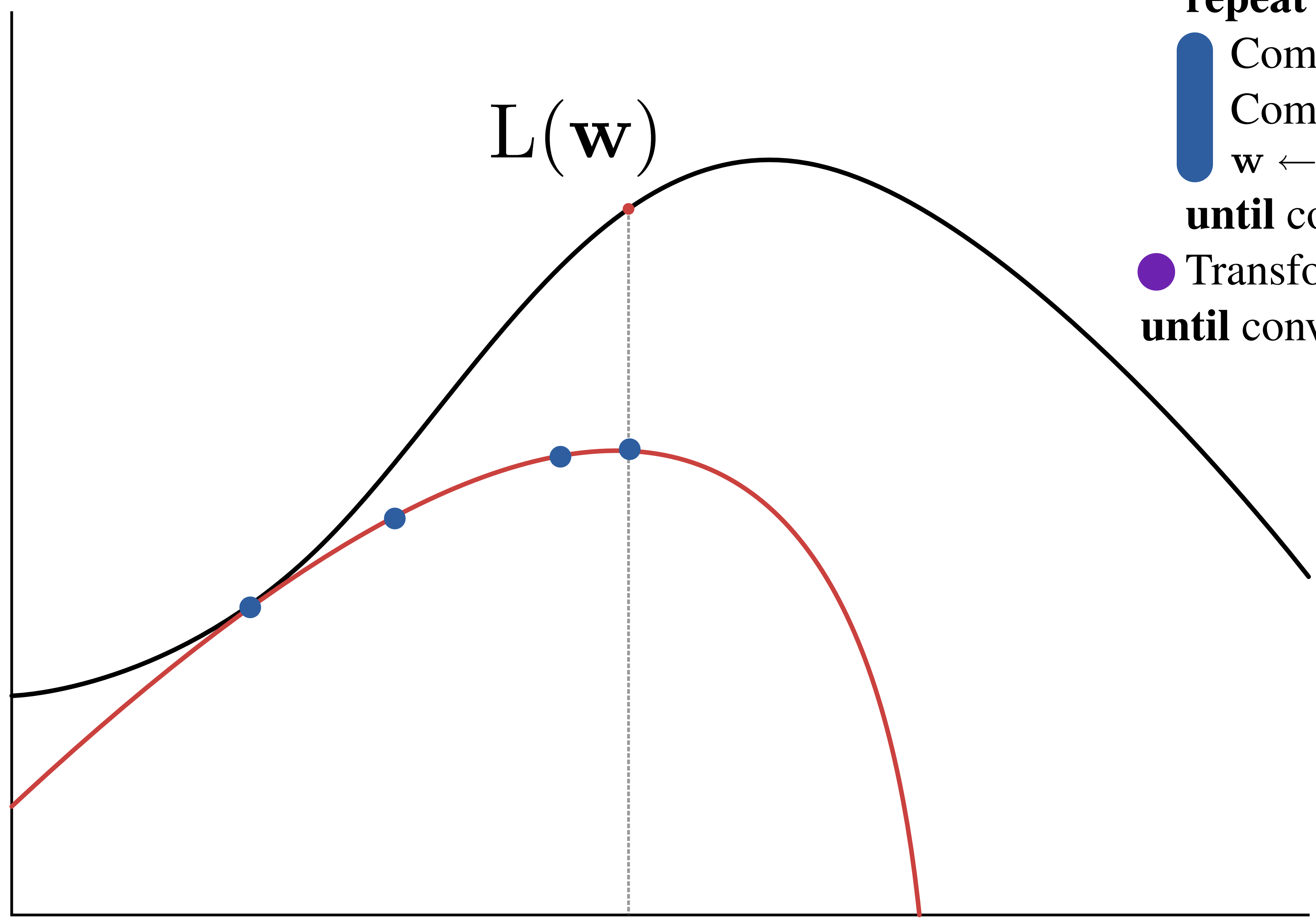
$\mathbf{w} \leftarrow \mathrm{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

