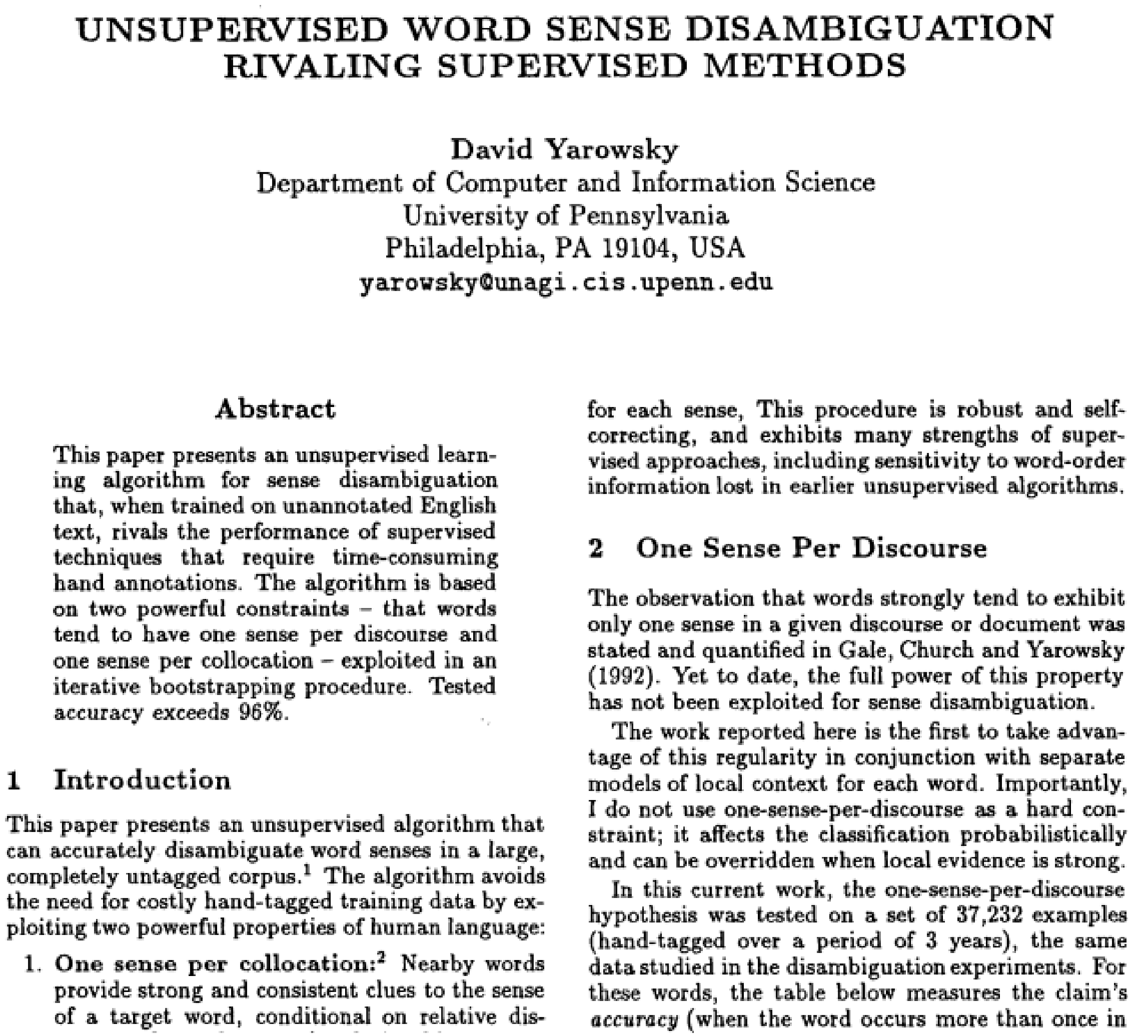
# Label Propagation

Minoo Jafarlou

# Lecture Outline

- Darvid Yarowskey, 1995, influential solution for unsupervised word sense disambiguation
- Label Propagation and Semi-Supervised Learning
- Yarowskey Algorithm influences other machine learning realms

# Yarowsky algorithm as an initiation



## UNSUPERVISED WORD SENSE DISAMBIGUATION RIVALING SUPERVISED METHODS

David Yarowsky
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
yarowsky@unagi.cis.upenn.edu

### Abstract

This paper presents an unsupervised learning algorithm for sense disambiguation that, when trained on unannotated English text, rivals the performance of supervised techniques that require time-consuming hand annotations. The algorithm is based on two powerful constraints – that words tend to have one sense per discourse and one sense per collocation – exploited in an iterative bootstrapping procedure. Tested accuracy exceeds 96%.

### 1 Introduction

This paper presents an unsupervised algorithm that can accurately disambiguate word senses in a large, completely untagged corpus.[1] The algorithm avoids the need for costly hand-tagged training data by exploiting two powerful properties of human language:

1. **One sense per collocation:**[2] Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative dis-

for each sense, This procedure is robust and self-correcting, and exhibits many strengths of supervised approaches, including sensitivity to word-order information lost in earlier unsupervised algorithms.

