Adapting Natural Language Processing Systems to New Domains

John Blitzer

Joint with: Shai Ben-David, Koby Crammer, Mark Dredze, Fernando Pereira, Alex Kulesza, Ryan McDonald, Jenn Wortman

NLP models – Single domain setting

Model estimation (training)


THE WALL STREET JOURNAL.

Cracking the Neanderthal Code

A Mystery Multiplies Over a Math Prize

What's News—

Burning Debate

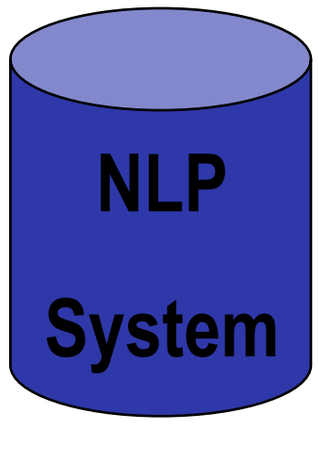
As Emission Restrictions Loom, Texas Utility Bets Big on Coal

TXU to Build 11 New Plants As Rivals Try Alternatives; Global Warming Is Concern

Power Source

Mr. Wilder Cites Demand

- data annotation
- training procedure



Syntactic analysis

Information extraction

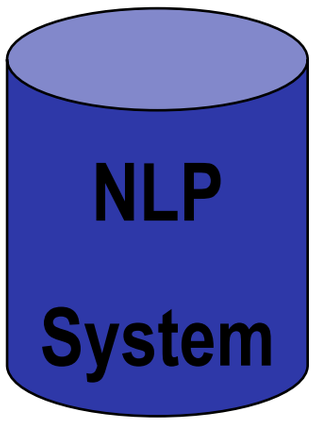
Content-based advertisement

Model application (testing)


THE WALL STREET JOURNAL.

The Iraq Connection

Was Saddam Hussein complicit in the 1995 Oklahoma City and the 1993 World Trade Center attacks? Two dossiers paint the possibility.



NLP models – Single domain setting

Training and testing samples are from the same distribution

Theoretical guarantees for large training samples

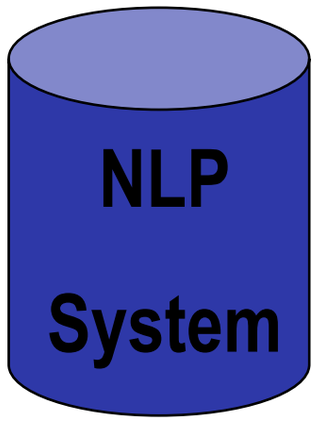
In practice, state-of-the art models have low error

NLP models, different domains

Model estimation (training)

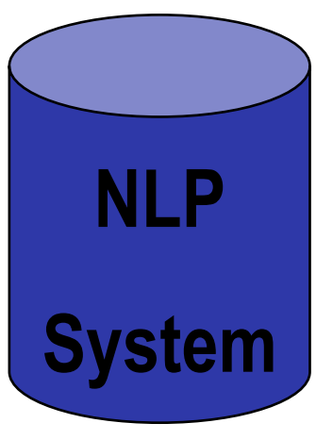
Source domain

• data annotation
• training procedure



Model application (testing)

Target domain



NLP models, different domains

When we apply models in different domains, we encounter differences in vocabulary

No theoretical guarantees for large source samples

State of the art models more than double in error

2-part talk

1. **Structural correspondence learning (SCL)**

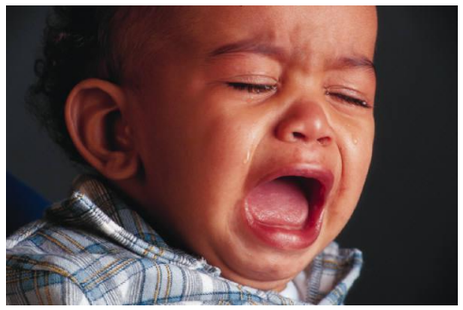
2. **A formal analysis of domain adaptation**

Sentiment classification

Product Review

Linear classifier

Positive



Negative

Multiple Domains

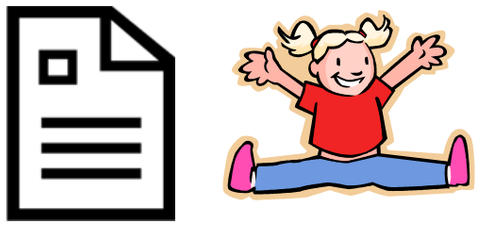
books

kitchen appliances

??