until convergence

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

EM with Features

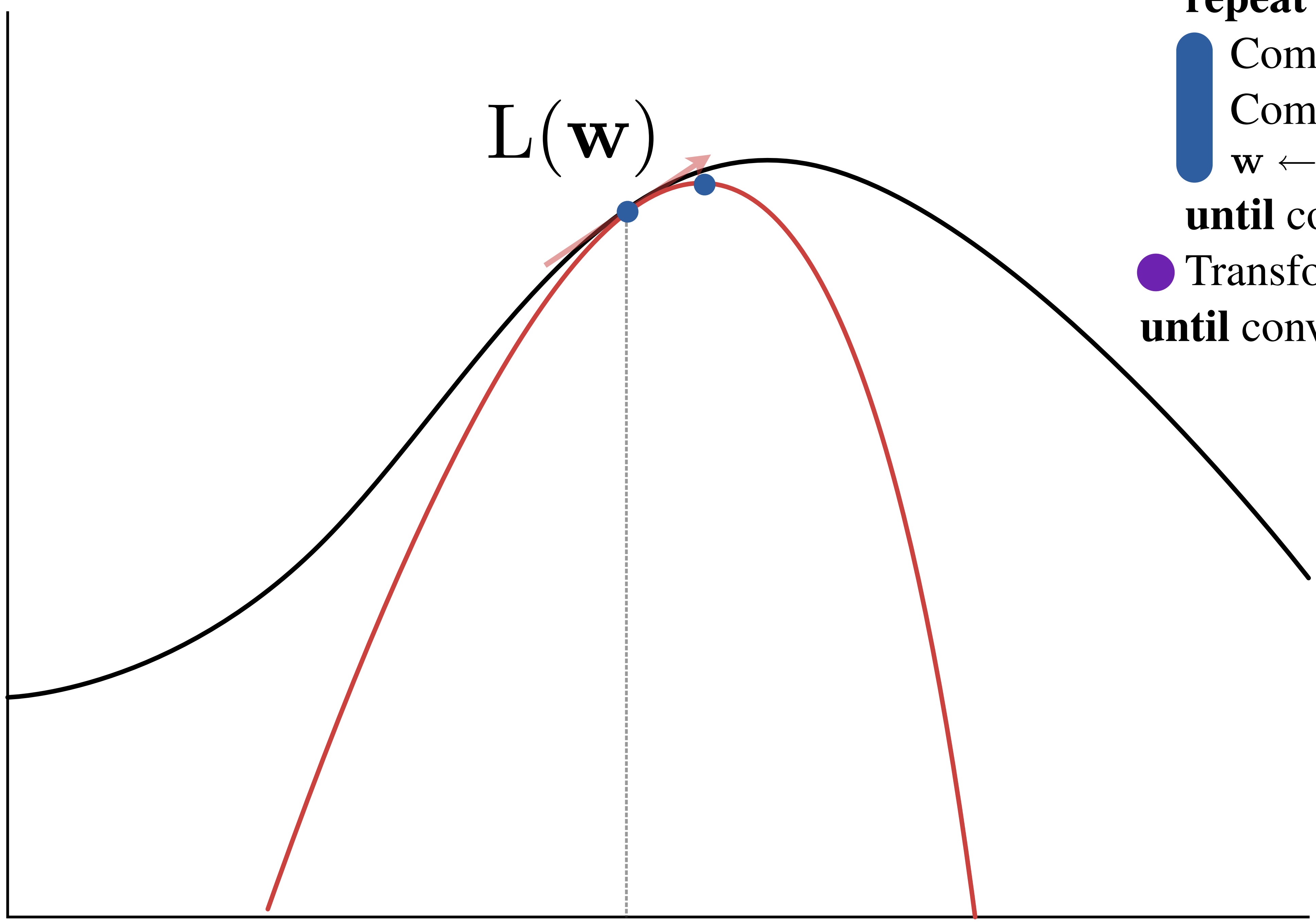


Initialize weights $\mathbf{w}$
repeat
- Compute expected counts $\mathbf{e}$
  repeat
  - Compute $\ell(\mathbf{w}, \mathbf{e})$
  - Compute $\nabla \ell(\mathbf{w}, \mathbf{e})$
  - $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla \ell(\mathbf{w}, \mathbf{e}))$

  until convergence
- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

EM with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

repeat

Compute $\ell(\mathbf{w}, \mathbf{e})$

Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$

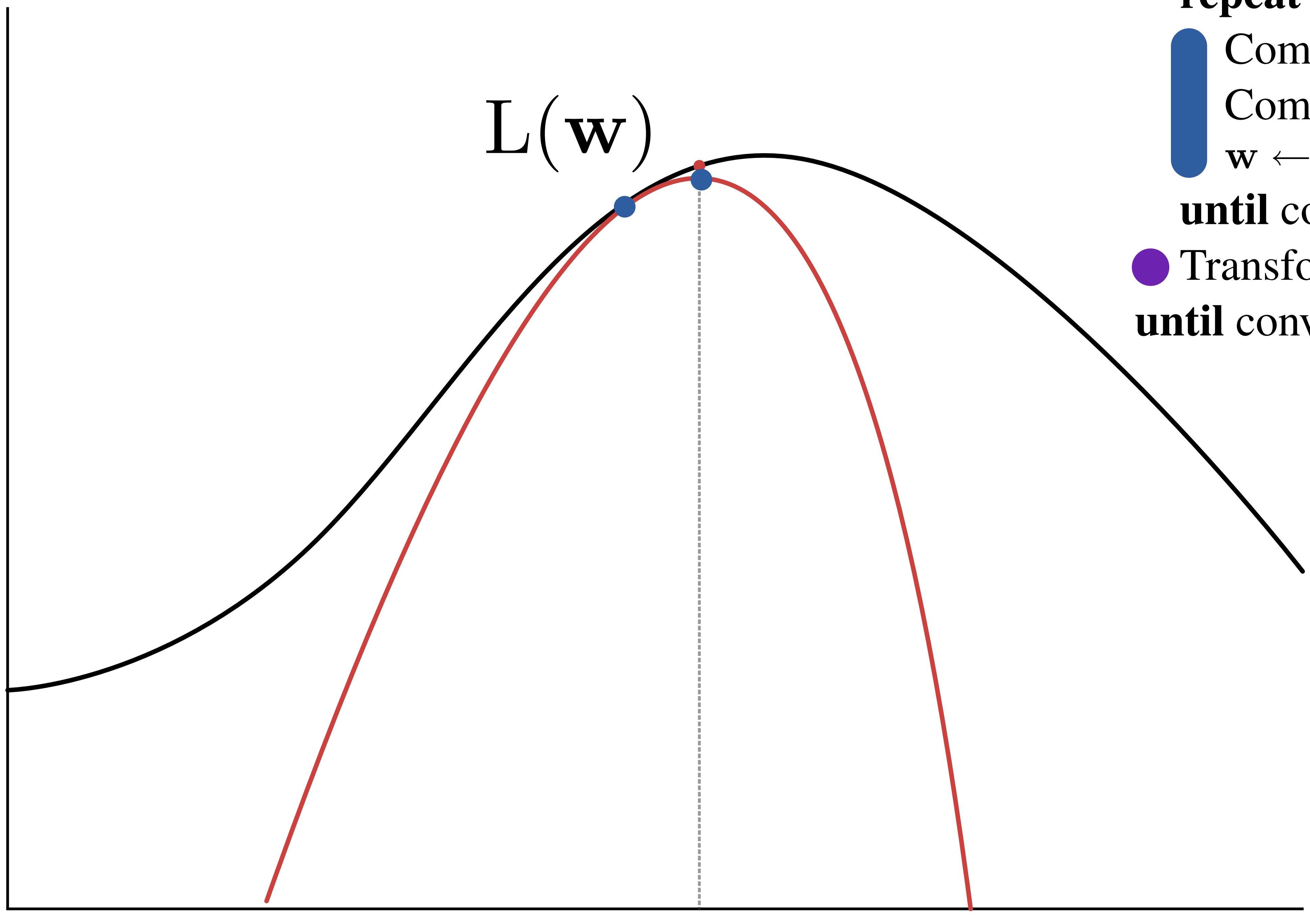
$\mathbf{w} \leftarrow \mathrm{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

until convergence

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

EM with Features



Initialize weights $\mathbf{w}$

repeat

- Compute expected counts $\mathbf{e}$

  repeat

  - Compute $\ell(\mathbf{w}, \mathbf{e})$
  - Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$
  - $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

  until convergence

- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

Direct Gradient with Features

EM w/ Features

Initialize weights $\mathbf{w}$
repeat
- Compute expected counts $\mathbf{e}$
  repeat
  - Compute $\ell(\mathbf{w}, \mathbf{e})$
  - Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$
  - $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

