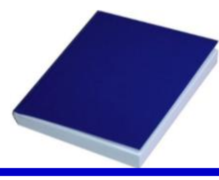
Domain Adaptation

John Blitzer and Hal Daumé III

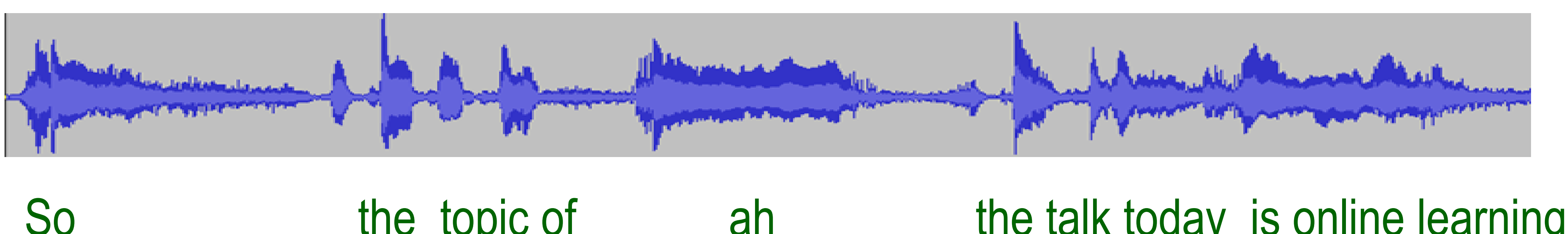
Classical "Single-domain" Learning

Predict: $x \rightarrow y$ $\quad\quad (x, y) \sim \Pr[x, y]$



Running with Scissors **Title:** Horrible book, horrible.

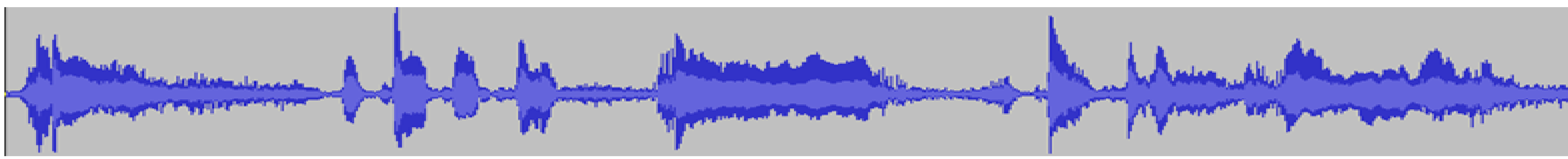
This book was horrible. I read half, suffering from a headache the entire time, and eventually i lit it on fire. 1 less copy in the world. Don't waste your money. I wish i had the time spent reading this book back. It wasted my life



Domain Adaptation

$$(x, y) \sim \Pr_S[x, y]$$

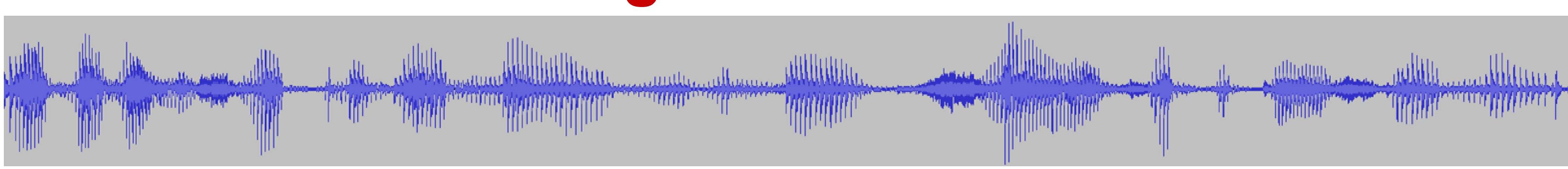
Training **Source**



So the topic of ah the talk today is online learning

$$(x, y) \sim \Pr_T[x, y]$$

Testing **Target**



Everything is happening online. Even the slides are produced on-line

Domain Adaptation

Natural Language Processing

Packed with fascinating info

A breeze to clean up

Cracking the Neanderthal Code

A Mystery Multiplies Over a Math Prize

THE WALL STREET JOURNAL

What's News—

Burning Debate

As Emission Restrictions Loom, Texas Utility Bets Big on Coal

Visual Object Recognition

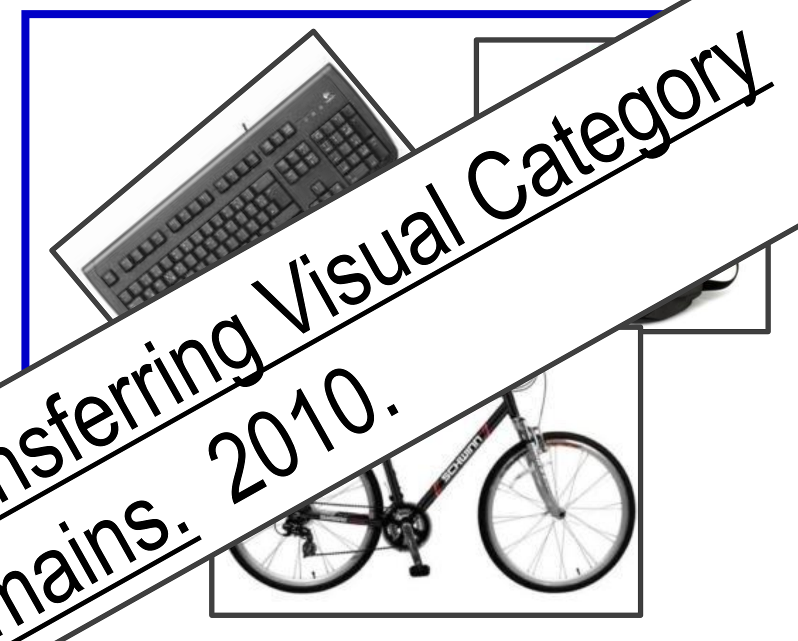
Domain Adaptation

Natural Language Processing

Packed with fascinating info

A breeze to clean up

Visual Object Recognition



K. Saenko et al. Transferring Visual Category Models to New Domains. 2010.

Tutorial Outline

1. Domain Adaptation: Common Concepts

2. Semi-supervised Adaptation
   - Learning with Shared Support
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

4. Open Questions and Uncovered Algorithms

Classical vs Adaptation Error

Classical Test Error:

$$\epsilon_{\text{test}} \le \hat{\epsilon}_{\text{train}} + \sqrt{\frac{\text{complexity}}{n}}$$

Measured on the same distribution!

Adaptation Target Error:

$$\epsilon_{\text{test}} \le ??$$

Measured on a **new** distribution!

Common Concepts in Adaptation

Covariate Shift

$$\Pr_{\color{blue}S}[y|x] \;=\; \Pr_{\color{red}T}[y|x]$$

 understands both  & 

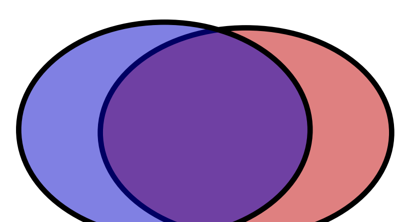
Single Good Hypothesis

$$\exists h^*, \epsilon_{\color{blue}S}(h^*),\; \epsilon_{\color{red}T}(h^*) \text{ small}$$

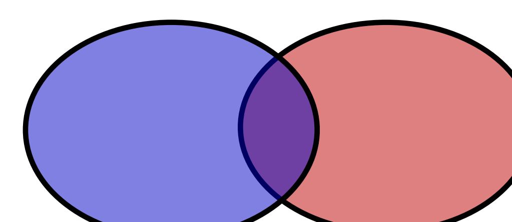
 understands both  & 

Domain Discrepancy and Error

Easy



Hard



Tutorial Outline

1. Notation and Common Concepts

2. Semi-supervised Adaptation
   - Covariate shift with Shared Support
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

4. Open Questions and Uncovered Algorithms

A bound on the adaptation error

Let h be a binary hypothesis. If $\Pr_S(y|x) = \Pr_T(y|x)$, then

$$\epsilon_T(h) \quad \leq \quad \epsilon_S(h) + \int_{\mathcal{X}} \left|\Pr_T(x) - \Pr_S(x)\right| dx$$

Minimize the total variation

Covariate Shift with Shared Support

Assumption: Target & Source Share Support

$$\forall x,\ \Pr_S[x] \neq 0 \text{ iff } \Pr_T[x] \neq 0$$

Reweight source instances to minimize discrepancy

Source Instance Reweighting

Defining Error

$$\epsilon_T(h) = \mathbb{E}_{\Pr_T[x]}\ \mathbb{E}_{\Pr[y|x]}\left[h(x) \neq y\right]$$

Using Definition of Expectation

$$= \sum_x \Pr_T[x]\, \mathbb{E}_{\Pr[y|x]}\left[h(x) \neq y\right]$$

Multiplying by 1

$$= \sum_x \frac{\Pr_S[x]}{\Pr_S[x]} \Pr_T[x] \mathbb{E}_{\Pr[y|x]}\left[h(x) \neq y\right]$$

per-instance weights w

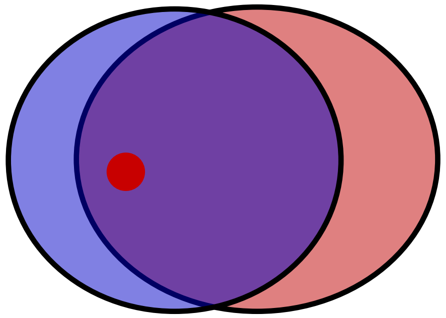
Rearranging

$$\epsilon_T(h) = \epsilon_S(h, w) = \mathbb{E}_{\Pr_S[x]}\ \frac{\Pr_t[x]}{\Pr_s[x]}\ \mathbb{E}_{\Pr[y|x]}\left[h(x) \neq y\right]$$

Sample Selection Bias

Another Way to View

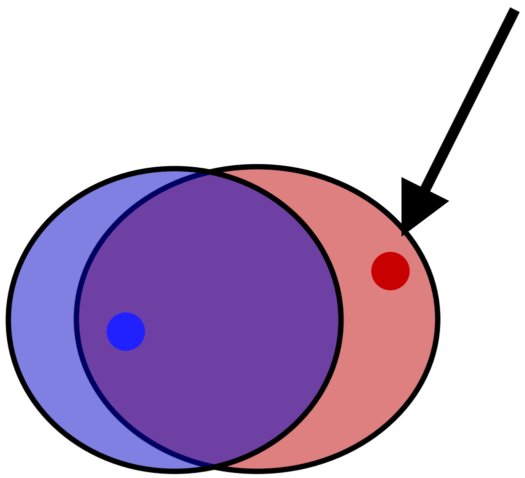
1) Draw from the target $\Pr_T[x]$



Sample Selection Bias

Redefine the source distribution

1) Draw from the target $\Pr_T[x]$
2) Select into the source with $\Pr[\sigma = 1|x]$



$$\Pr_S[x] = \frac{\Pr_T[x]\Pr[\sigma = 1|x]}{\Pr[\sigma = 1]}$$

Rewriting Source Risk

$$\Pr_S[x] = \frac{\Pr_T[x]\Pr[\sigma = 1|x]}{\Pr[\sigma = 1]}$$

Rearranging

$$\frac{\Pr_T[x]}{\Pr_S[x]} = \frac{\Pr[\sigma = 1]}{\Pr[\sigma = 1|x]}$$

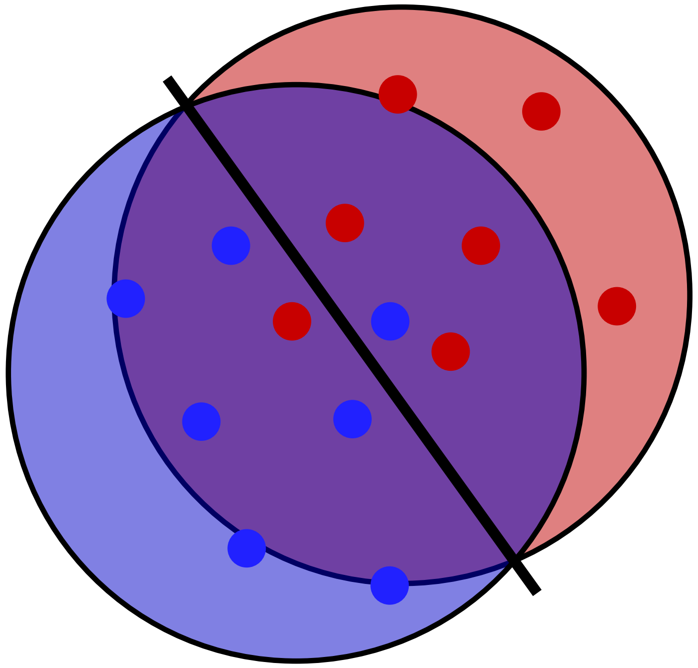
per-instance weights w

$\Pr[\sigma = 1]$ not dependent on x

$$\epsilon_S(h, w) \propto \mathbb{E}_{\Pr_S[x]} \frac{1}{\Pr[\sigma = 1|x]} \mathbb{E}_{\Pr[y|x]} [h(x) \neq y]$$

Logistic Model of Source Selection

$$\Pr[\sigma = 1 | x] = \frac{1}{1+\exp(\theta^\top x + b)}$$



Training Data

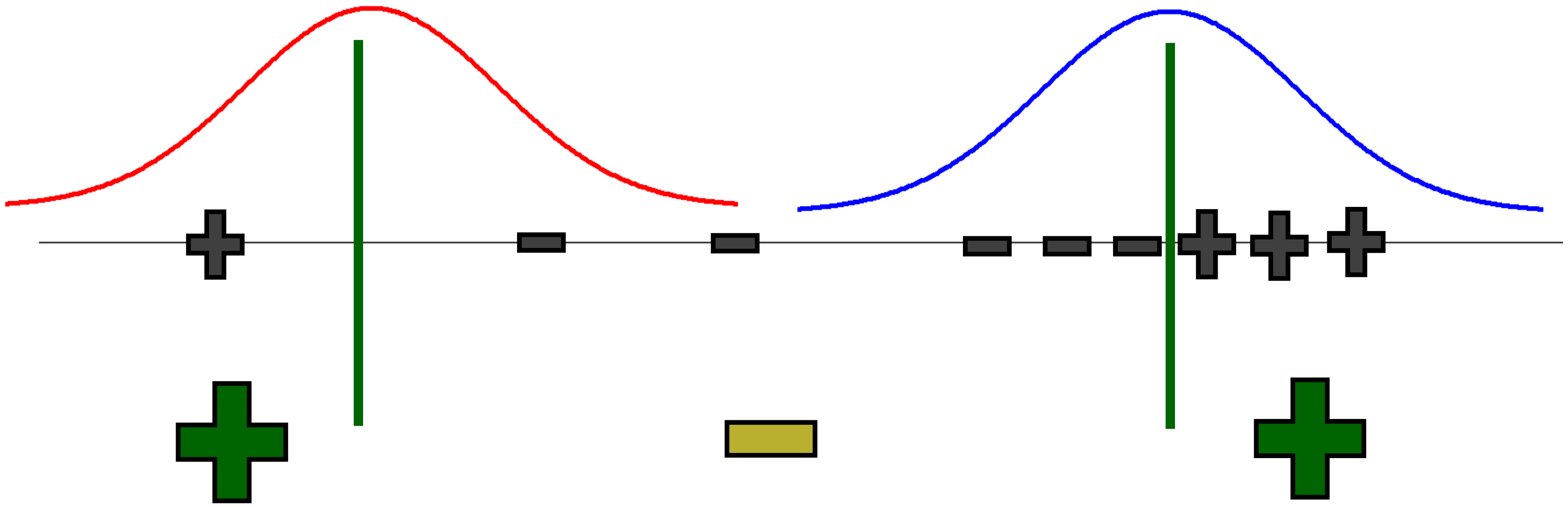
Source instances, $\sigma = 1$

Target unlabeled instances, $\sigma = 0$

Selection Bias Correction Algorithm

Input:

Labeled source data

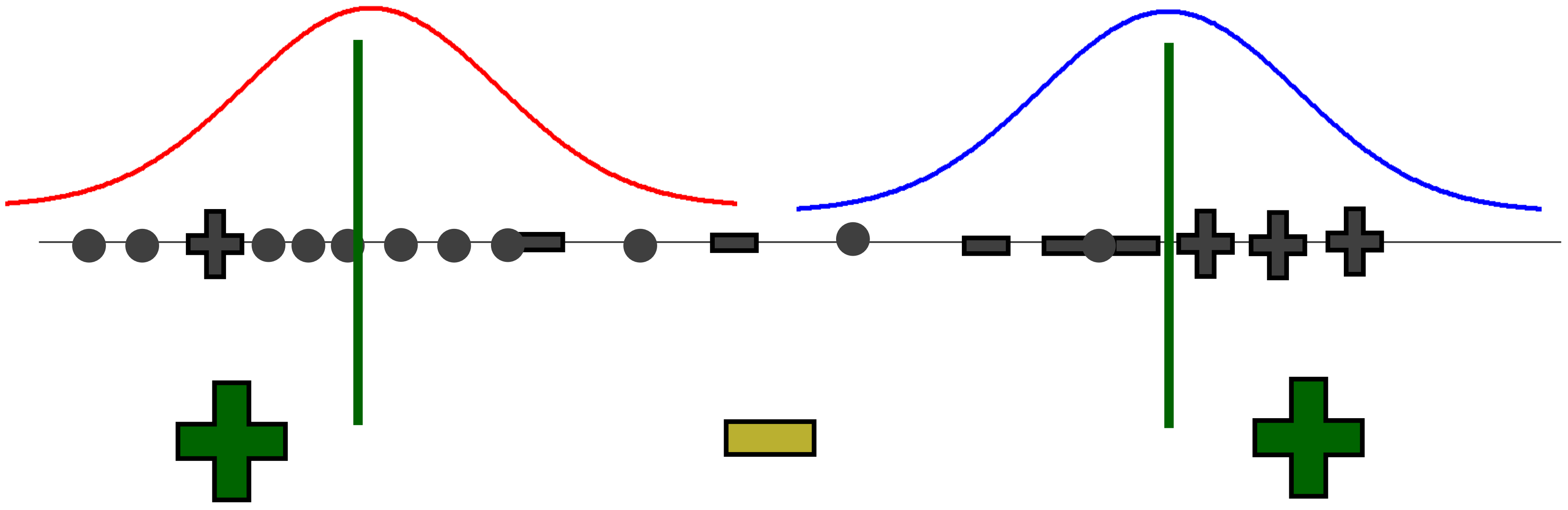


Selection Bias Correction Algorithm

Input:

Labeled source data

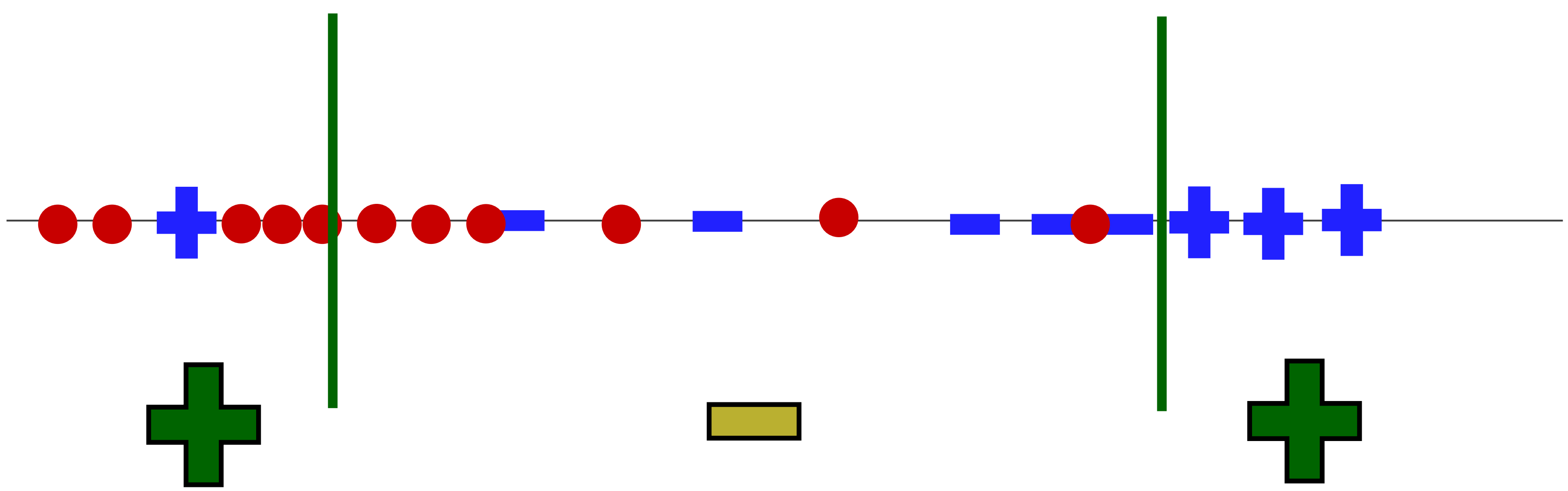
Unlabeled target data



Selection Bias Correction Algorithm

Input: Labeled source and unlabeled target data

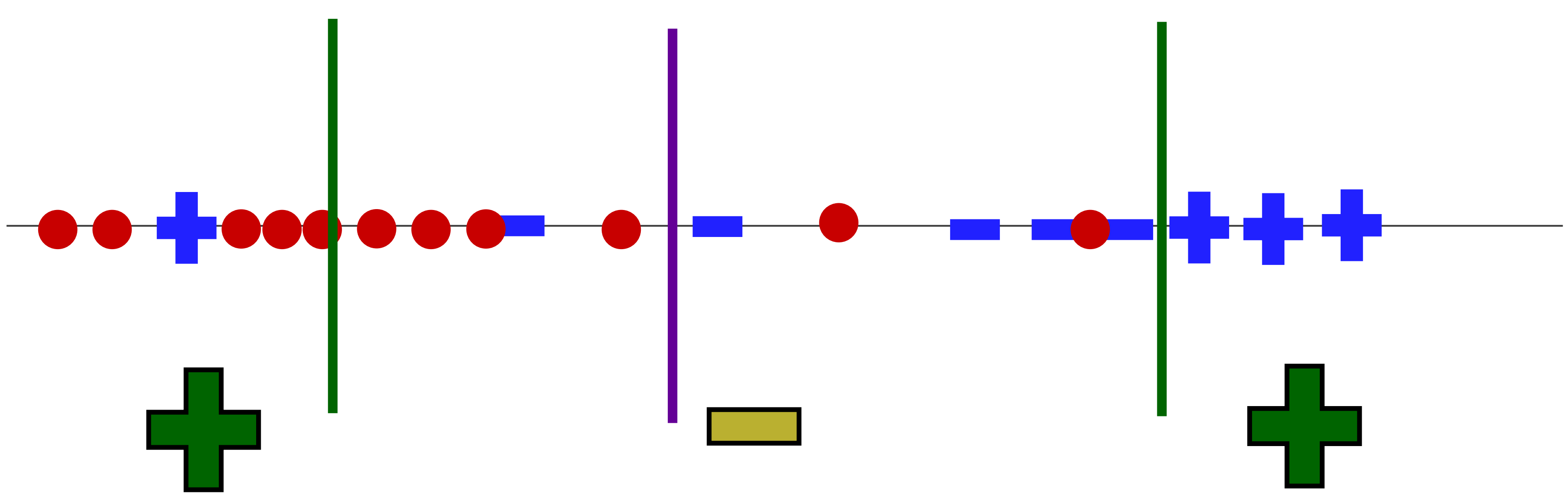
1) Label source instances as $\sigma = 1$, target as $\sigma = 0$



Selection Bias Correction Algorithm

Input: Labeled source and unlabeled target data

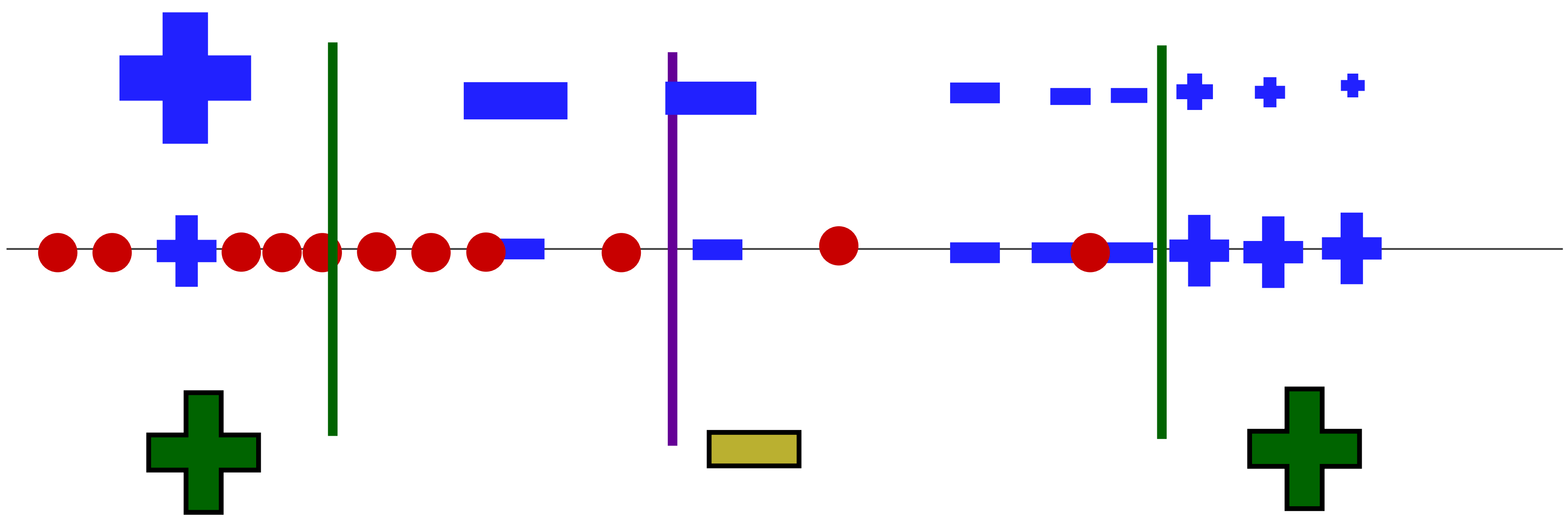
1) Label source instances as $\sigma = 1$, target as $\sigma = 0$
2) Train predictor $\Pr[\sigma = 1 | x] = \frac{1}{1+\exp(\theta^\top x + b)}$



Selection Bias Correction Algorithm

Input: Labeled source and unlabeled target data

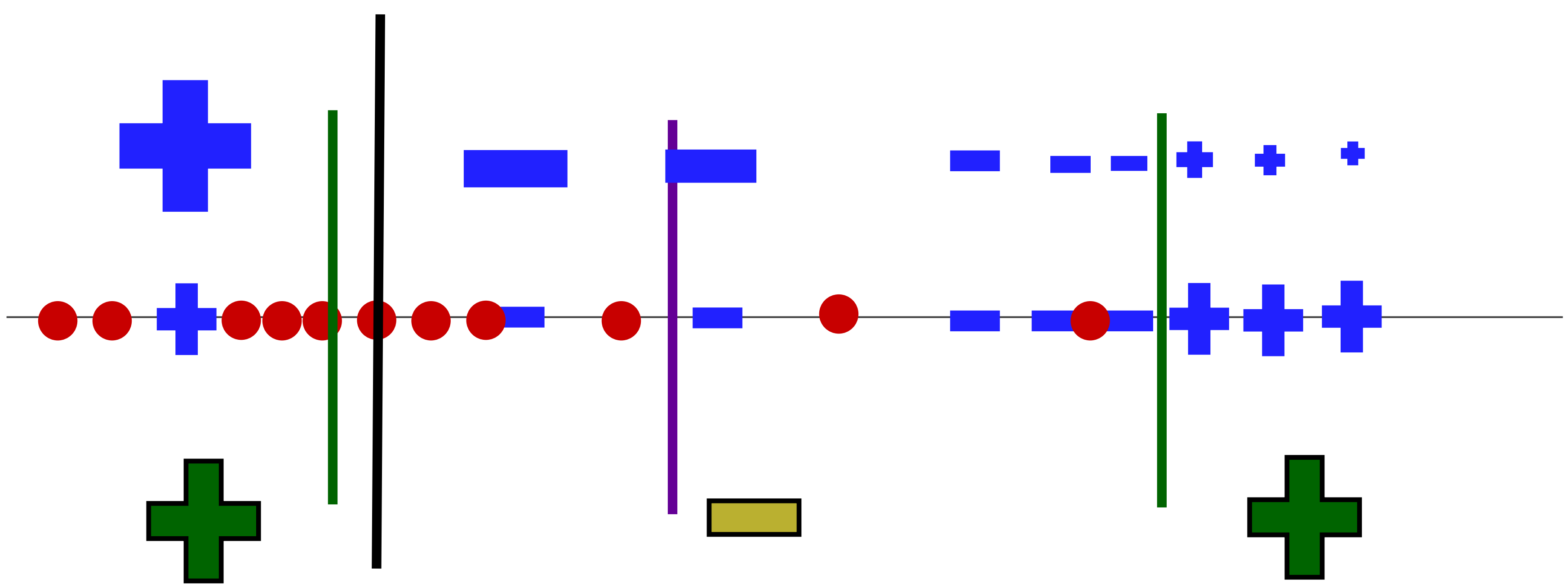
1) Label source instances as $\sigma = 1$, target as $\sigma = 0$
2) Train predictor $\Pr[\sigma = 1 | x] = \frac{1}{1+\exp(\theta^\top x + b)}$
3) Reweight source instances



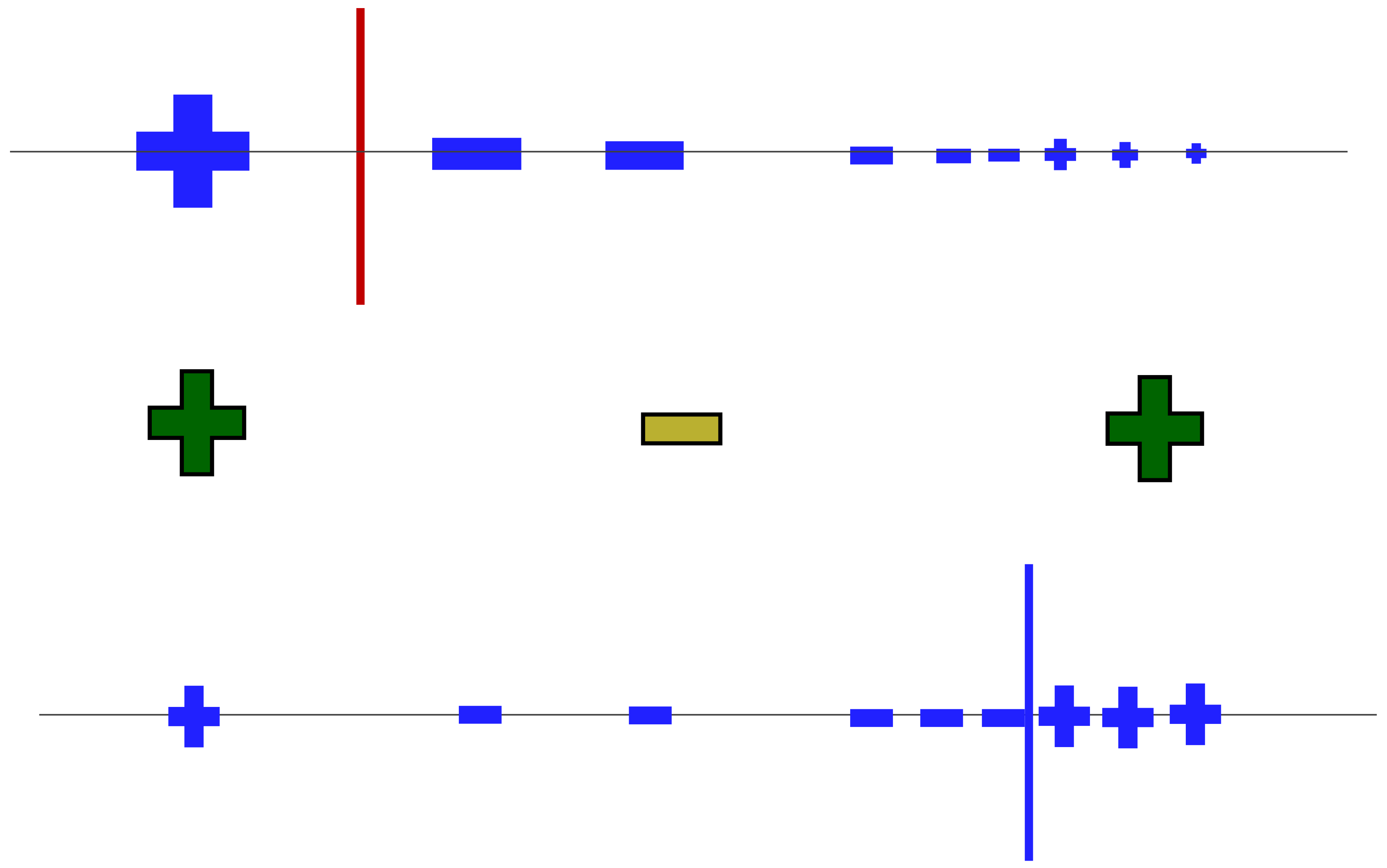
Selection Bias Correction Algorithm

Input: Labeled source and unlabeled target data

1) Label source instances as $\sigma = 1$, target as $\sigma = 0$
2) Train predictor $\Pr[\sigma = 1 | x] = \frac{1}{1+\exp(\theta^\top x + b)}$
3) Reweight source instances
4) Train target predictor



How Bias gets Corrected



Rates for Re-weighted Learning

$\hat{\epsilon}_s^n(h, w)$: weighted source test error on sample of size n

With probability $1 - \delta$, for every h

$$|\hat{\epsilon}_S^n(h, w) - \epsilon_T(h)| \leq \sqrt{\frac{O\left(\frac{1}{\delta}\right) + O\left(\max_{x \in \mathcal{X}} w(x)^2\right)}{n}}$$

Adapted from Gretton et al.