??

. . .



??

Books & kitchen appliances

Running with Scissors: A Memoir

Title: Horrible book, horrible.

This book was horrible. I read half of it, suffering from a headache the fire. One less copy in the world...don't waste your money. I wish i had the time spent reading this book back so i could use it for better purposes. This book wasted my life

Avante Deep Fryer, Chrome & Black

Title: lid does not work well...

I love the way the Tefal deep fryer 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 will not be purchasing this one again.

Error increase: 13% → 26%

SCL: 2-step learning process

Step 1: Unlabeled – Learn correspondence mapping

Unlabeled. Learn Φ

$\mathbf{x}_S$

1
0
⋮
0
3

$\mathbf{x}_T$

0
1
⋮
0
7

$\Phi(\mathbf{x})$

0.3
-1.0
⋮
0.7
-2.1

Step 2: Labeled – Learn weight vector

Labeled. Learn $\mathbf{v}$

$$\Phi(\mathbf{x}) \Rightarrow \operatorname{sgn}(\mathbf{v} \cdot \Phi(\mathbf{x}))$$

- **Φ should make the domains look as similar as possible**
- **But Φ should also allow us to classify well**

SCL: making domains look similar

Incorrect classification of kitchen review

defective lid

Unlabeled **kitchen** contexts	Unlabeled **books** contexts
• Do **not buy** the Shark portable steamer …. Trigger mechanism is **defective**.	• The book is so **repetitive** that I found myself yelling …. I will definitely **not buy** another.
• the very nice lady assured me that I must have a **defective** set …. What a **disappointment**!	• A **disappointment** …. Ender was talked about for **<#> pages** altogether.
• Maybe mine was **defective** …. The directions were **unclear**	• it's **unclear** …. It's repetitive and **boring**

SCL: pivot features

- Occur frequently in both domains
- Characterize the task we want to do
- Number in the hundreds or thousands
- Choose using labeled **source**, unlabeled **source** & **target** data

Words & bigrams that occur frequently in both domains	Frequency together with conditional entropy on labels
book one <num> so all very about they like good when	**a_must a_wonderful loved_it weak don't_waste awful highly_recommended and_easy**

SCL unlabeled step: pivot predictors

Use **pivot features** to align other features

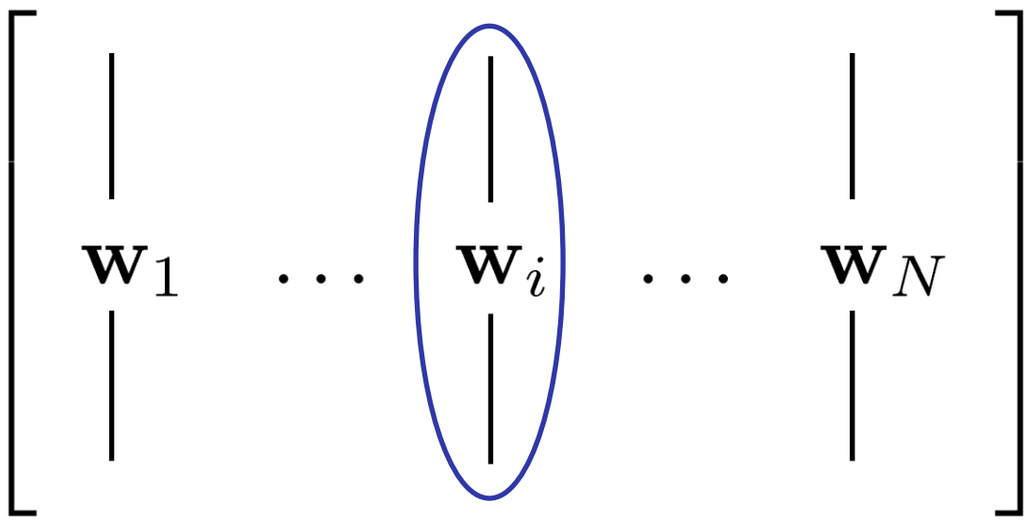
(1) The book is so **repetitive** that I found myself yelling …. I will definitely [redacted] another.

(2) Do [redacted] the Shark portable steamer …. Trigger mechanism is **defective**.