  until convergence
- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

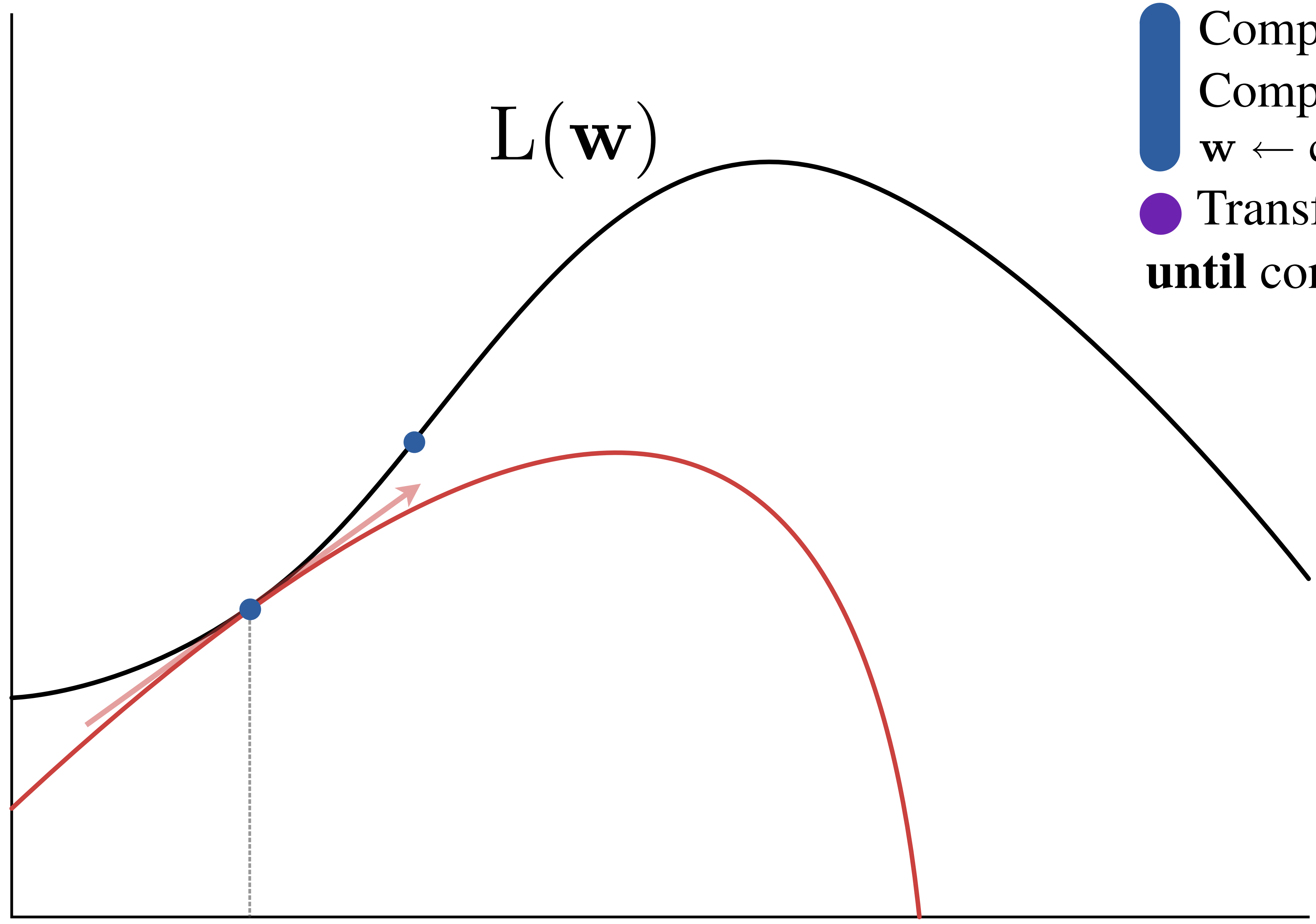
until convergence

DG w/ Features

Initialize weights $\mathbf{w}$
repeat
- Compute expected counts $\mathbf{e}$
- Compute $L(\mathbf{w})$
- Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$
- $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, L(\mathbf{w}), \nabla\ell(\mathbf{w}, \mathbf{e}))$
- Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

Direct Gradient with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

Compute $L(\mathbf{w})$

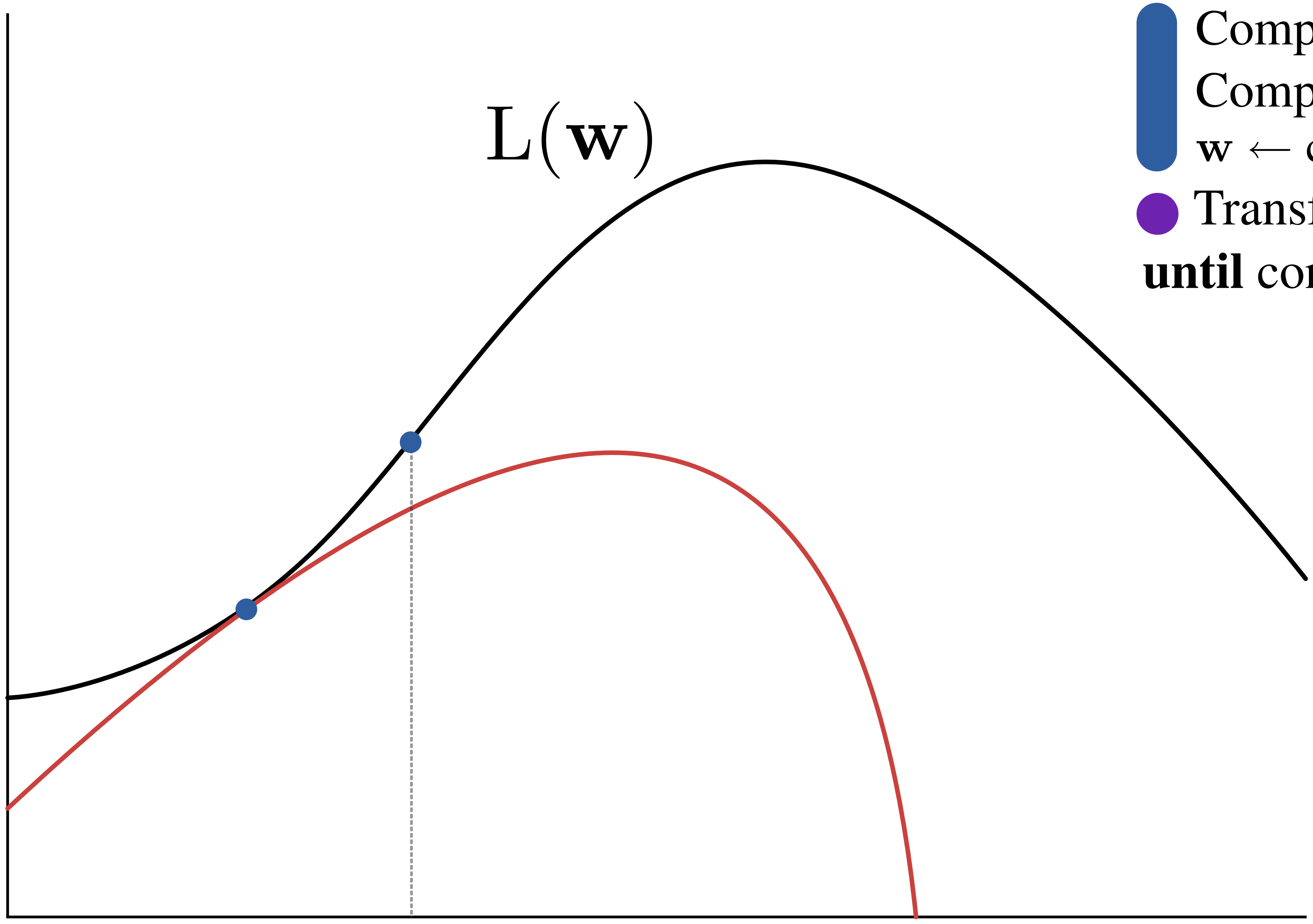
Compute $\nabla \ell(\mathbf{w}, \mathbf{e})$