Sample Selection Bias Summary

Two Key Assumptions

1) Covariate shift: $\Pr_S[y|x] \;=\; \Pr_T[y|x]$

2) Shared support: $\forall x,\ \Pr_S[x] \neq 0$ iff $\Pr_T[x] \neq 0$

Advantage

$$\hat{\epsilon}^n_S(h, w) \xrightarrow[\infty]{n} \epsilon_T(h)$$

Optimal target predictor
without labeled target data

Sample Selection Bias Summary

Two Key Assumptions

1) Covariate shift: $\Pr_S[y|x] = \Pr_T[y|x]$

2) Shared support: $\forall x,\ \Pr_S[x] \neq 0$ iff $\Pr_T[x] \neq 0$

Advantage

$$\hat{\epsilon}^n_S(h,w) \xrightarrow[\infty]{n} \epsilon_T(h)$$

Disadvantage

Convergence to $\epsilon_T(h)$ depends on $\max_x \frac{\Pr_T(x)}{\Pr_S(x)}$

Sample Selection Bias References

http://adaptationtutorial.blitzer.com/references/

[1] J. Heckman. Sample Selection Bias as a Specification Error. 1979.

[2] A. Gretton et al. Covariate Shift by Kernel Mean Matching. 2008.

[3] C. Cortes et al. Sample Selection Bias Correction Theory. 2008

[4] S. Bickel et al. Discriminative Learning Under Covariate Shift. 2009.

Tutorial Outline

1. Notation and Common Concepts

2. Semi-supervised Adaptation
   - Covariate shift
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

4. Open Questions and Uncovered Algorithms

Unshared Support in the Real World

amazon.com

Running with Scissors

Title: Horrible book, horrible.

This book was horrible. I read half, suffering from a headache the entire time, and eventually i lit it on fire. 1 less copy in the world. Don't waste your money. I wish i had the time spent reading this book back. It wasted my life

Avante Deep Fryer; Black

Title: lid does not work well...

I love the way the Tefal deep fryer cooks, however, I am returning my second one due to a defective lid closure. The lid may close initially, but after a few uses it no longer stays closed. I won't be buying this one again.

Unshared Support in the Real World

amazon.com

Running with Scissors

Title: Horrible book, horrible.

This book was horrible. I read half

time, and eventually i lit it on fire. 1 less copy in the world. Don't waste your money. I wish i had the time spent reading this book back. It wasted my life

Avante Deep Fryer; Black

Title: lid **does not work** well...

I love the way the Tefal deep fryer

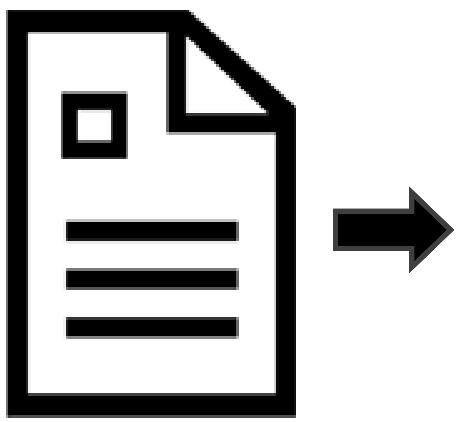
second one due to a **defective** lid closure. The lid may close initially, but after a few uses it no longer stays closed. I won't be buying this one again.

Error increase: 13% → 26%

Linear Regression for Rating Prediction

$$h(x) = \mathrm{sgn}\left(\theta^\top x\right)$$

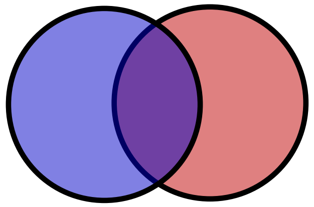
$h(x) \in \{$  $\}$



	excellent				great				fascinating
x	**3**	0	$\cdots$	0	**1**	0	$\cdots$	0	1
θ	0.5	1	$\cdots$	-1.1	2	-0.3	$\cdots$	0.1	1.5

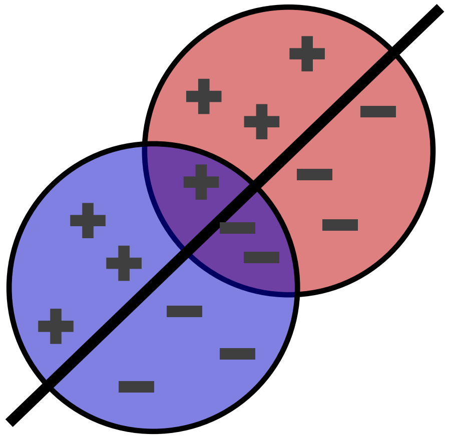
Coupled Subspaces

No Shared Support



Single Good Linear Hypothesis

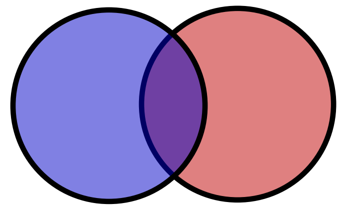
$$\exists \theta^*, \quad \epsilon_S(\theta^*) + \epsilon_T(\theta^*) \quad \text{small}$$



Stronger than $\quad \Pr_S[y|x] \; = \; \Pr_T[y|x]$

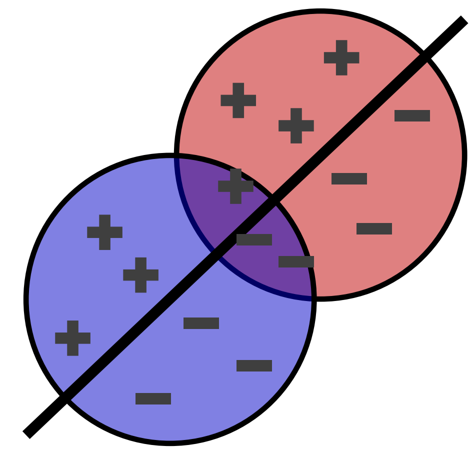
Coupled Subspaces

No Shared Support



Single Good Linear Hypothesis

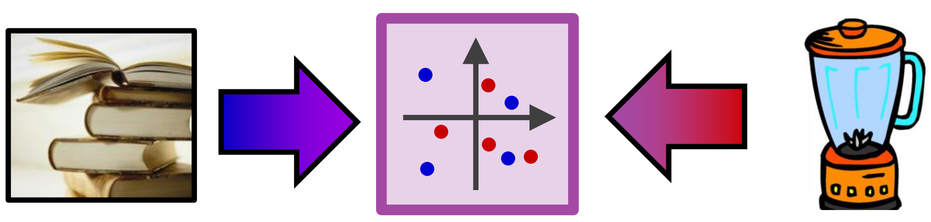
$$\exists \theta^*, \quad \epsilon_S(\theta^*) + \epsilon_T(\theta^*) \quad \text{small}$$



Coupled Representation Learning

Px couples domains

Bound target error $\epsilon_{P,T}(\theta)$



Single Good Linear Hypothesis?

$$\exists \theta^*, \quad \epsilon_S(\theta^*) + \epsilon_T(\theta^*) \quad \text{small}$$

Adaptation Squared Error

Source \ Target	Books	Kitchen
Books	1.35	
Kitchen		1.19
Both		

Single Good Linear Hypothesis?

$$\exists \theta^*, \quad \epsilon_S(\theta^*) + \epsilon_T(\theta^*) \quad \text{small}$$

Adaptation Squared Error

Source \ Target	Books	Kitchen
Books	1.35	
Kitchen		1.19
Both	1.38	1.23

Single Good Linear Hypothesis?

$$\exists \theta^*, \quad \epsilon_S(\theta^*) + \epsilon_T(\theta^*) \quad \text{small}$$

Adaptation Squared Error

Target / Source	Books	Kitchen
Books	**1.35**	1.68
Kitchen	1.80	**1.19**
Both	**1.38**	**1.23**

A bound on the adaptation error

Let h be a binary hypothesis. If $\Pr_S(Y|x) = \Pr_T(Y|x)$, then

$$\epsilon_T(h) \quad \leq \quad \epsilon_S(h) + \int_{\mathcal{X}} \left|\Pr_T(x) - \Pr_S(x)\right| dx$$

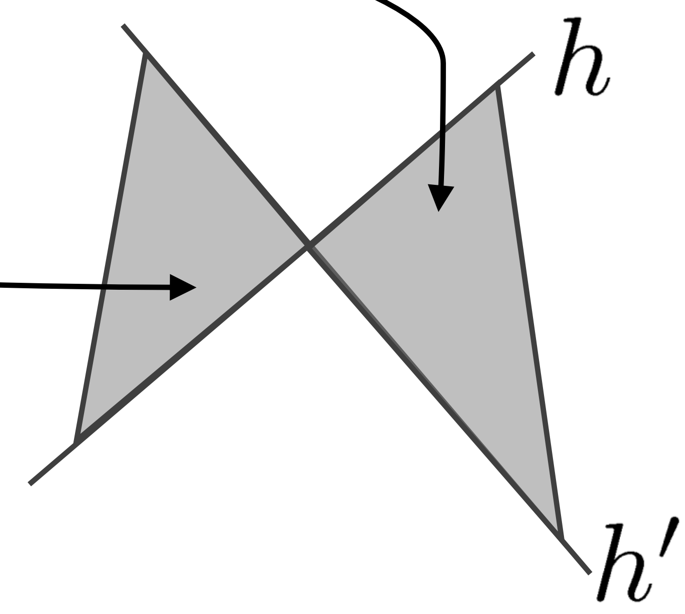
What if a single good hypothesis exists?

A better discrepancy than total variation?

A generalized discrepancy distance

Measure how hypotheses make mistakes

Linear, binary discrepancy region

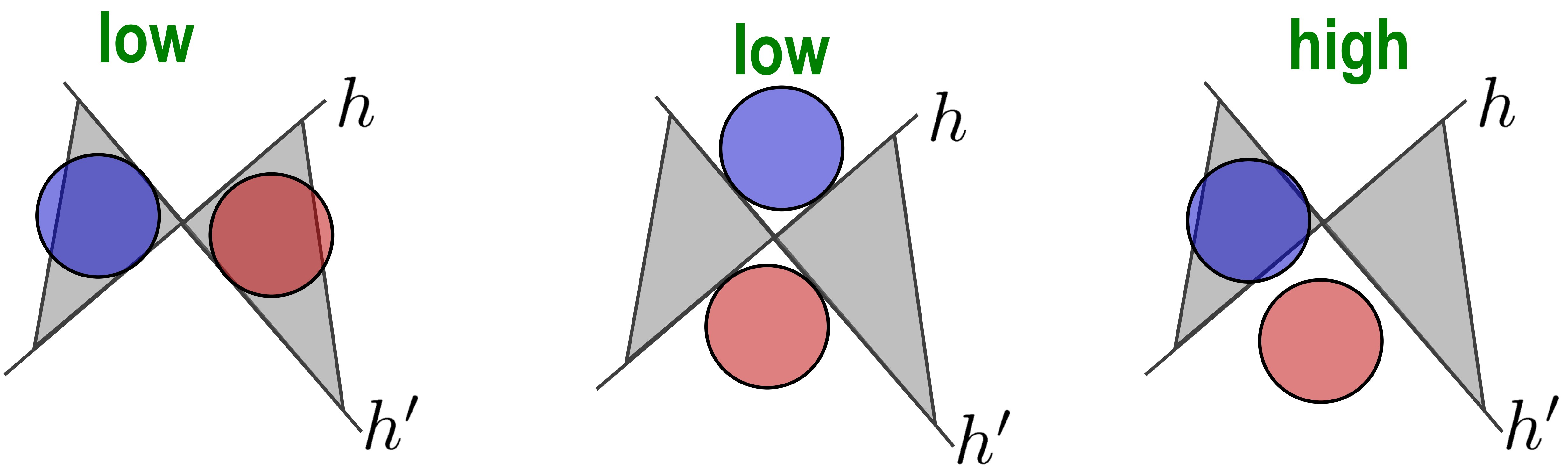


A generalized discrepancy distance

Measure how hypotheses make mistakes

$$\mathrm{disc}_H(Q,P) =$$

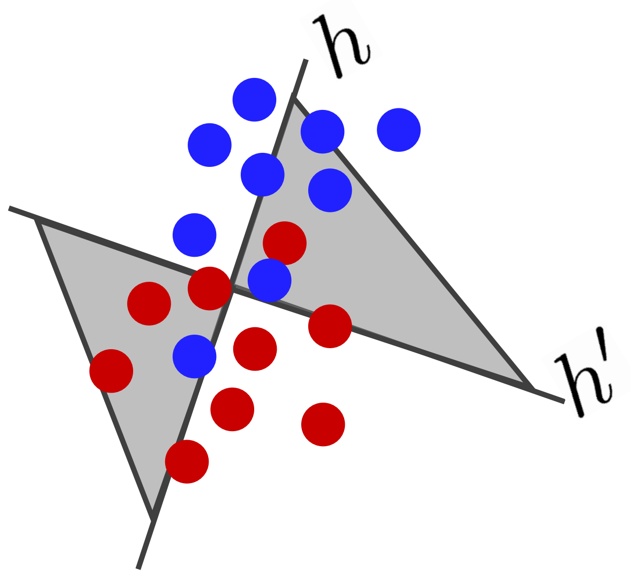
$$\max_{h,h' \in H} \left| E_Q[h(x) \neq h'(x)] - E_P[h(x) \neq h'(x)] \right|$$



Discrepancy vs. Total Variation

Discrepancy

Computable from finite samples.



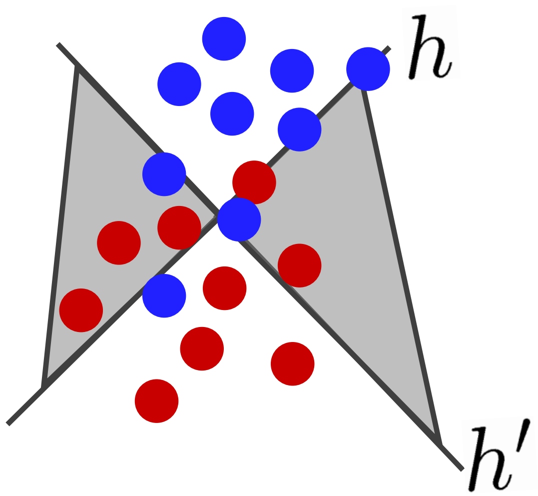
Total Variation

Not computable in general

Discrepancy vs. Total Variation

Discrepancy

Computable from finite samples.



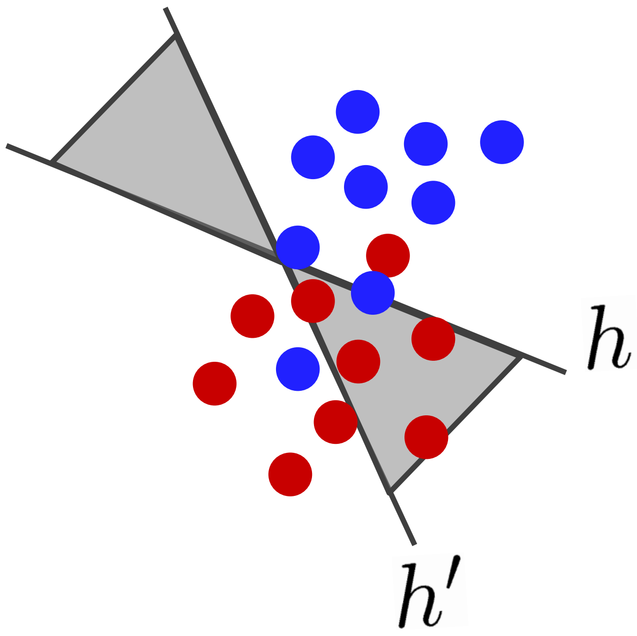
Total Variation

Not computable in general

Discrepancy vs. Total Variation

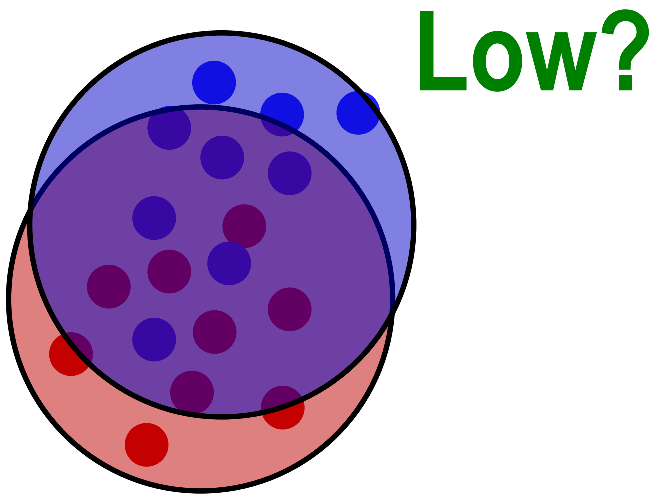
Discrepancy

Computable from finite samples.



Total Variation

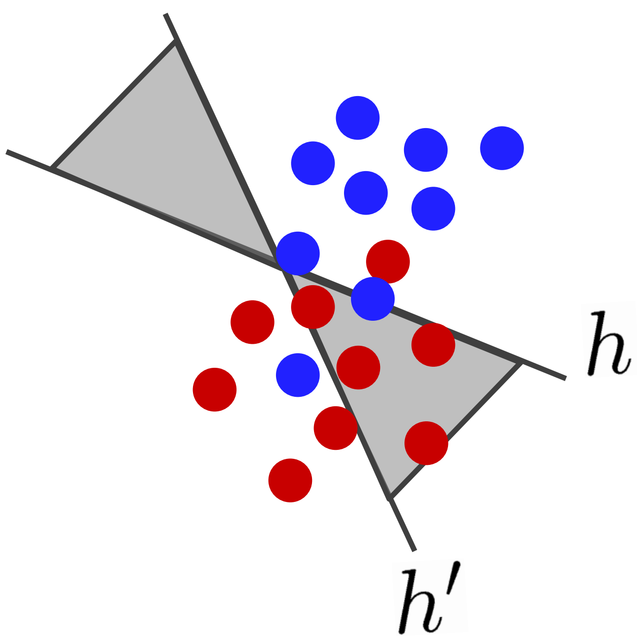
Not computable in general



Discrepancy vs. Total Variation

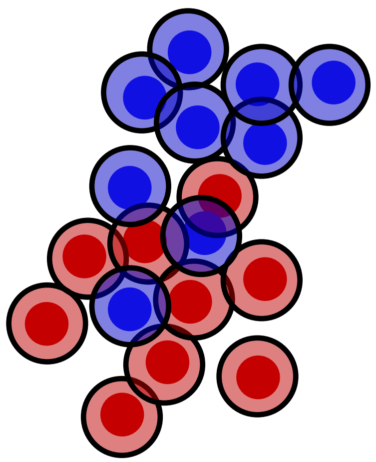
Discrepancy

Computable from finite samples.



Total Variation

Not computable in general

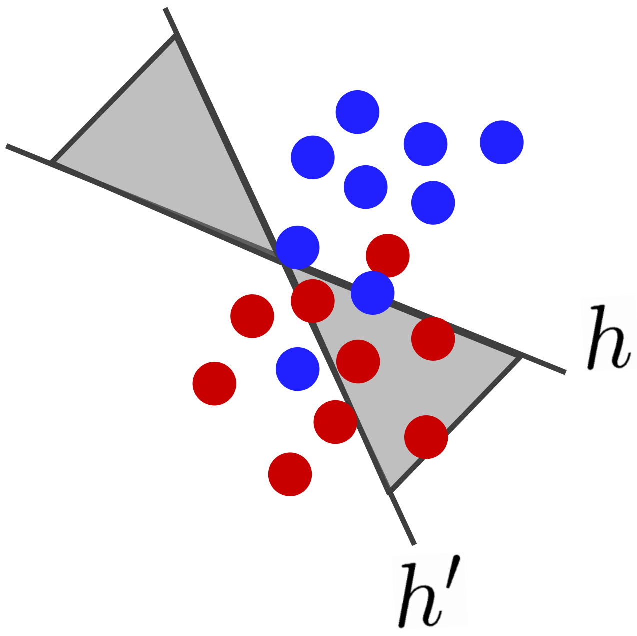


High?

Discrepancy vs. Total Variation

Discrepancy

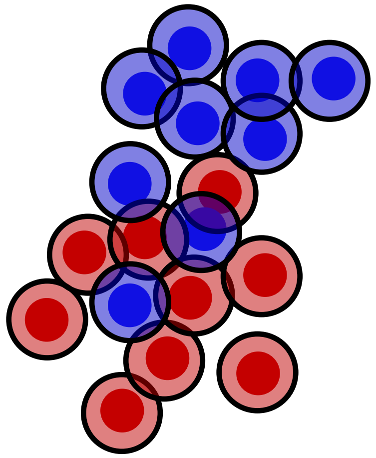
Computable from finite samples.



Related to hypothesis class

Total Variation

Not computable in general



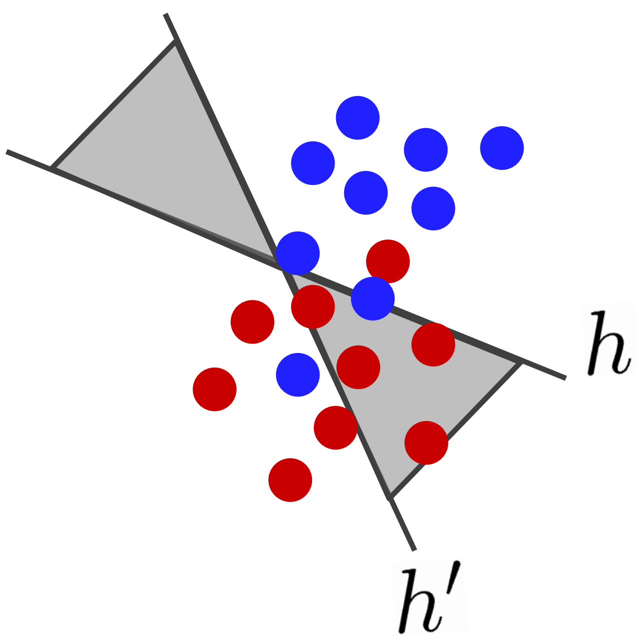
High?

Unrelated to hypothesis class

Discrepancy vs. Total Variation

Discrepancy

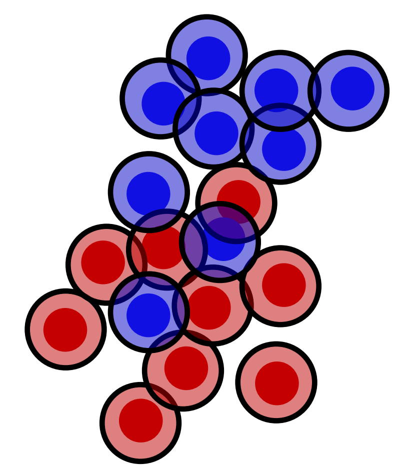
Computable from finite samples.



Related to hypothesis class

Total Variation

Not computable in general



High?

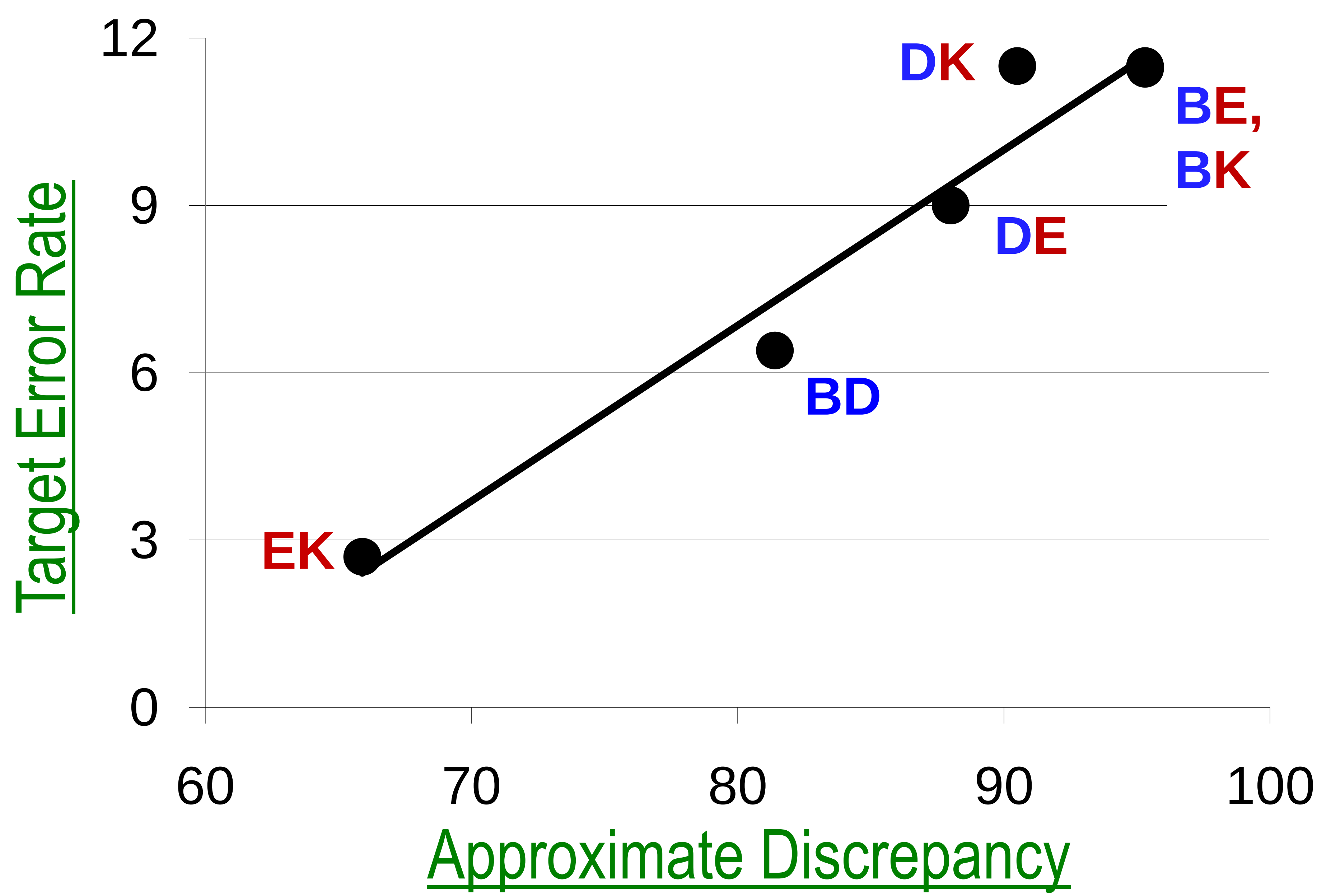
Unrelated to hypothesis class

Bickel covariate shift algorithm heuristically minimizes both measures

Is Discrepancy Intuitively Correct?