### 2 One Sense Per Discourse

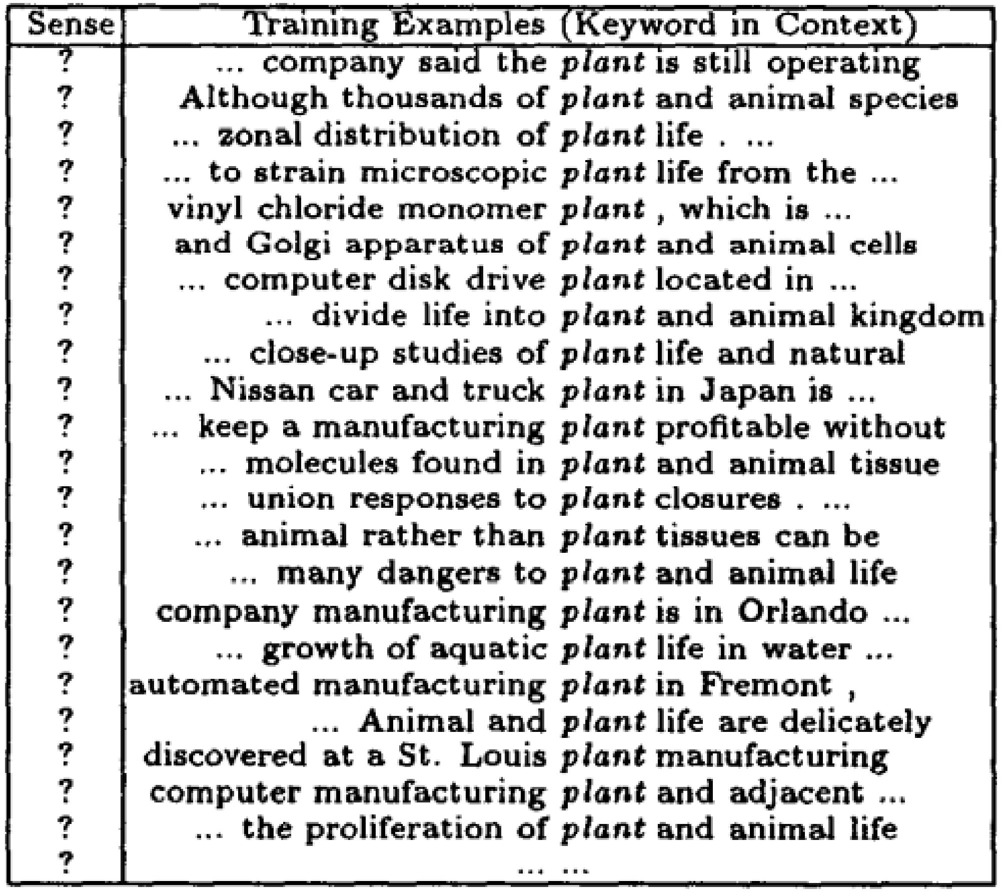
The observation that words strongly tend to exhibit only one sense in a given discourse or document was stated and quantified in Gale, Church and Yarowsky (1992). Yet to date, the full power of this property has not been exploited for sense disambiguation.

The work reported here is the first to take advantage of this regularity in conjunction with separate models of local context for each word. Importantly, I do not use one-sense-per-discourse as a hard constraint; it affects the classification probabilistically and can be overridden when local evidence is strong.

In this current work, the one-sense-per-discourse hypothesis was tested on a set of 37,232 examples (hand-tagged over a period of 3 years), the same data studied in the disambiguation experiments. For these words, the table below measures the claim's *accuracy* (when the word occurs more than once in

# Yarowsky Algorithm

- Step 1: a large corpus, identify all examples of the given polysemous word, storing their contexts as lines in an initially untagged training set



| Sense | Training Examples (Keyword in Context) |
|---|---|
| ? | ... company said the *plant* is still operating |
| ? | Although thousands of *plant* and animal species |
| ? | ... zonal distribution of *plant* life . ... |
| ? | ... to strain microscopic *plant* life from the ... |
| ? | vinyl chloride monomer *plant* , which is ... |
| ? | and Golgi apparatus of *plant* and animal cells |
| ? | ... computer disk drive *plant* located in ... |
| ? | ... divide life into *plant* and animal kingdom |
| ? | ... close-up studies of *plant* life and natural |
| ? | ... Nissan car and truck *plant* in Japan is ... |
| ? | ... keep a manufacturing *plant* profitable without |
| ? | ... molecules found in *plant* and animal tissue |
| ? | ... union responses to *plant* closures . ... |
| ? | ... animal rather than *plant* tissues can be |
| ? | ... many dangers to *plant* and animal life |
| ? | company manufacturing *plant* is in Orlando ... |
| ? | ... growth of aquatic *plant* life in water ... |
| ? | automated manufacturing *plant* in Fremont , |
| ? | ... Animal and *plant* life are delicately |
| ? | discovered at a St. Louis *plant* manufacturing |
| ? | computer manufacturing *plant* and adjacent ... |
| ? | ... the proliferation of *plant* and animal life |
| ? | ... ... |

# Yarowsky Algorithm

- Step 2: For each possible sense of the word, identify a relatively small number of training examples representative of that sense.