Pivot predictors implictly align source & target features

- **Mask** pivot features and predict them using other features
- N pivots → train N **linear predictors**
  - One for each binary problem
  - Let $\mathbf{w}_i$ be the weight vector for the ith predictor

SCL: dimensionality reduction



- $\mathbf{W}^T\mathbf{x}$ gives N new features

- value of i[th] feature is the propensity to see **"not buy"** in the same document

- **Many pivot predictors give similar information**
  - **"horrible", "terrible", "awful"**
- **Hard to solve optimization with N dense features per instance**
- **Compute SVD of W & use top k left singular vectors Φ**
  - **Top orthonormal principal pivot predictors**
  - **If we chose our pivots well, then $\Phi^T\mathbf{x}$ will give us good features for classification in both domains**

Back to labeled training / testing

$\mathbf{x}$

1
0
⋮
0
3

Classifier $\operatorname{sgn}\left[\mathbf{w}\cdot\mathbf{x}+\mathbf{v}\cdot\mathbf{\Phi}^{T}\mathbf{x}\right]$

- **Source** training: Learn $\mathbf{w}$ & $\mathbf{v}$ together

$\mathbf{\Phi}^{T}\mathbf{x}$

0.3
-1.0
⋮
0.7
-2.1

- **Target** testing: First apply $\mathbf{w}$, then apply $\mathbf{v}$ and $\mathbf{\Phi}$

Using labeled target data

50 instances of labeled target domain data

Source data, save weights for SCL features $\mathbf{v}_S$

Target data, regularize weights $\mathbf{v}_T$ to be close to $\mathbf{v}_S$

Chelba & Acero, EMNLP 2004

$$\min_{\mathbf{w},\mathbf{v}_T} \sum_j L(\mathbf{w}\cdot\mathbf{x}_j + \mathbf{v}_T\cdot\mathbf{x}_j, y_j) + \lambda||\mathbf{w}||^2 + \mu||\mathbf{v}_T - \mathbf{v}_S||^2$$

Huberized hinge loss

Keep SCL weights close to source weights

Avoid using high-dimensional features

Inspirations for SCL

1. Alternating Structural Optimization (ASO)

- **Ando & Zhang** (JMLR 2005)
- Training predictors using unlabeled data

2. Correspondence Dimensionality Reduction

- **Ham, Lee, & Saul** (AISTATS 2003)
- Learn a low-dimensional representation from high-dimensional correspondences

Sentiment classification data

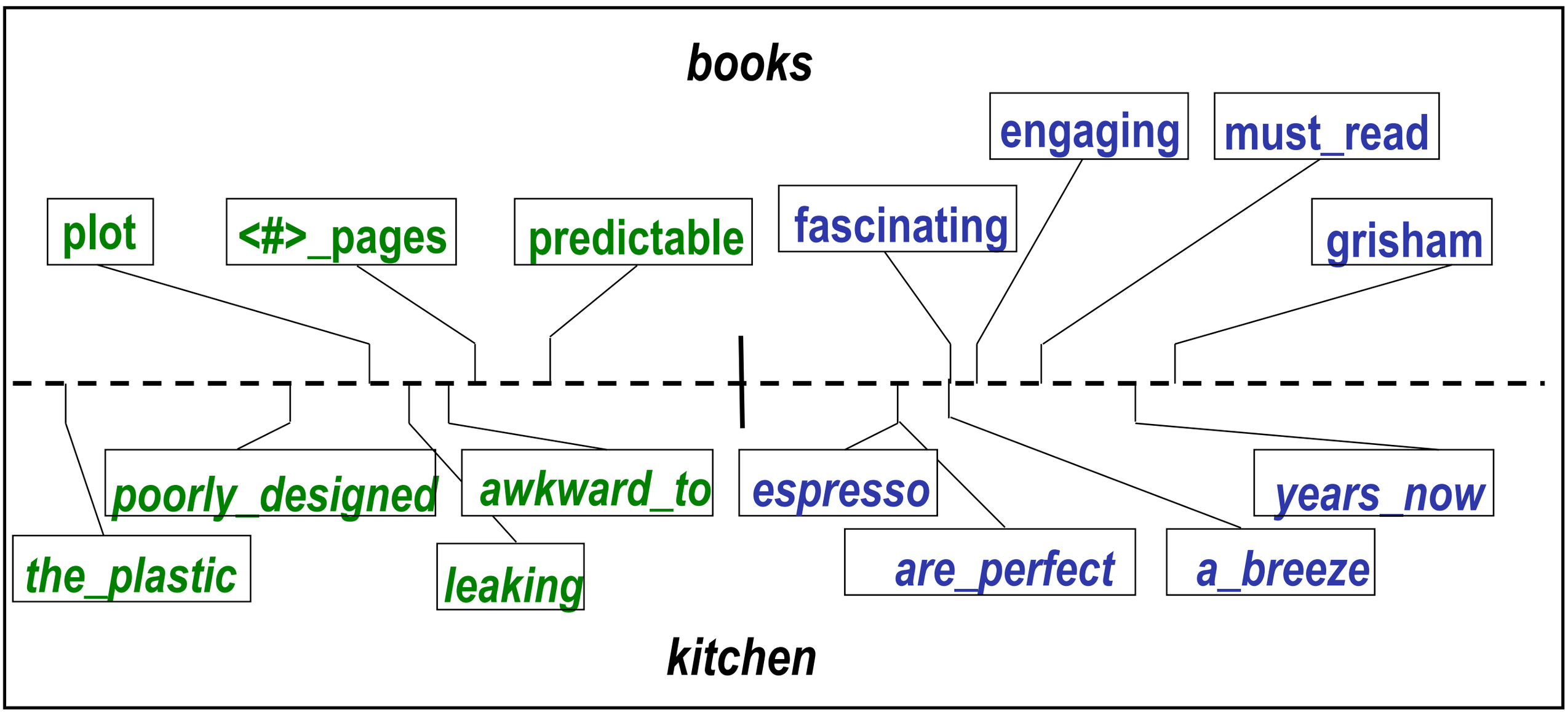
- **Product reviews from Amazon.com**
  - Books, DVDs, Kitchen Appliances, Electronics
  - 2000 labeled reviews from each domain
  - 3000 – 6000 unlabeled reviews