4 domains: **B**ooks, **D**VDs, **E**lectronics, **K**itchen   B&D, E&K   Shared Vocabulary

B&D: *fascinating, boring*   **E&K**: *super easy, bad quality*



A new adaptation bound

S, T: Source and target $\mathcal{H}$: Hypothesis class n: Sample size

$\hat{S}$: Labeled S sample $\hat{T}$: Unlabeled T sample

$\mathcal{R}_{\hat{S}}(\mathcal{H})$, $\mathcal{R}_{\hat{T}}(\mathcal{H})$: Rademacher complexities

With probability $1-\delta$, for h the ERM of $\hat{S}$:

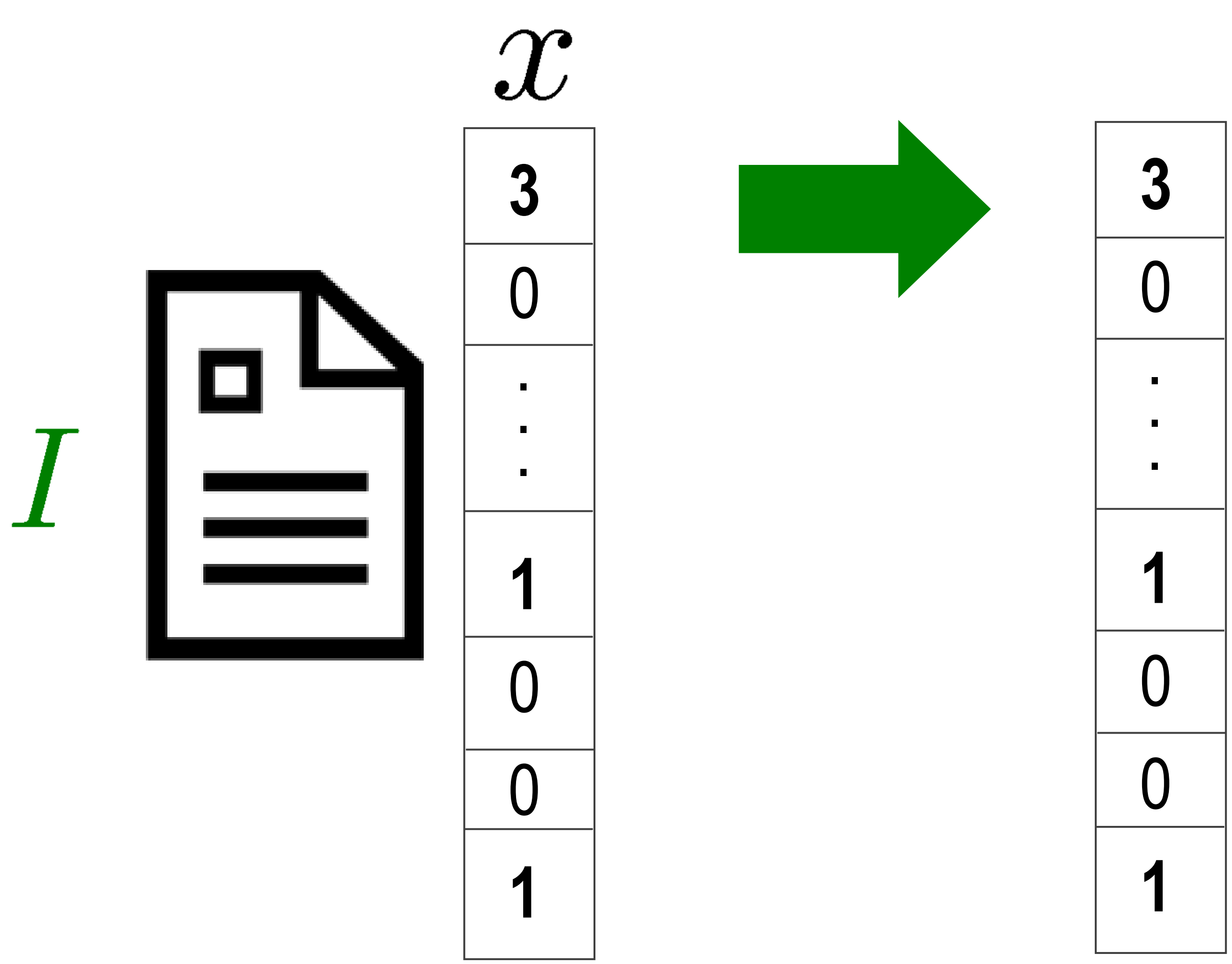
$$\epsilon_T(h) - \epsilon_T(h^*) \leq \epsilon_{\hat{S}}(h, h^*) + O\left(\mathcal{R}_{\hat{S}}(\mathcal{H}) + \mathcal{R}_{\hat{T}}(\mathcal{H})\right) + O\left(\sqrt{\frac{\log \frac{1}{\delta}}{n}}\right) + \mathrm{disc}_{\mathcal{H}}(\hat{S}, \hat{T})$$

Representations and the Bound

Linear Hypothesis Class: $h(x) = \mathrm{sgn}\left(\theta^\top x\right)$

$P = I$

Hypothesis classes from projections P : $\theta^\top P x$

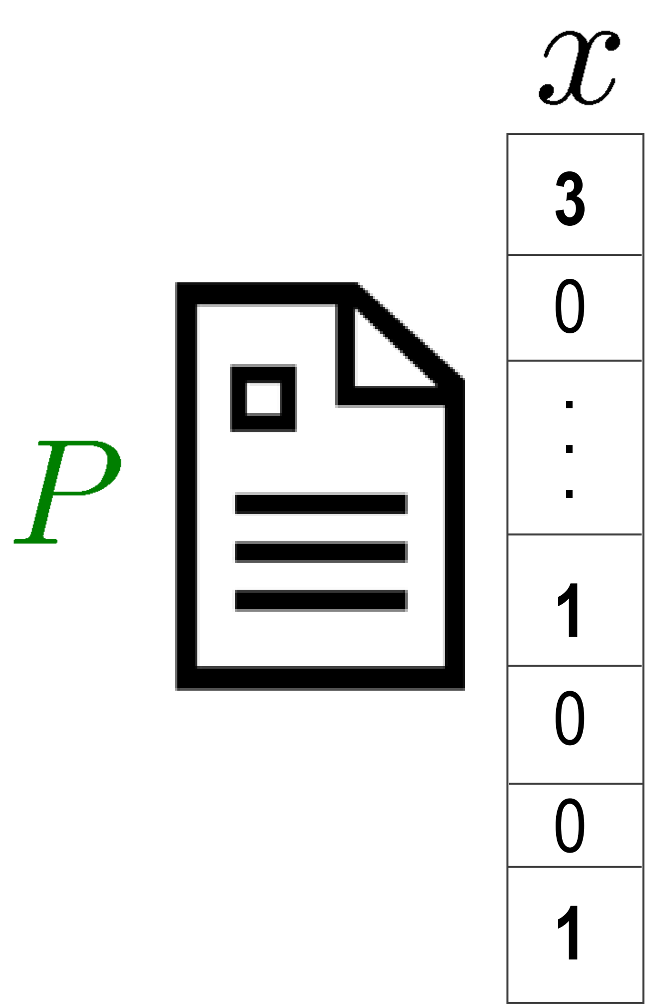


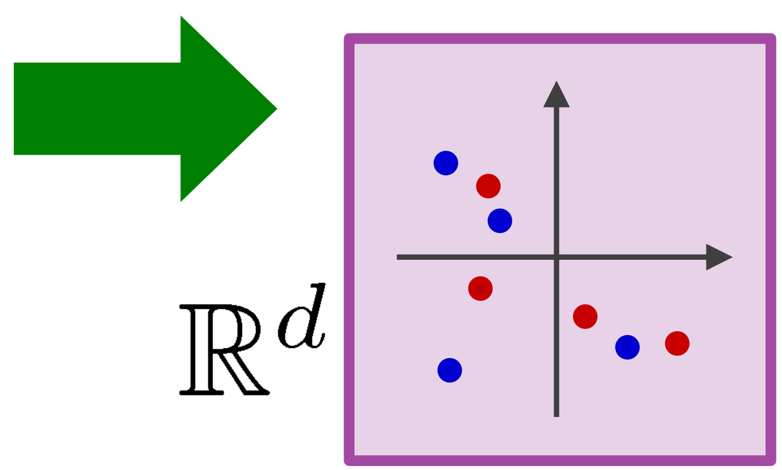
Representations and the Bound

Linear Hypothesis Class: $h(x) = \operatorname{sgn}\left(\theta^\top x\right)$

P

Hypothesis classes from projections P: $\theta^\top P x$

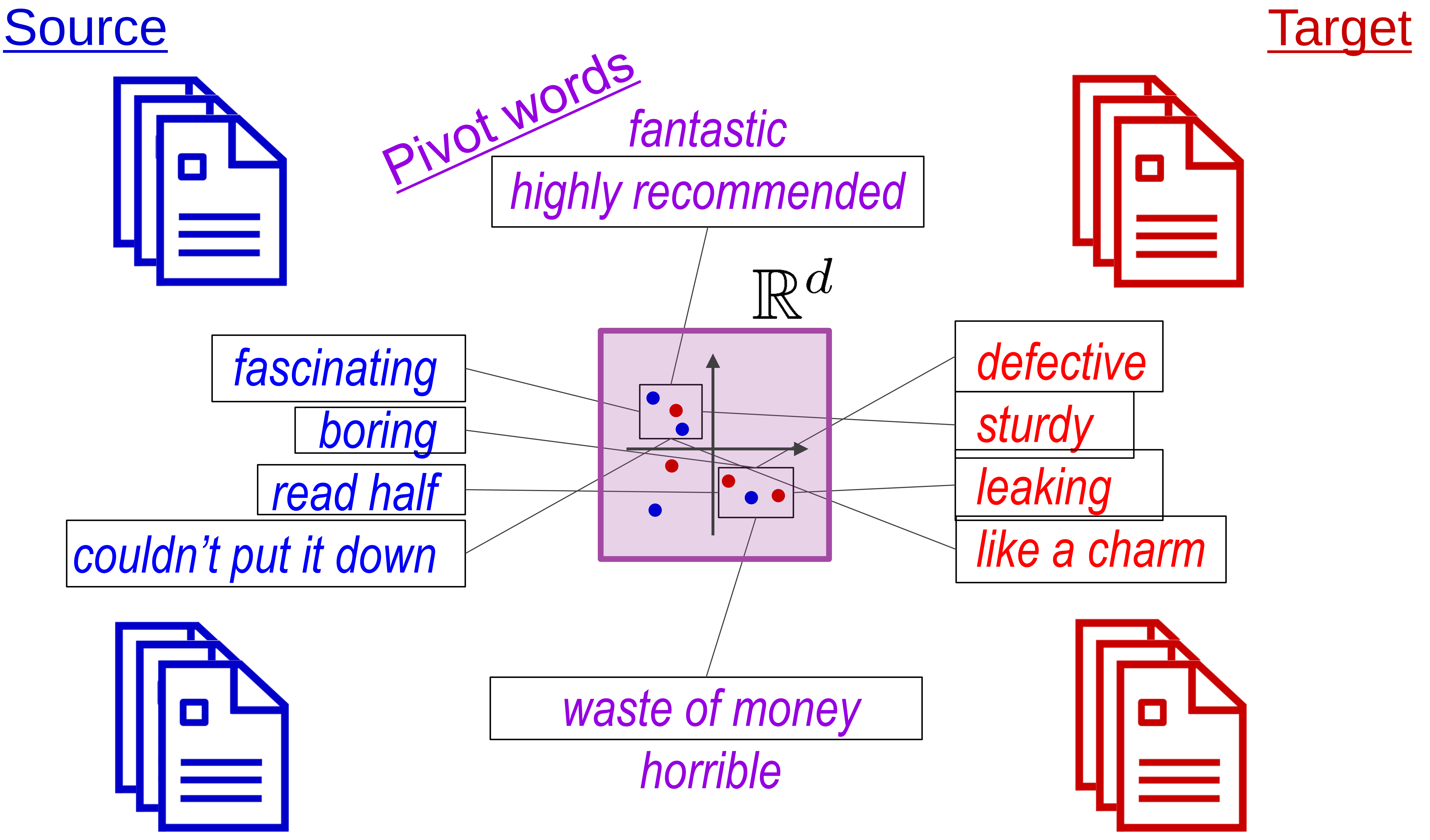




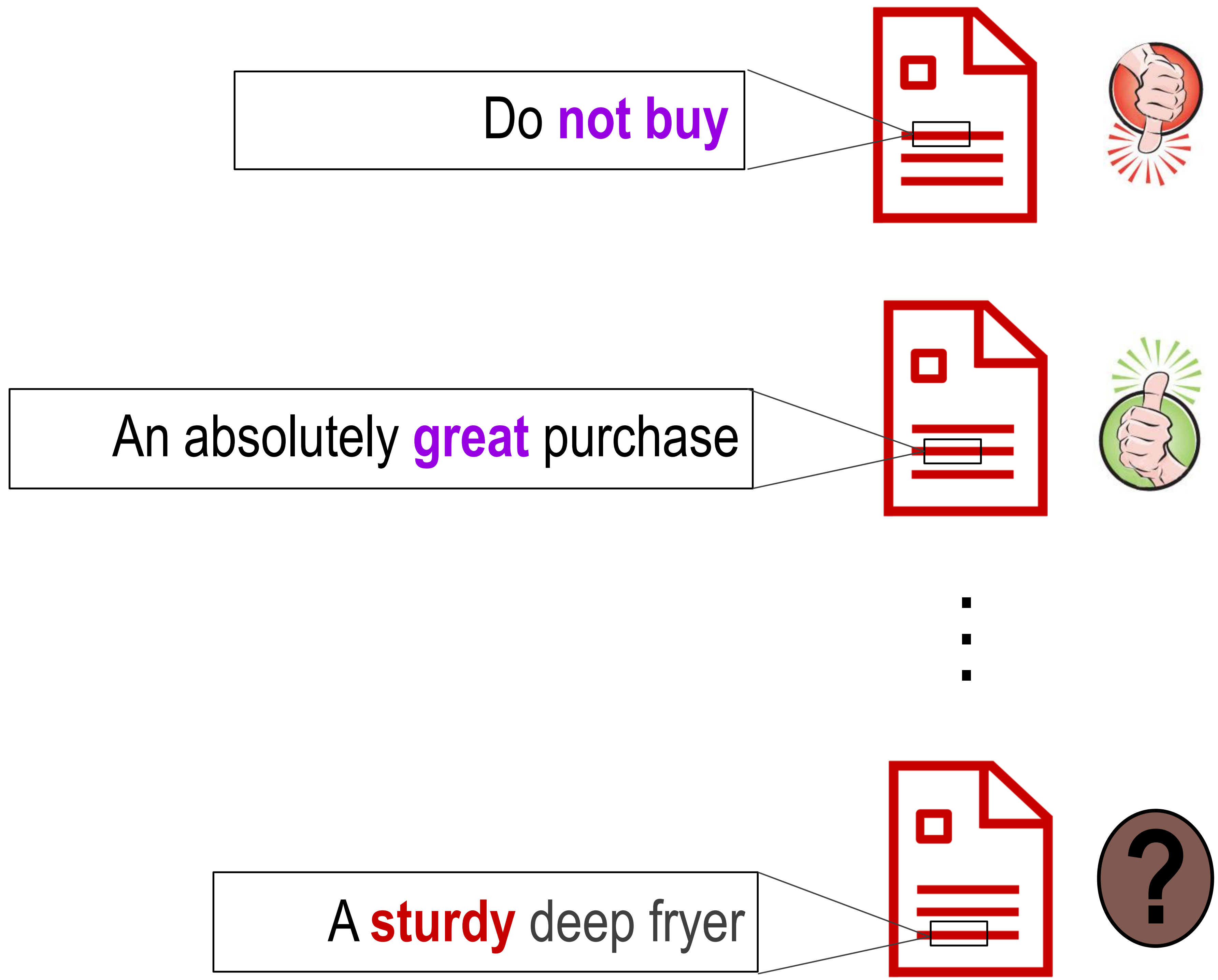
Goals for P

1) Minimize divergence
2) $\epsilon_{P,T}(\theta^*) - \epsilon_{I,T}(\theta^*)$ small

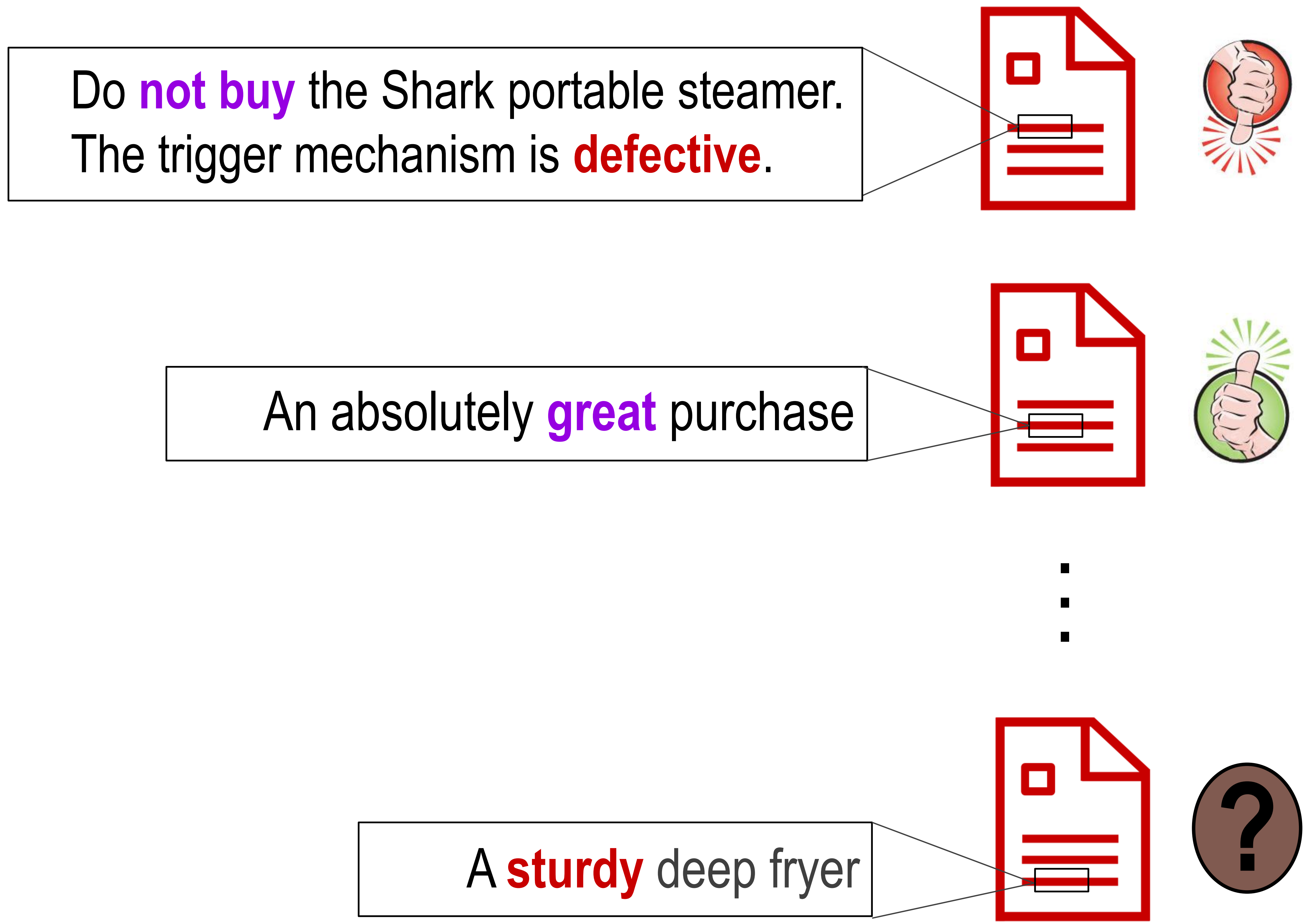
Learning Representations: Pivots



Predicting pivot word presence



Predicting pivot word presence



Predicting pivot word presence

Do **not buy** the Shark portable steamer. The trigger mechanism is **defective**.

An absolutely **great** purchase. . . . This blender is incredibly **sturdy**.

⋮

Predict presence of pivot words

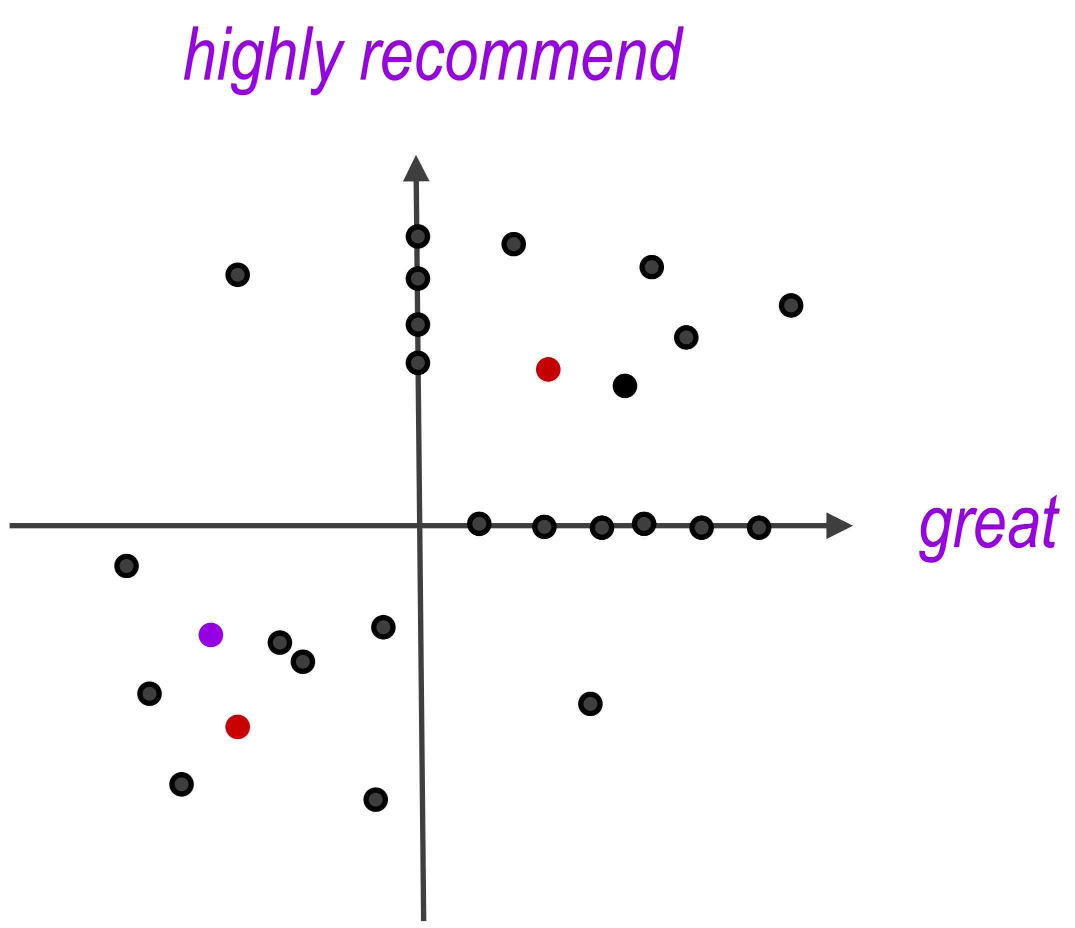
$$p_{w(\mathit{great})}(\mathit{great} \mid x) \propto \exp\{\langle x, w(\mathit{great})\rangle\}$$

A **sturdy** deep fryer

Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

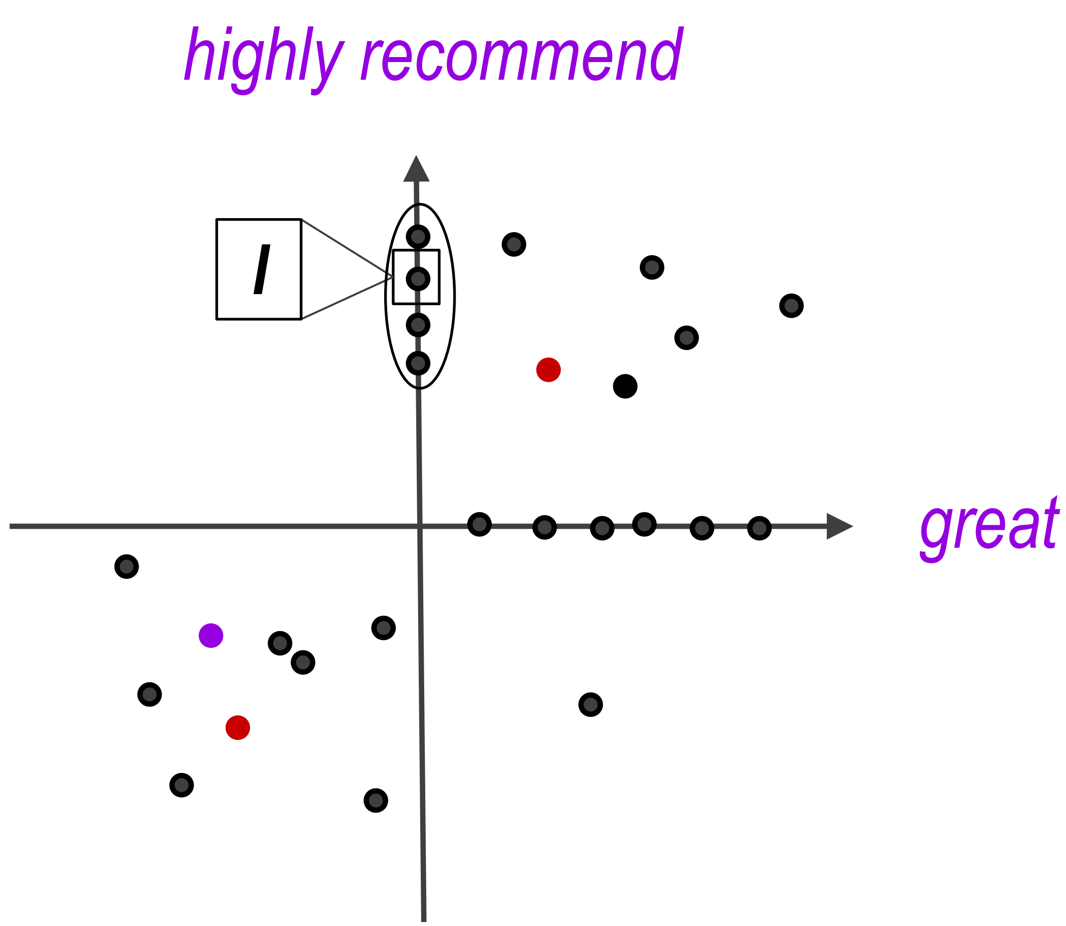
- $p_W(\textit{pivots} \mid x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend} \mid x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

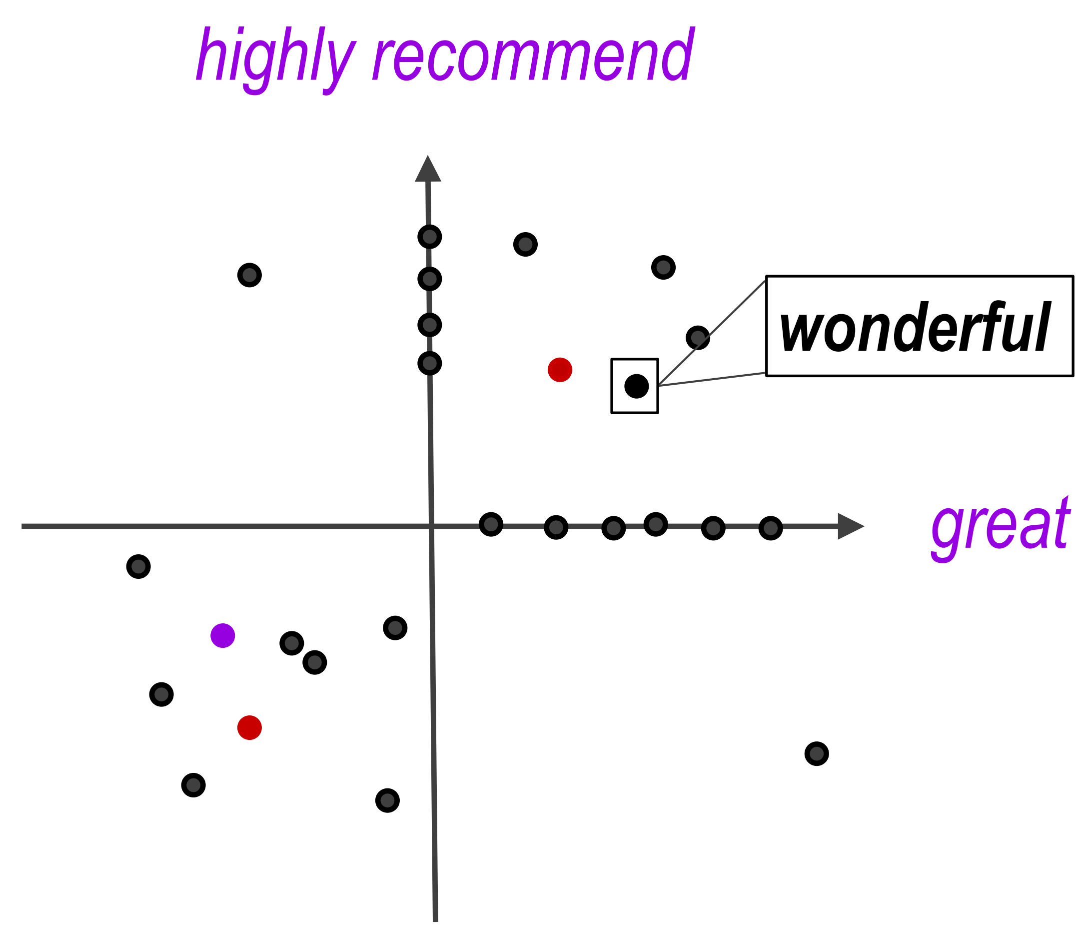
- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\text{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

- $p_W(\text{pivots}|x)$ generates N new features
- $p_{w(\text{highly recommend})}(\text{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information

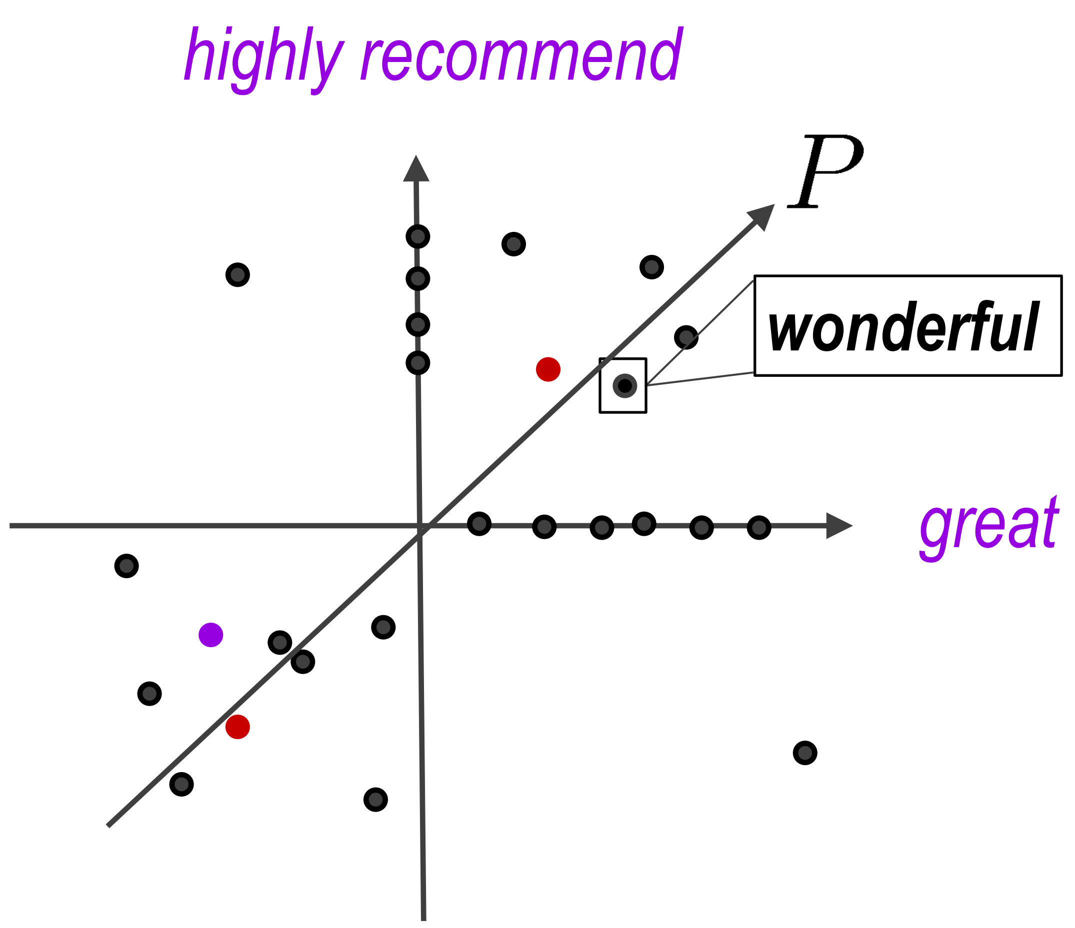


Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

- Let P be a basis for the subspace of best fit to W

- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information

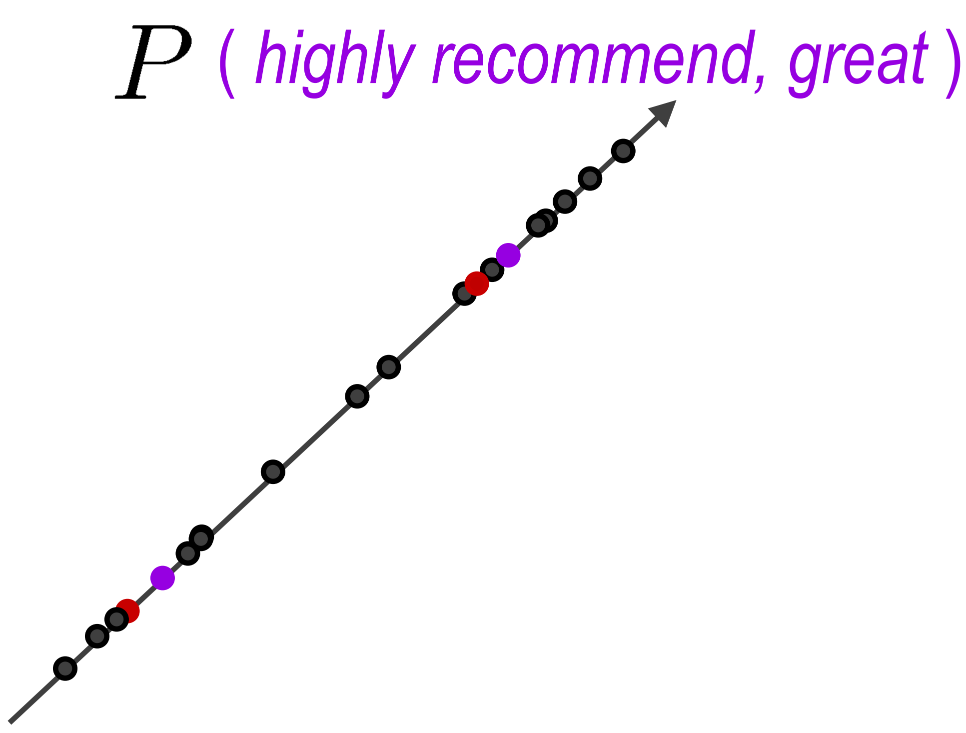


Finding a shared sentiment subspace

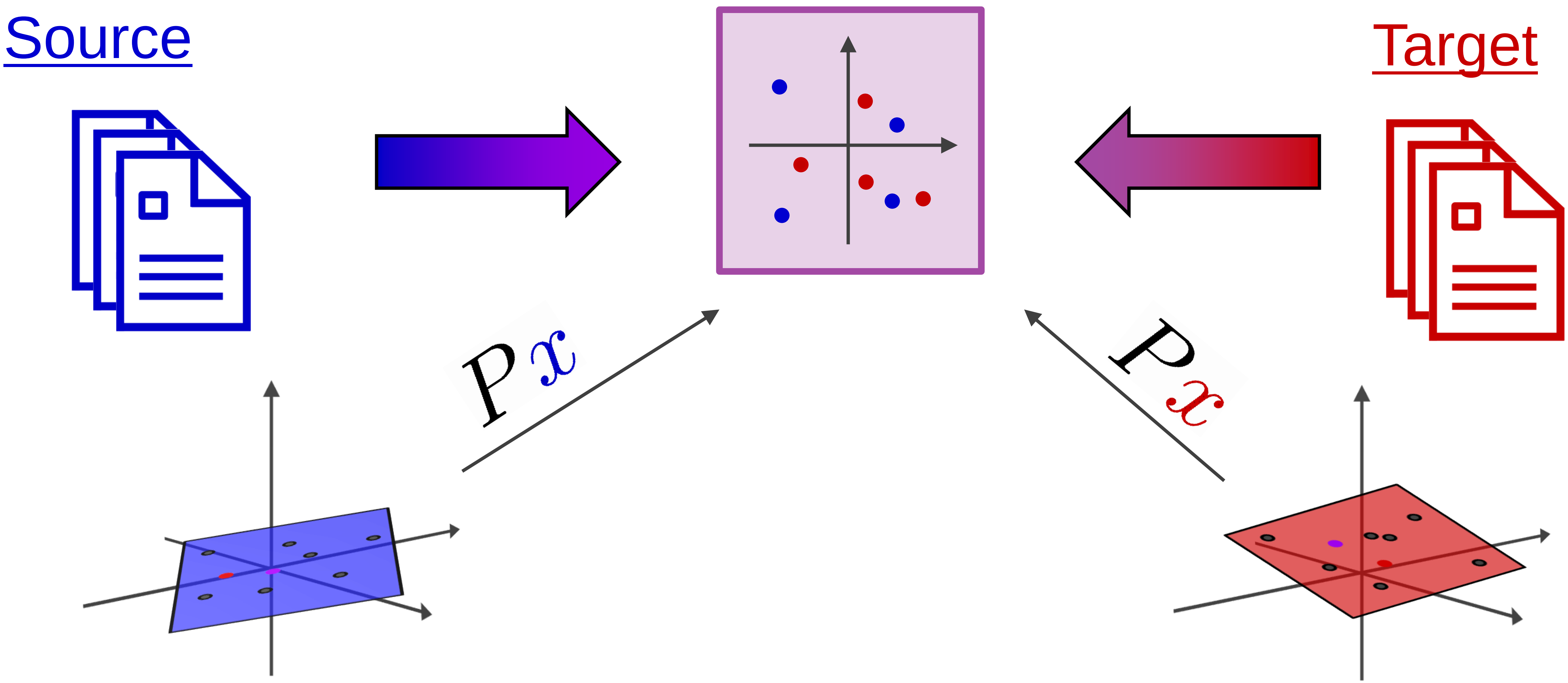
$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

- Let P be a basis for the subspace of best fit to W
- P captures sentiment variance in W

- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information

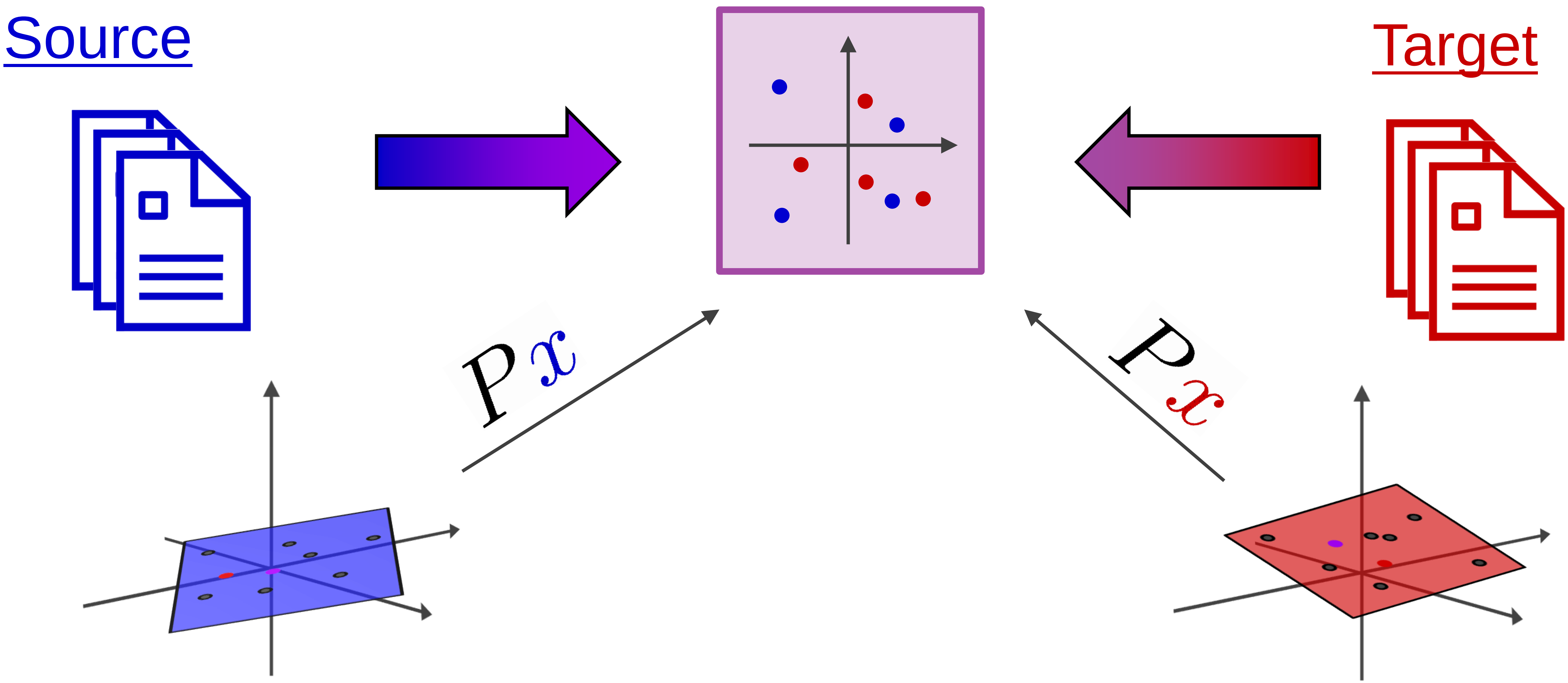


P projects onto shared subspace



$$p_{\tilde{\theta}}(\;\;|x) \propto \exp\left\{\langle \phi(\;\;, Px), \tilde{\theta} \rangle\right\}$$

P projects onto shared subspace



$$h(x) = \operatorname{sgn}\left(\theta^\top P x\right)$$

Correlating Pieces of the Bound

$$\epsilon_T(h) - \epsilon_T(h^*) \quad \leq \quad \epsilon_{\hat{S}}(h, h^*) + O\left(\mathcal{R}_{\hat{S}}(\mathcal{H}) + \mathcal{R}_{\hat{T}}(\mathcal{H})\right)$$

$$+O\left(\sqrt{\frac{\log \frac{1}{\delta}}{n}}\right) + \mathrm{disc}_{\mathcal{H}}(\hat{S}, \hat{T})$$

Component / Projection	Discrepancy	Source Huber Loss	Target Error
Idenitity	1.796	**0.003**	0.253

Correlating Pieces of the Bound

$$\epsilon_T(h) - \epsilon_T(h^*) \quad \leq \quad \epsilon_{\hat{S}}(h, h^*) + O\left(\mathcal{R}_{\hat{S}}(\mathcal{H}) + \mathcal{R}_{\hat{T}}(\mathcal{H})\right)$$

$$+O\left(\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right) + \mathrm{disc}_{\mathcal{H}}(\hat{S}, \hat{T})$$

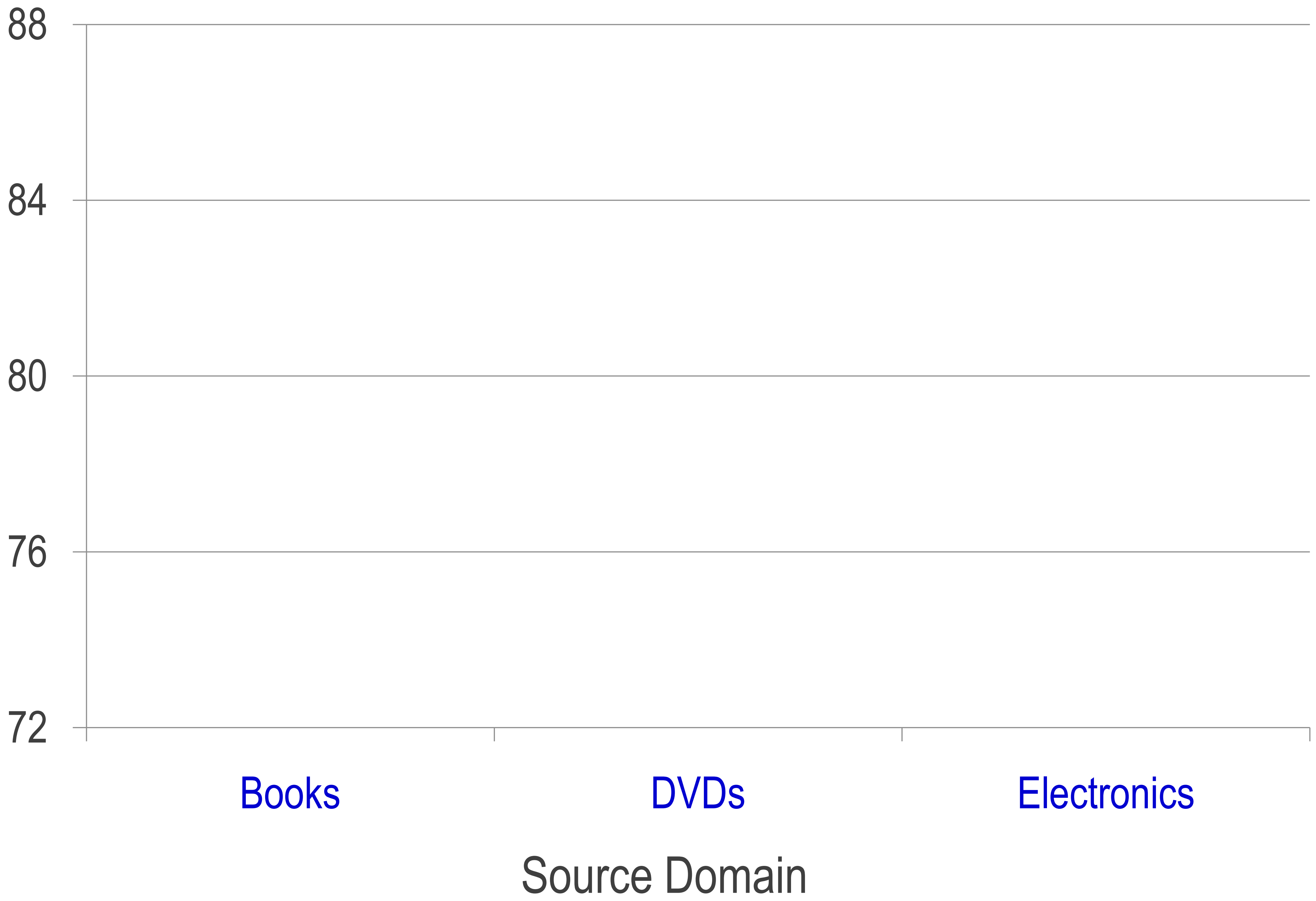
Component / Projection	Discrepancy	Source Huber Loss	Target Error
Idenitity	1.796	**0.003**	0.253
Random	0.223	0.254	0.561

Correlating Pieces of the Bound

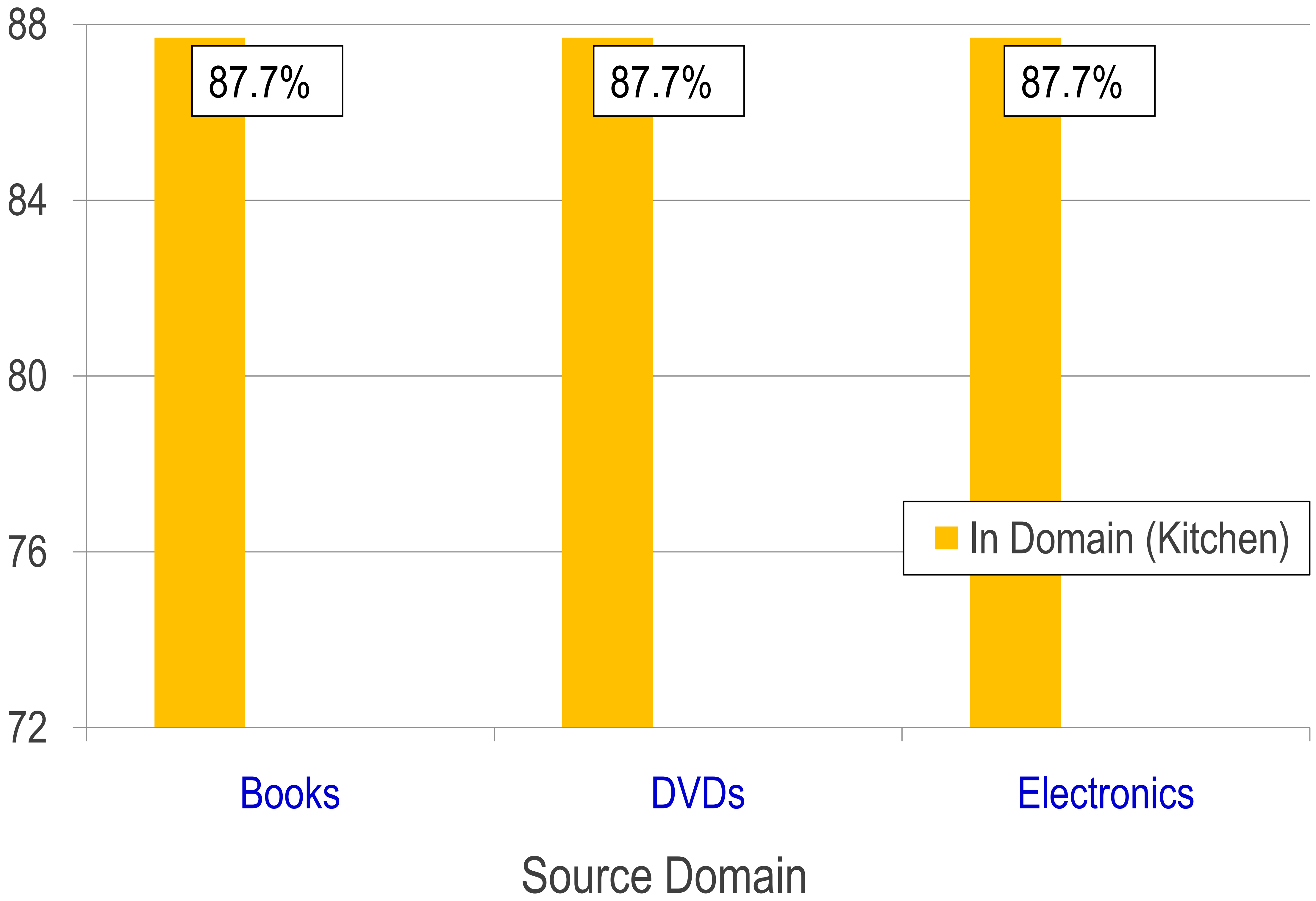
$$\epsilon_T(h) - \epsilon_T(h^*) \quad \leq \quad \epsilon_{\hat{S}}(h, h^*) + O\left(\mathcal{R}_{\hat{S}}(\mathcal{H}) + \mathcal{R}_{\hat{T}}(\mathcal{H})\right)$$
$$+ O\left(\sqrt{\frac{\log \frac{1}{\delta}}{n}}\right) + \mathrm{disc}_{\mathcal{H}}(\hat{S}, \hat{T})$$

Projection \ Component	Discrepancy	Source Huber Loss	Target Error
Idenitity	1.796	**0.003**	0.253
Random	0.223	0.254	0.561
Coupled Projection	**0.211**	0.07	**0.216**

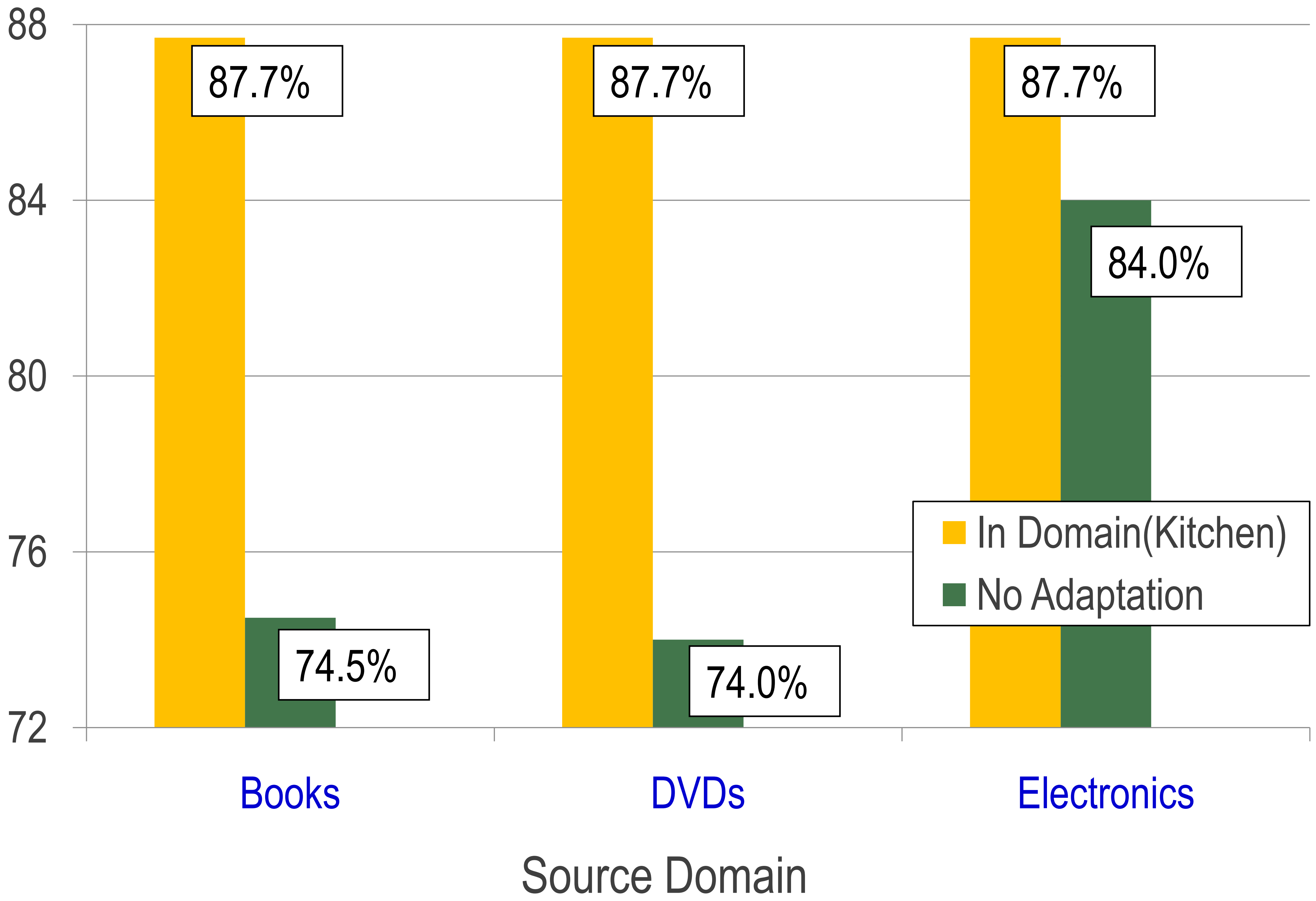
Target Accuracy: Kitchen Appliances



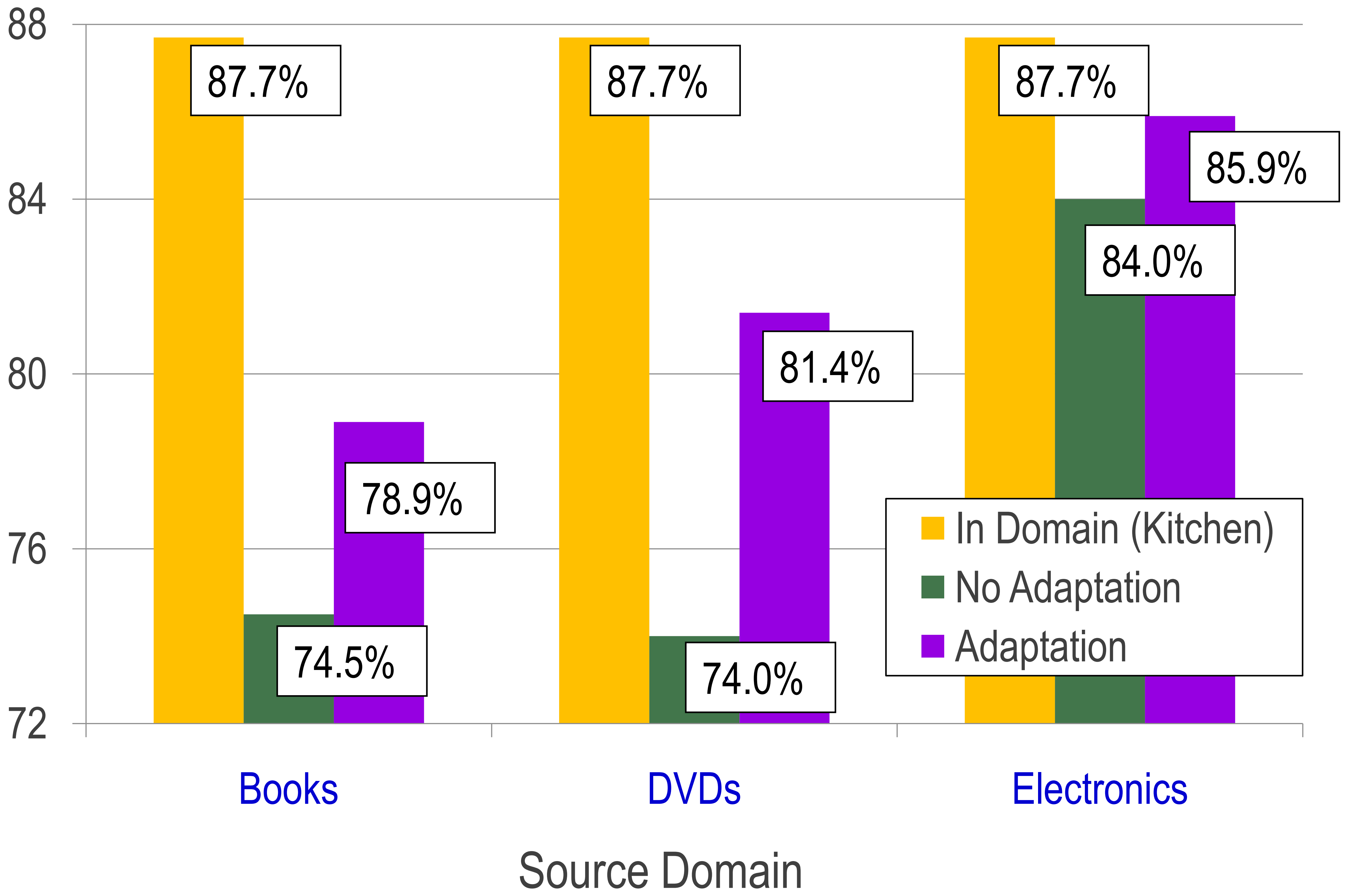
Target Accuracy: Kitchen Appliances



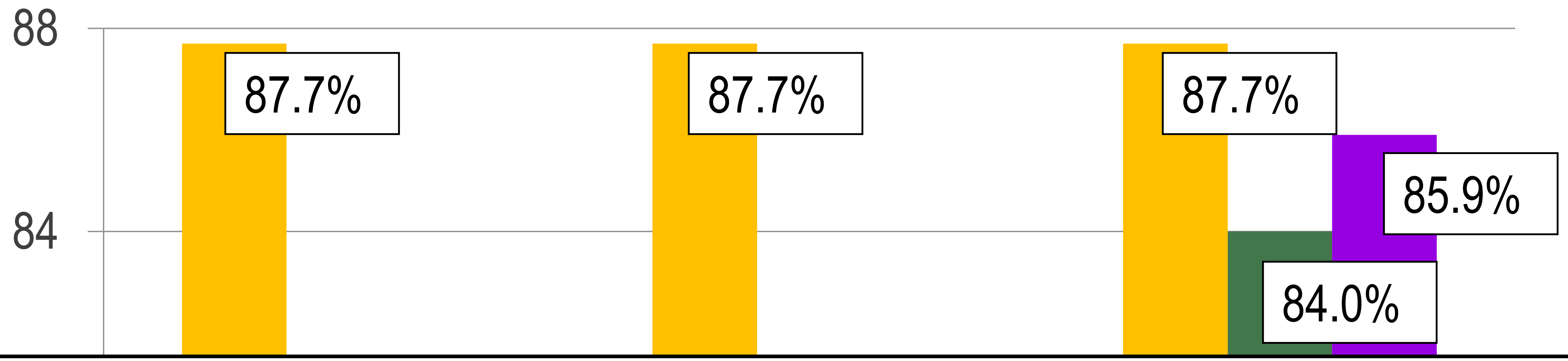
Target Accuracy: Kitchen Appliances



Target Accuracy: Kitchen Appliances



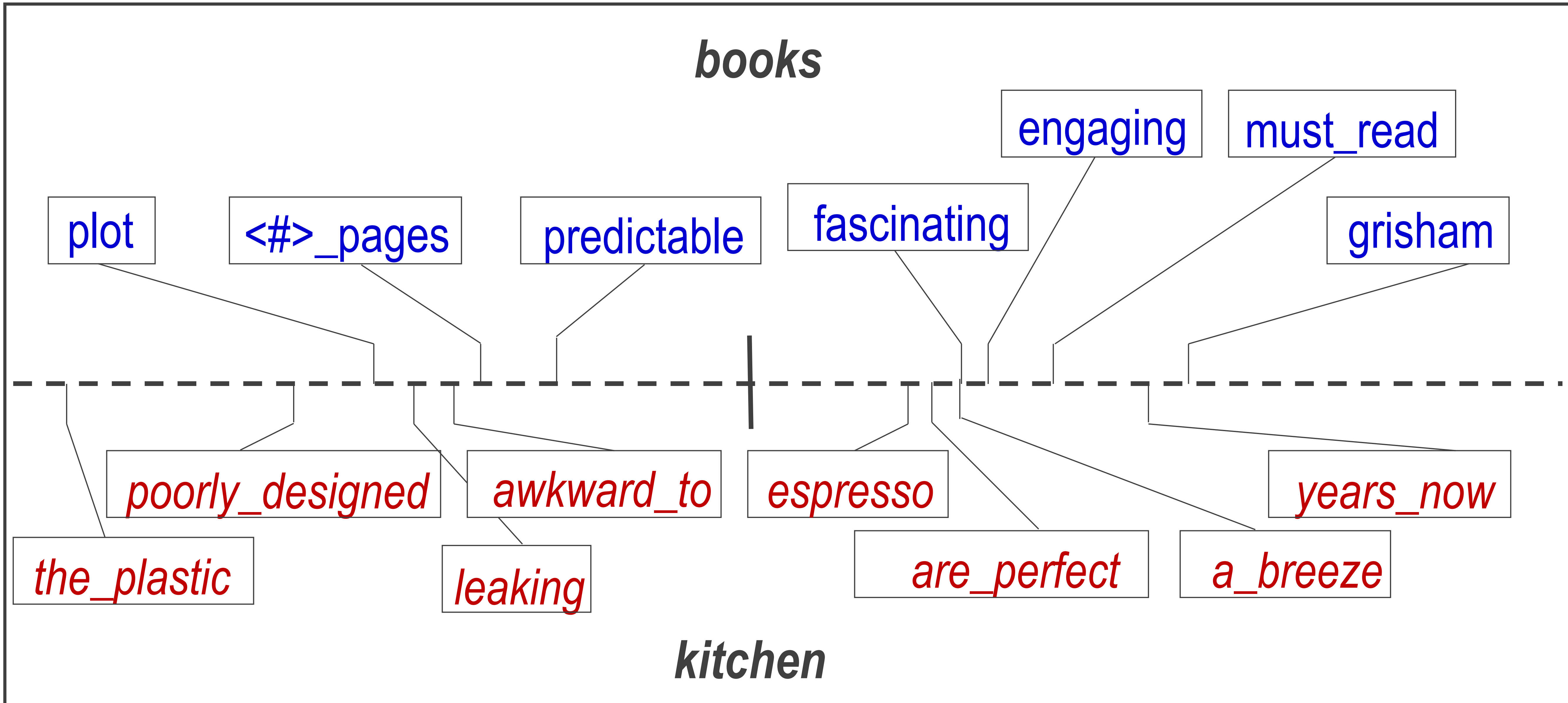
Adaptation Error Reduction



<u>36% reduction in error due to adaptation</u>

Visualizing P (books & kitchen)

negative vs. positive



Representation References

http://adaptationtutorial.blitzer.com/references/

[1] Blitzer et al. Domain Adaptation with Structural Correspondence Learning. 2006.