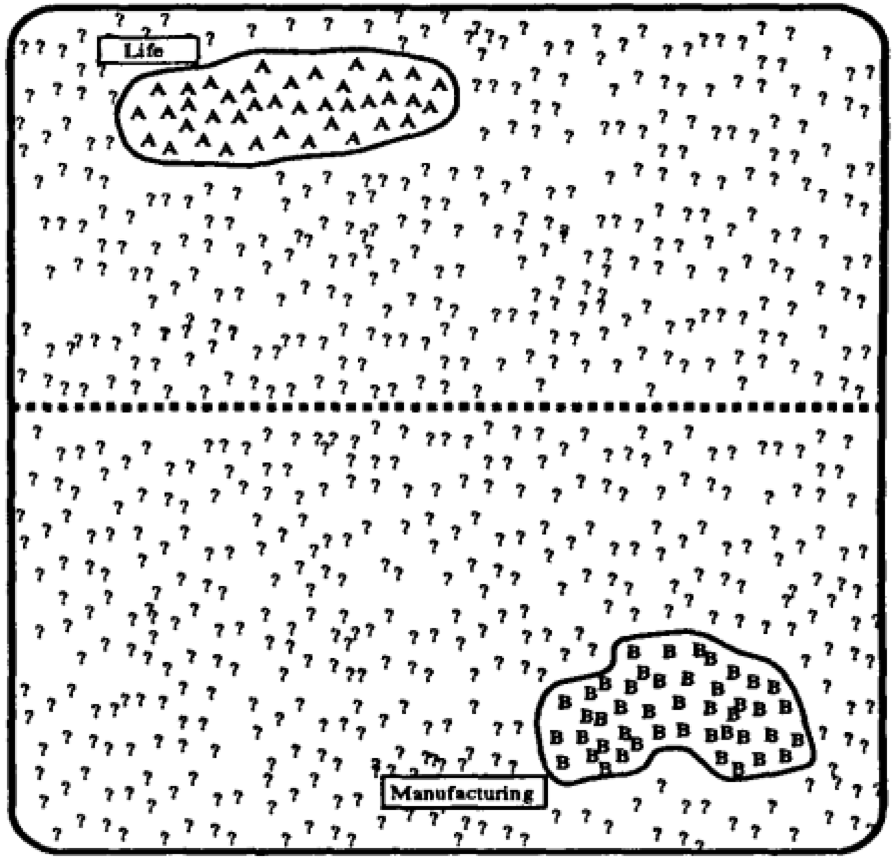
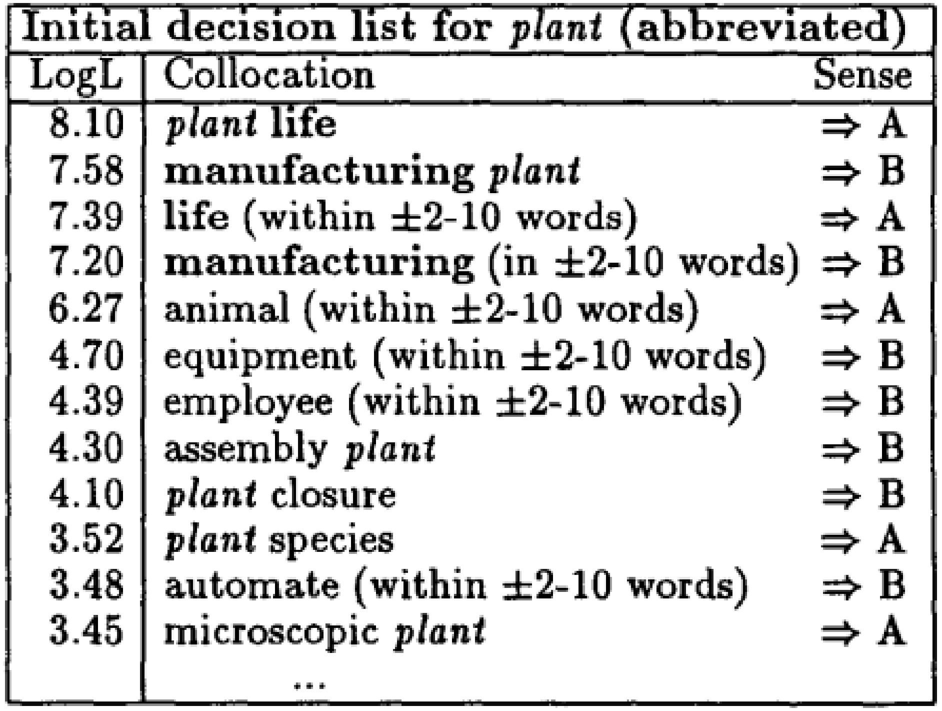


**Figure 1: Sample Initial State**

A = SENSE-A training example
B = SENSE-B training example
? = currently unclassified training example
Life = Set of training examples containing the collocation "life".

# Yarowsky Algorithm

- Step 3:Train the supervised classification algorithm on the SENSE-A/SENSE-B seed sets. The decision-list algorithm used here (Yarowsky, 1994) identifies other collocations that reliably partition the seed training data, ranked by the purity of the distribution.



Initial decision list for *plant* (abbreviated)

| LogL | Collocation | Sense |
|---|---|---|
| 8.10 | *plant* life | ⇒ A |
| 7.58 | manufacturing *plant* | ⇒ B |
| 7.39 | life (within ±2-10 words) | ⇒ A |
| 7.20 | manufacturing (in ±2-10 words) | ⇒ B |
| 6.27 | animal (within ±2-10 words) | ⇒ A |
| 4.70 | equipment (within ±2-10 words) | ⇒ B |
| 4.39 | employee (within ±2-10 words) | ⇒ B |
| 4.30 | assembly *plant* | ⇒ B |
| 4.10 | *plant* closure | ⇒ B |
| 3.52 | *plant* species | ⇒ A |
| 3.48 | automate (within ±2-10 words) | ⇒ B |
| 3.45 | microscopic *plant* | ⇒ A |
| | ... | |

- Decision list algorithm:

$$\log\left(\frac{\Pr(\mathrm{Sense}_A \mid \mathrm{Collocation}_i)}{\Pr(\mathrm{Sense}_B \mid \mathrm{Collocation}_i)}\right)$$

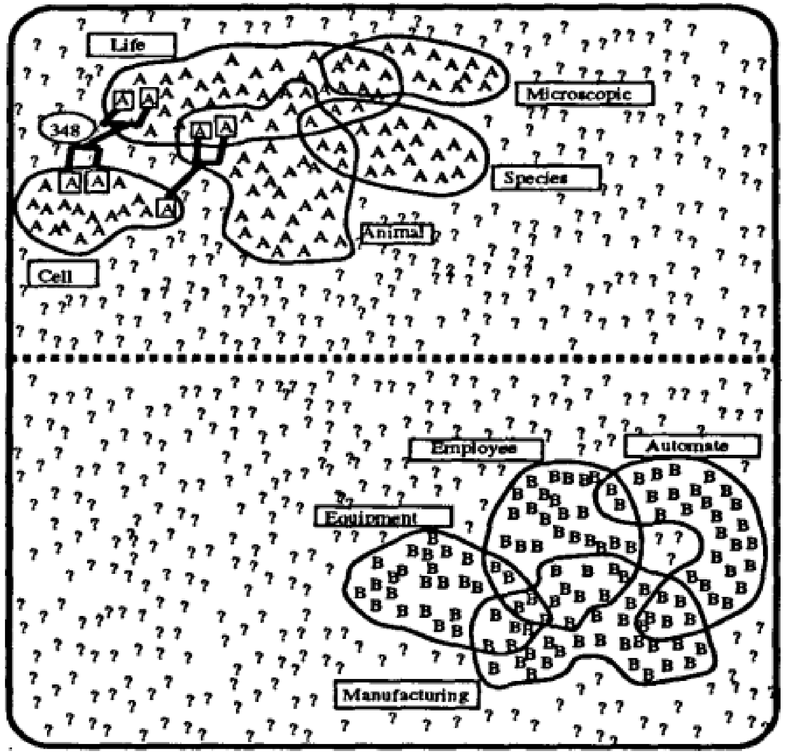


**Figure 2: Sample Intermediate State**
(following Steps 3b and 3c)

# Yarowsky Algorithm

- Step 4: Stop. When the training parameters are held constant, the algorithm will converge on a stable residual set