- **Binary classification problem**
  - Positive if 4 stars or more, negative if 2 or fewer

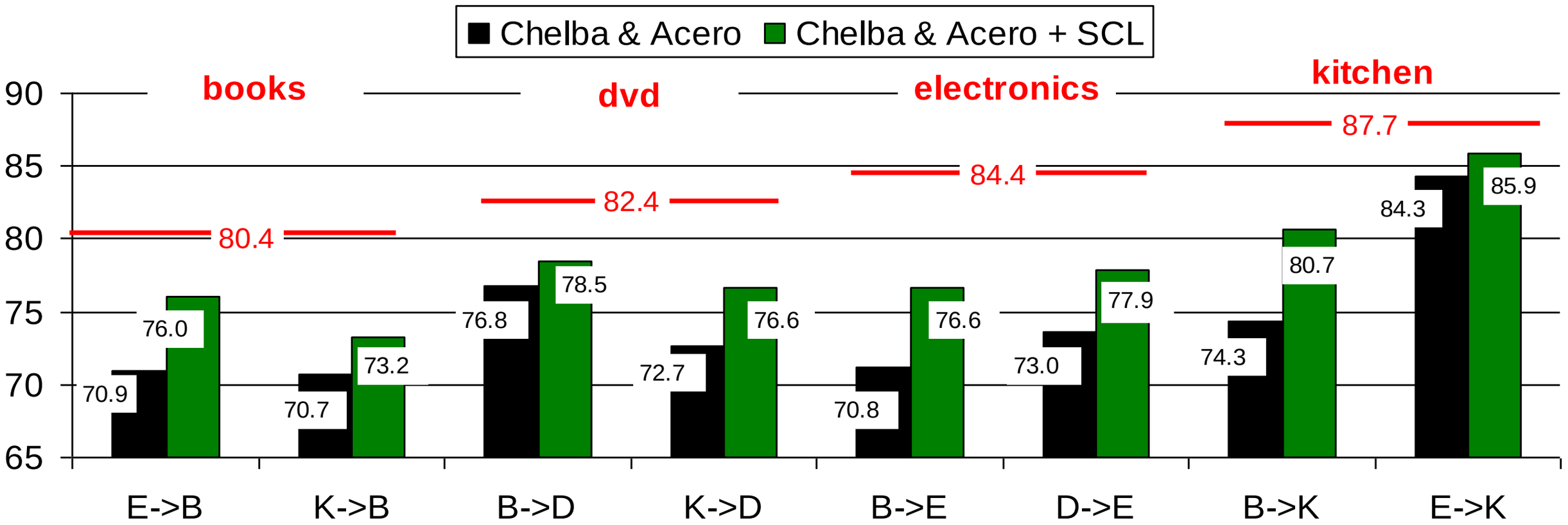
- **Features:** unigrams & bigrams

Visualizing Φ (books & kitchen)

negative **vs.** **positive**



Results: 50 labeled target instances



- With 50 labeled target instances, SCL **always** improves over baseline.
- Overall relative reduction is 36% relative