$\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, L(\mathbf{w}), \nabla \ell(\mathbf{w}, \mathbf{e}))$

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

Direct Gradient with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

Compute $L(\mathbf{w})$

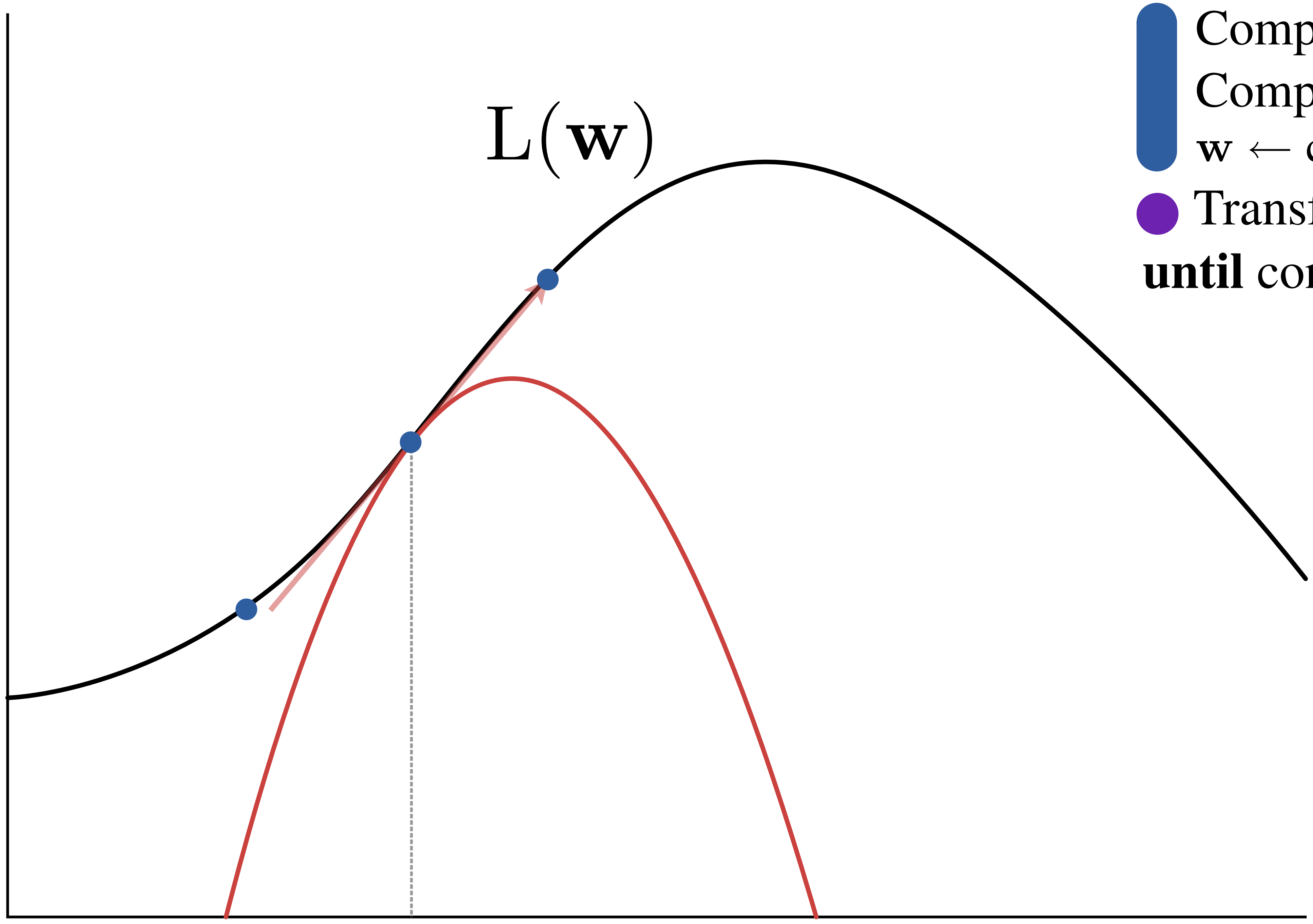
Compute $\nabla \ell(\mathbf{w}, \mathbf{e})$

$\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, L(\mathbf{w}), \nabla \ell(\mathbf{w}, \mathbf{e}))$