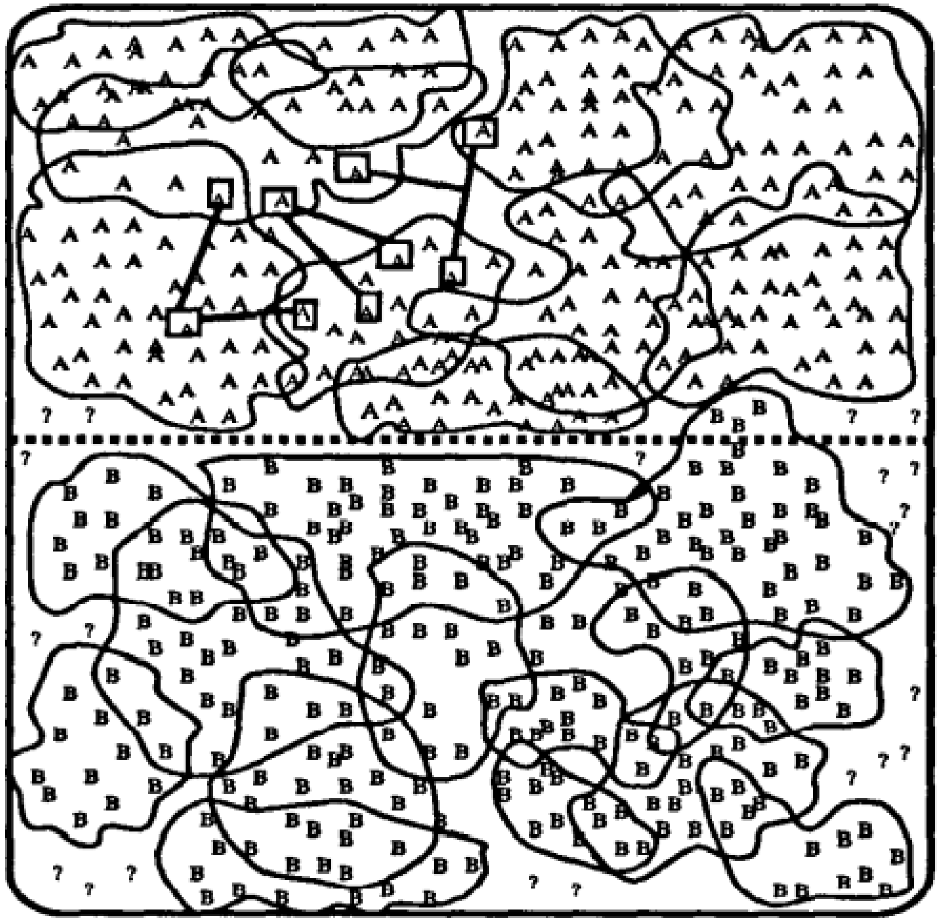
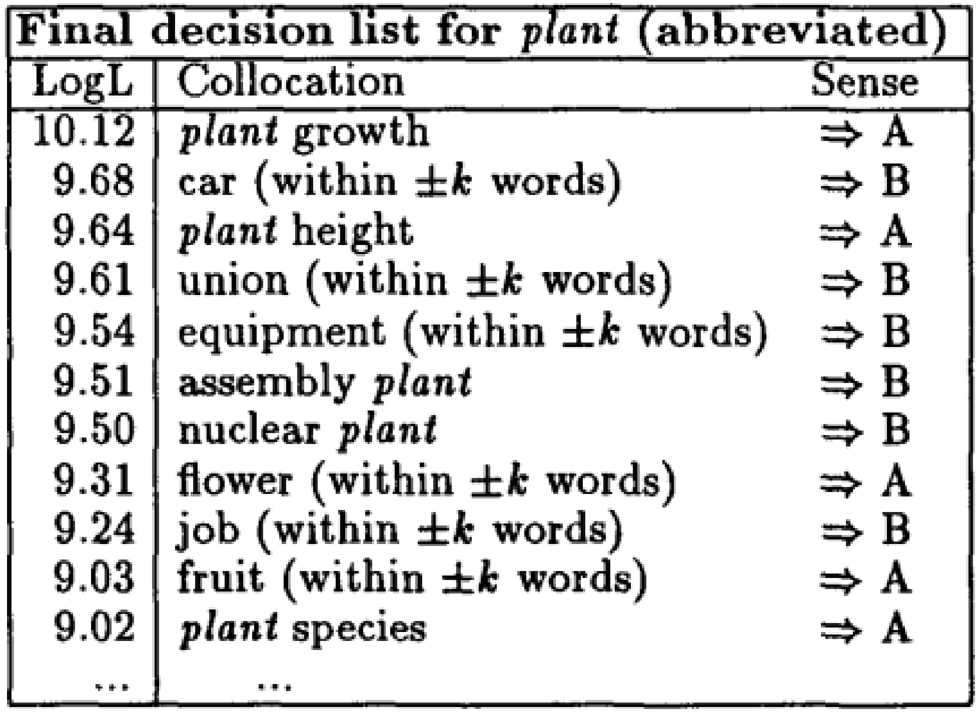