[2] S. Ben-David et al. Analysis of Representations for Domain Adaptation. 2007.

[3] J. Blitzer et al. Domain Adaptation for Sentiment Classification. 2008.

[4] Y. Mansour et al. Domain Adaptation: Learning Bounds and Algorithms. 2009.

Tutorial Outline

1. Notation and Common Concepts

2. Semi-supervised Adaptation
   - Covariate shift
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

4. Open Questions and Uncovered Algorithms

Feature-based approaches

<u>Cell-phone domain:</u>
“horrible” is **bad**
“small” is **good**

<u>Hotel domain:</u>
“horrible” is **bad**
“small” is **bad**

Key Idea:
Share some features (“horrible”)
Don't share others (“small”)

(and let an arbitrary *learning algorithm* decide which are which)

In feature-vector lingo:

x → ‹x, x, 0›   (for source domain)

x → ‹x, 0, x›   (for target domain)

	The phone is small	The hotel is small
Original Features	W:the W:phone W:is W:small	W:the W:hotel W:is W:small
Augmented Features	S:the S:phone S:is S:small	
		T:the T:hotel T:is T:small

A Kernel Perspective

In feature-vector lingo:

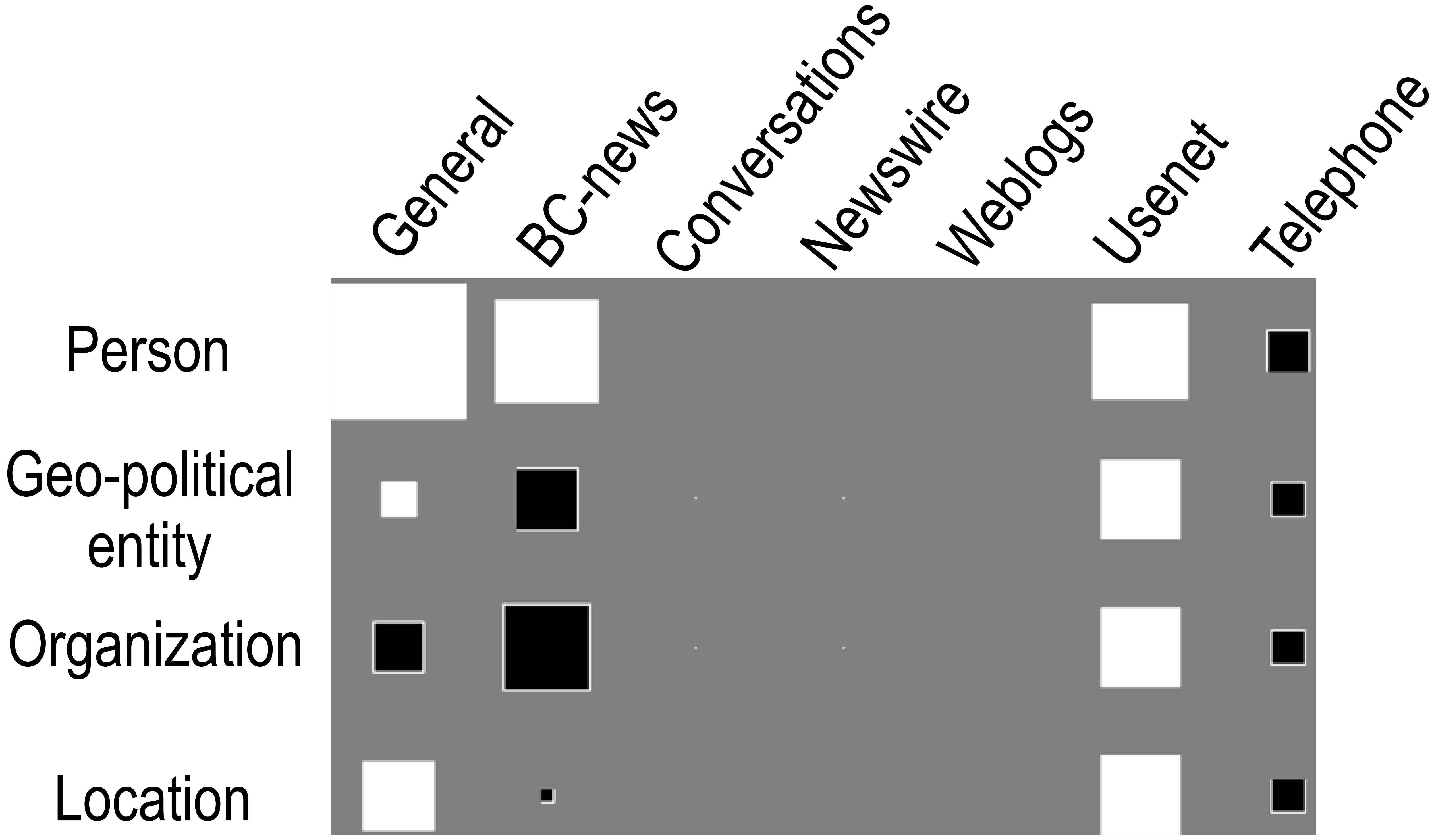
$x \rightarrow \langle x, x, 0 \rangle$ (for source domain)

$x \rightarrow \langle x, 0, x \rangle$ (for target domain)

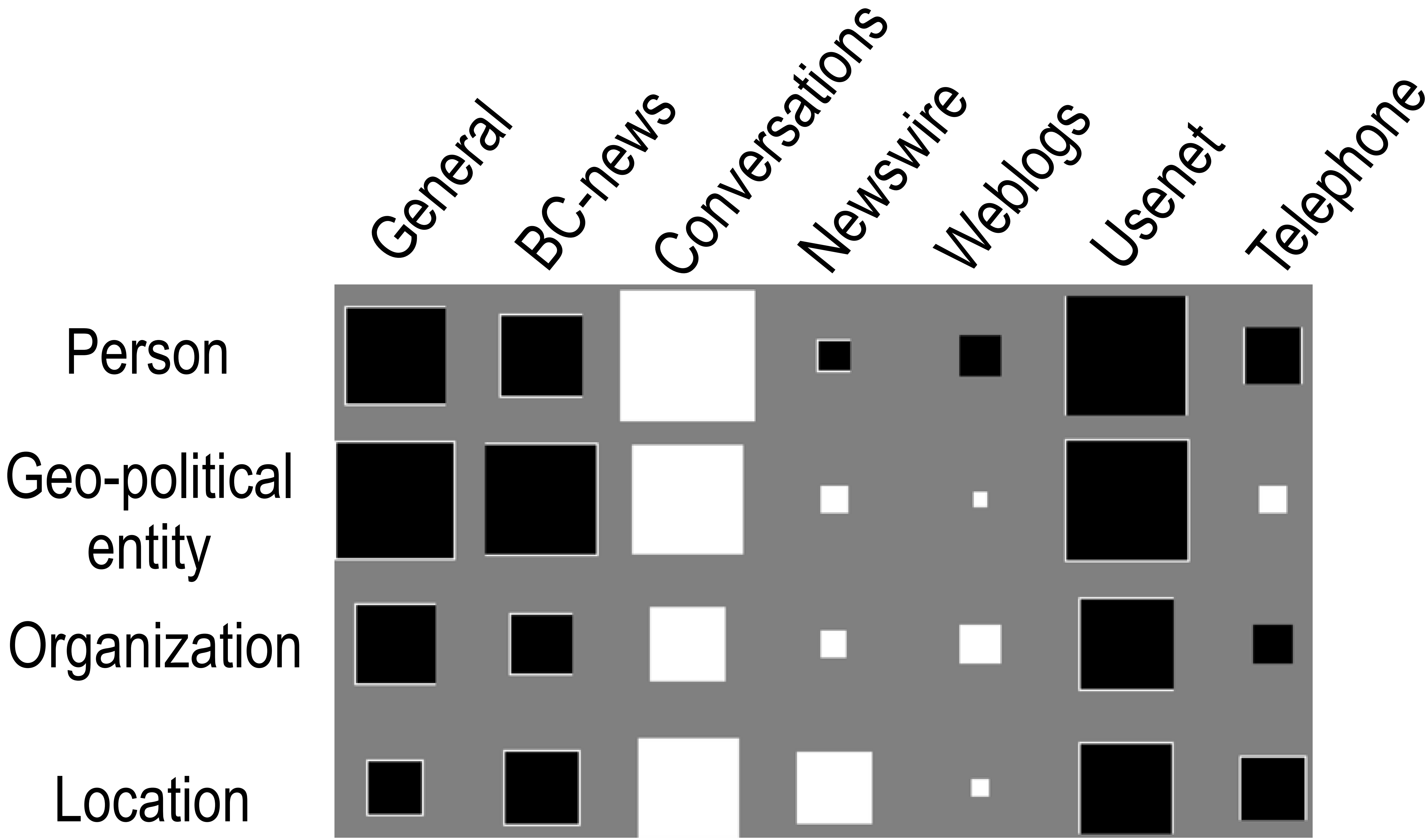
$$K^{aug}(x,z) = \begin{cases} 2K(x,z) & \text{if } x,z \text{ from same domain} \\ K(x,z) & \text{othe} \end{cases}$$

We have *ensured* SGH & *destroyed* shared support

Named Entity Rec.: /bush/



Named Entity Rec.: p=/the/



Some experimental results

Task	Dom	SrcOnly	TgtOnly	Baseline	Prior	Augment
ACE-NER	bn	4.98	2.37	2.11 (pred)	2.06	**1.98**
	bc	4.54	4.07	3.53 (weight)	**3.47**	**3.47**
	nw	4.78	3.71	3.56 (pred)	3.68	**3.39**
	wl	2.45	2.45	**2.12** (all)	2.41	**2.12**
	un	3.67	2.46	2.10 (linint)	2.03	**1.91**
	cts	2.08	0.46	0.40 (all)	**0.34**	**0.32**
CoNLL	tgt	2.49	2.95	**1.75** (wgt/li)	1.89	**1.76**
PubMed	tgt	12.02	4.15	3.95 (linint)	3.99	**3.61**
CNN	tgt	10.29	3.82	3.44 (linint)	**3.35**	**3.37**
Tree bank-Chunk	wsj	6.63	4.35	4.30 (weight)	4.27	**4.11**
	swbd3	15.90	4.15	4.09 (linint)	3.60	**3.51**
	br-cf	5.16	6.27	**4.72** (linint)	5.22	5.15
	br-cg	4.32	5.36	**4.15** (all)	4.25	4.90
	br-ck	5.05	6.32	**5.01** (prd/li)	5.27	5.41
	br-cl	5.66	6.60	**5.39** (wgt/prd)	5.99	5.73
	br-cm	3.57	6.59	**3.11** (all)	4.08	4.89
	br-cn	4.60	5.56	**4.19** (prd/li)	4.48	4.42
	br-cp	4.82	5.62	**4.55** (wgt/prd/li)	4.87	4.78
	br-cr	5.78	9.13	**5.15** (linint)	6.71	6.30
Treebank-	brown	6.35	5.75	4.72 (linint)	4.72	**4.65**

Some Theory

- Can bound expected target error:

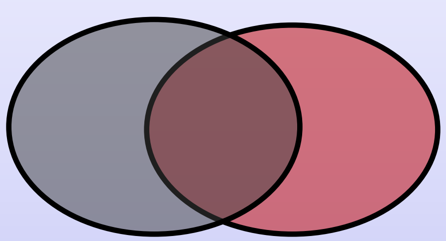
Average training error

$$\epsilon_t \le \frac{1}{2}\left(\hat{\epsilon}_s + \hat{\epsilon}_t\right) + O(complexity)$$

$$+\left(\frac{1}{\sqrt{N_s}} + \frac{1}{\sqrt{N_t}}\right) O\left(\frac{1}{\delta}\right) + O\left(disc_H(S,T)\right)$$

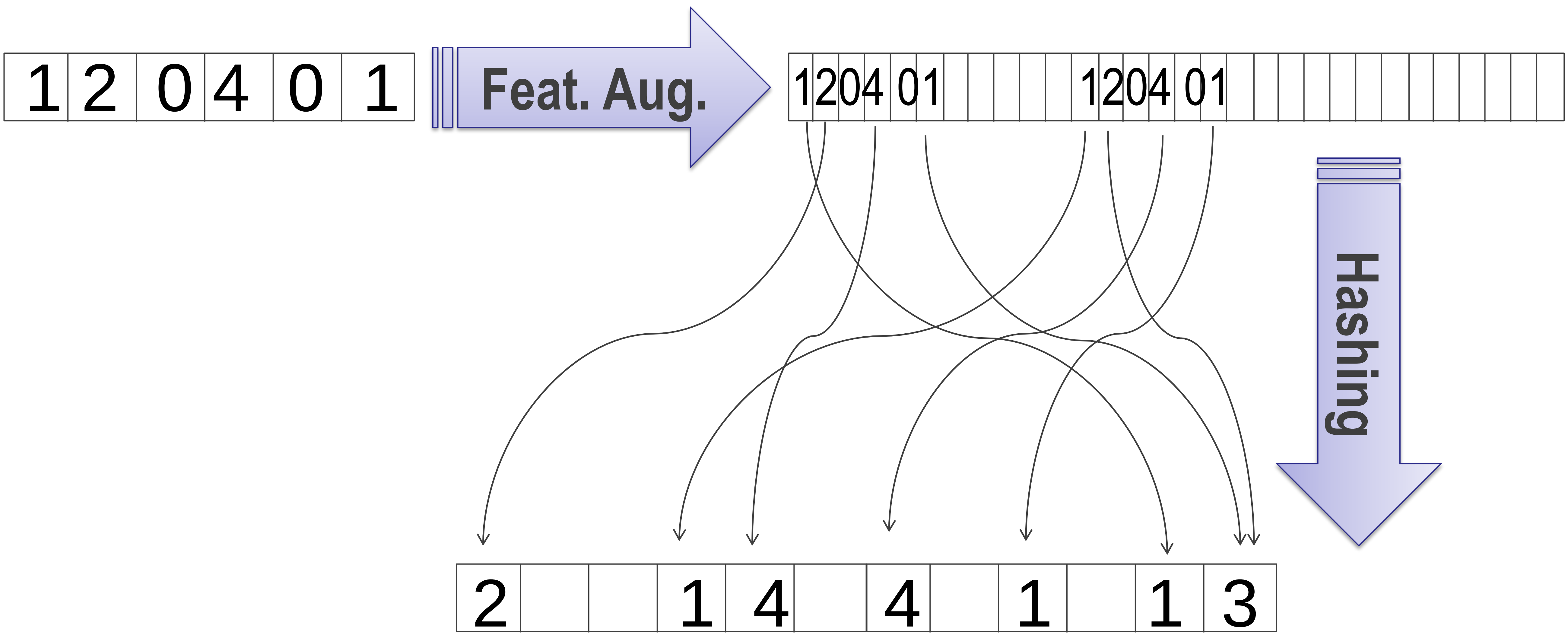
Number of source examples

Number of target examples

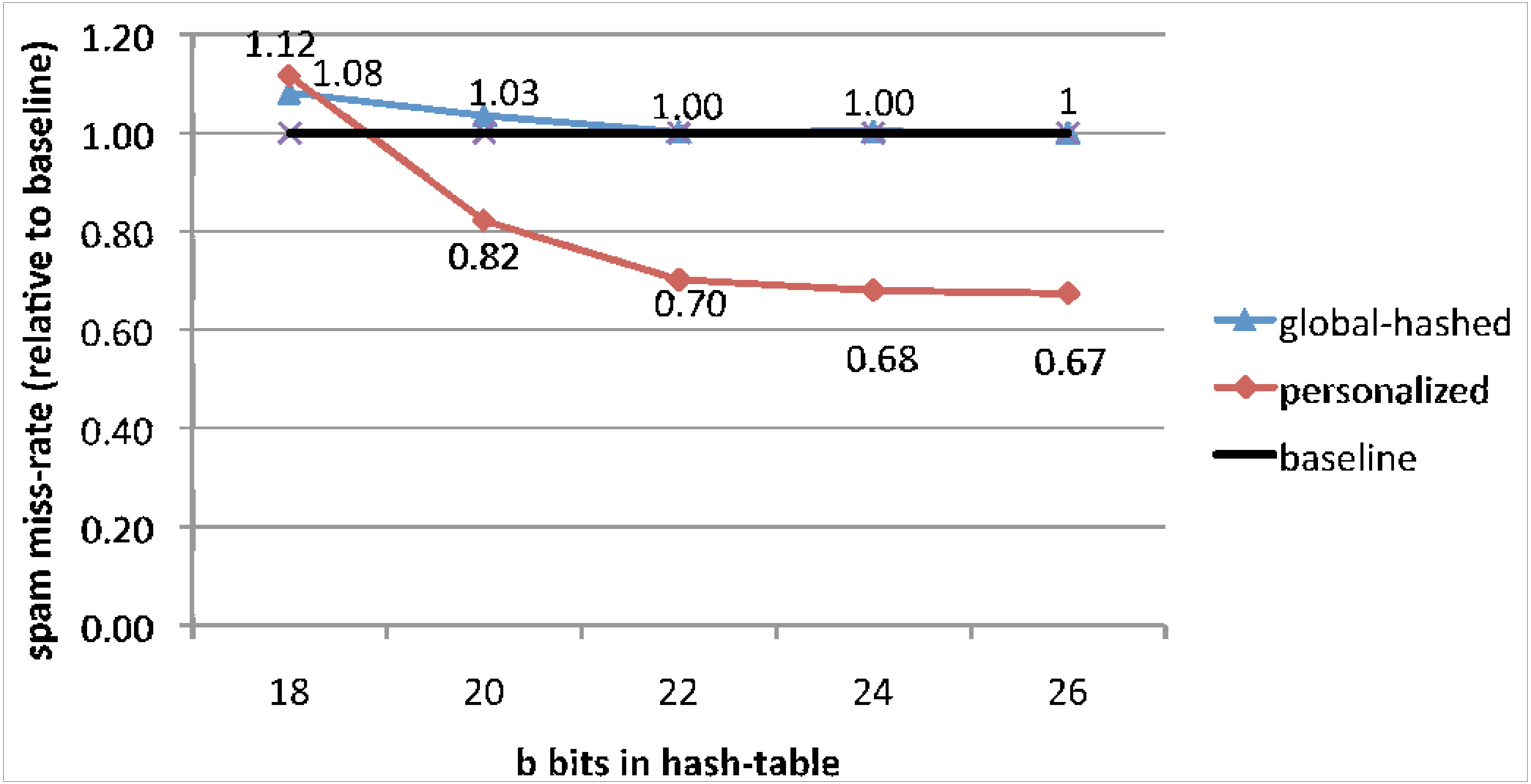


Feature Hashing

- Feature augmentation creates (K+1)D parameters
- Too big if K>>20, but *very sparse!*



Hash Kernels



Semi-sup Feature Augmentation

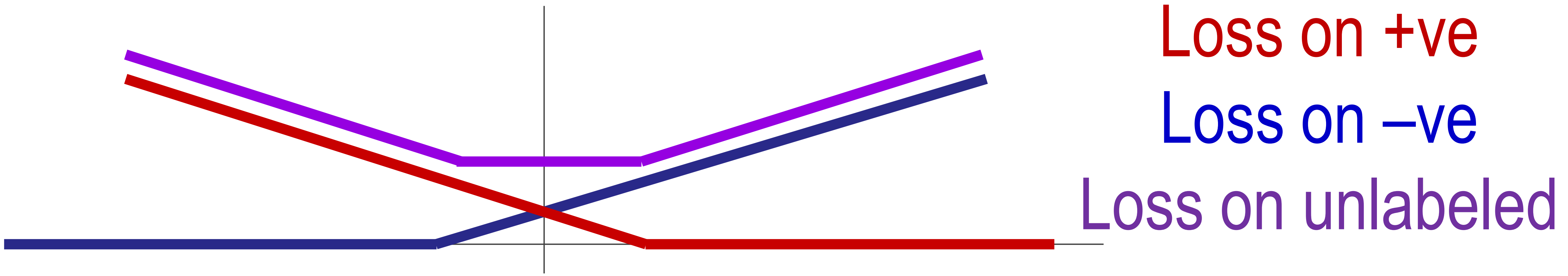
- For labeled data:
  - (y, x) → (y, ‹x, x, 0›) (for source domain)
  - (y, x) → (y, ‹x, 0, x›) (for target domain)
- What about unlabeled data?

- Encourage agreement:

$$\left[h_t(x) = h_s(x)\right] \Leftrightarrow \left[w_t \circ x - w_s \circ x = 0\right]$$

Semi-sup Feature Augmentation

- For labeled data:
  - (y, x) → (y, ‹x, x, 0›) (for source domain)
  - (y, x) → (y, ‹x, 0, x›) (for target domain)
- What about unlabeled data?
  - (x) → { (+1, ‹0, x, -x›) , (-1, ‹0, x, -x›) }



- Encourages *agreement* on unlabeled data
  - Akin to *multiview learning*
  - Reduces generalization bound

Feature-based References

- T. Evgeniou and M. Pontil. Regularized Multi-task Learning (2004).
- H. Daumé III, Frustratingly Easy Domain Adaptation. 2007.
- K. Weinberger, A. Dasgupta, J. Langford, A. Smola, J. Attenberg. Feature Hashing for Large Scale Multitask Learning. 2009.
- A. Kumar, A. Saha and H. Daumé III, Frustratingly Easy Semi-Supervised Domain Adaptation. 2010.

Tutorial Outline

1. Notation and Common Concepts

2. Semi-supervised Adaptation
   - Covariate shift
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

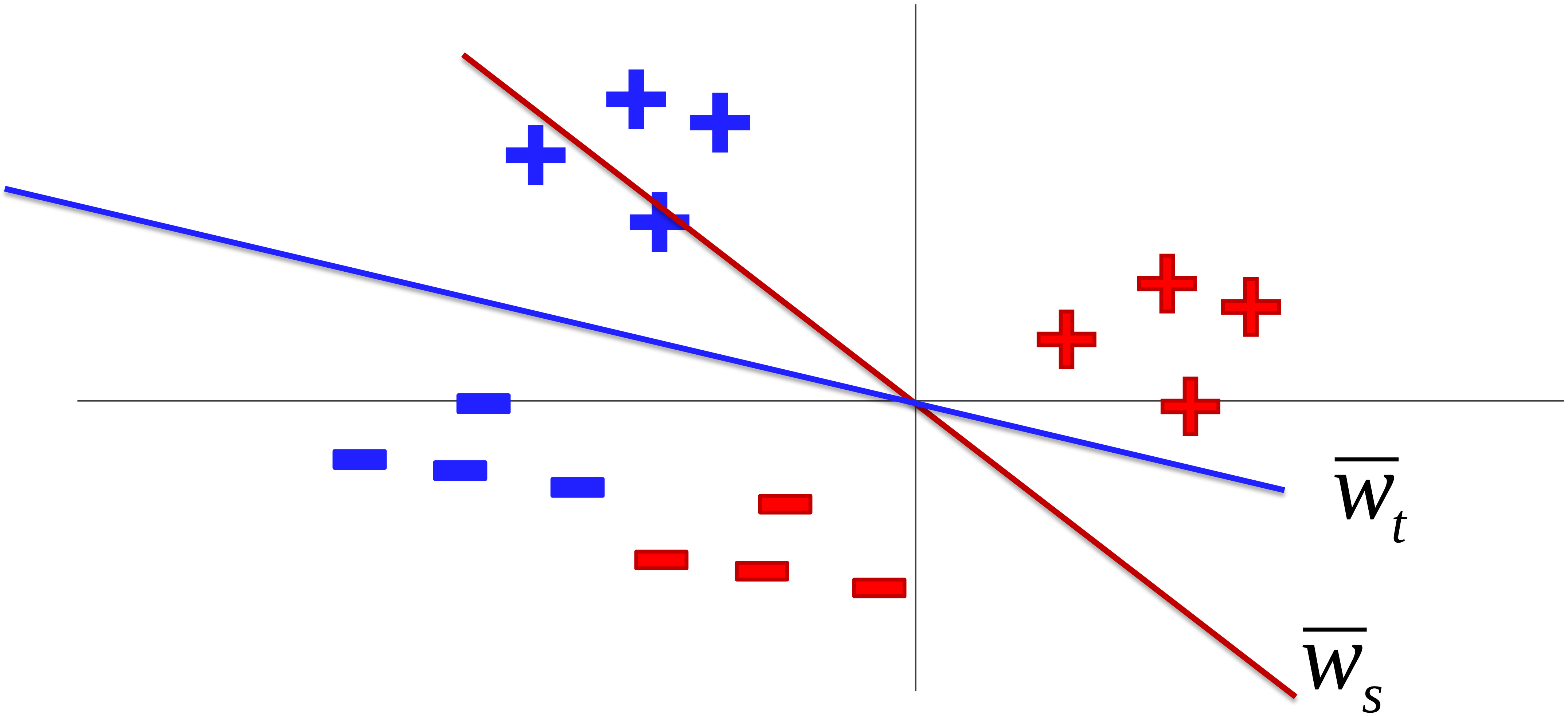
4. Open Questions and Uncovered Algorithms

A Parameter-based Perspective

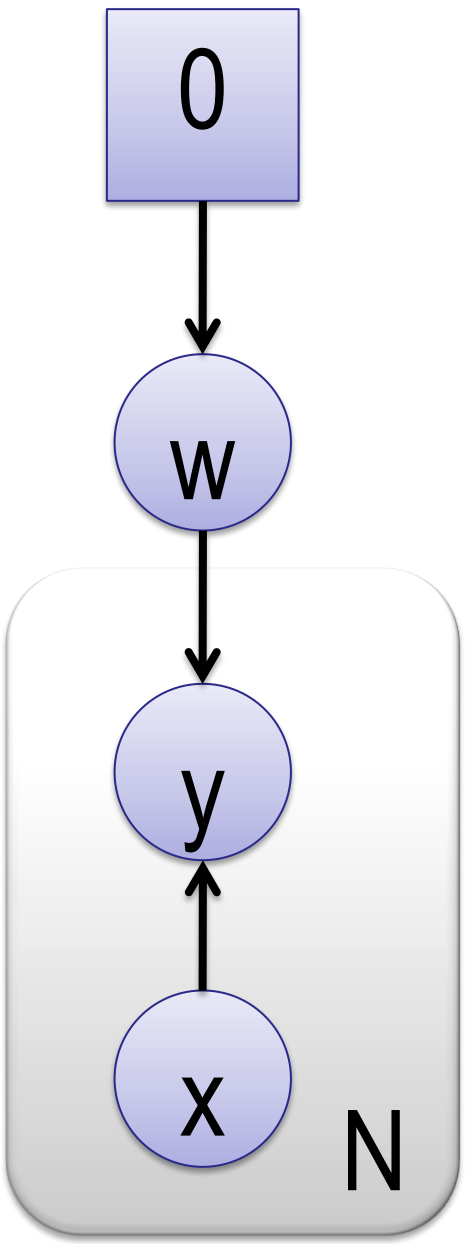
- Instead of duplicating features, write:

$$\overline{w}_t = \overline{w}_s + \overline{v}$$

- And *regularize* $\overline{w}_s$ and $\overline{v}$ toward zero

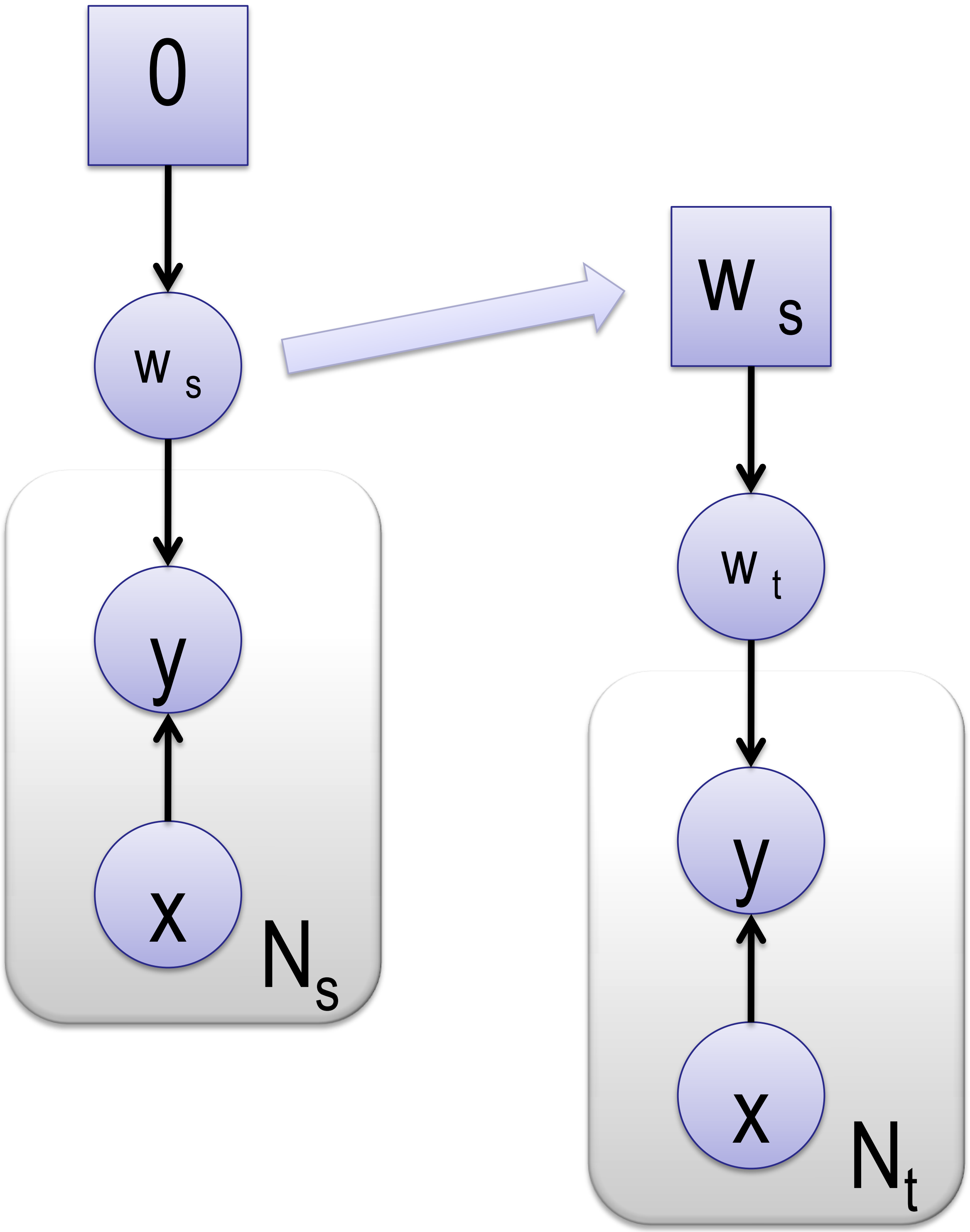


Sharing Parameters via Bayes



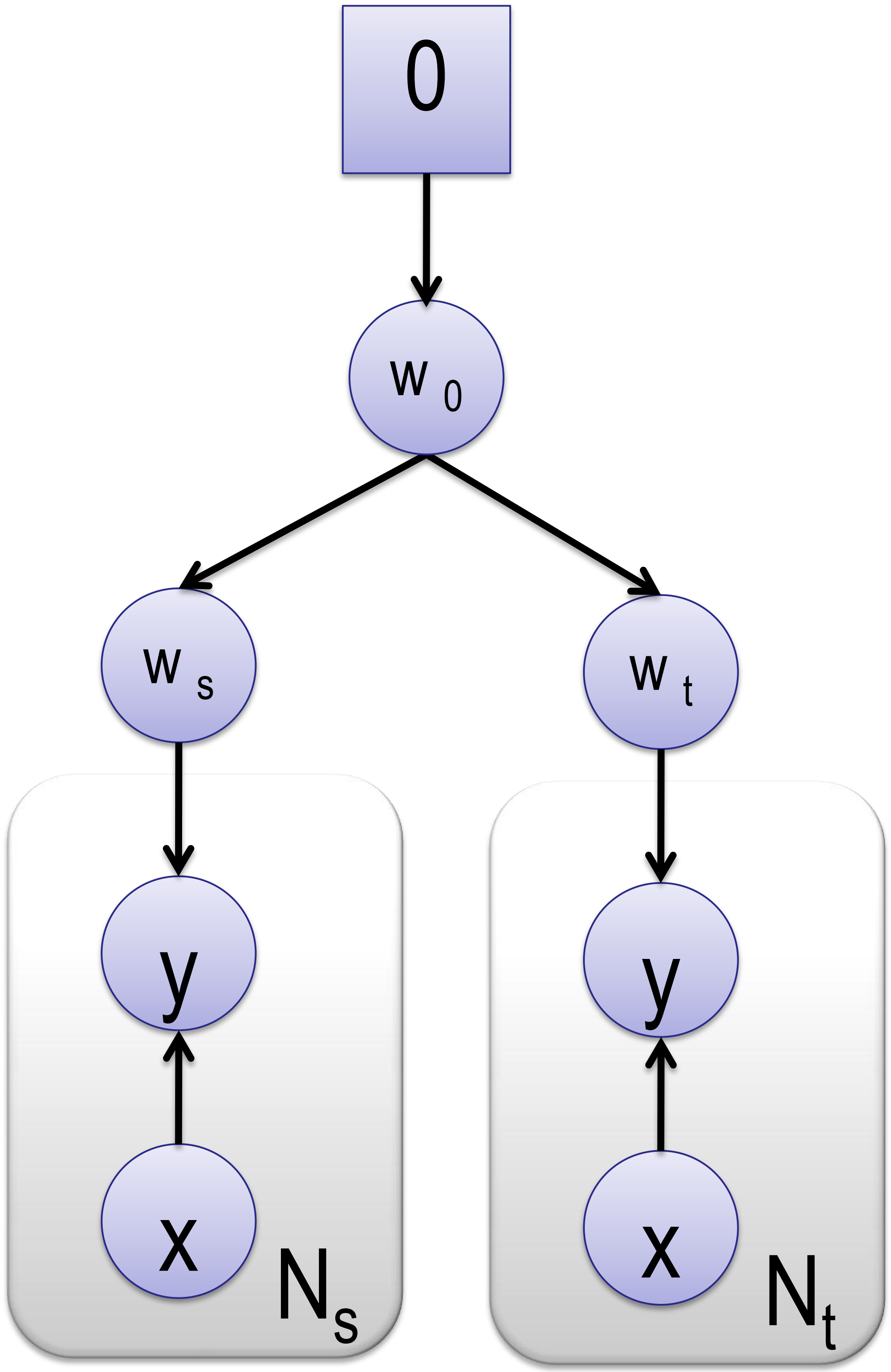
- N data points
- w is regularized to zero
- Given x and w,
  we predict y

Sharing Parameters via Bayes



- Train model on source domain, regularized toward zero
- Train model on target domain, regularized toward source domain

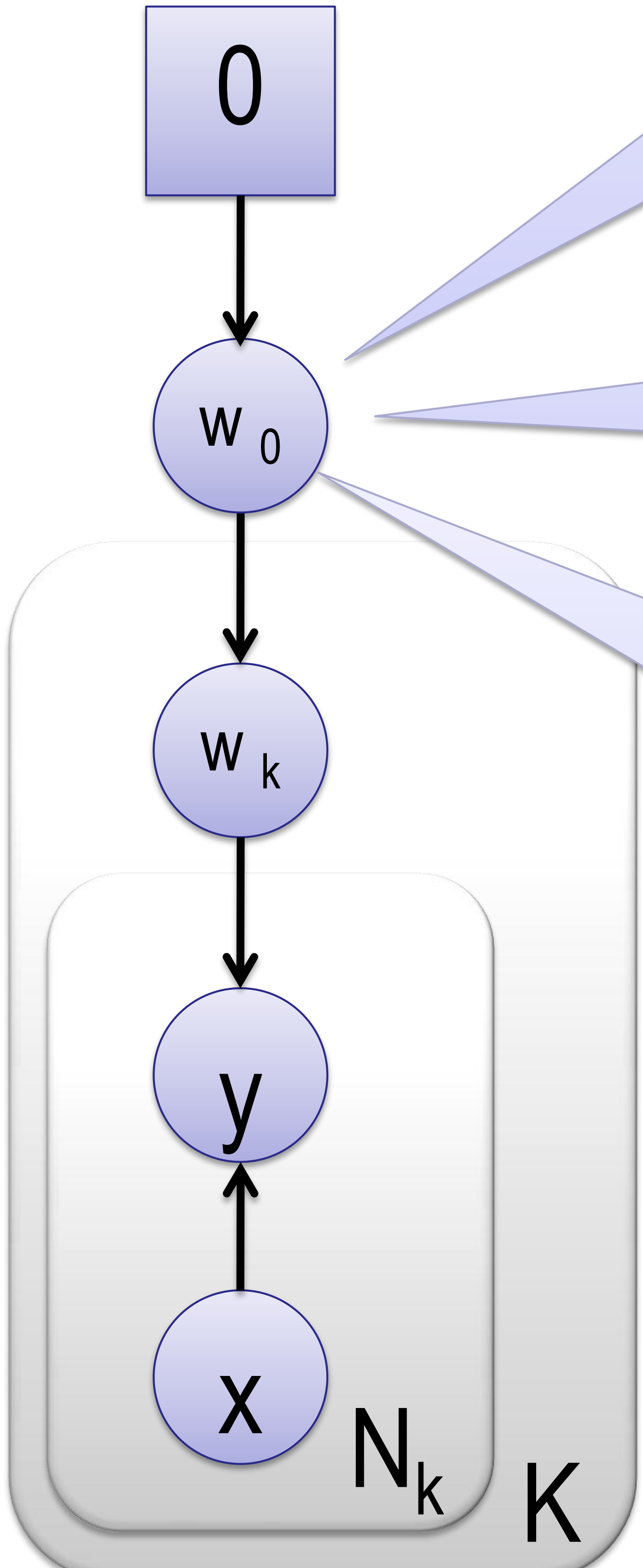
Sharing Parameters via Bayes



- Separate weights for each domain
- Each regularized toward w_0
- w_0 regularized toward zero

Sharing P… via Bayes

$w_0 \sim \mathrm{Nor}(0, \rho^2)$
$w_k \sim \mathrm{Nor}(w_0, \sigma^2)$



- … weights for … in
- …arized … toward w_0

Non-linearize:
$w_0 \sim \mathrm{GP}(0, K)$
$w_k \sim \mathrm{GP}(w_0, K)$

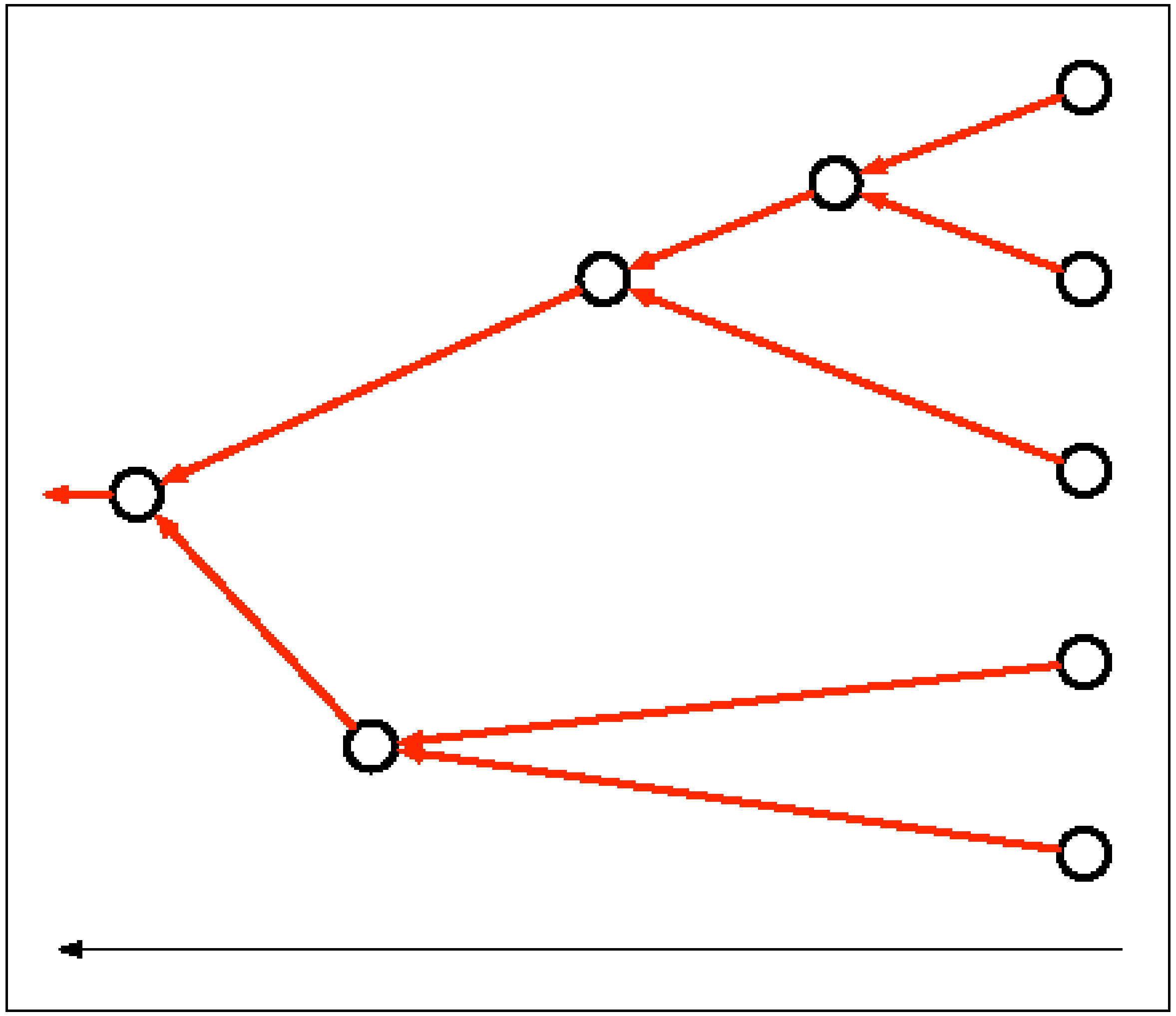
Cluster Tasks:
$w_0 \sim \mathrm{Nor}(0, \sigma^2)$
$w_k \sim \mathrm{DP}(\mathrm{Nor}(w_0, \rho^2), \alpha)$

- Can derive "augment" as an approximation to this model

Not all domains created equal

- Would like to infer tree structure automatically
- Tree structure should be good for the *task*
- Want to simultaneously infer tree structure and parameters

Kingman's Coalescent

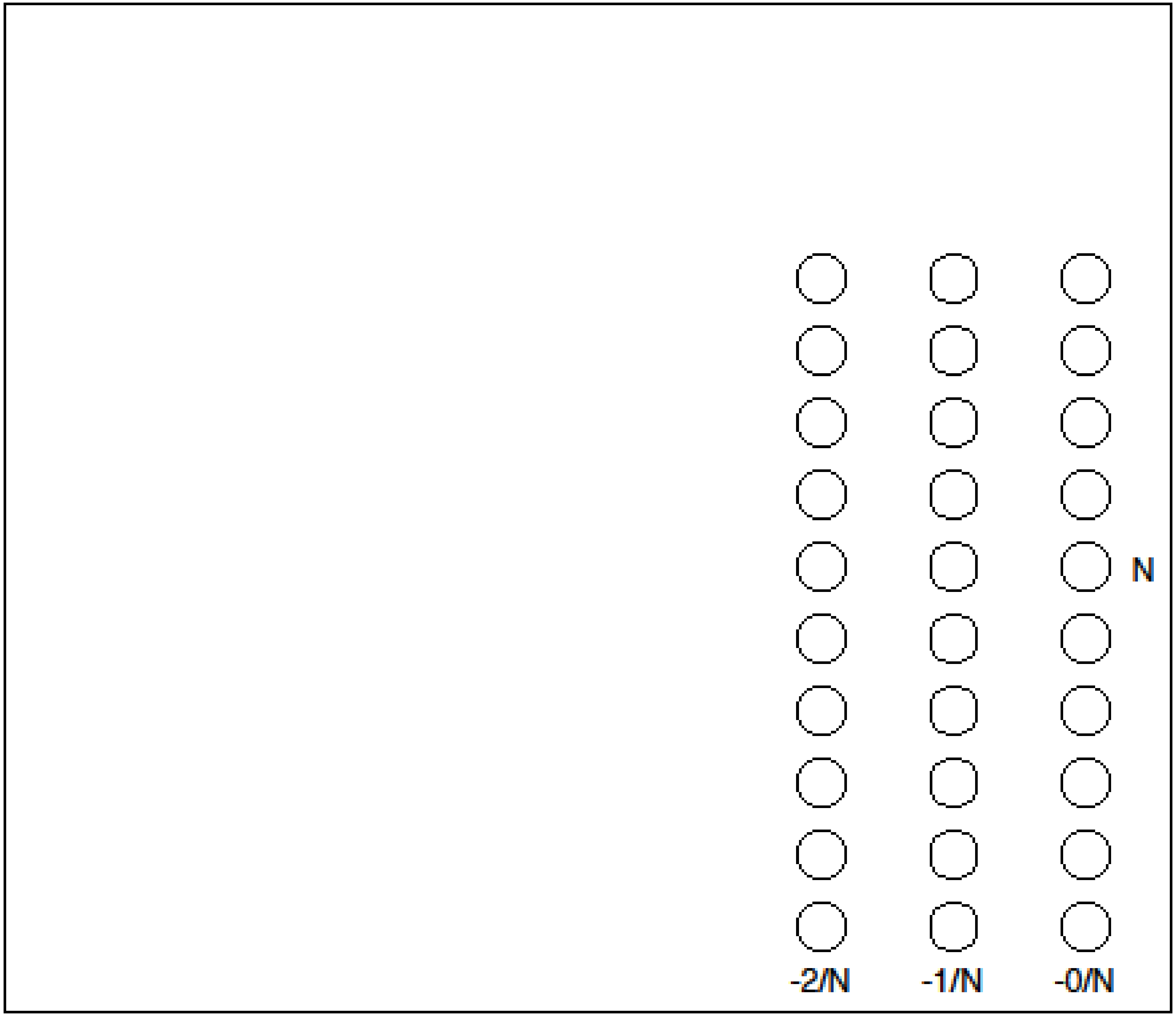


- A standard model for the genealogy of a population
- Each organism has exactly one parent (haploid)
- Thus, the genealogy is a tree