Theoretical Analysis: Using labeled data from multiple domains

Study the tradeoff between accurate but scarce target data and plentiful but biased source data

Analyze algorithms which minimize convex combinations of source & target risk

Give a generalization bound that is computable from finite labeled & unlabeled samples

Relating source & target error

A basic bound:

Let h be a binary hypothesis from class $\mathcal{H}$ and $\mathcal{D}_S, \mathcal{D}_T$ be source and target distributions. Then

$$\epsilon_{\mathcal{D}_T}(h) \leq \epsilon_{\mathcal{D}_S}(h) + \mathrm{dist}(\mathcal{D}_S, \mathcal{D}_T)$$

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \quad + \quad \lambda$$

- **Measureable from finite unlabeled samples**
- **Related to hypothesis class $\mathcal{H}$**

- **Not measurable from unlabeled samples**
- **Small for realistic NLP problems**

The $\mathcal{H}\Delta\mathcal{H}$ distance

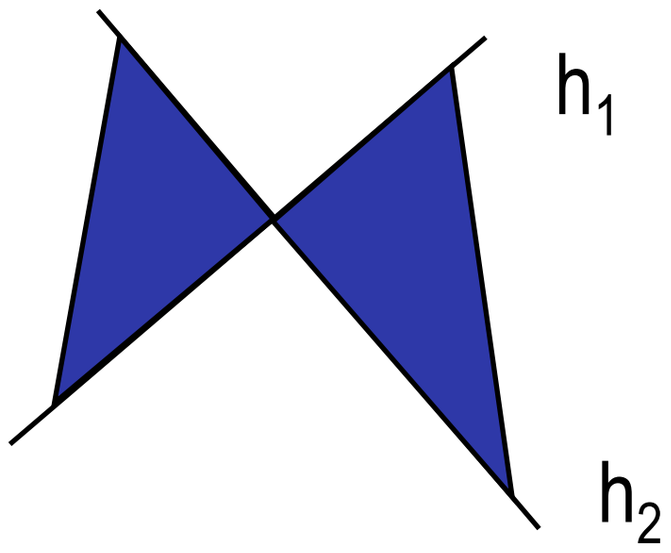
Idea: Measure subsets where hypotheses in $\mathcal{H}$ disagree

Let $\mathcal{H}$ be a hypothesis class. Denote by $\mathcal{H}\Delta\mathcal{H}$ the set of subsets of $\mathcal{X}$ where two hypotheses in $\mathcal{H}$ disagree.

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2 \sup_{A \in \mathcal{H}\Delta\mathcal{H}} \left| \int_A p_T(\mathbf{x}) - p_S(\mathbf{x}) d\mathbf{x} \right|$$

Subsets A are **symmetric differences** of two hypotheses

Where does h_1 make errors with respect to h_2?



The $\mathcal{H}\Delta\mathcal{H}$ distance

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2 \sup_{A \in \mathcal{H}\Delta\mathcal{H}} \left| \int_A p_T(\mathbf{x}) - p_S(\mathbf{x}) d\mathbf{x} \right|$$

1. Always lower than L_1

2. Computable from finite **unlabeled** samples.

3. Easy to compute: train classifier to discriminate between source and target instances

For unlabeled samples $\mathcal{U}_S, U_T$, we write $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$

Domain Adaptation Assumption

There exists some h^* which performs well on both domains

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \, \epsilon_{\mathcal{D}_S}(h) + \epsilon_{\mathcal{D}_T}(h)$$

$$\lambda = \epsilon_{\mathcal{D}_S}(h^*) + \epsilon_{\mathcal{D}_T}(h^*)$$

λ must be small in order to learn from only source labeled data

Combining source & target labeled data

The α-risk: $\epsilon_\alpha(h) = \alpha\epsilon_{D_T}(h) + (1-\alpha)\epsilon_{D_S}(h)$

We investigate algorithms which minimize the empirical α-risk

Let h be a binary hypothesis. Then

$$|\epsilon_\alpha(h) - \epsilon_{\mathcal{D}_T}(h)| \leq (1-\alpha)\left(d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda\right)$$

A bound on the target risk

Let $\hat{h}_\alpha$ and h^*_T indicate the empirical α-risk and labeled target and source examples, respectively.