Figure 3: Sample Final State

# Yarowsky Algorithm

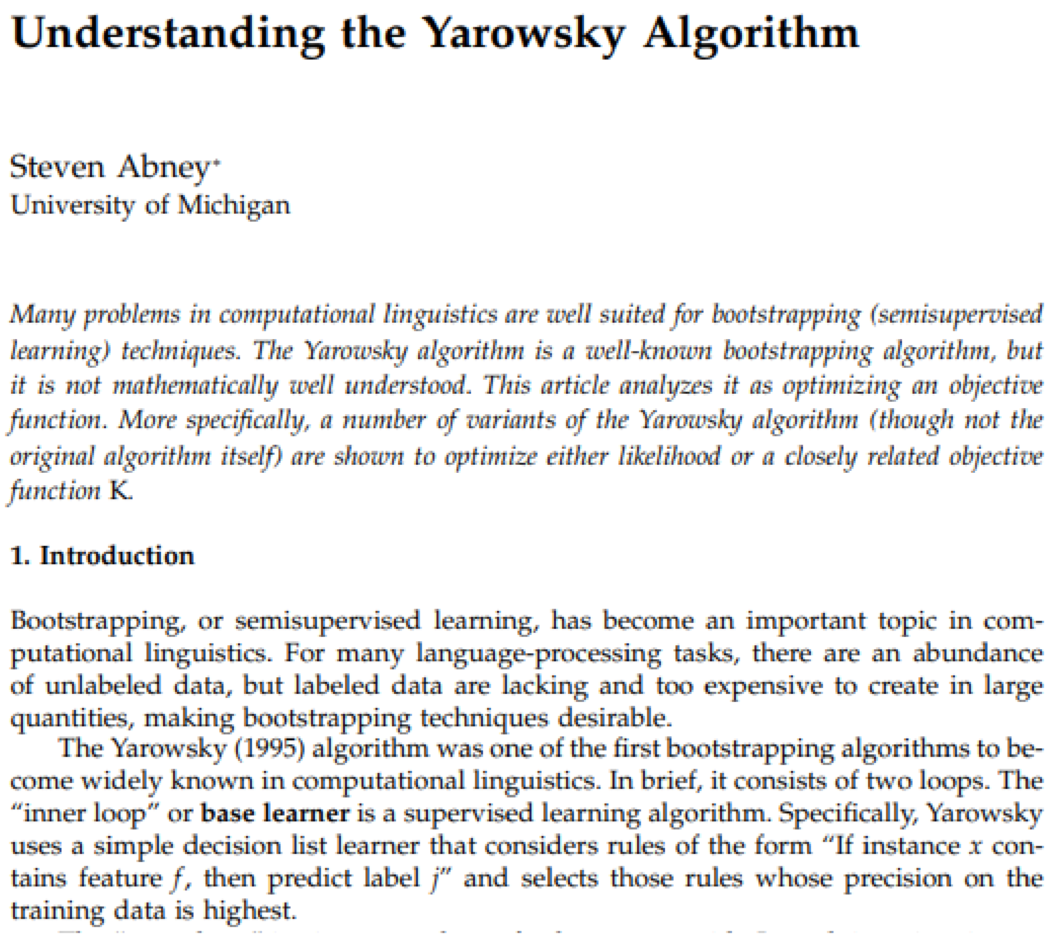
- Step 5: The classification procedure learned from the final supervised training step may now be applied to new data, and used to annotate the original untagged corpus with sense tags and probabilities.



Final decision list for *plant* (abbreviated)

| LogL | Collocation | Sense |
|---|---|---|
| 10.12 | *plant* growth | ⇒ A |
| 9.68 | car (within ±*k* words) | ⇒ B |
| 9.64 | *plant* height | ⇒ A |
| 9.61 | union (within ±*k* words) | ⇒ B |
| 9.54 | equipment (within ±*k* words) | ⇒ B |
| 9.51 | assembly *plant* | ⇒ B |
| 9.50 | nuclear *plant* | ⇒ B |
| 9.31 | flower (within ±*k* words) | ⇒ A |
| 9.24 | job (within ±*k* words) | ⇒ B |
| 9.03 | fruit (within ±*k* words) | ⇒ A |
| 9.02 | *plant* species | ⇒ A |
| ... | ... | |

# Yarowsky algorithm’s other variations



## Understanding the Yarowsky Algorithm

Steven Abney[*]
University of Michigan

*Many problems in computational linguistics are well suited for bootstrapping (semisupervised learning) techniques. The Yarowsky algorithm is a well-known bootstrapping algorithm, but it is not mathematically well understood. This article analyzes it as optimizing an objective function. More specifically, a number of variants of the Yarowsky algorithm (though not the original algorithm itself) are shown to optimize either likelihood or a closely related objective function* K.

### 1. Introduction

Bootstrapping, or semisupervised learning, has become an important topic in computational linguistics. For many language-processing tasks, there are an abundance of unlabeled data, but labeled data are lacking and too expensive to create in large quantities, making bootstrapping techniques desirable.

The Yarowsky (1995) algorithm was one of the first bootstrapping algorithms to become widely known in computational linguistics. In brief, it consists of two loops. The “inner loop” or **base learner** is a supervised learning algorithm. Specifically, Yarowsky uses a simple decision list learner that considers rules of the form “If instance $x$ contains feature $f$, then predict label $j$” and selects those rules whose precision on the training data is highest.