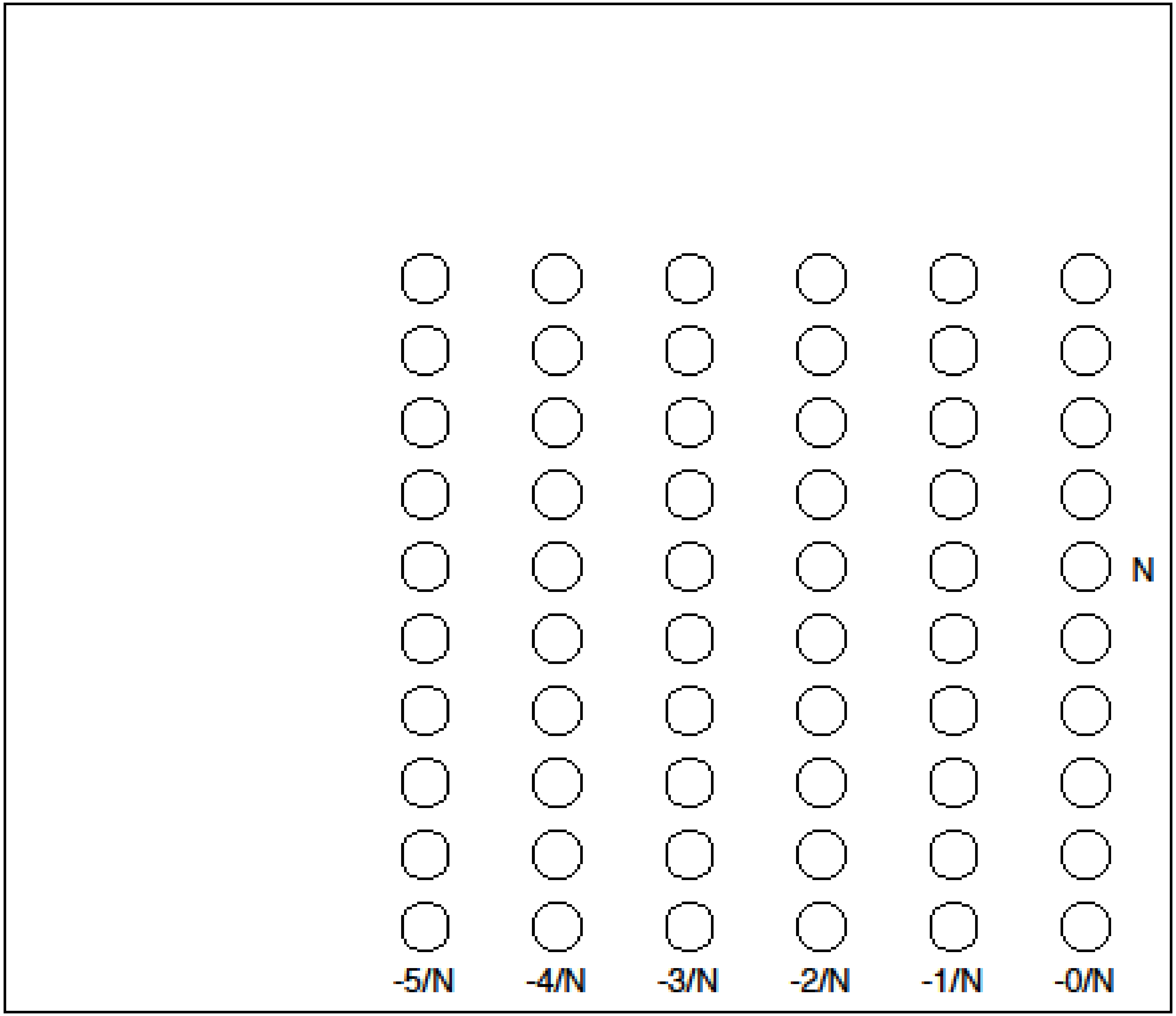
A distribution over trees

A distribution over trees

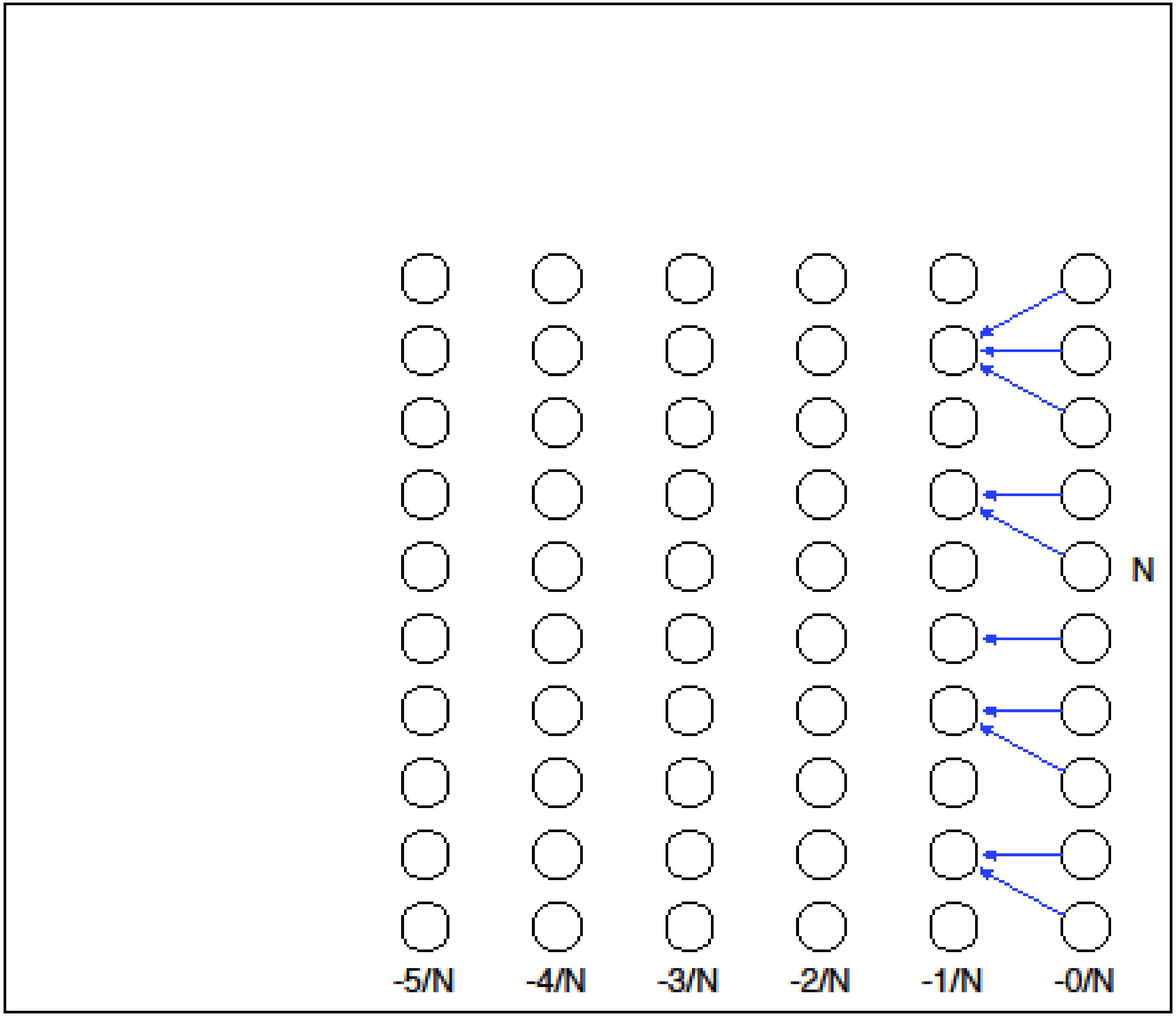
A distribution over trees



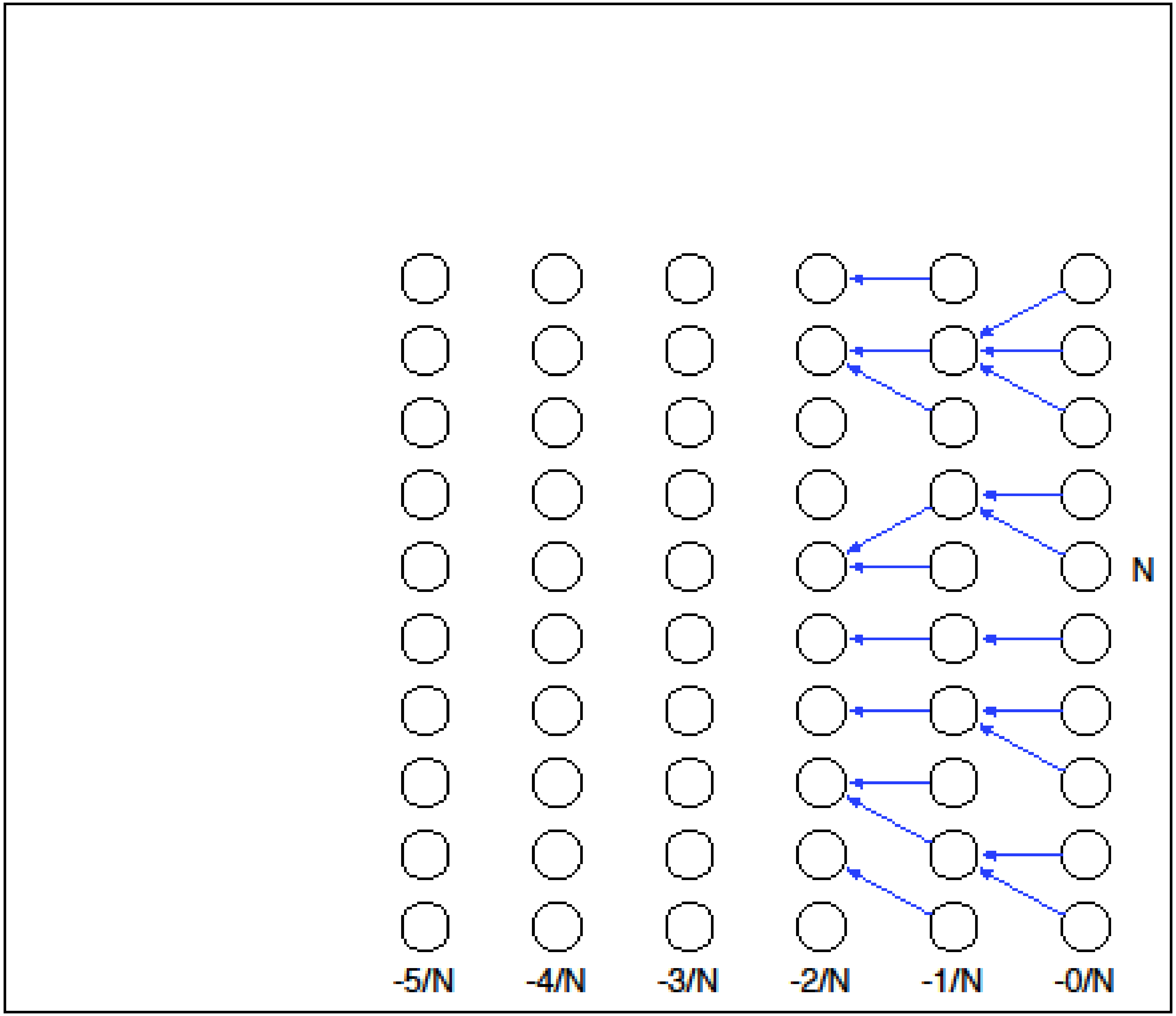
A distribution over trees



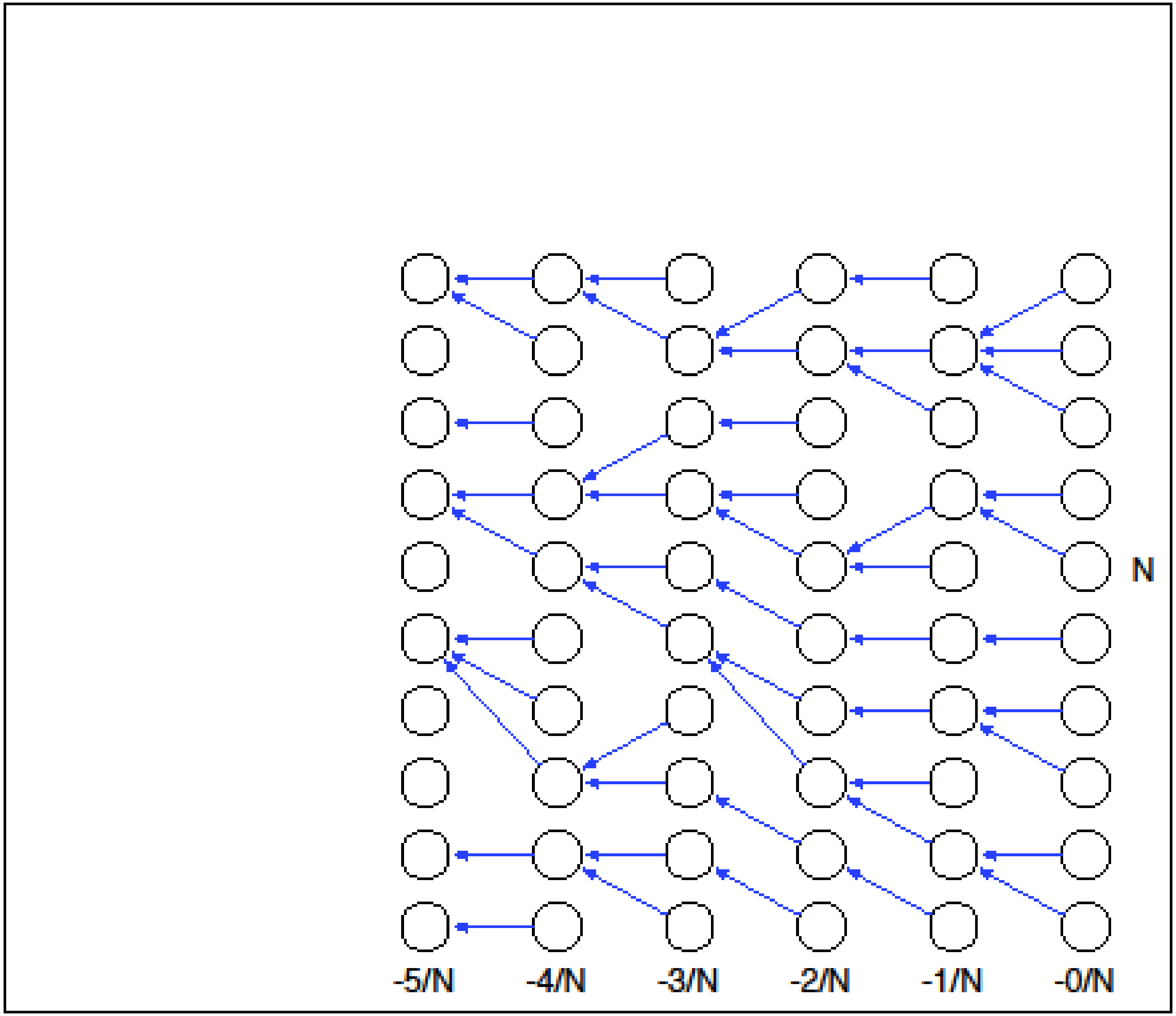
A distribution over trees



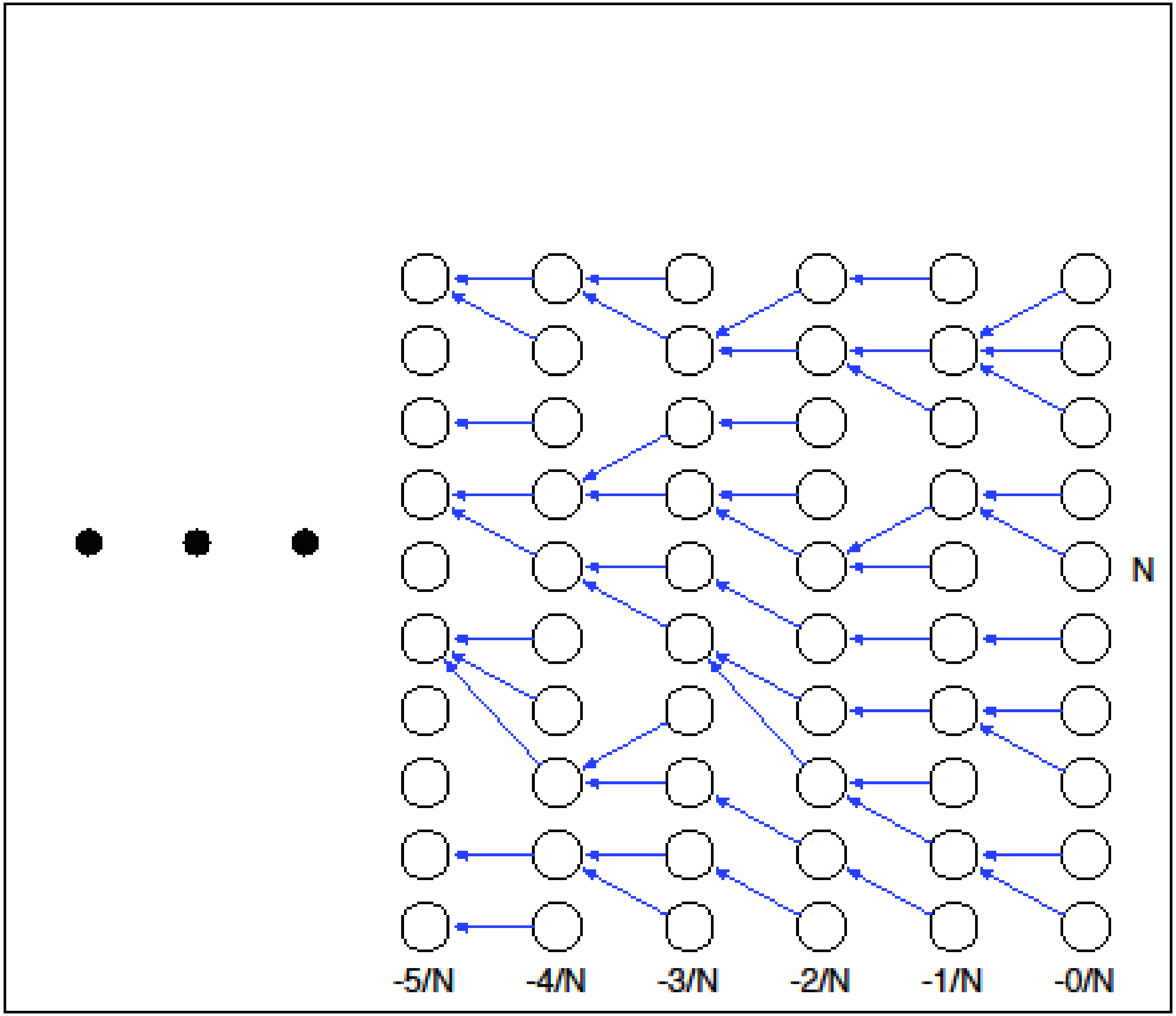
A distribution over trees



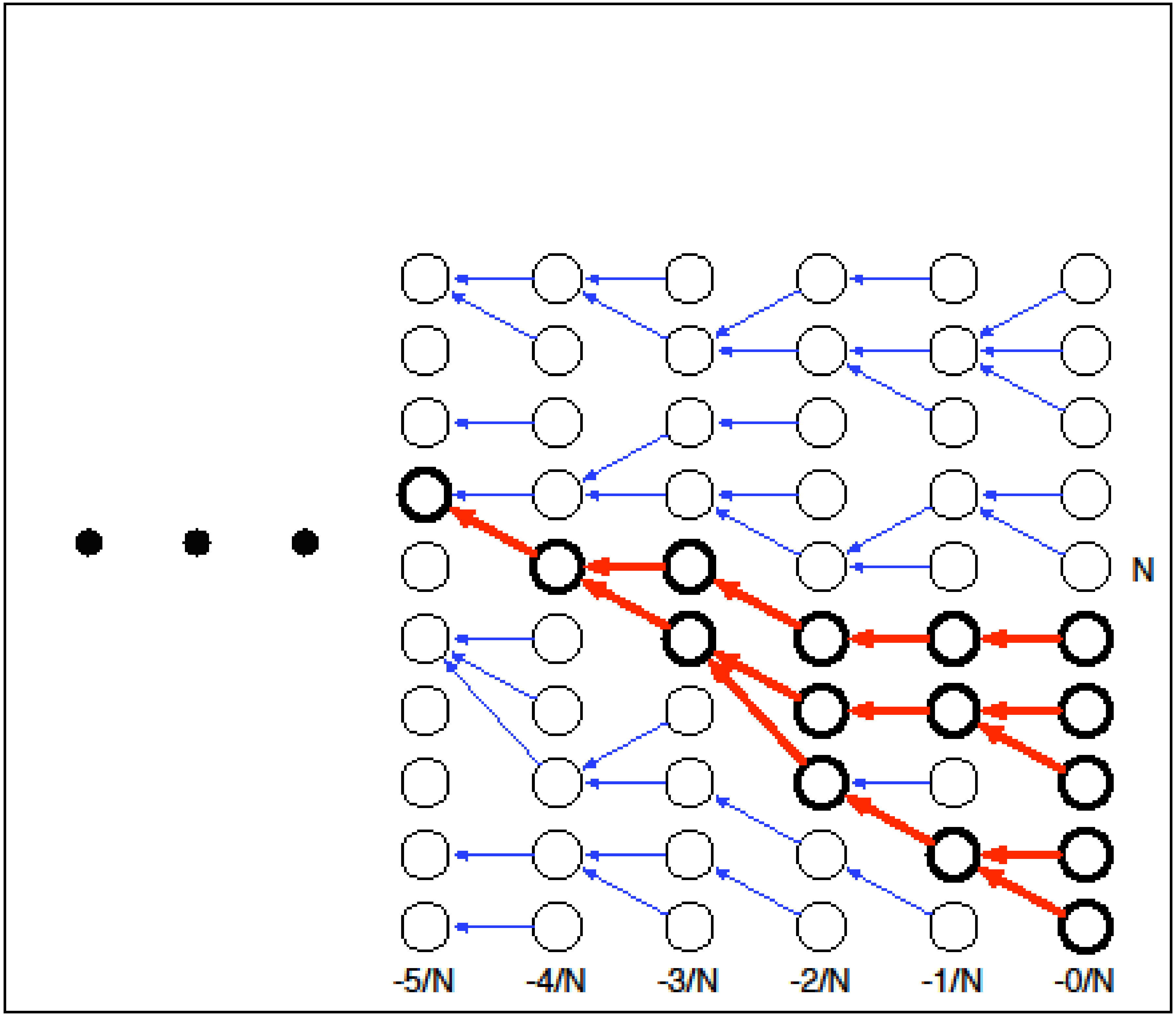
A distribution over trees



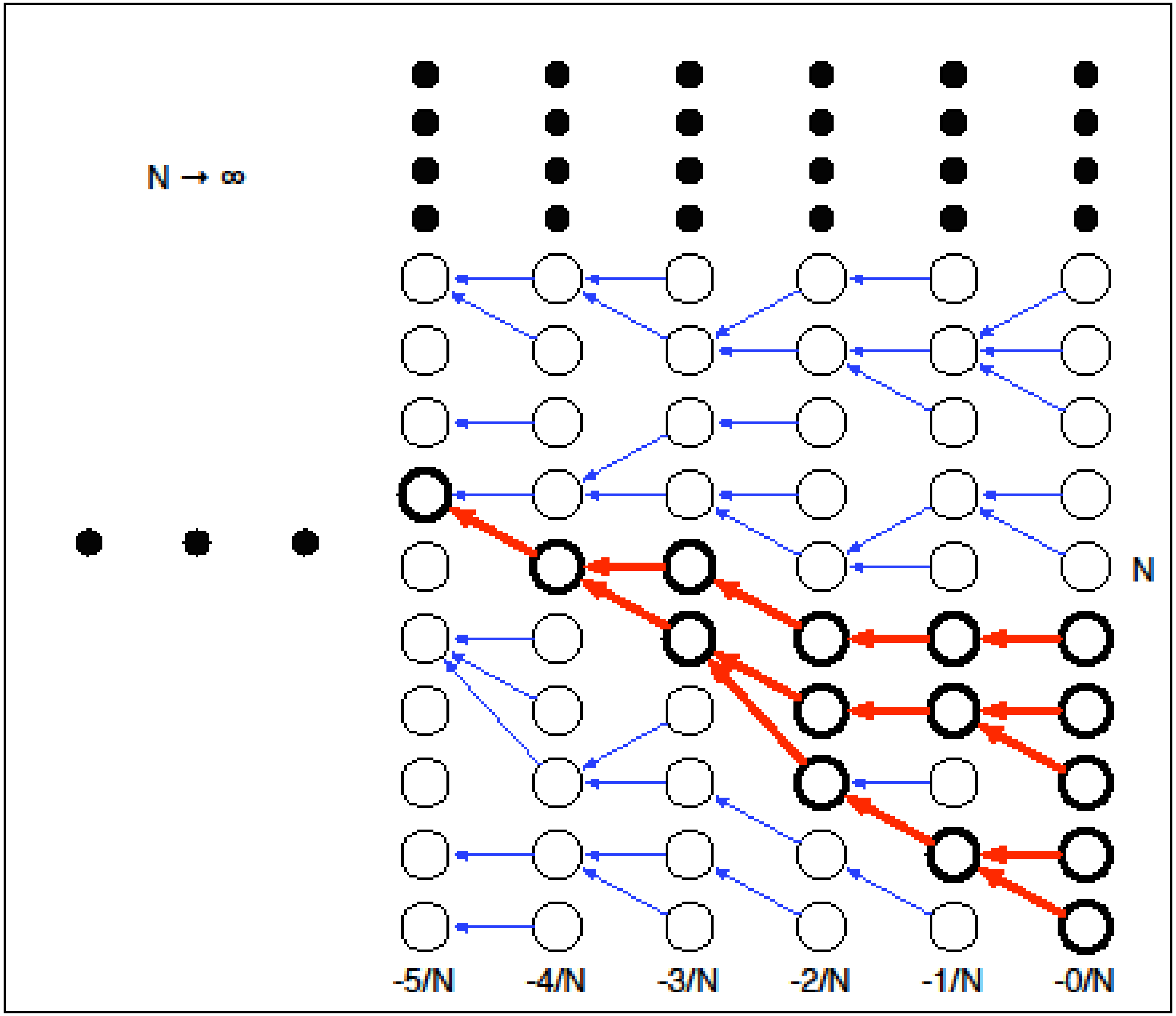
A distribution over trees



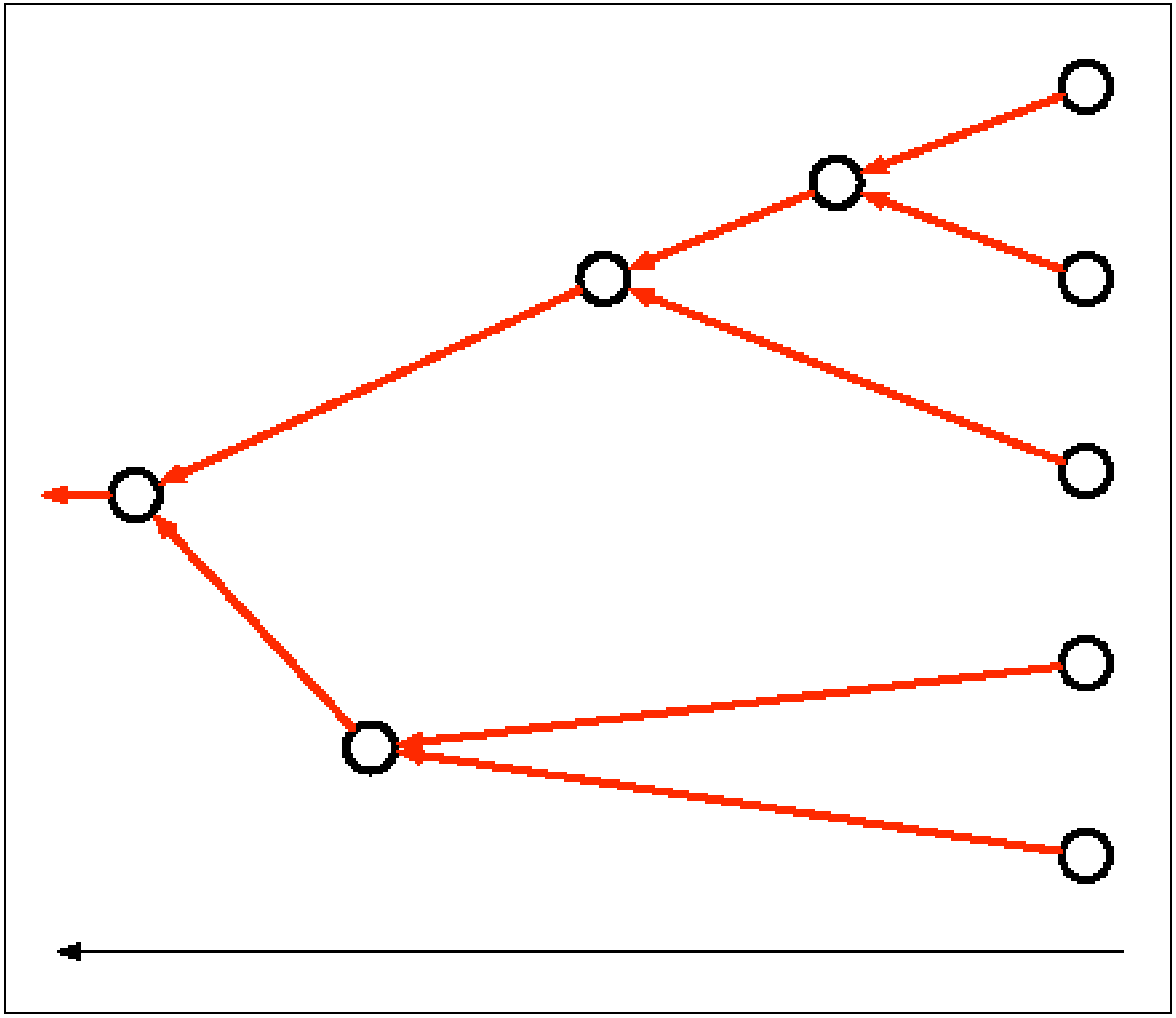
A distribution over trees



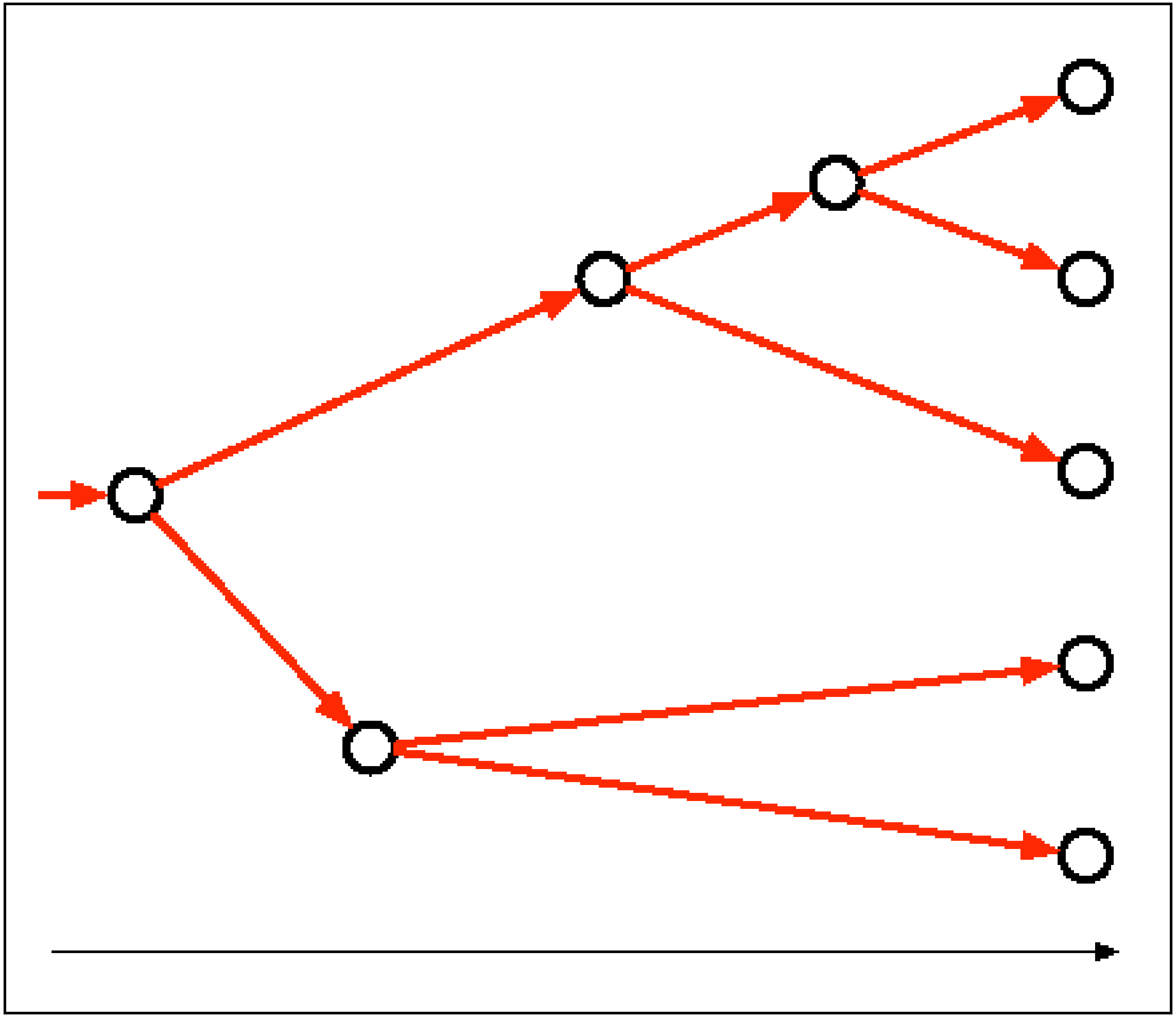
A distribution over trees



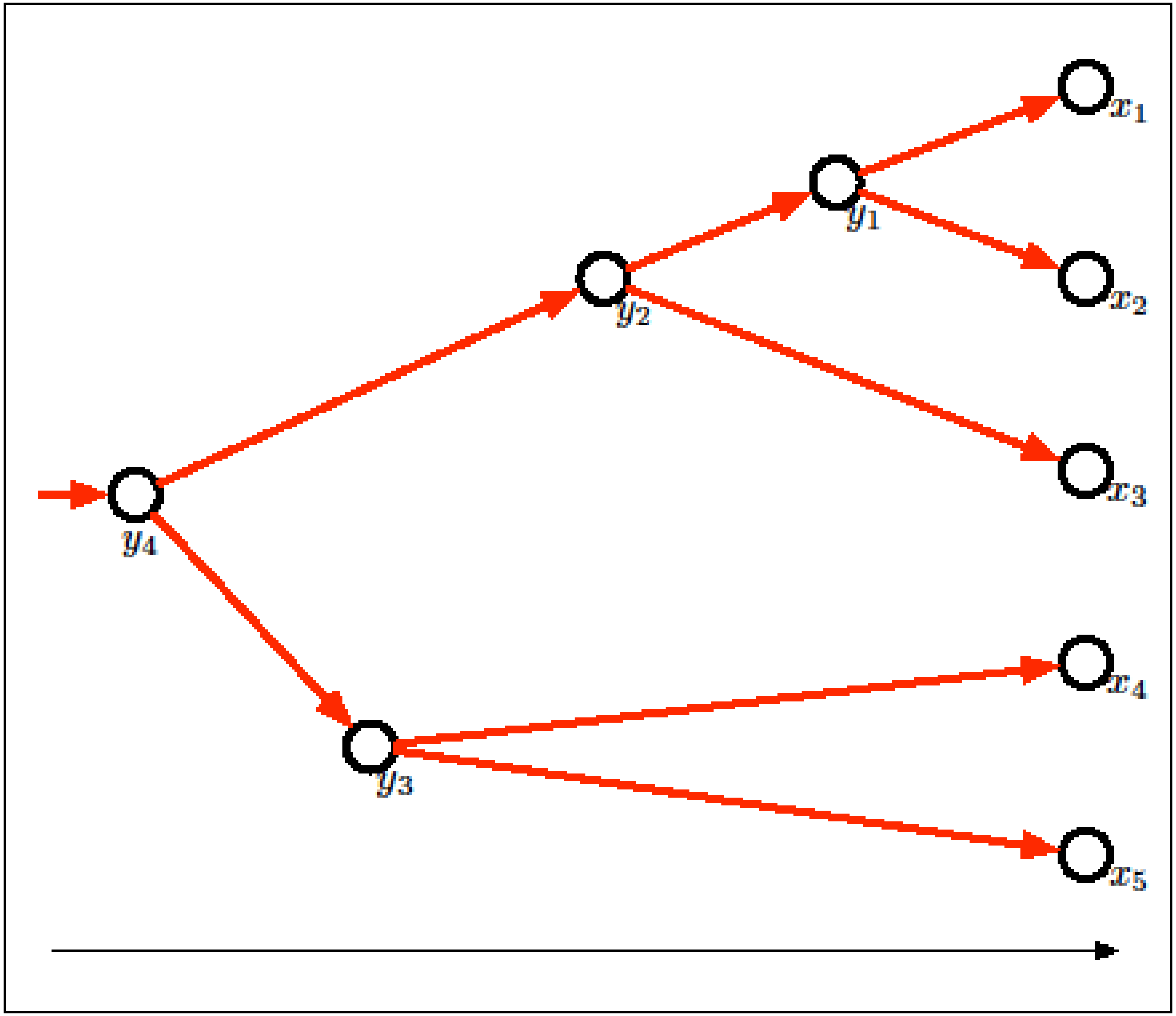
Coalescent as a graphical model



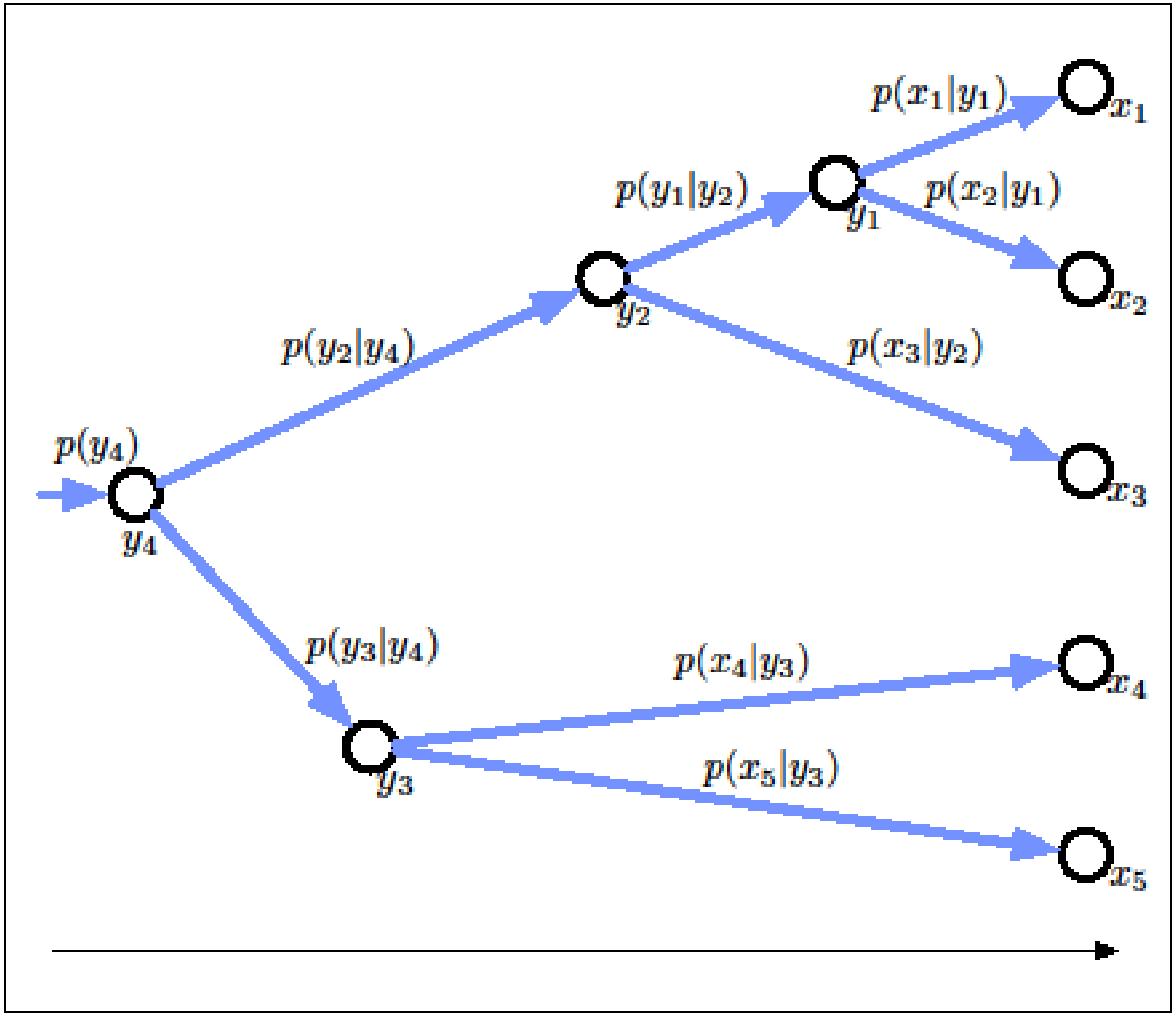
Coalescent as a graphical model



Coalescent as a graphical model

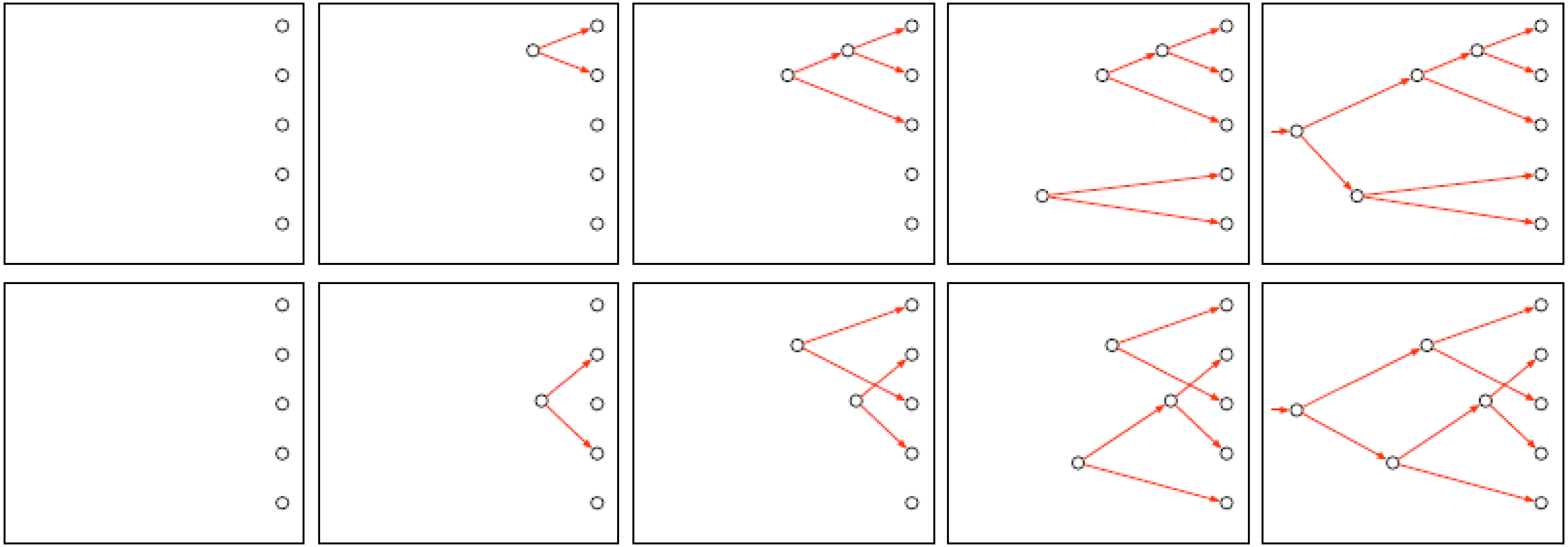


Coalescent as a graphical model



Efficient Inference

- Construct trees in a bottom-up manner



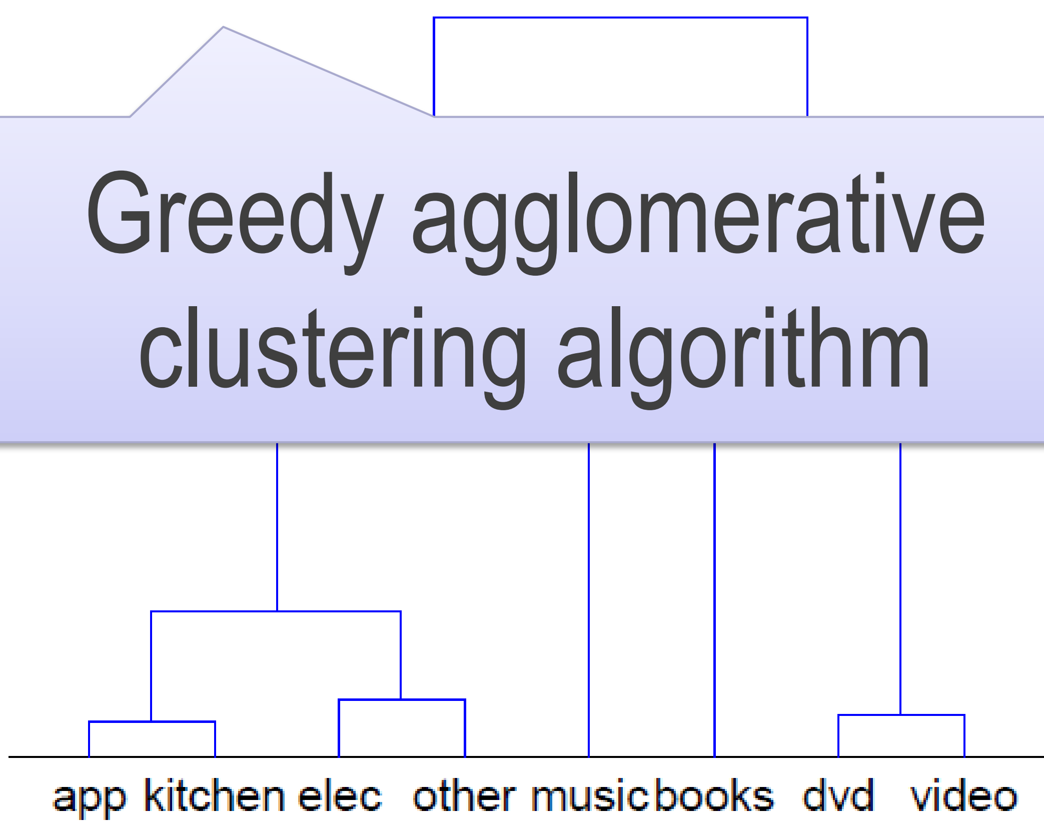
- Greedy: At each step, pick optimal pair (maximizes joint likelihood) and time to coalesce (branch length)
- Infer values of internal nodes by belief propagation

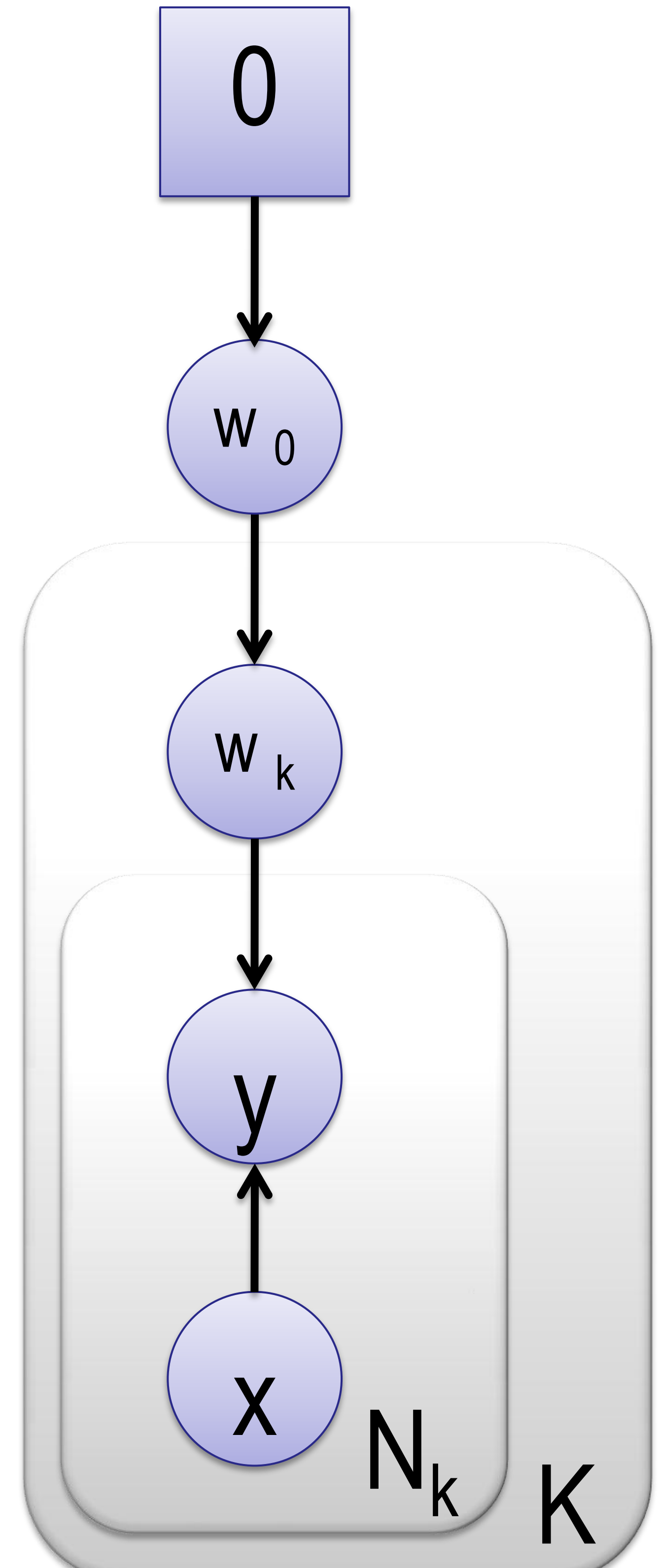
N ated equal

Message passing on coalescent tree; efficiently done by belief propagation

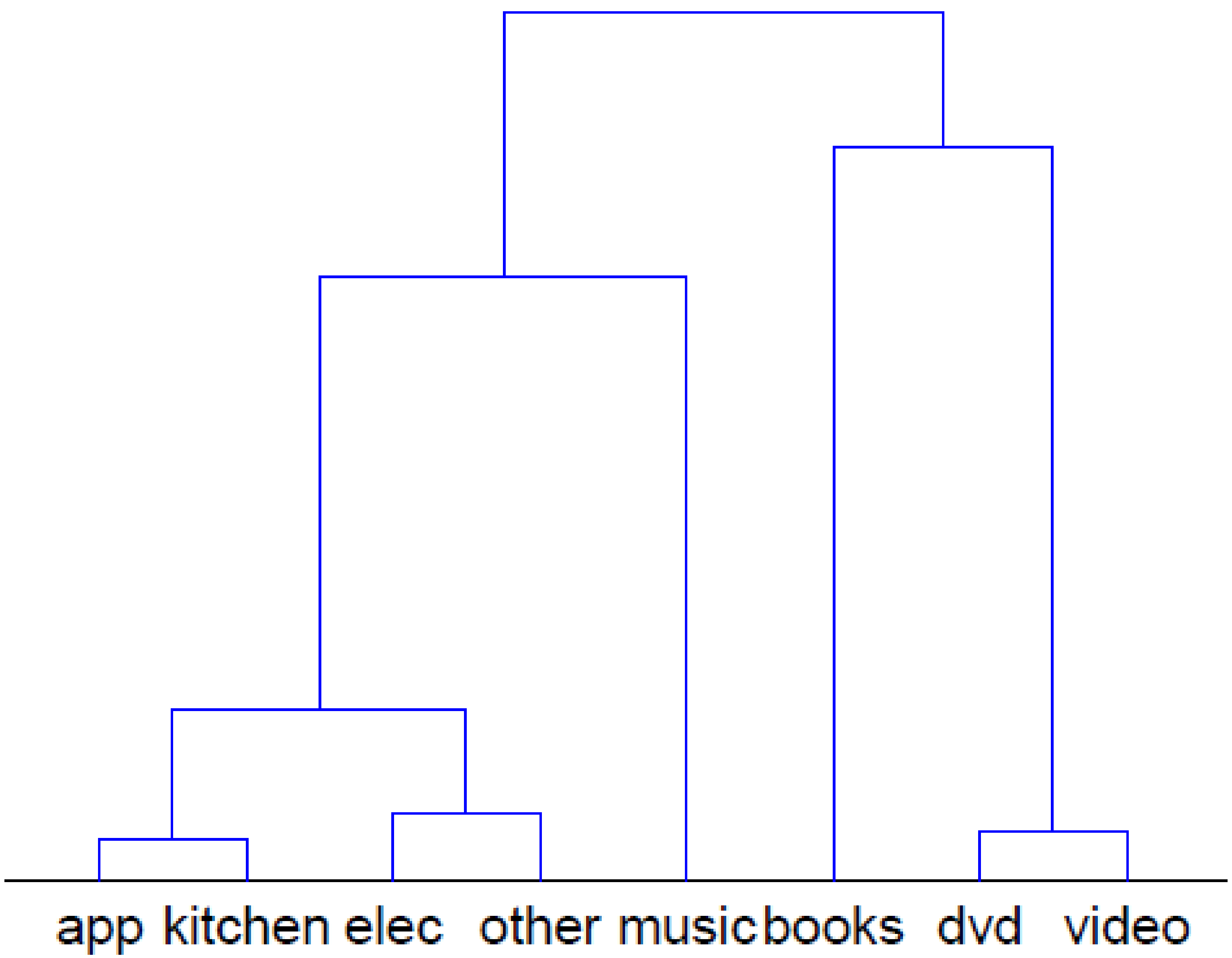
- Inferen by EM:
- E: compute expectations over weights
- M: maximize tree structure

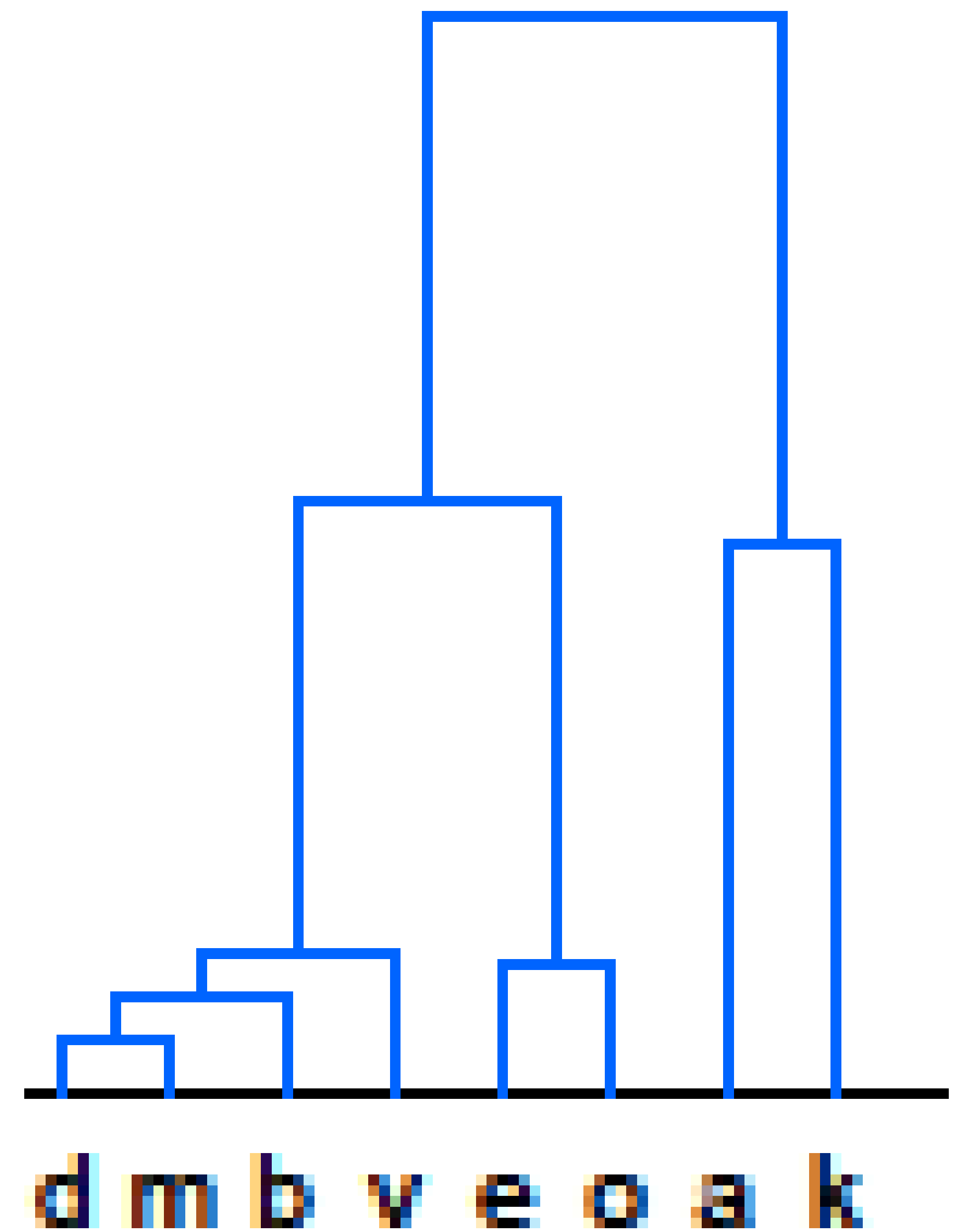
Greedy agglomerative clustering algorithm



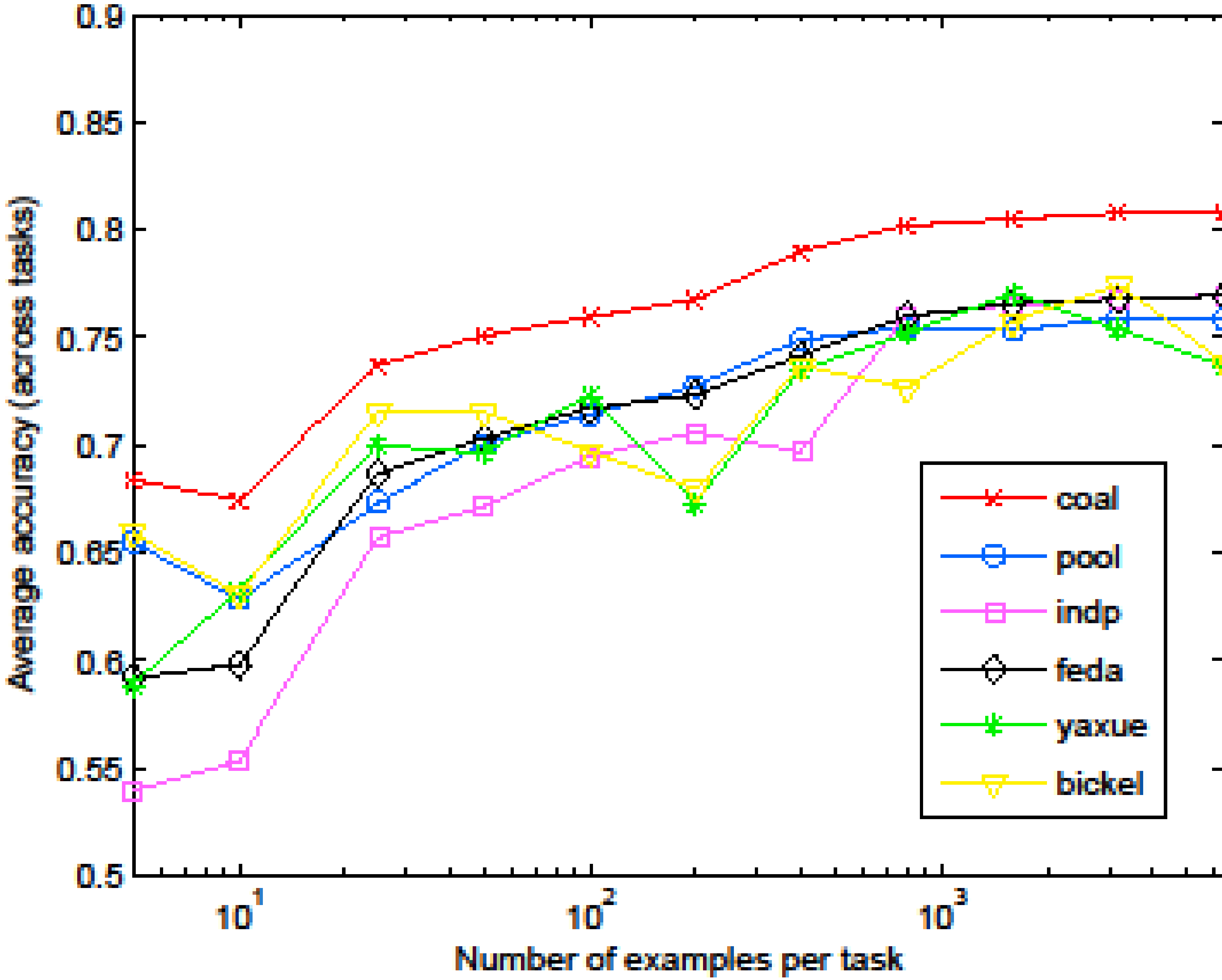


Data tree *versus* inferred tree





Some experimental results



Parameter-based References

- O. Chapelle and Z. Harchaoui. A machine learning approach to conjoint analysis. NIPS, 2005.
- K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. ICML, 2005.
- Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classication with Dirichlet process priors. JMLR, 2007.
- H. Daumé III. Bayesian Multitask Learning with Latent Hierarchies. UAI 2009.

Tutorial Outline

1. Notation and Common Concepts

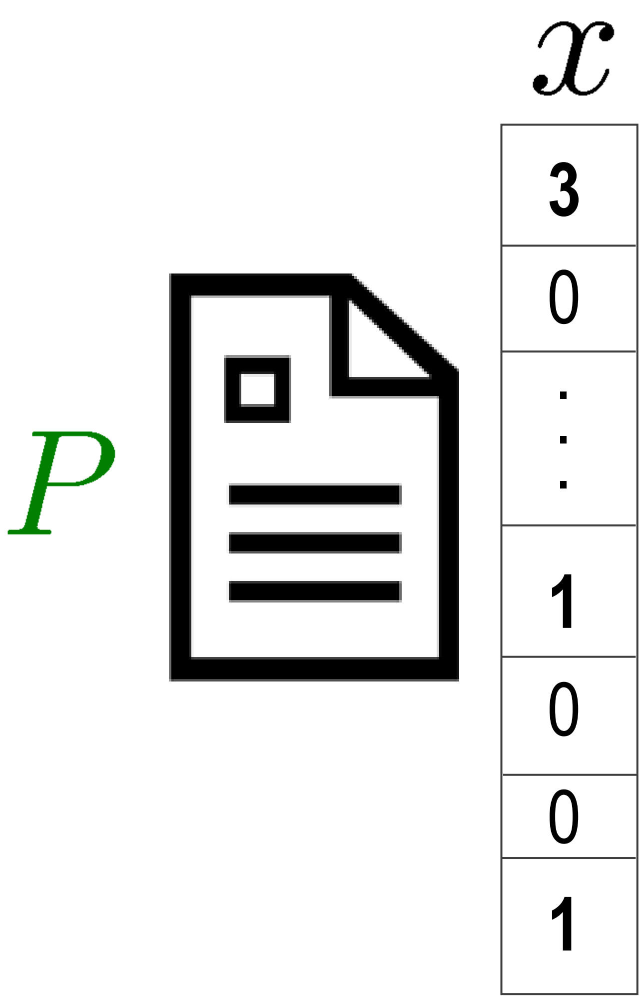
2. Semi-supervised Adaptation
   - Covariate shift
   - Learning Shared Representations

3. Supervised Adaptation
   - Feature-Based Approaches
   - Parameter-Based Approaches

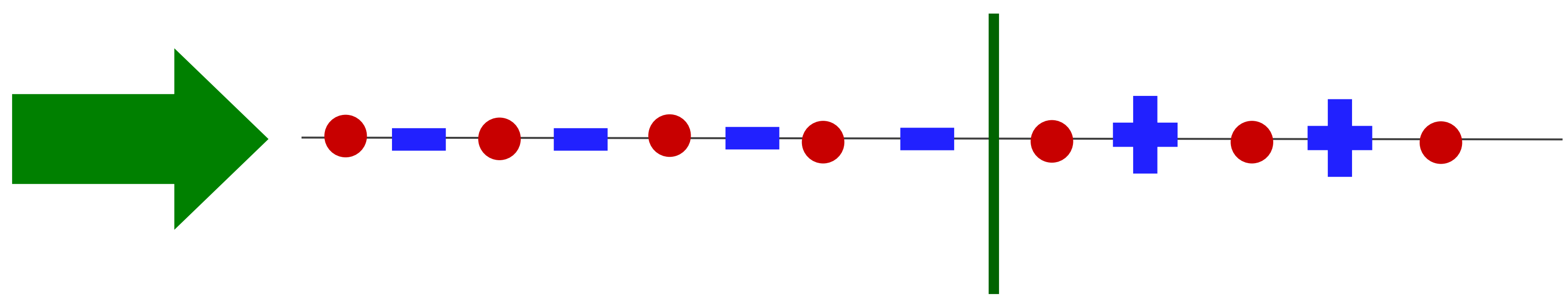
4. Open Questions and Uncovered Algorithms

Theory and Practice

Hypothesis classes from projections P : $\theta^\top P x$



$P = \hat{\theta}\hat{\theta}^\top \quad \hat{\theta} = \text{source ERM}$



1) Minimize divergence
2) $\hat{\epsilon}_S(\theta)$ small

$\cancel{\epsilon_{P,T}(\theta^*) - \epsilon_{I,T}(\theta^*) \ \text{small}}$

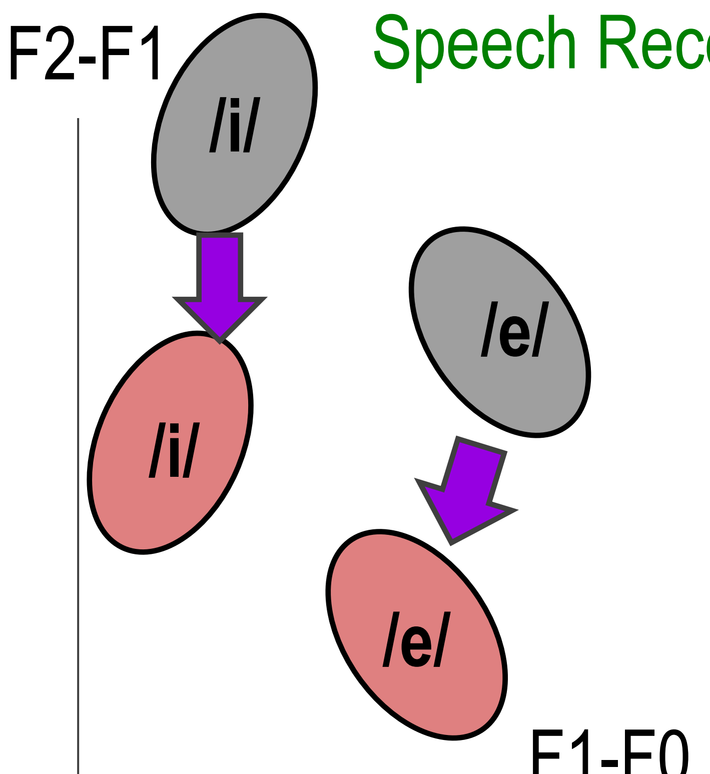
Open Questions

Matching Theory and Practice

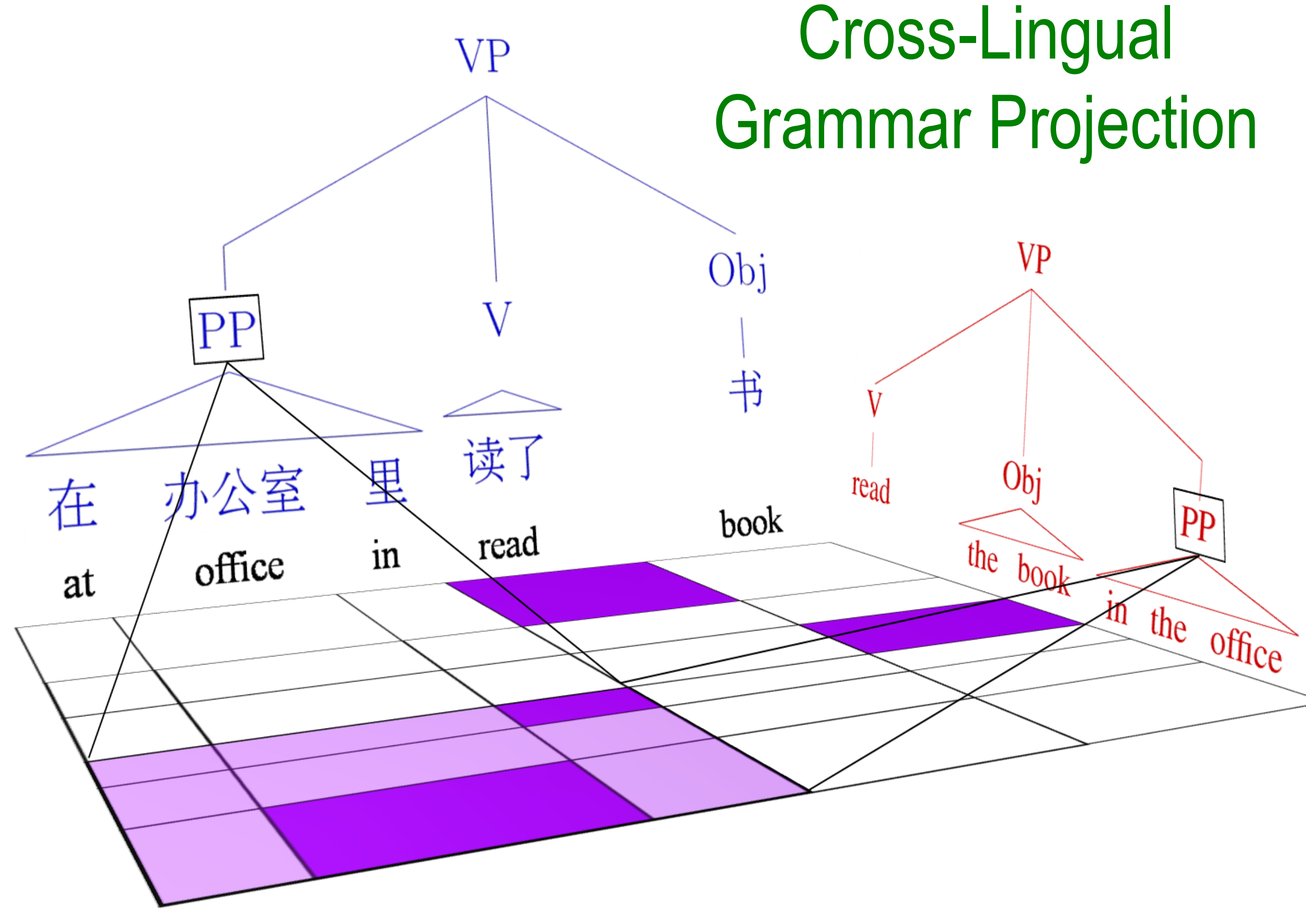
Theory does not exactly suggest what practitioners do

Prior Knowledge

Speech Recognition



Cross-Lingual Grammar Projection



More Semi-supervised Adaptation

http://adaptationtutorial.blitzer.com/references/

Self-training and Co-training

[1] D. McClosky et al. Reranking and Self-Training for Parser Adaptation. 2006.

[2] K. Sagae & J. Tsuji. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. 2007.

Structured Representation Learning

[3] F. Huang and A. Yates. Distributional Representations for Handling Sparsity in Supervised Sequence Labeling. 2009.

What is a domain anyway?

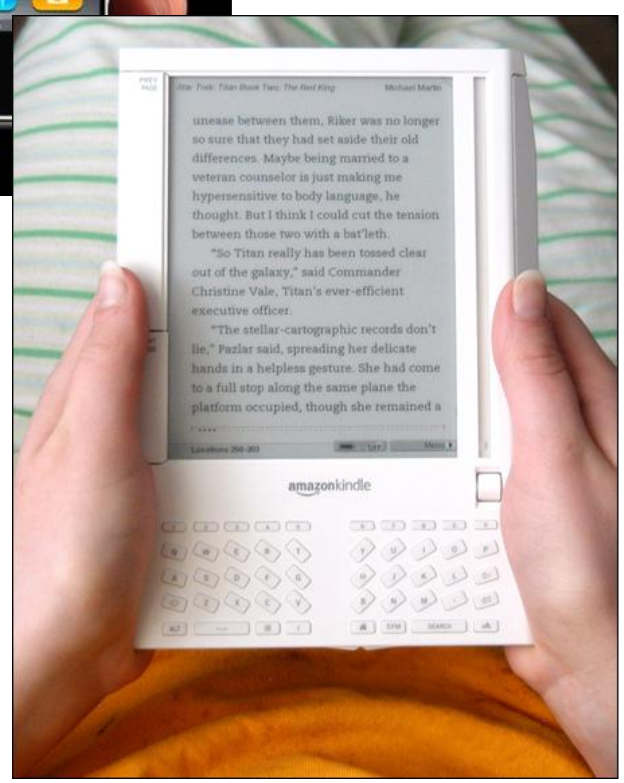
- Time?
  - News the day I was born *vs* news today?
  - News yesterday *vs* news today?
- Space?
  - News back home *vs* news in Haifa?
  - News in Tel Aviv *vs* news in Haifa?

Suggest a continuous structure

- Do my data even come with a domain specified?

Stream of <*x*,*y*,*d*> data with *y* and *d* sometimes hidden?

We’re *all* domains: personalization



- adapt learn across millions of “domains”?
- share enough information to be useful?
- share little enough information to be safe?
- avoid negative transfer?
- avoid DAAM (domain adaptation spam)?

Thanks

Questions?