Tradeoff: Complexity term increases as α moves away from β
Divergence term increases as α moves away from 1

$$\epsilon_{\mathcal{D}_T}(\hat{h}_\alpha) \leq \epsilon_{\mathcal{D}_T}(h^*_T) + 2(1-\alpha)\left(d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda\right)$$
$$+ 2\sqrt{\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}} \times \tilde{O}\left(\sqrt{\frac{d}{m}}\right)$$

Computing the terms in the bound

$$\epsilon_{\mathcal{D}_T}(\hat{h}_\alpha) \leq \epsilon_{\mathcal{D}_T}(h_T^*) + 2(1-\alpha)\left(d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda\right) + 2\sqrt{\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}} \times \tilde{O}\left(\sqrt{\frac{d}{m}}\right)$$

- **Given as input**
- **Computable from unlabeled data**
- **Assumed to be small**

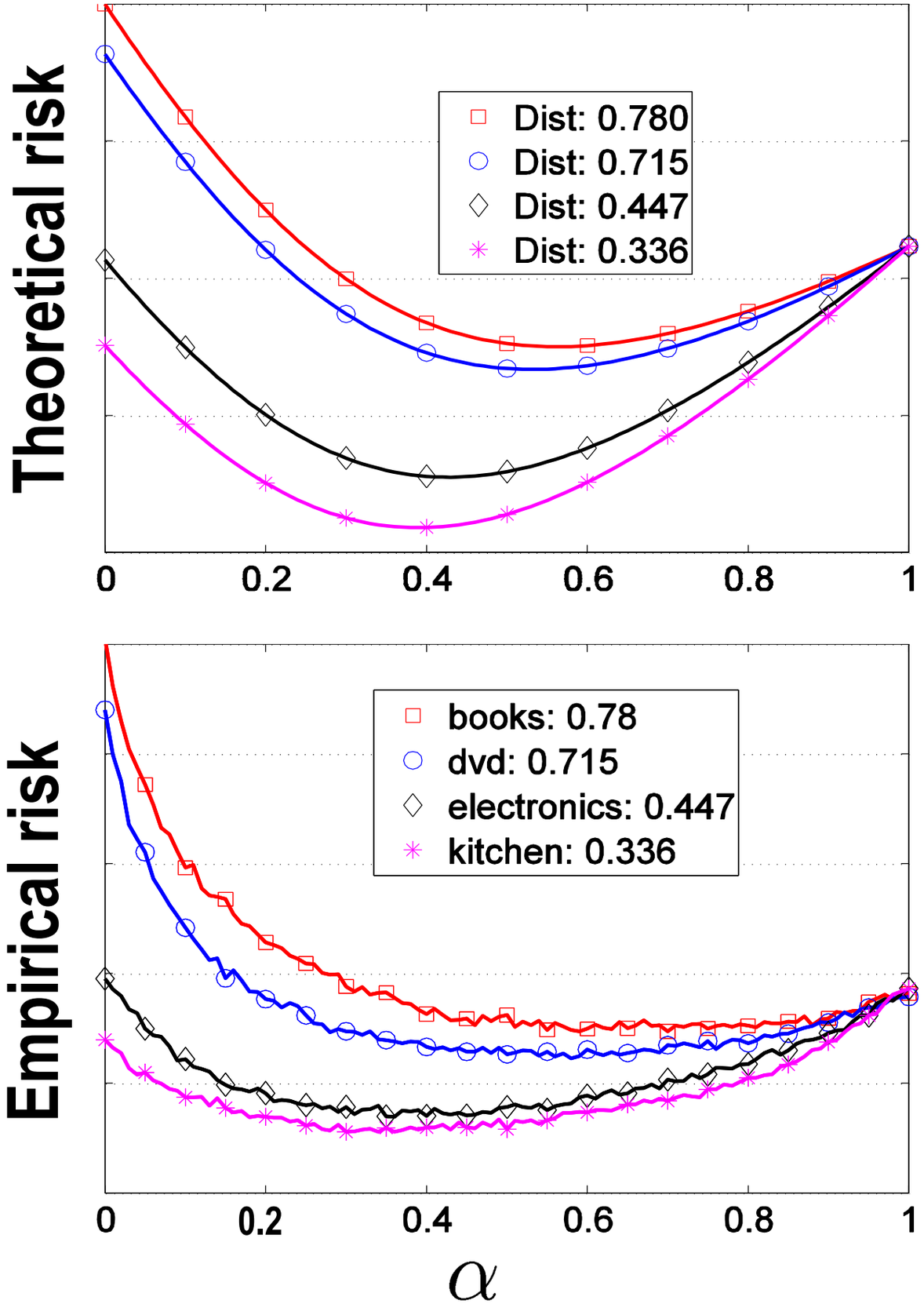
Evaluating the bound: parameters

- **Look at the shape of the bound vs. empirical error for different values of α**