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

Direct Gradient with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

Compute $L(\mathbf{w})$

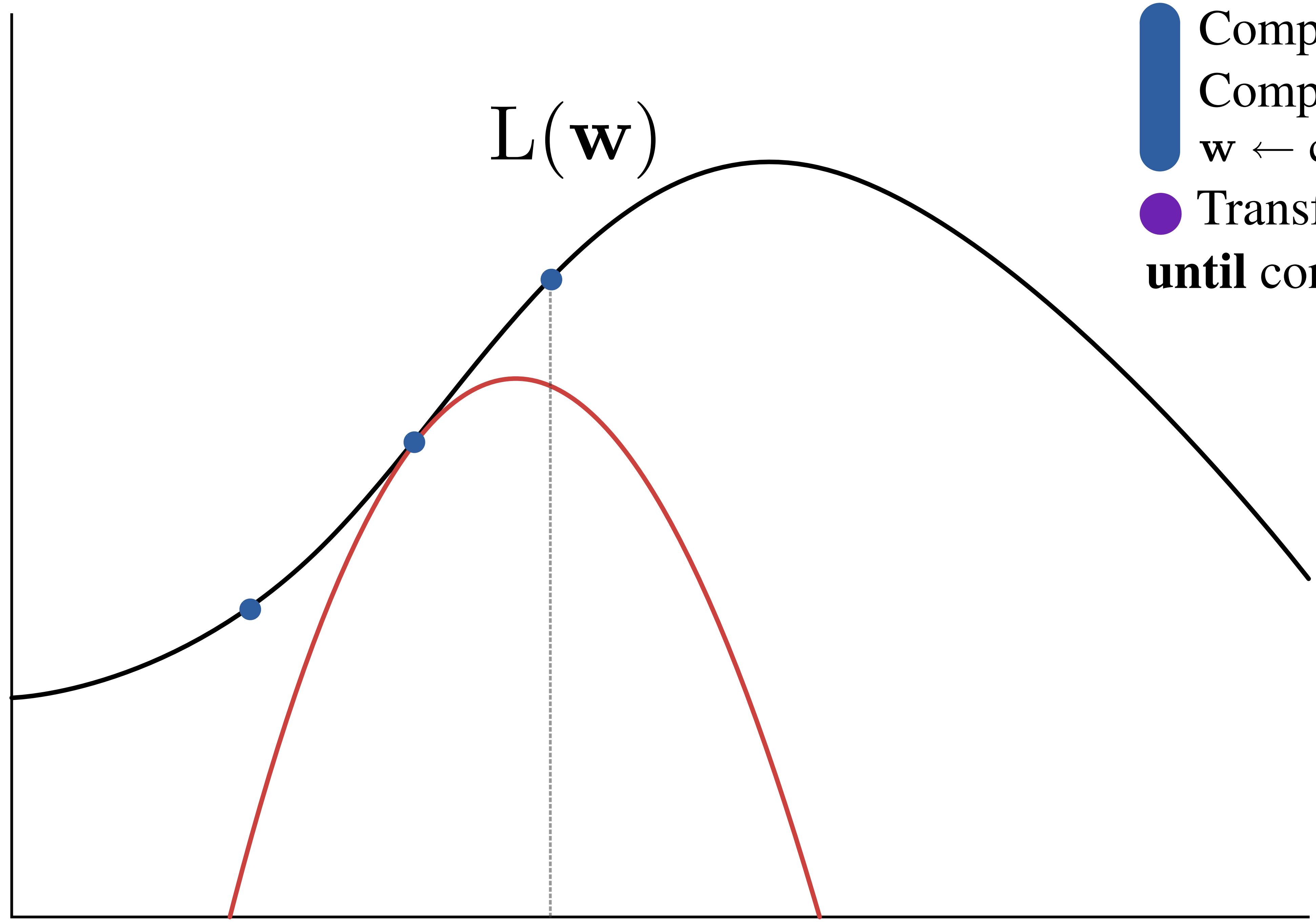
Compute $\nabla \ell(\mathbf{w}, \mathbf{e})$

$\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, L(\mathbf{w}), \nabla \ell(\mathbf{w}, \mathbf{e}))$

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

Direct Gradient with Features



Initialize weights $\mathbf{w}$

repeat

Compute expected counts $\mathbf{e}$

Compute $L(\mathbf{w})$

Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$

$\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, L(\mathbf{w}), \nabla\ell(\mathbf{w}, \mathbf{e}))$

Transform $\mathbf{w}$ to $\boldsymbol{\theta}$

until convergence

POS Induction Results

DT	JJ	NN	VBZ	IN	NN
The	green	cat	sleeps	at	home.

Features:

BASIC:	$\mathbb{1}(y = \cdot, z = \cdot)$
CONTAINS-DIGIT:	Check if y contains digit and conjoin with z: $\mathbb{1}(\text{containsDigit}(y) = \cdot, z = \cdot)$
CONTAINS-HYPHEN:	$\mathbb{1}(\text{containsHyphen}(x) = \cdot, z = \cdot)$
INITIAL-CAP:	Check if the first letter of y is capitalized: $\mathbb{1}(\text{isCap}(y) = \cdot, z = \cdot)$
N-GRAM:	Indicator functions for character n-grams of up to length 3 present in y.

POS Induction Results

DT	JJ	NN	VBZ	IN	NN
The	green	cat	sleeps	at	home.

Features:

BASIC:	$\mathbb{1}(y = \cdot, z = \cdot)$
CONTAINS-DIGIT:	Check if y contains digit and conjoin with z: $\mathbb{1}(\text{containsDigit}(y) = \cdot, z = \cdot)$
CONTAINS-HYPHEN:	$\mathbb{1}(\text{containsHyphen}(x) = \cdot, z = \cdot)$
INITIAL-CAP:	Check if the first letter of y is capitalized: $\mathbb{1}(\text{isCap}(y) = \cdot, z = \cdot)$
N-GRAM:	Indicator functions for character n-grams of up to length 3 present in y.