# Labeling is expensive

Labeled Data

Unlabeled Data

# Two major approaches for semi-supervised learning

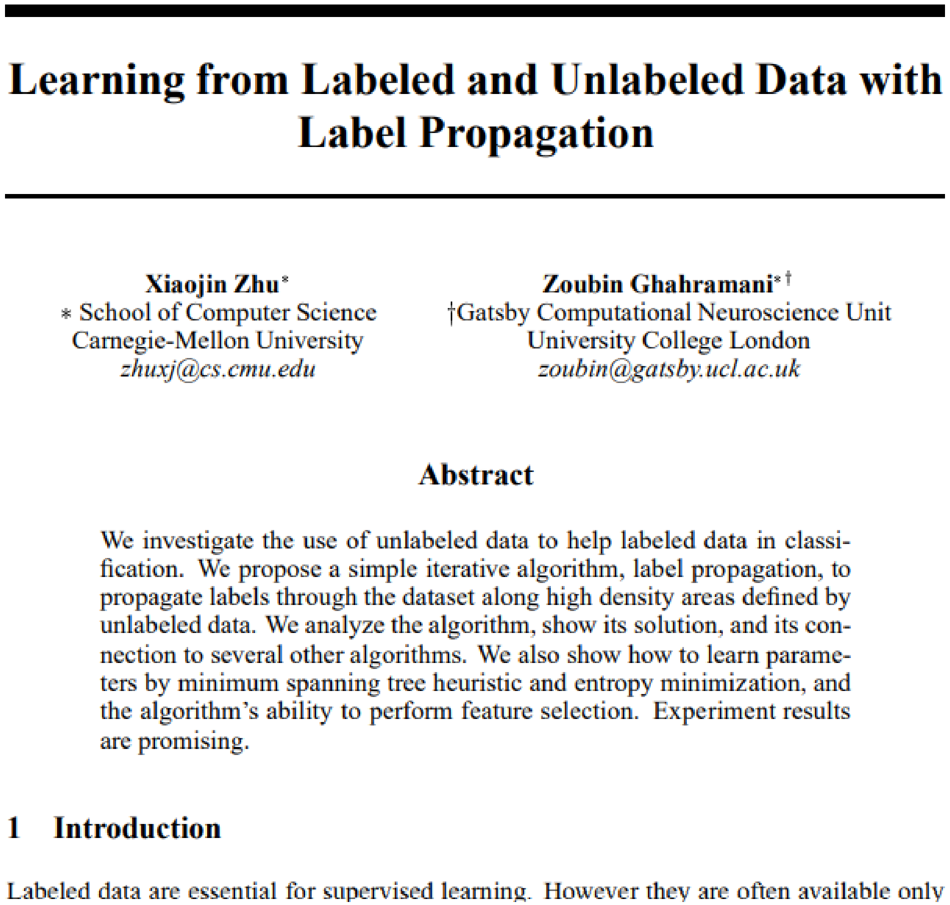
1. Graph-based

- Label propagation
- Graph convolution networks

1. Consistency-based

- Entropy minimization
- Pseudo-label
- Unsupervised data augmentation

# Label propagation (Zhu et al., 2002)



## Learning from Labeled and Unlabeled Data with Label Propagation

**Xiaojin Zhu**[*]
[*] School of Computer Science
Carnegie-Mellon University
*zhuxj@cs.cmu.edu*

**Zoubin Ghahramani**[*][†]
[†]Gatsby Computational Neuroscience Unit
University College London
*zoubin@gatsby.ucl.ac.uk*

### Abstract

We investigate the use of unlabeled data to help labeled data in classification. We propose a simple iterative algorithm, label propagation, to propagate labels through the dataset along high density areas defined by unlabeled data. We analyze the algorithm, show its solution, and its connection to several other algorithms. We also show how to learn parameters by minimum spanning tree heuristic and entropy minimization, and the algorithm's ability to perform feature selection. Experiment results are promising.

### 1 Introduction

Labeled data are essential for supervised learning. However they are often available only

# Label propagation problem definition

- Intuitively, we want data points that are close to have similar labels. We create a fully connected graph where the nodes are all data points, both labeled and unlabeled. The edge between any nodes i; j is weighted so that the closer the nodes are in local Euclidean distance, the larger the weight. The weights are controlled by a parameter :

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) = \exp\left(-\frac{\sum_{d=1}^{D}(x_i^d - x_j^d)^2}{\sigma^2}\right)$$

- All nodes have soft labels which can be interpreted as distributions over labels. We let the labels of a node propagate to all nodes through the edges. Larger edge weights allow labels to travel through more easily.

$$T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}$$

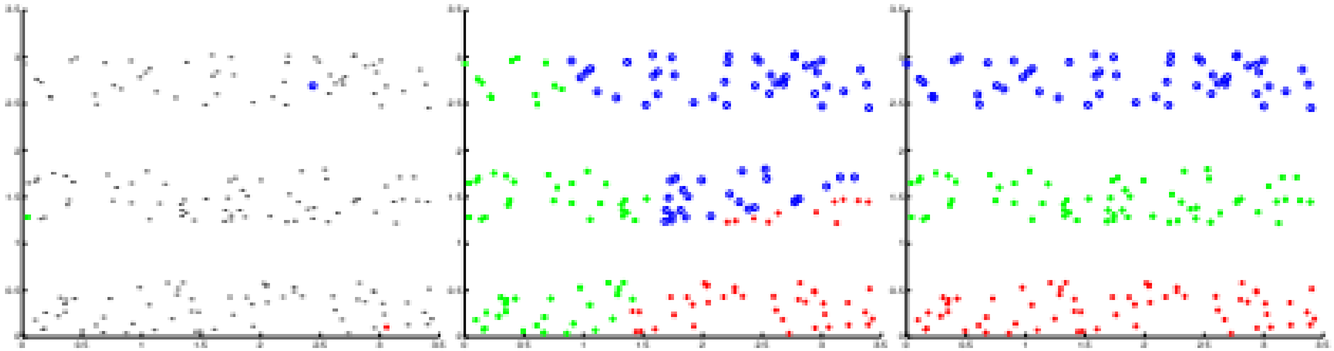
# Label Propagation Formulation

- 1. All nodes propagate labels for one step
- 2. Row-normalize Y to maintain the class probability interpretation.
- 3. Clamp the labeled data. Repeat from step 2 until Y converges

$$\bar{T} = \begin{bmatrix} \bar{T}_{ll} & \bar{T}_{lu} \\ \bar{T}_{ul} & \bar{T}_{uu} \end{bmatrix}$$

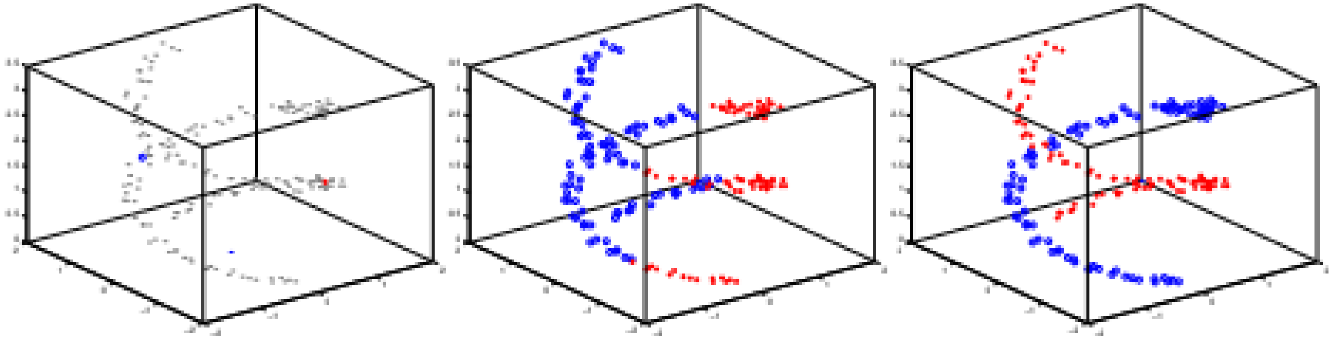
$$Y_U = (I - \bar{T}_{uu})^{-1}\bar{T}_{ul}Y_L$$