- **Vary input parameters:**
  1. **distance between source and target**
  2. **amount of source data**

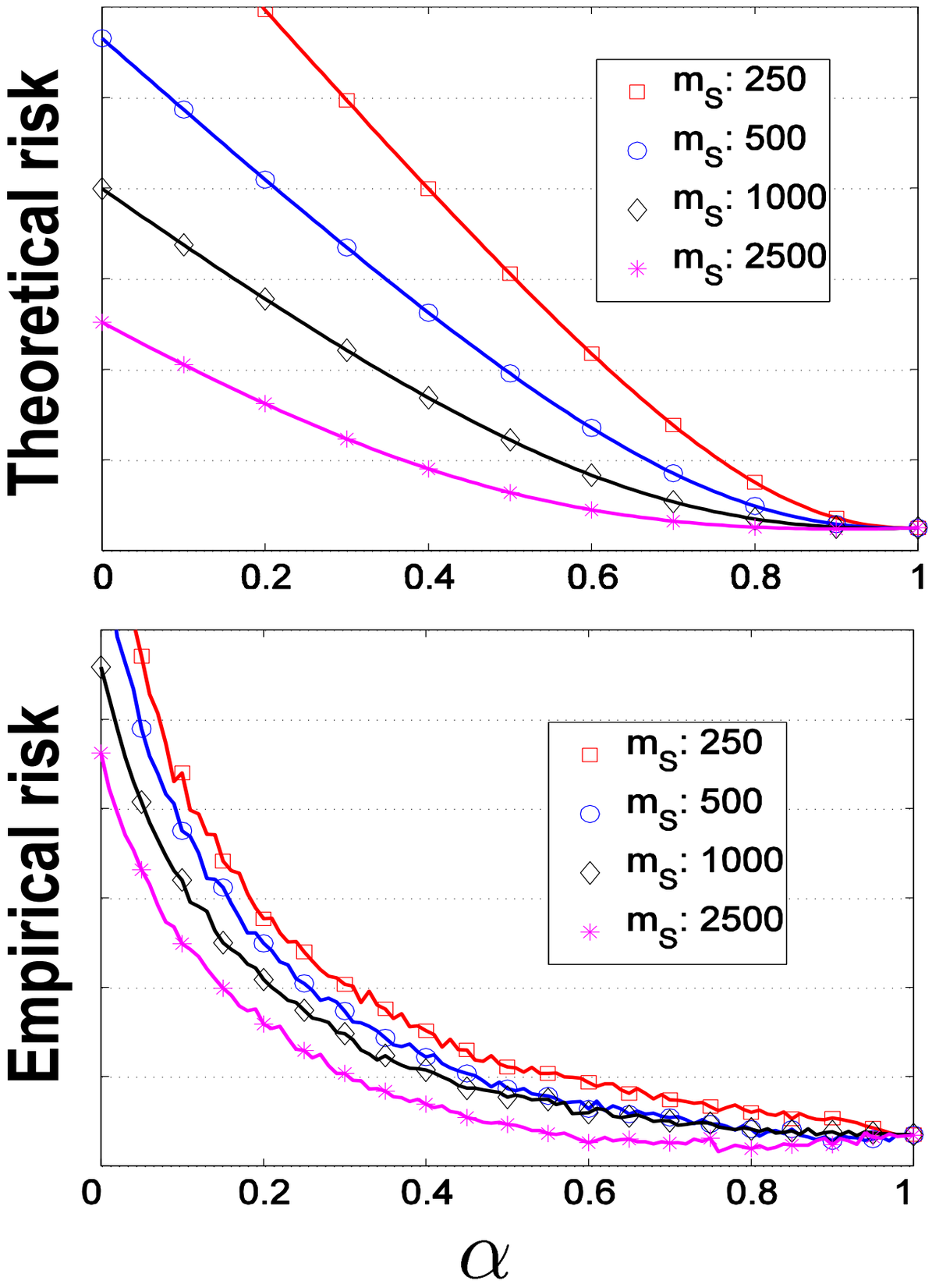
Increasing m_S corresponds to increasing m and decreasing β