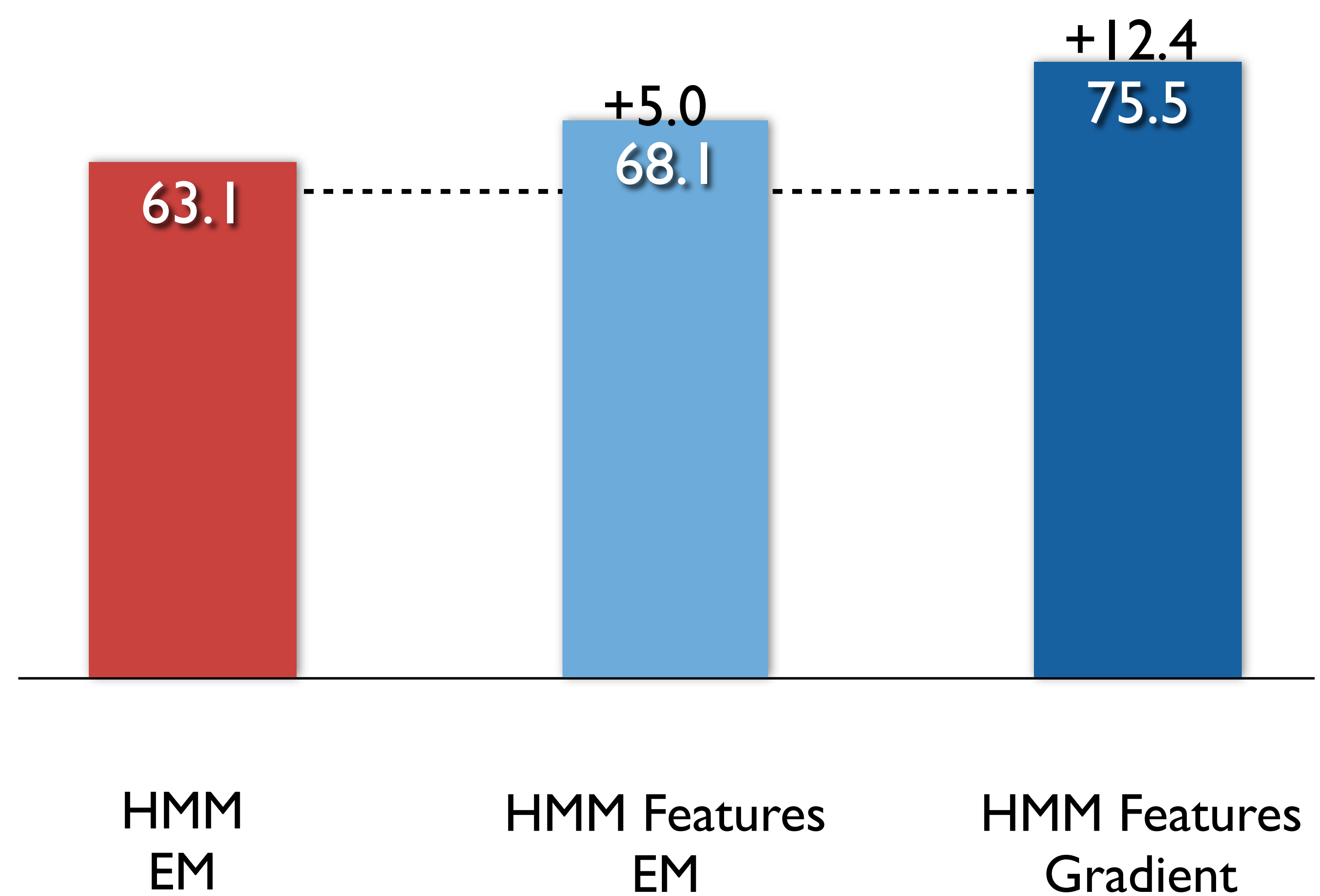
Data:

Train and test on entire WSJ

No tagging dictionary

45 POS tags

Many-to-1 Accuracy



POS Induction Results

DT	JJ	NN	VBZ	IN	NN
The	green	cat	sleeps	at	home.

Features:

BASIC:	$\mathbb{1}(y = \cdot, z = \cdot)$
CONTAINS-DIGIT:	Check if y contains digit and conjoin with z: $\mathbb{1}(\text{containsDigit}(y) = \cdot, z = \cdot)$
CONTAINS-HYPHEN:	$\mathbb{1}(\text{containsHyphen}(x) = \cdot, z = \cdot)$
INITIAL-CAP:	Check if the first letter of y is capitalized: $\mathbb{1}(\text{isCap}(y) = \cdot, z = \cdot)$
N-GRAM:	Indicator functions for character n-grams of up to length 3 present in y.

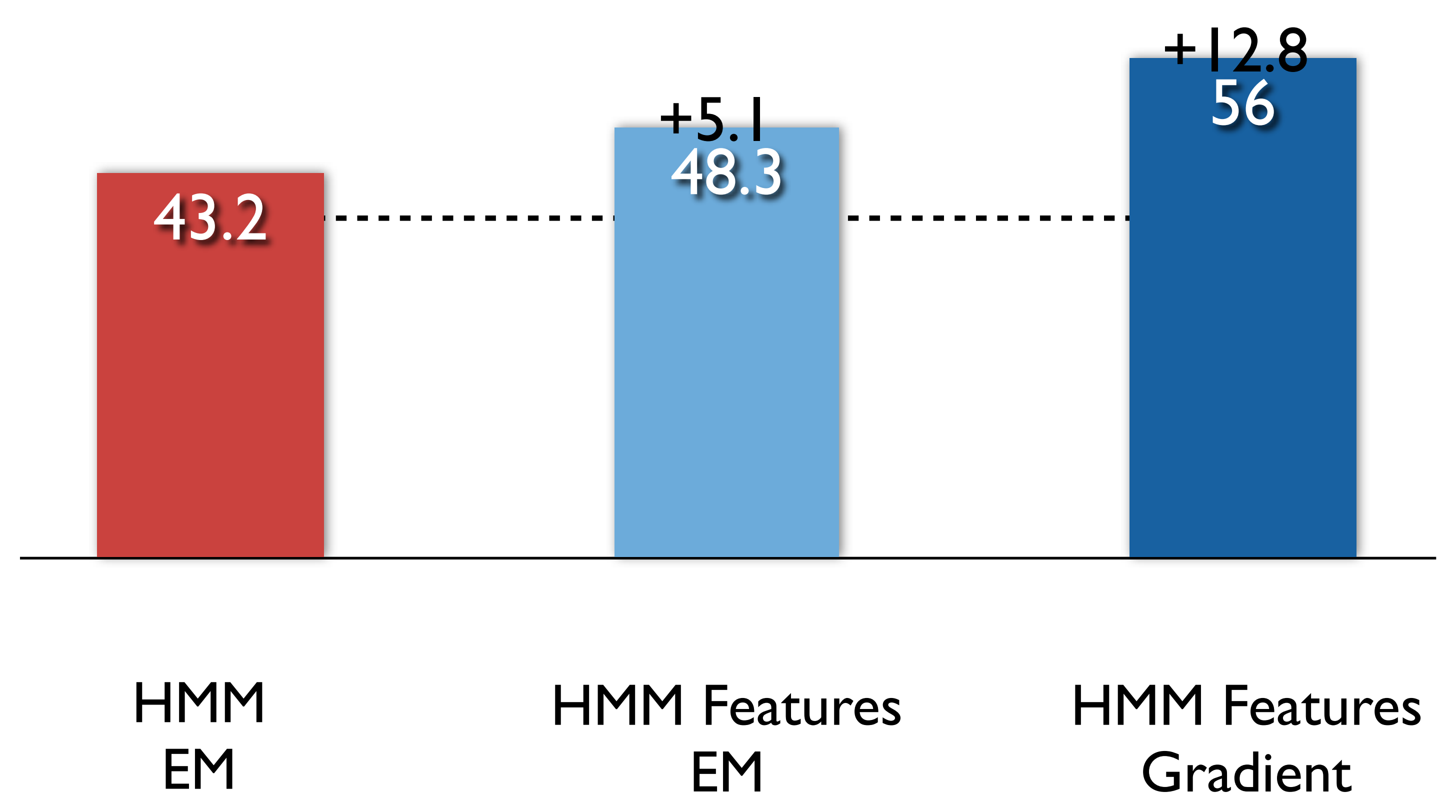
Data:

Train and test on entire WSJ

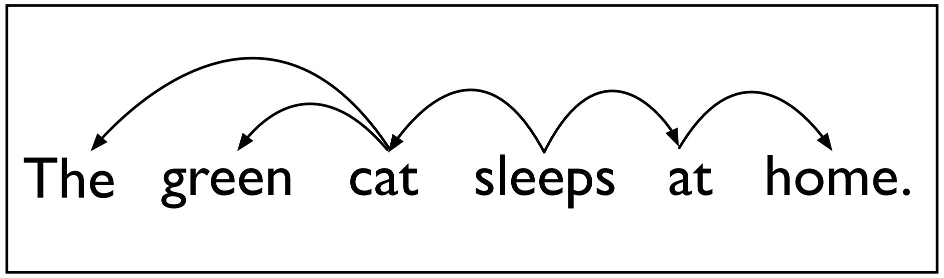
No tagging dictionary

45 POS tags

1-to-1 Accuracy



Grammar Induction Results

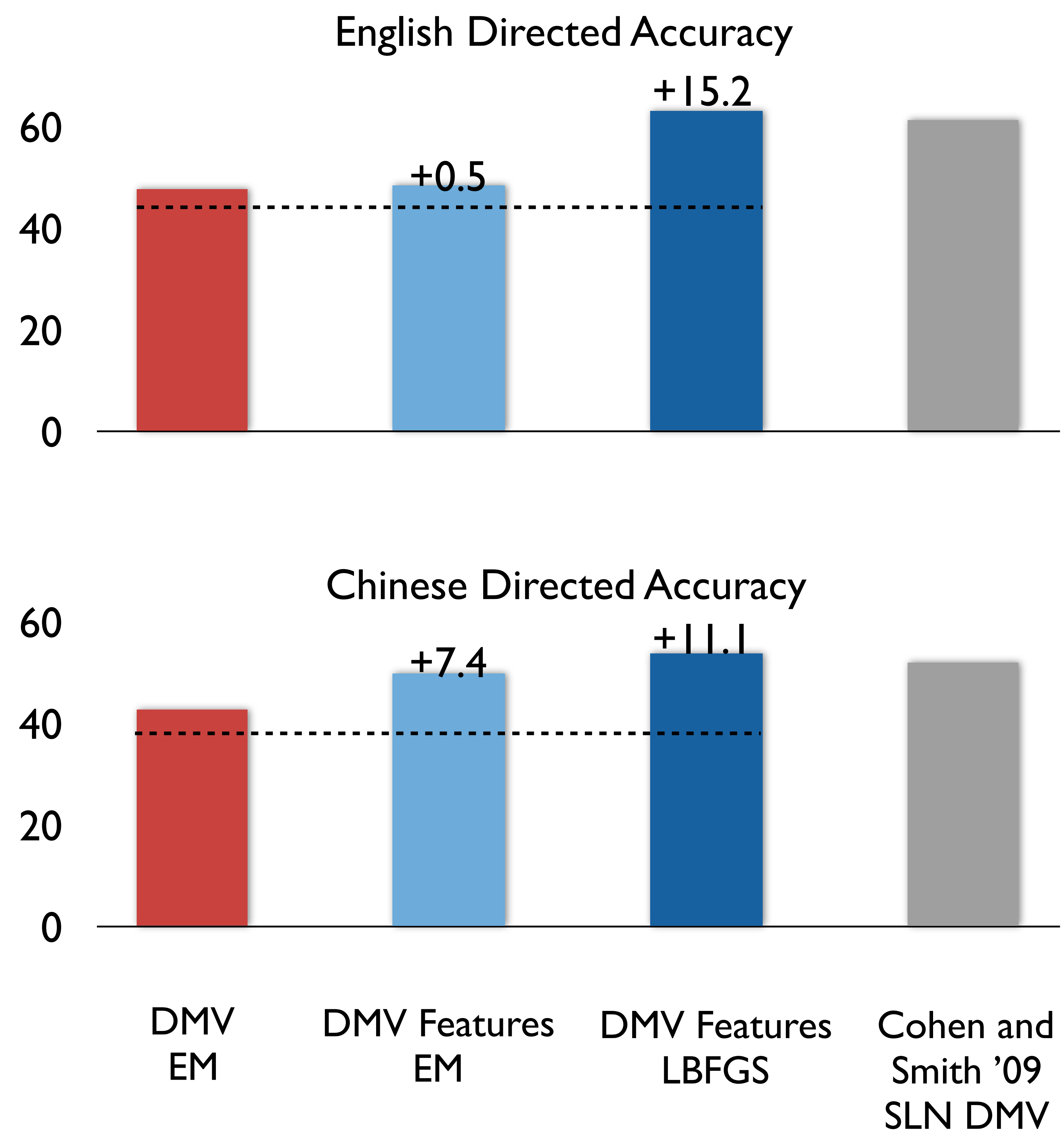


Data:

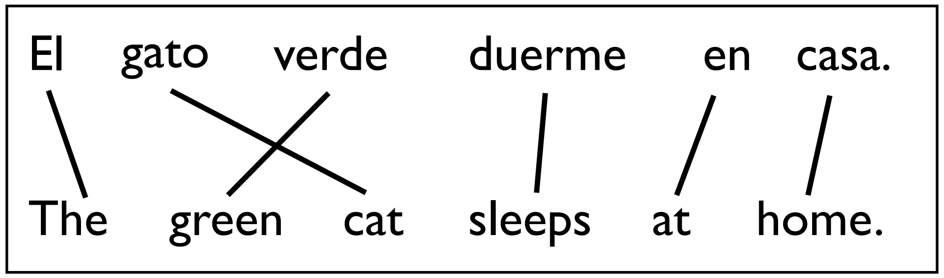
Train	WSJ10 Sec. 2-21 CTB10 Sec. 1-270
Tune	WSJ10 Sec. 22 CTB10 Sec. 400-454
Test	WSJ10 Sec. 23 CTB10 Sec. 271-300

Features:

BASIC:	$\mathbb{1}(a = \cdot, h = \cdot, \delta = \cdot)$
NOUN:	Generalize the morphological variants of nouns by using isNoun$(\cdot)$: $\mathbb{1}(a = \cdot, \text{isNoun}(h) = \cdot, \delta = \cdot)$ $\mathbb{1}(\text{isNoun}(a) = \cdot, h = \cdot, \delta = \cdot)$ $\mathbb{1}(\text{isNoun}(a) = \cdot, \text{isNoun}(h) = \cdot, \delta = \cdot)$
VERB:	Same as above, generalizing verbs instead of nouns by using isVerb$(\cdot)$
NOUN-VERB:	Same as above, generalizing with isVerbOrNoun$(\cdot)$ = isVerb$(\cdot) \vee$ isNoun$(\cdot)$
BACK-OFF:	We add versions of all other features that ignore direction or adjacency.



Word Alignment Results



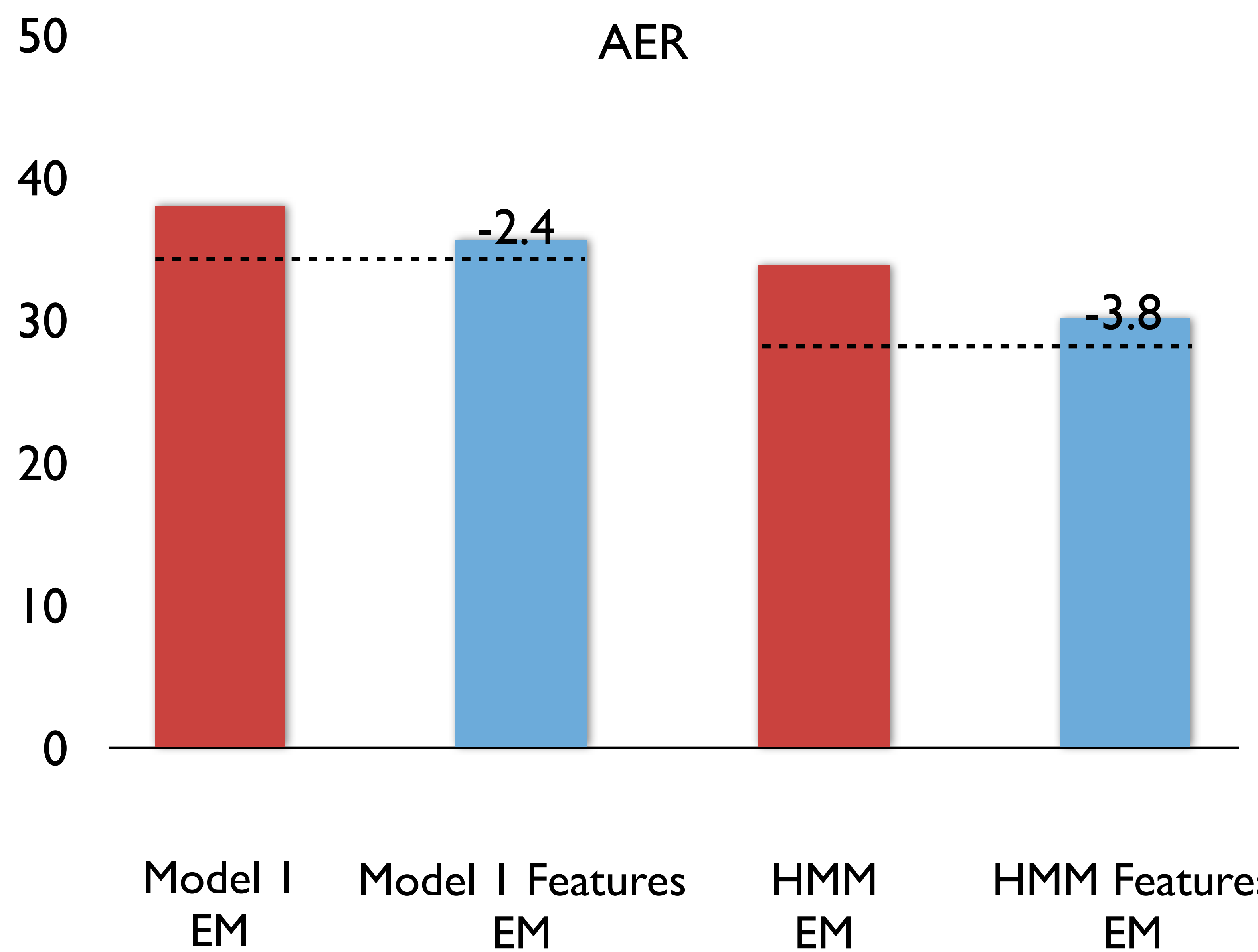
Data:

Train 10K sentences of FIBIS Ch-En newswire

Test NIST 2002 Ch-En dev set

Features:

BASIC:	$\mathbb{1}(e = \cdot, y = \cdot)$
EDIT-DISTANCE:	$\mathbb{1}(\text{dist}(y, e) = \cdot)$
DICTIONARY:	$\mathbb{1}((y, e) \in D)$ for dictionary D.
STEM:	$\mathbb{1}(\text{stem}(e) = \cdot, y = \cdot)$ for Porter stemmer.
PREFIX:	$\mathbb{1}(\text{prefix}(e) = \cdot, y = \cdot)$ for prefixes of length 4.
CHARACTER:	$\mathbb{1}(e = \cdot, \text{charAt}(y, i) = \cdot)$ for index i in the Chinese word.



Word Segmentation Results

[T h e][g r e e n][c a t]

Data:

Train and test on phonetic version of Bernstein-Ratner corpus

Features:

BASIC:	$\mathbb{1}(z = \cdot)$
LENGTH:	$\mathbb{1}(\text{length}(z) = \cdot)$
NUMBER-VOWELS:	$\mathbb{1}(\text{numVowels}(z) = \cdot)$
PHONO-CLASS-PREF:	$\mathbb{1}(\text{prefix}(\text{coarsePhonemes}(z)) = \cdot)$
PHONO-CLASS-PREF:	$\mathbb{1}(\text{suffix}(\text{coarsePhonemes}(z)) = \cdot)$