# Label Propagation (Zhu et al., 2002)



(a) The data (b) kNN (c) Label propagation

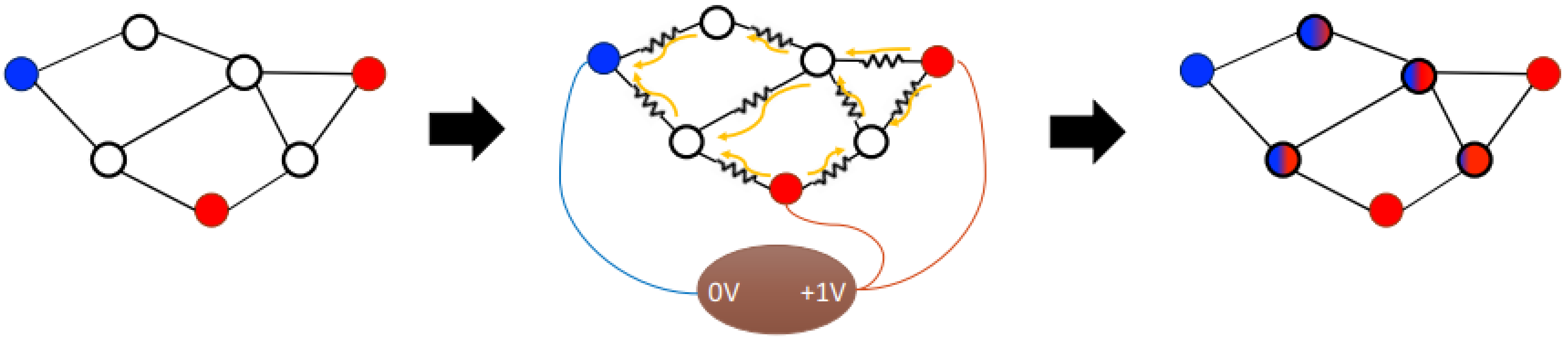
Figure 1: The 3 Bands dataset. Labeled data are color symbols and unlabeled data are dots in (a). kNN ignores unlabeled data structure, while label propagation uses it.



(a) The data (b) kNN (c) Label propagation

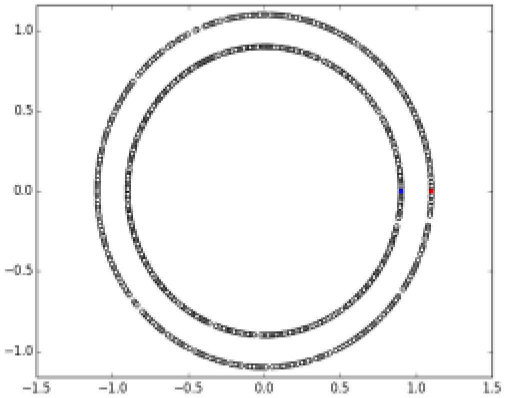
Figure 2: The Springs dataset.

# Label propagation

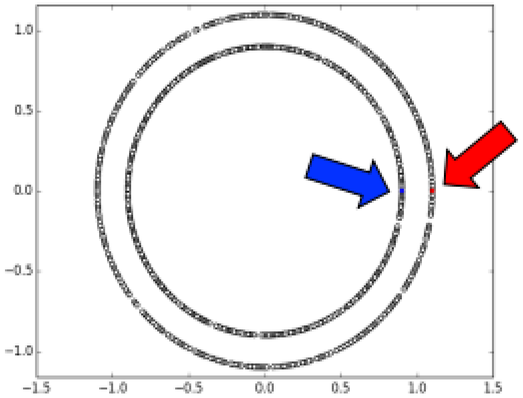


$$\min_{\mathbf{f}_u} \; \frac{1}{2} \sum_{(x,y)\in E} w_{x,y}(\mathbf{f}(x) - \mathbf{f}(y))^2.$$