Vary distance, $m_S = 2500$, $m_T = 1000$



- Same relative ordering: higher distance means higher risk
- Same convex shape: error-minimizing alpha reflects distance
- Very different actual numbers: empirical error much lower

Vary source size, $m_T = 2500$, dist $= 0.715$



- Same relative ordering: more source data is better

- With enough target data, it's always better to set α=1, regardless of amount of source data

University of California
C A L
N L P
Berkeley

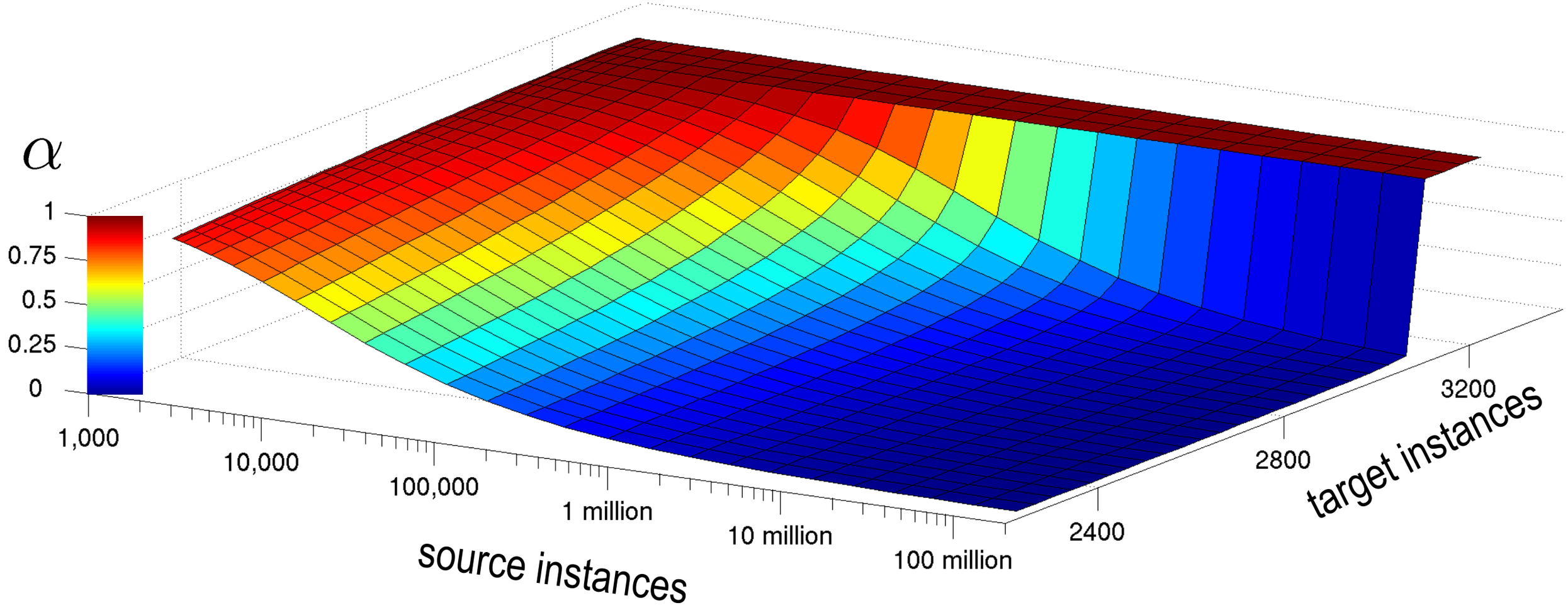
A phase transition in the optimal α

After a fixed amount of target data, adding source data is not helpful

Plot optimal α for varying amounts of source and target data

With 3130 or more target instances, the optimal α is **always** 1

$$d_{H\Delta\mathcal{H}} = 0.715$$



Domain Adaptation: Theory and practice

- Our theory shows that decreasing $d_{\mathcal{H}\Delta\mathcal{H}}$ can lead to a decrease in error due to adaptation
  - But we have no theory that suggests an algorithm for using unlabeled data in domain adaptation

- What if we have many source domains
  - There exists a kind of hierarchical structure on sources
  - Can we design an algorithm which has low regret with respect to the best model from each one?

Thanks

Thanks