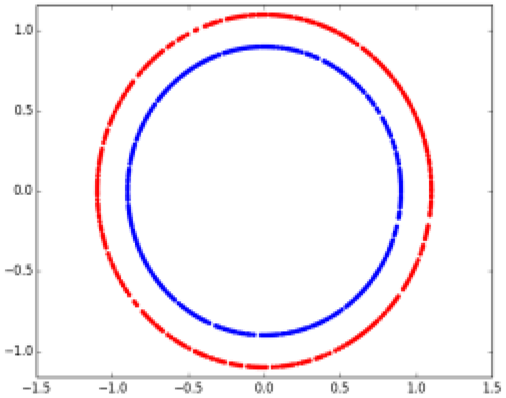
# Example 1: Separate two rings



*input*



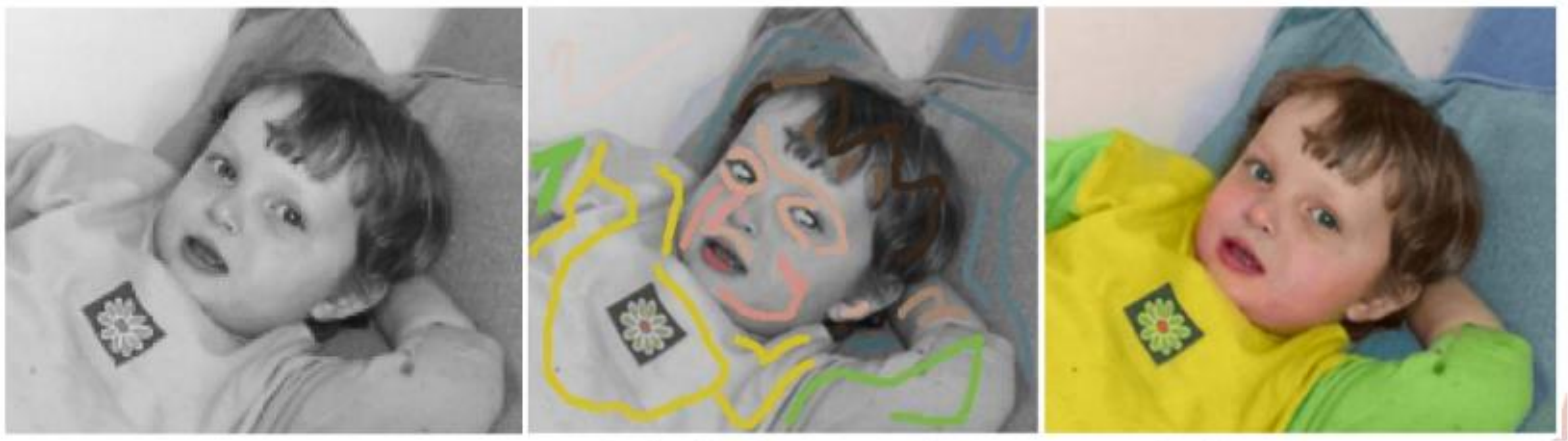
*labeled examples*



*Label Propagation*

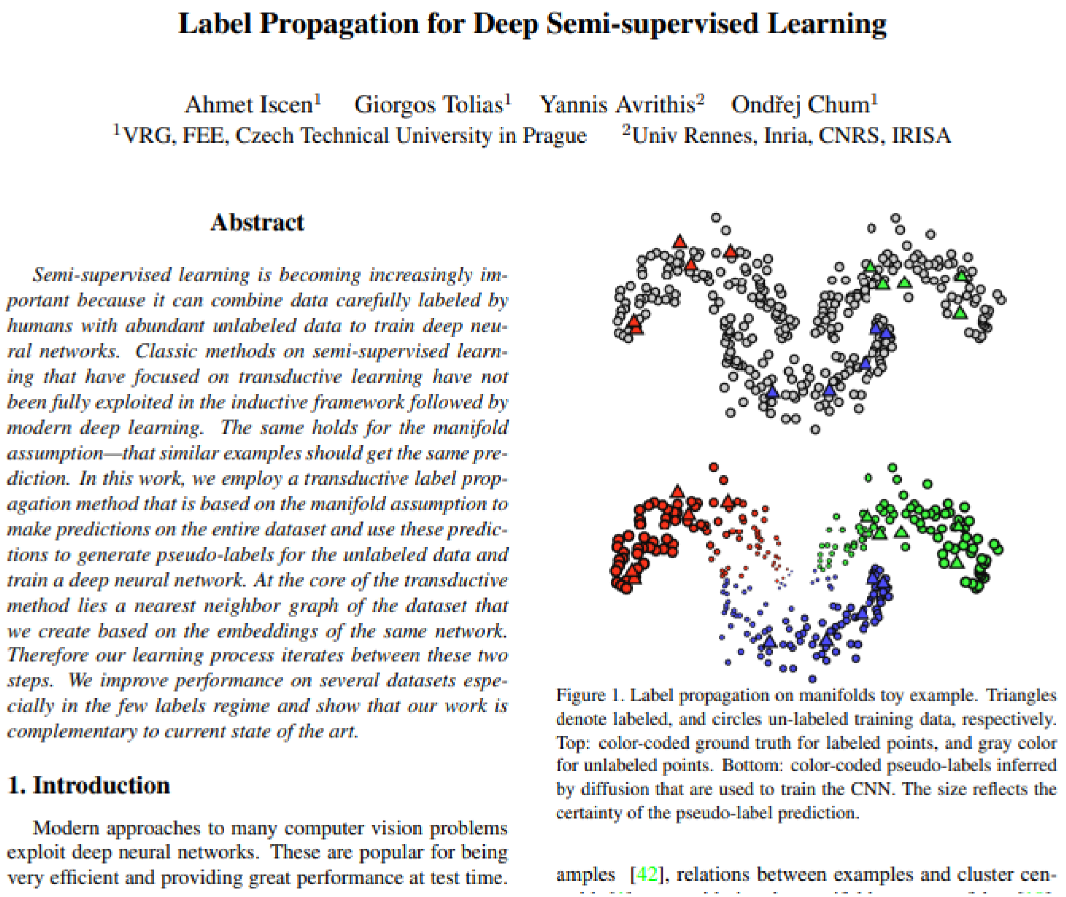
# Example 2: Coloring grayscale image

[Levin, Lischinski, Weiss, SIGGRAPH 2004]



*input* *labeled examples* *Label Propagation*

# Label Propagation for Deep Semi-supervised Learning



## Label Propagation for Deep Semi-supervised Learning

Ahmet Iscen[1] Giorgos Tolias[1] Yannis Avrithis[2] Ondřej Chum[1]
[1]VRG, FEE, Czech Technical University in Prague [2]Univ Rennes, Inria, CNRS, IRISA

### Abstract

*Semi-supervised learning is becoming increasingly important because it can combine data carefully labeled by humans with abundant unlabeled data to train deep neural networks. Classic methods on semi-supervised learning that have focused on transductive learning have not been fully exploited in the inductive framework followed by modern deep learning. The same holds for the manifold assumption—that similar examples should get the same prediction. In this work, we employ a transductive label propagation method that is based on the manifold assumption to make predictions on the entire dataset and use these predictions to generate pseudo-labels for the unlabeled data and train a deep neural network. At the core of the transductive method lies a nearest neighbor graph of the dataset that we create based on the embeddings of the same network. Therefore our learning process iterates between these two steps. We improve performance on several datasets especially in the few labels regime and show that our work is complementary to current state of the art.*

### 1. Introduction

Modern approaches to many computer vision problems exploit deep neural networks. These are popular for being very efficient and providing great performance at test time.

Figure 1. Label propagation on manifolds toy example. Triangles denote labeled, and circles un-labeled training data, respectively. Top: color-coded ground truth for labeled points, and gray color for unlabeled points. Bottom: color-coded pseudo-labels inferred by diffusion that are used to train the CNN. The size reflects the certainty of the pseudo-label prediction.

amples [42], relations between examples and cluster cen-

# Refrences

[1] Unsupervised word sense disambiguation rivaling supervised methods

[2] Semi-Supervised Learning on Data Streams via Temporal Label Propagation

[3] Label Propagation for Deep Semi-supervised Learning

[4] Semi-supervised learning on data streams via temporal label propagation

[5] Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions