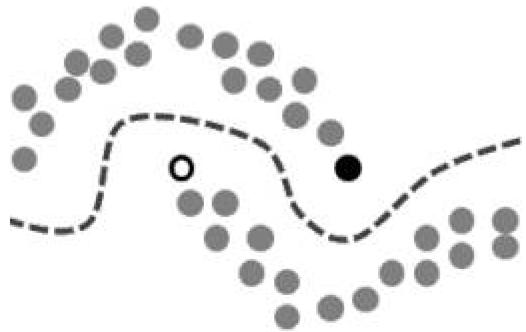
Semi-Supervised Learning

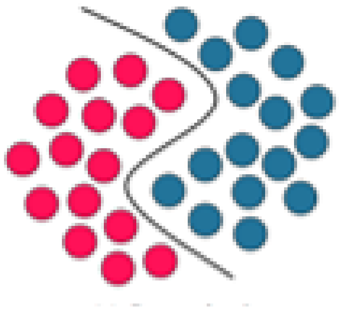


Readings:

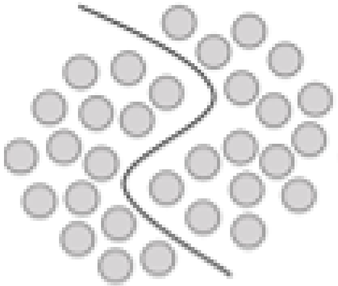
- Semi-Supervised Learning. Encyclopedia of Machine Learning. Jerry Zhu, 2010
- Combining Labeled and Unlabeled Data with Co-Training. Avrim Blum, Tom Mitchell. COLT 1998.

Machine Learning Paradigms

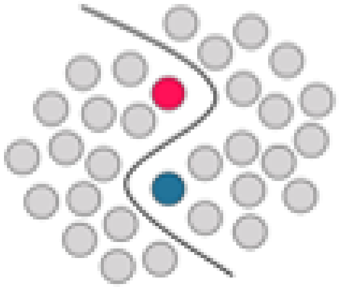
Supervised Learning



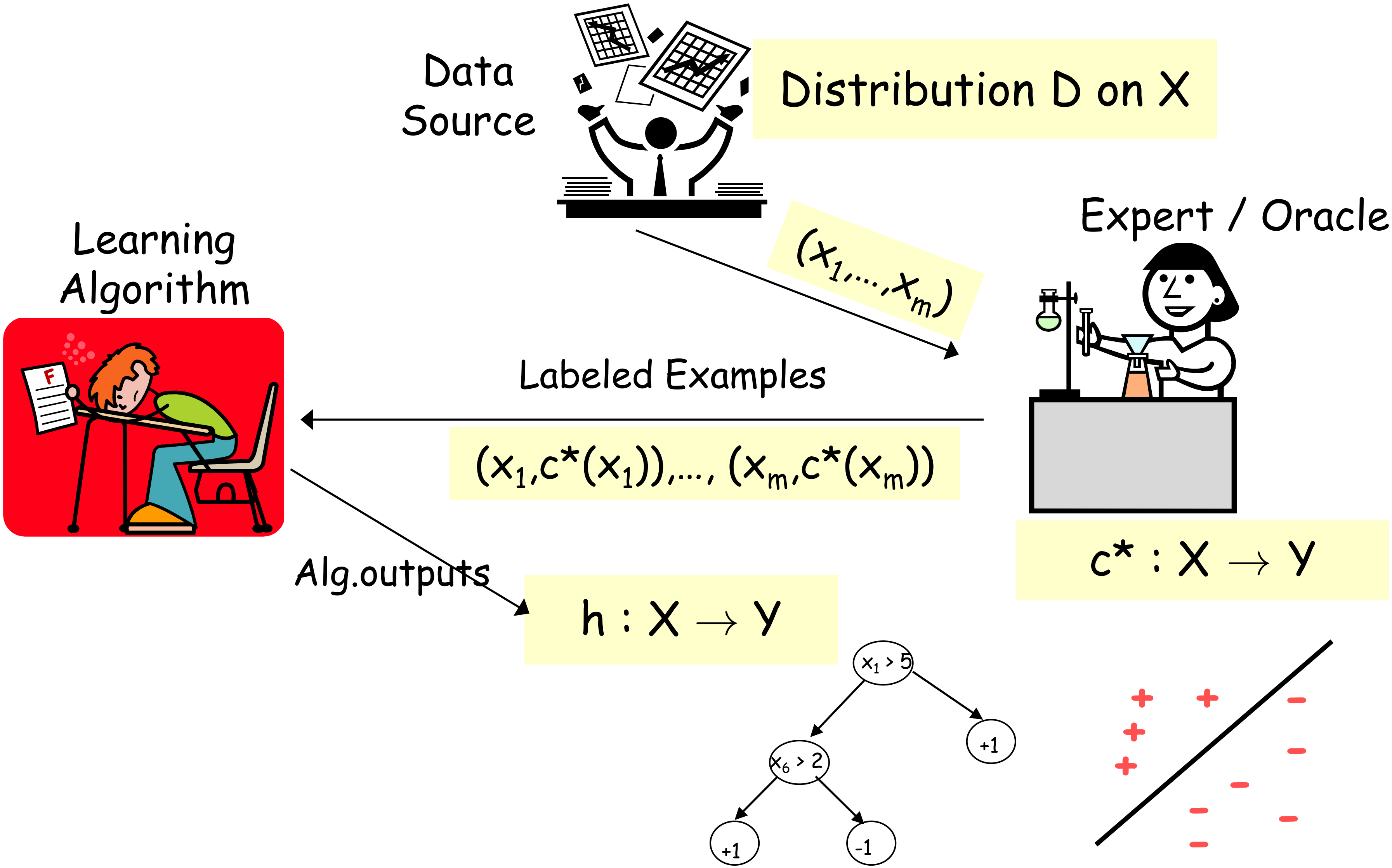
Unsupervised Learning



Semi-Supervised Learning



Fully Supervised Learning



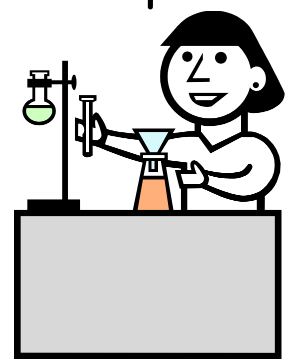
Fully Supervised Learning

Data Source

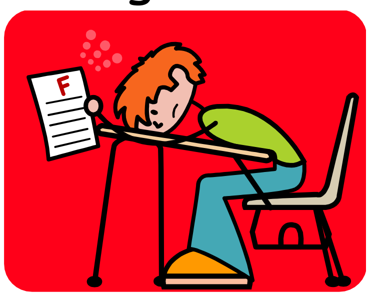
Distribution D on X

$(x_1,\ldots,x_m)$

Expert / Oracle



Learning Algorithm



Labeled Examples

$(x_1,c^*(x_1)),\ldots, (x_m,c^*(x_m))$

$c^* : X \to Y$

Alg.outputs

$h : X \to Y$

$S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

Goal: h has small error over D.

$$err_D(h) = \Pr_{x\sim D}(h(x) \neq c^*(x))$$

Two Core Aspects of Supervised Learning

Algorithm Design. How to optimize?

Computation

Automatically generate rules that do well on observed data.

- E.g.: Naïve Bayes, logistic regression, SVM, Adaboost, etc.

Confidence Bounds, Generalization

(Labeled) Data

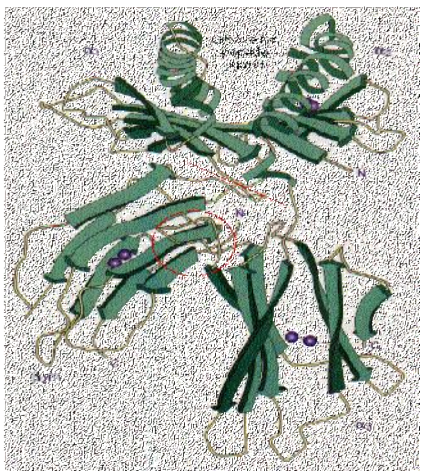
Confidence for rule effectiveness on future data.

- VC-dimension, Rademacher complexity, margin based bounds, etc.

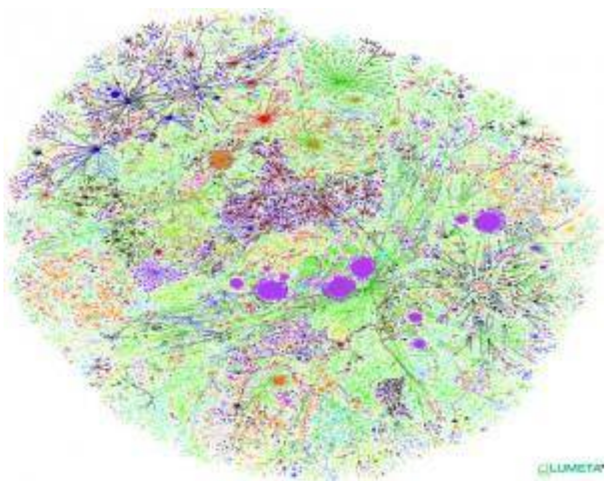
Classic Paradigm Insufficient Nowadays

Modern applications: massive amounts of raw data.

Only a tiny fraction can be annotated by human experts.



Protein sequences



Billions of webpages

Images

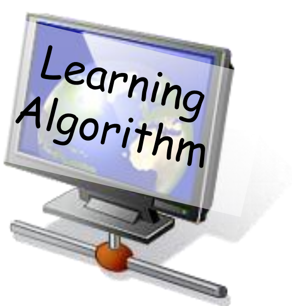
Modern ML: New Learning Approaches

Modern applications: massive amounts of raw data.

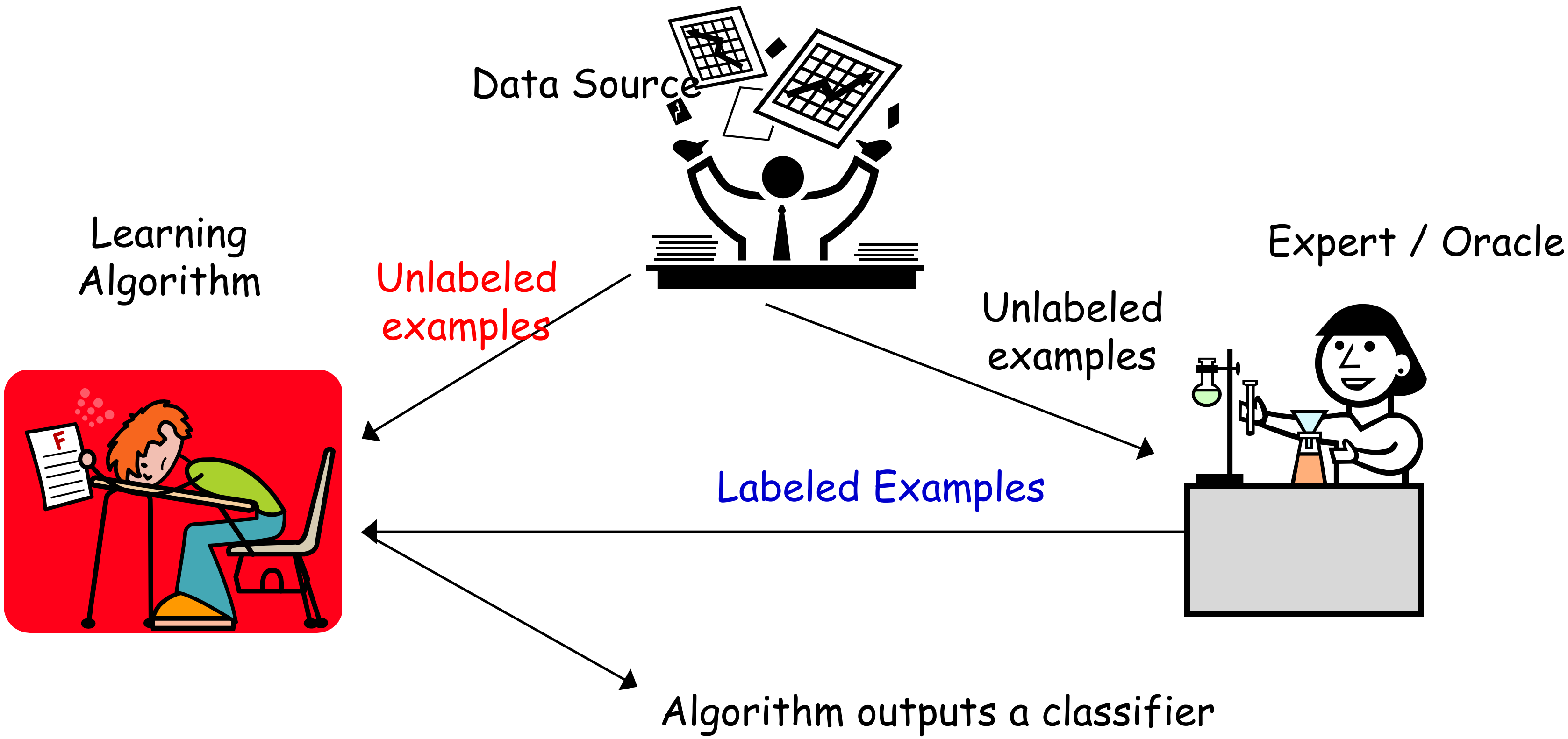
Techniques that best utilize data, minimizing need for expert/human intervention.

Paradigms where there has been great progress.

- Semi-supervised Learning, (Inter)active Learning.



Semi-Supervised Learning



$S_l=\{(x_1,y_1),\ldots,(x_{m_l},y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

$S_u=\{x_1,\ldots,x_{m_u}\}$ drawn i.i.d from D

Goal: h has small error over D.

$$\mathrm{err}_D(h) = \Pr_{x\sim D}(h(x) \neq c^*(x))$$

Semi-supervised Learning

- Major topic of research in ML.
- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:
  - Transductive SVM [Joachims '99]
  - Co-training [Blum & Mitchell '98]
  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

Workshops [ICML '03, ICML' 05, …]

Books:
- Semi-Supervised Learning, MIT 2006

  O. Chapelle, B. Scholkopf and A. Zien (eds)
- Introduction to Semi-Supervised Learning, Morgan & Claypool, 2009 Zhu & Goldberg

Semi-supervised Learning

- Major topic of research in ML.
- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:
  - Transductive SVM [Joachims '99]
  - Co-training [Blum & Mitchell '98]
  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

Both wide spread applications and solid foundational understanding!!!

Semi-supervised Learning

- Major topic of research in ML.
- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:
  - Transductive SVM [Joachims '99]
  - Co-training [Blum & Mitchell '98]
  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

Today: discuss these methods.

Very interesting, they all exploit unlabeled data in different, very interesting and creative ways.

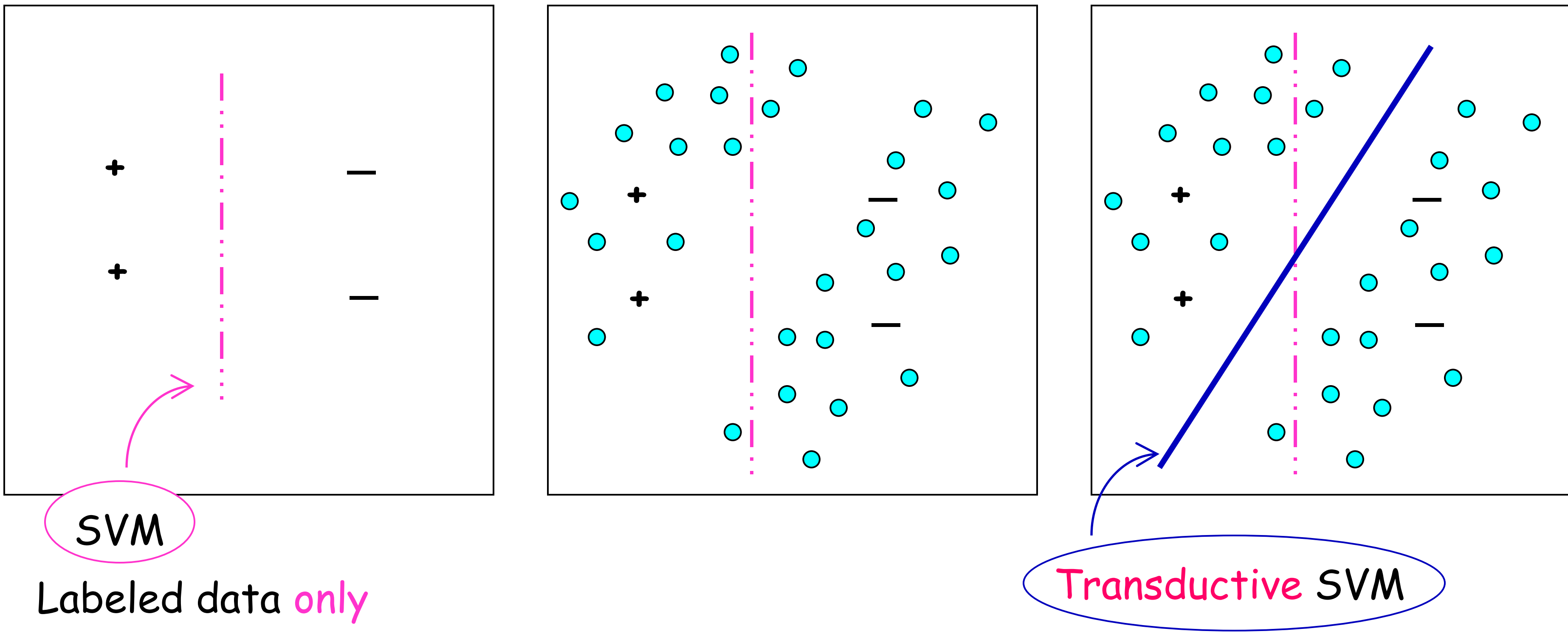
Semi-supervised SVM

[Joachims '99]

Margins based regularity

Target goes through low density regions (large margin).

- assume we are looking for linear separator
- belief: should exist one with large separation



Transductive Support Vector Machines

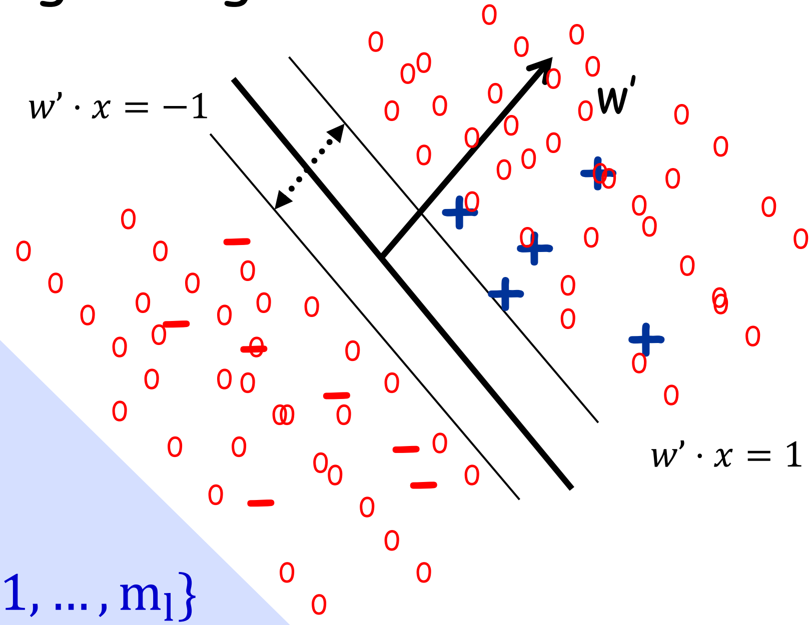
Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]

<u>Input</u>: $S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

$S_u = \{x_1, \ldots, x_{m_u}\}$

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]



Input: $S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

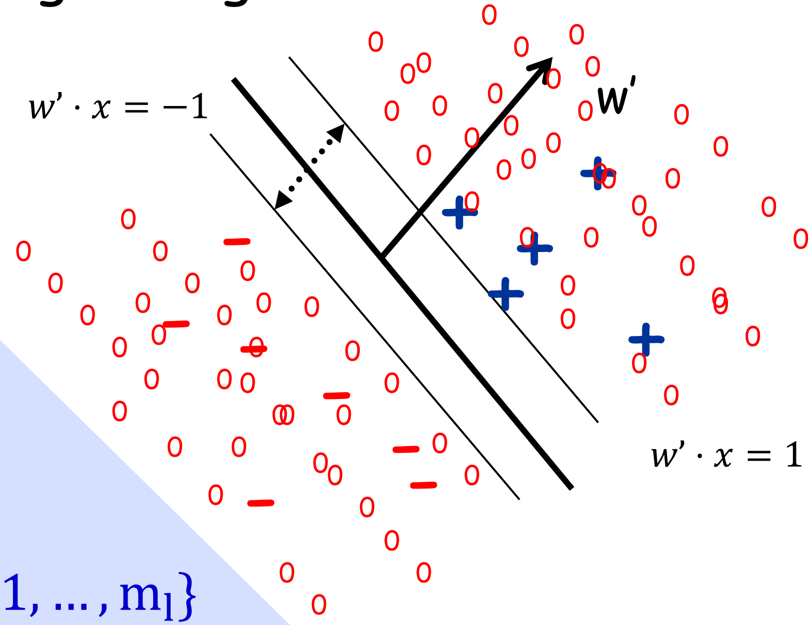
$S_u=\{x_1, \ldots,x_{m_u}\}$

$\mathrm{argmin}_w \; ||w||^2$ s.t.:

- $y_i \, w \cdot x_i \geq 1$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]



Input: $S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

$S_u=\{x_1, \ldots,x_{m_u}\}$

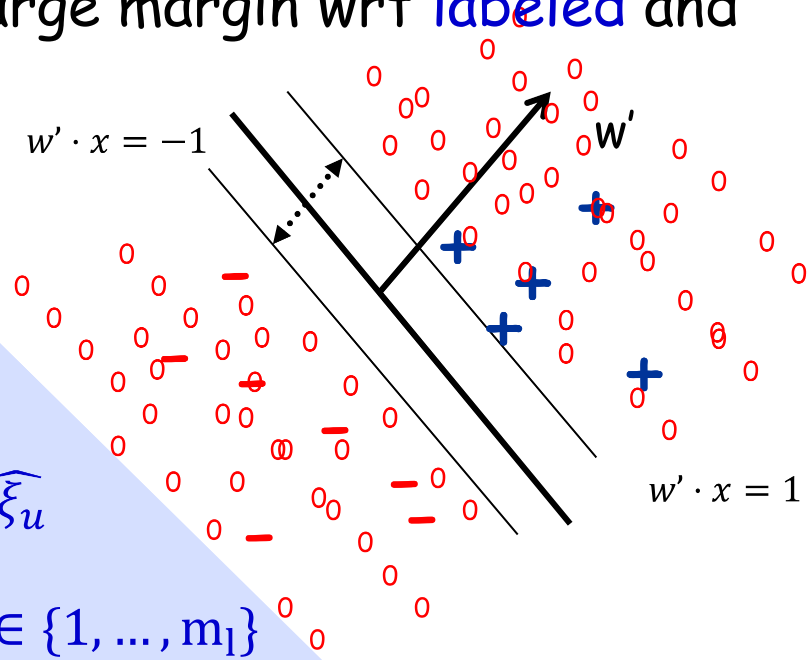
$\text{argmin}_w \, ||w||^2$ s.t.:

- $y_i \, w \cdot x_i \geq 1$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

Find a labeling of the unlabeled sample and w s.t. w separates both labeled and unlabeled data with maximum margin.

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]



<u>Input</u>: $S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

$S_u = \{x_1, \ldots, x_{m_u}\}$

$$\mathrm{argmin}_w \, \|w\|^2 + C \sum_i \xi_i + C \sum_u \widehat{\xi_u}$$

- $y_i \, w \cdot x_i \geq 1 - \xi_i$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

Find a labeling of the unlabeled sample and w s.t. w separates both labeled and unlabeled data with maximum margin.

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

Input: $S_l=\{(x_1,y_1), \ldots,(x_{m_l},y_{m_l})\}$

$S_u=\{x_1, \ldots,x_{m_u}\}$

$$\mathrm{argmin}_w \, ||w||^2 + C\sum_i \xi_i + C\sum_u \widehat{\xi_u}$$

- $y_i \, w \cdot x_i \geq 1-\xi_i$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

NP-hard….. Convex only after you guessed the labels… too many possible guesses…

Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

<u>Heuristic (Joachims) high level idea:</u>

- First maximize margin over the labeled points
- Use this to give initial labels to unlabeled points based on this separator.
- Try flipping labels of unlabeled points to see if doing so can increase margin

Keep going until no more improvements. Finds a locally-optimal solution.

Co-training

[Blum & Mitchell '98]

Different type of underlying regularity assumption:

Consistency or Agreement Between Parts

Co-training: Self-consistency

Agreement between two parts : co-training [Blum-Mitchell98].

- examples contain two sufficient sets of features, $x = \langle x_1, x_2 \rangle$
- belief: the parts are consistent, i.e. $\exists$ c_1, c_2 s.t. $c_1(x_1)=c_2(x_2)=c^*(x)$

For example, if we want to classify web pages: $x = \langle x_1, x_2 \rangle$ as faculty member homepage or not

Prof. Avrim Blum My Advisor

x - Link info & Text info

x_1- Text info

Prof. Avrim Blum My Advisor

x_2- Link info

Iterative Co-Training

Idea: Use small labeled sample to learn initial rules.

- E.g., "my advisor" pointing to a page is a good indicator it is a faculty home page.
- E.g., "I am teaching" on a page is a good indicator it is a faculty home page.

Idea: Use unlabeled data to propagate learned information.

my advisor

Avrim Blum's home page Page 1 of 1

Avrim Blum
Professor of Computer Science

Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
avrim at cs.cmu.edu

Office: Wean 4130
Tel: (412) 268-6452
Fax: (412) 268-5576
Admin assist: Nicole Stenger, Wean 4116, 268-3779
Check out our new faculty members Ryan O'Donnell and Luis von Ahn.

My main research interests are machine learning theory, approximation algorithms, on-line algorithms, and algorithmic game theory. I was/am on the Program Committees for FOCS 2008 (Symp. Foundations of Computer Science), ACM-EC 2008 (Electronic Commerce), and COLT 2007 (Conference on Learning Theory), and was recently local organizer for COLT 2006 and FOCS 2005. I also co-organized the 2005 Foundations of Computational Mathematics Workshop on Algorithmic Game Theory and Metric Embeddings. A while back I served as Program Chair for FOCS 2000 and I've done some work in AI Planning. For more information on my research, see the publications and research interests links below. I am also affiliated with the Machine Learning department.

I am currently (Spring 2008) teaching 15-859(B) Machine Learning Theory.

- Publications
- Research Interests
- Survey Talks
- Courses
- ALADDIN, Algorithms and Complexity Group
- ACO Program Home Page
- Theory Seminars, Theory lunch ML lunch
- Family pictures, Other pictures, My Startup Page
- My Tutorial on Machine Learning Theory given at FOCS 2003 and a short essay.

My advisees: Aaron Roth, Katrina Ligett, Nina Balcan, Mugizi Robert Rwebangira, Shobha Venkataraman.

Past advisees: Prasad Chalasani, Santosh Vempala, Carl Burch, Adam Kalai, John Langford, Nikhil Bansal, Martin Zinkevich, Shuchi Chawla, Brendan McMahan.

Google

Iterative Co-Training

Idea: Use small labeled sample to learn initial rules.

- E.g., "my advisor" pointing to a page is a good indicator it is a faculty home page.
- E.g., "I am teaching" on a page is a good indicator it is a faculty home page.

Idea: Use unlabeled data to propagate learned information.

The co-training algorithm trains two predictors:

h(1) --> x(1) h(2) --> x(2)

If h(1) confidently predicts the label of an unlabeled instance x then the instance-label pair (x, h(1)(x)) is added to h(2)'s labeled data, and vice versa.

Note this promotes h(1) and h(2) to predict the same on x.

Co-training/Multi-view SSL: Direct Optimization of Agreement

<u>Input</u>: $S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$
$S_u = \{x_1, \ldots, x_{m_u}\}$

$$\mathrm{argmin}_{h_1, h_2} \sum_{l=1}^{2} \sum_{i=1}^{m_l} l(h_l(x_i), y_i) + C \sum_{i=1}^{m_u} \mathrm{agreement}(h_1(x_i), h_2(x_i))$$

Each of them has small labeled error

Regularizer to encourage agreement over unlabeled dat

Co-training/Multi-view SSL: Direct Optimization of Agreement

Input: $S_l=\{(x_1, y_1), \ldots,(x_{m_l}, y_{m_l})\}$
$S_u=\{x_1, \ldots,x_{m_u}\}$

$$\mathrm{argmin}_{h_1,h_2} \sum_{l=1}^{2}\sum_{i=1}^{m_l} l(h_l(x_i), y_i) + C\sum_{i=1}^{m_u} \mathrm{agreement}(h_1(x_i), h_2(x_i))$$

- $l(h(x_i), y_i)$ loss function
  - E.g., square loss $l(h(x_i), y_i) = (y_i - h(x_l))^2$
  - E.g., 0/1 loss $l(h(x_i), y_i) = 1_{y_i \neq h(x_i)}$

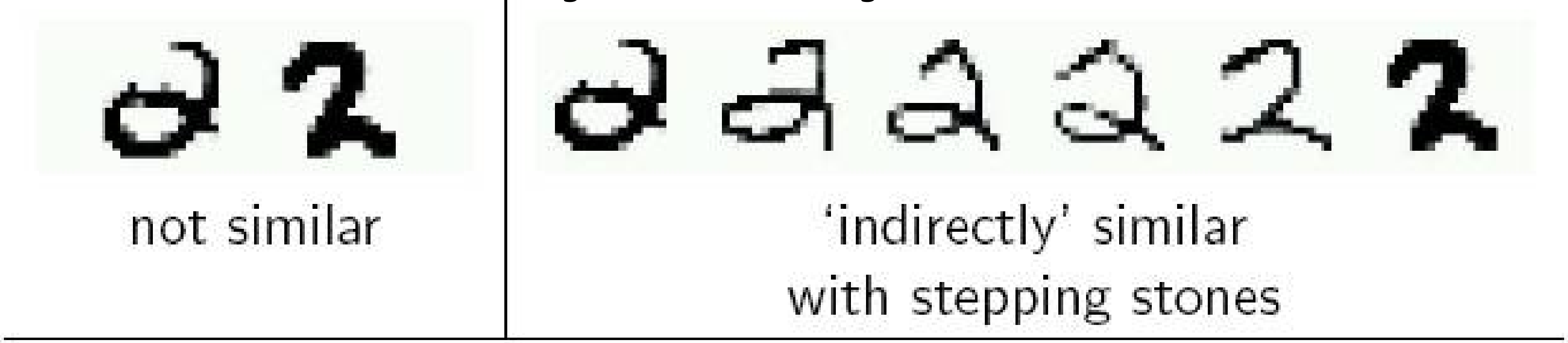
Similarity Based Regularity

[Blum&Chwala01], [ZhuGhahramaniLafferty03]

Graph-based Methods

- Assume we are given a pairwise similarity fnc and that very similar examples probably have the same label.
- If we have a lot of labeled data, this suggests a Nearest-Neighbor type of algorithm.
- If you have a lot of unlabeled data, perhaps can use them as "stepping stones".

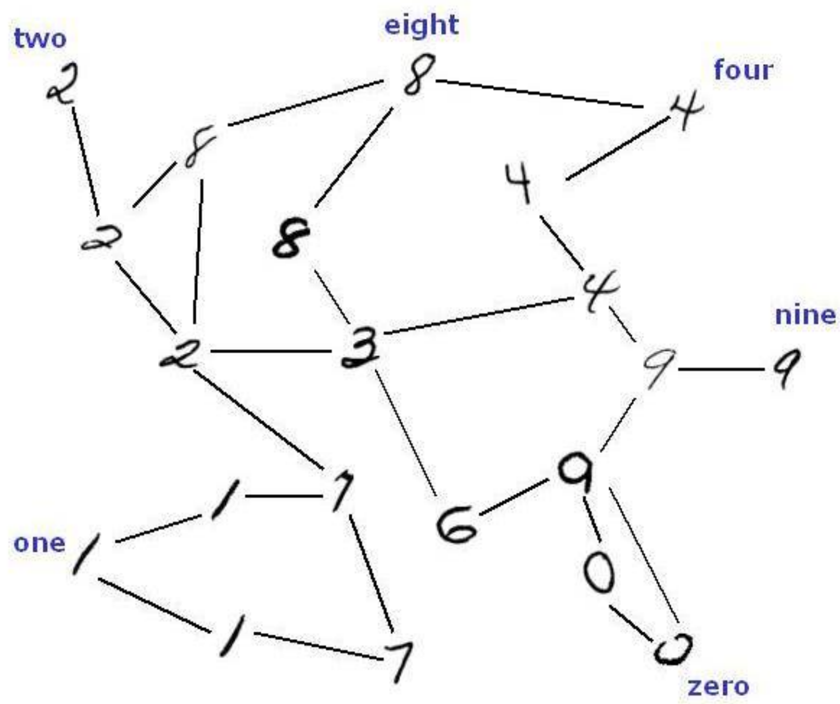
E.g., handwritten digits [Zhu07]:



Graph-based Methods

Idea: construct a graph with edges between very similar examples.

Unlabeled data can help "glue" the objects of the same class together.



Graph-based Methods

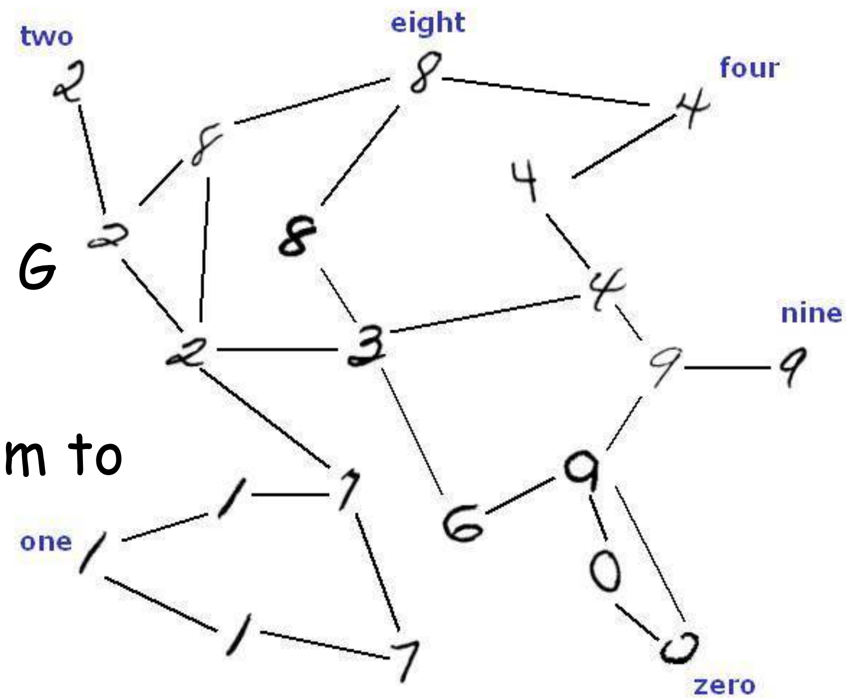
Often, transductive approach. (Given L + U, output predictions on U). Are alllowed to output any labeling of $L \cup U$.

Main Idea:

- Construct graph G with edges between very similar examples.

- Might have also glued together in G examples of different classes.

- Run a graph partitioning algorithm to separate the graph into pieces.



Several methods:

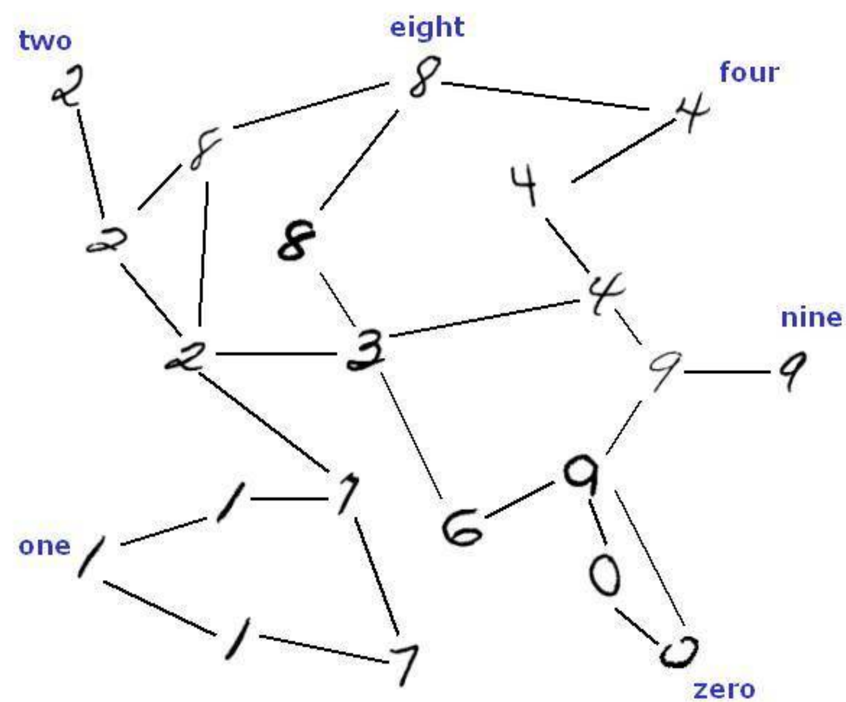
- Minimum/Multiway cut
- Minimum "soft-cut"
- Spectral partitioning
- …

Gaussian Fields and Harmonic Function
[ZhuGhahramaniLafferty'03]

graph G ={V, E, W}

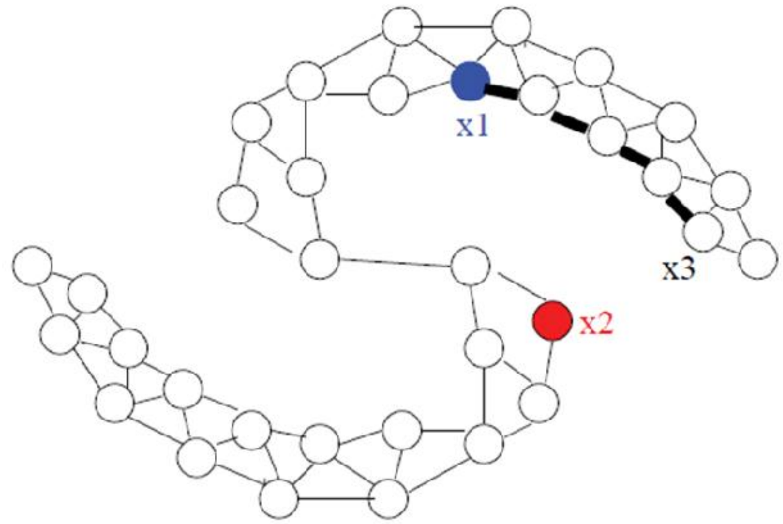
> vertices V are
the labeled and unlabeled instances

> The undirected edges E
connect instances i, j with weight wij



How to Create the Graph

- Empirically, the following works well:



1. Compute distance between i, j

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2$$

2. For each i, connect to its kNN. k very small but still connects the graph

3. Optionally put weights on (only) those edges

$$w_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2\right)$$

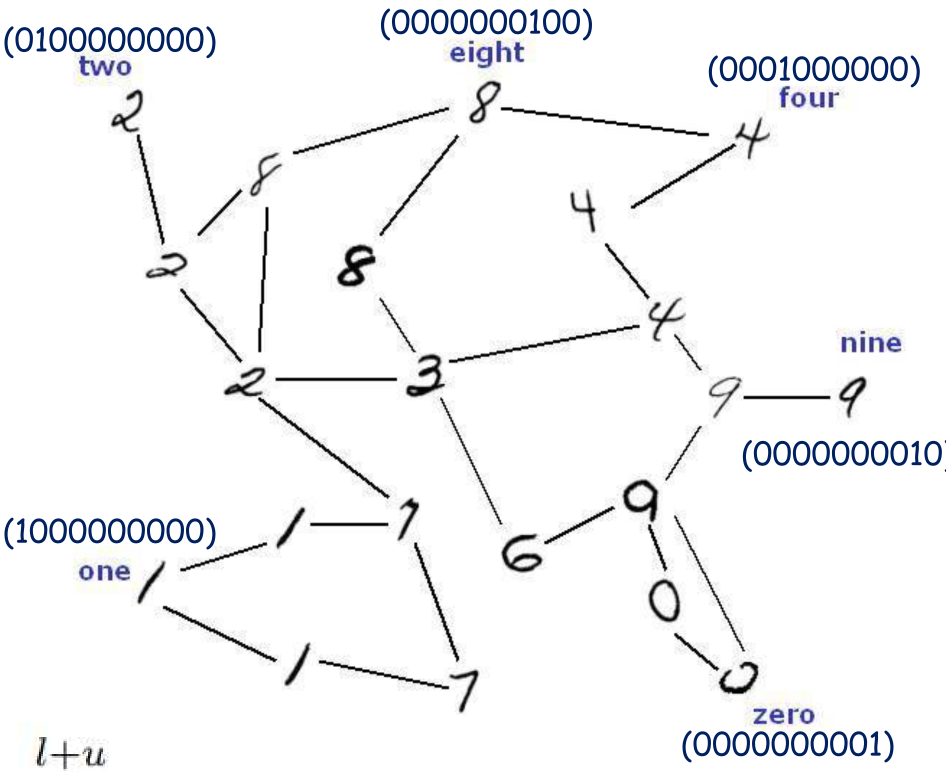
4. Tune σ

Gaussian Fields and Harmonic Function

[ZhuGhahramaniLafferty'03]

Large wij implies a preference for the predictions f(xi) and f(xj) to be the same.

$$\sum_{i,j=1}^{l+u} w_{ij} \left\| f(\mathbf{x}_i) - f(\mathbf{x}_j) \right\|^2$$



To find the f

$$\operatorname*{argmin}_f \frac{1}{l} \sum_{i=1}^{l} c(f(\mathbf{x}_i), y_i) + \lambda_1 \|f\|^2 + \lambda_2 \sum_{i,j=1}^{l+u} w_{ij} \left\| f(\mathbf{x}_i) - f(\mathbf{x}_j) \right\|^2$$

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

MixMatch: A Holistic Approach to Semi-Supervised Learning

David Berthelot
Google Research
dberth@google.com

Nicholas Carlini
Google Research
ncarlini@google.com

Ian Goodfellow
Work done at Google
ian-academic@mailfence.com

Avital Oliver
Google Research
avitalo@google.com

Nicolas Papernot
Google Research
papernot@google.com

Colin Raffel
Google Research
craffel@google.com

Background

Many recent approaches for semi-supervised learning add a loss term which is computed on unlabeled data and encourages the model to generalize better to unseen data.

Co-Training

$$\mathrm{argmin}_{h_1,h_2} \sum_{l=1}^{2} \sum_{i=1}^{m_l} l(h_l(x_i), y_i) + C \sum_{i=1}^{m_u} \mathrm{agreement}(h_1(x_i), h_2(x_i))$$

Each of them has small labeled error

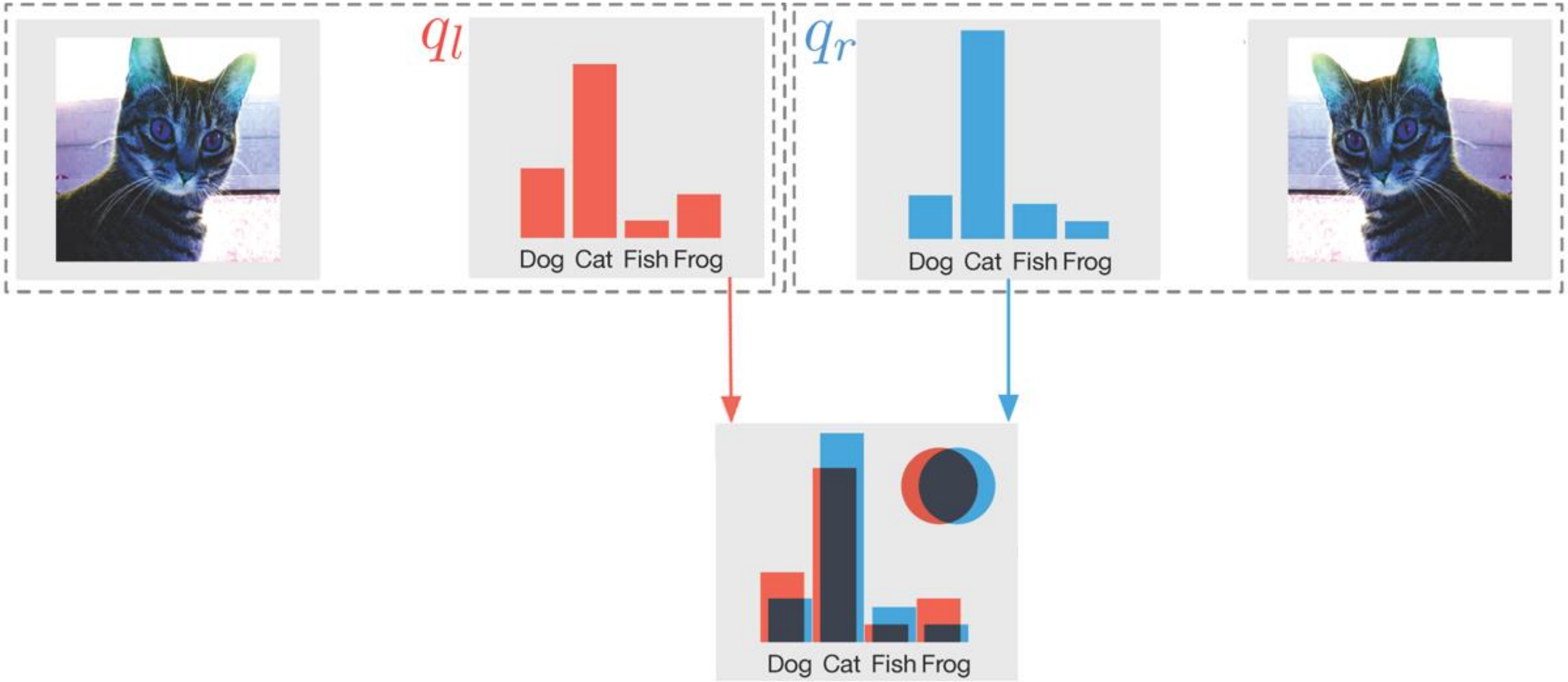
Regularizer to encourage agreement over unlabeled dat

In much recent work, the loss term falls into one of three classes:

- **Entropy minimization** encourages the model to output confident predictions on unlabeled data;
- **Consistency regularization** encourages the model to produce the same output distribution when its inputs are perturbed;
- **Generic regularization** encourages the model to generalize well and avoid overfitting the training data.

Background

1. Consistency Regularization



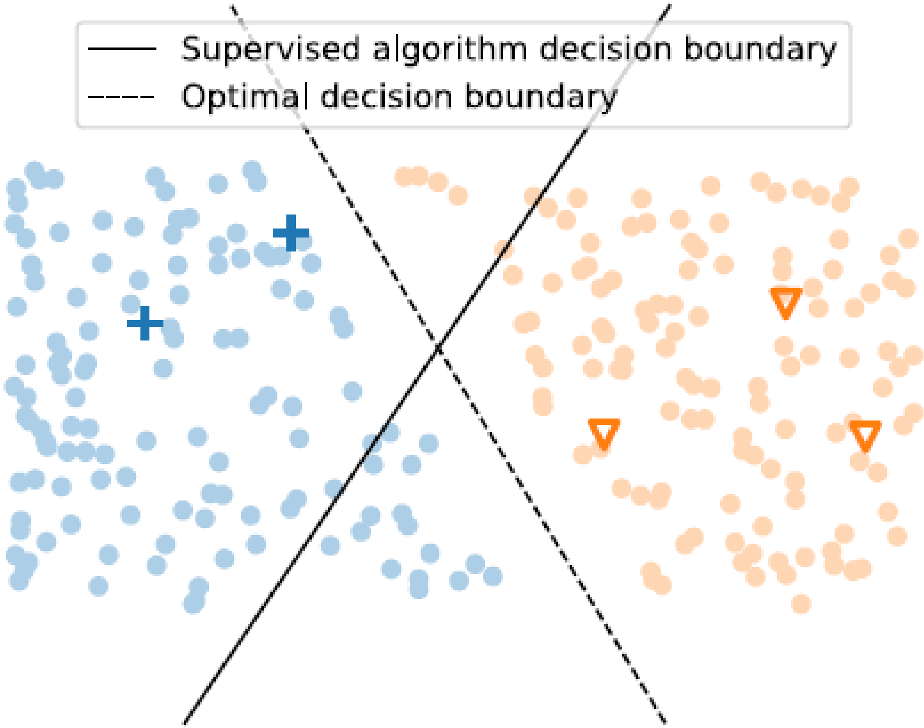
$$\|\mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta)\|_2^2.$$

Augment(*x*) is a stochastic transformation, so the two terms are not identical.

Background

2. Entropy Minimization

Density assumption: classifier's decision boundary should not pass through high-density regions.

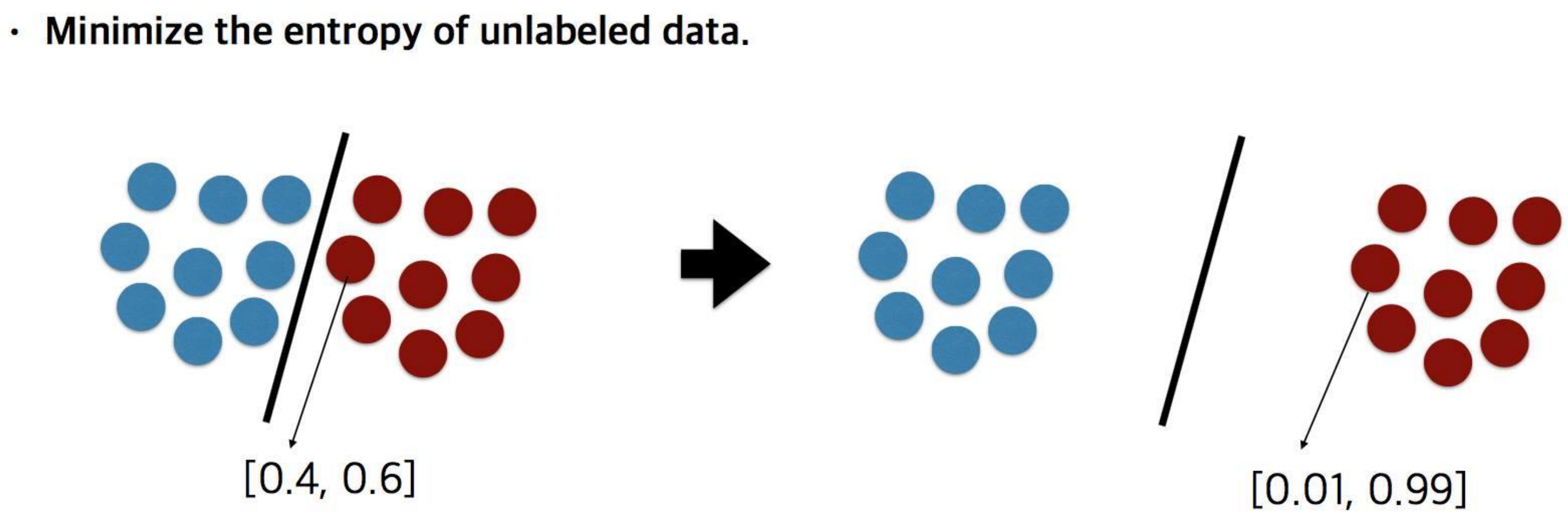


Background

2. Entropy Minimization

- One way to enforce this is to require that the classifier output low-entropy predictions on unlabeled data.

This is done explicitly with a loss term which minimizes the entropy of $p_{model}(y \mid x; \theta)$ for unlabeled data x.

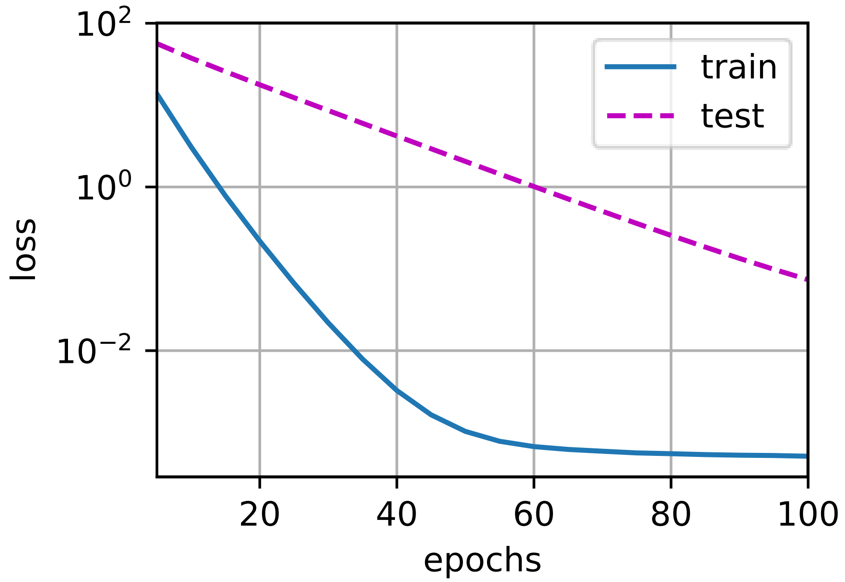


Background

3. Generic Regularization

Regularization refers to the general approach of imposing a constraint on a model to make it harder to memorize the training data and therefore hopefully make it generalize better to unseen data.

We use weight decay which penalizes the L_2 norm of the model parameters.



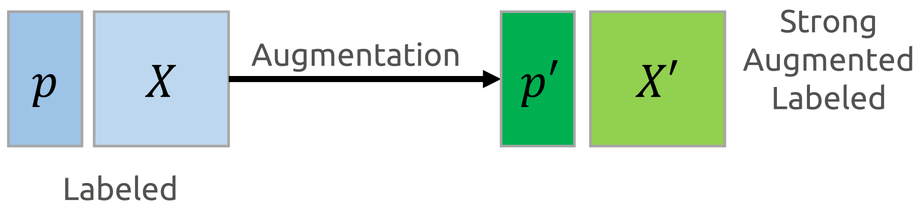
$$\min_{\theta} \sum_{x,p \in \mathcal{X}} \ell(p, \mathrm{p}_{model}(y|x;\theta)) + \lambda \|\theta\|_2^2$$

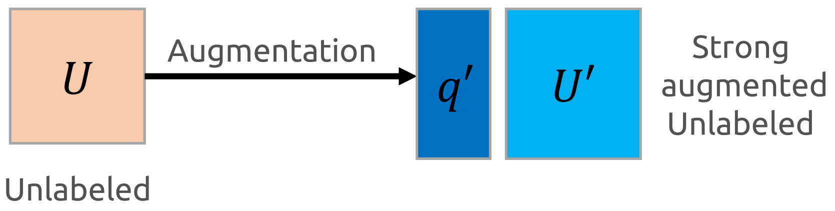
MixMatch

MixMatch introduces a unified loss term for unlabeled data that seamlessly reduces entropy while maintaining consistency and remaining compatible with traditional regularization techniques.

Given a batch X of labeled examples with one-hot targets (representing one of L possible labels) and an equally-sized batch U of unlabeled examples.

MixMatch produces a processed batch of augmented labeled examples X' and a batch of augmented unlabeled examples with "guessed" labels U'.





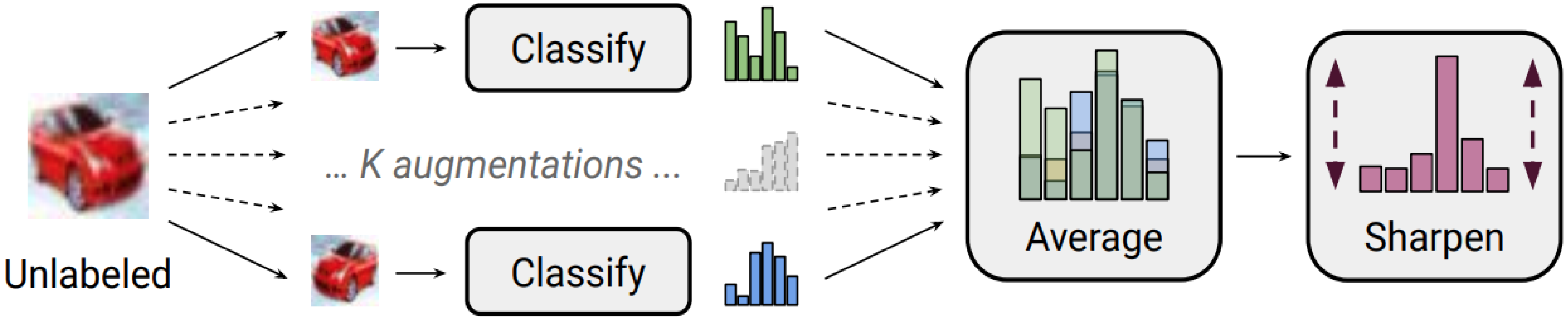
U' and X' are then used in computing separate labeled and unlabeled loss terms

MixMatch

1. Data Augmentation

For each unlabeled example in *U*, *MixMatch* produces a “guess” for the example’s label using the model’s predictions.

This guess is later used in the unsupervised loss term.



To do so, we compute the average of the model’s predicted class distributions across all the *K* augmentations of u_b by

$$\bar{q}_b = \frac{1}{K}\sum_{k=1}^{K} \mathrm{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$$

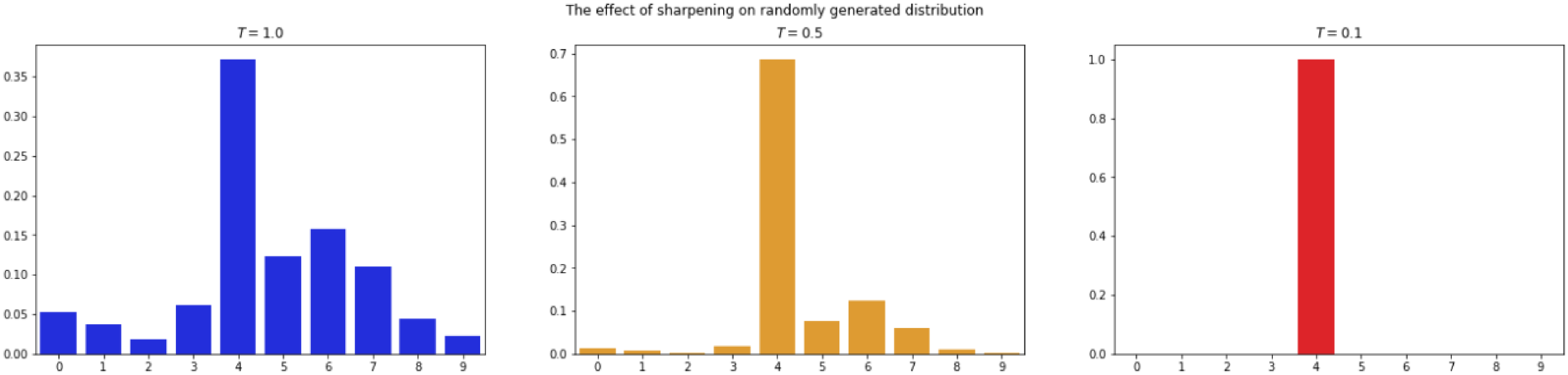
Using data augmentation to obtain an artificial target for an unlabeled example is common in consistency regularization methods.

MixMatch

2. Label Guessing and Sharpening

In generating a label guess, we perform one additional step inspired by the success of entropy minimization in semi-supervised learning.

Given the average prediction over augmentations $\bar{q}_b$, we apply a sharpening function to reduce the entropy of the label distribution.



$$\operatorname{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \bigg/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

MixUp

$$
\begin{aligned}
\hat{x} &= \lambda x_i + (1 - \lambda) x_j, \\
\hat{y} &= \lambda y_i + (1 - \lambda) y_j,
\end{aligned}
$$

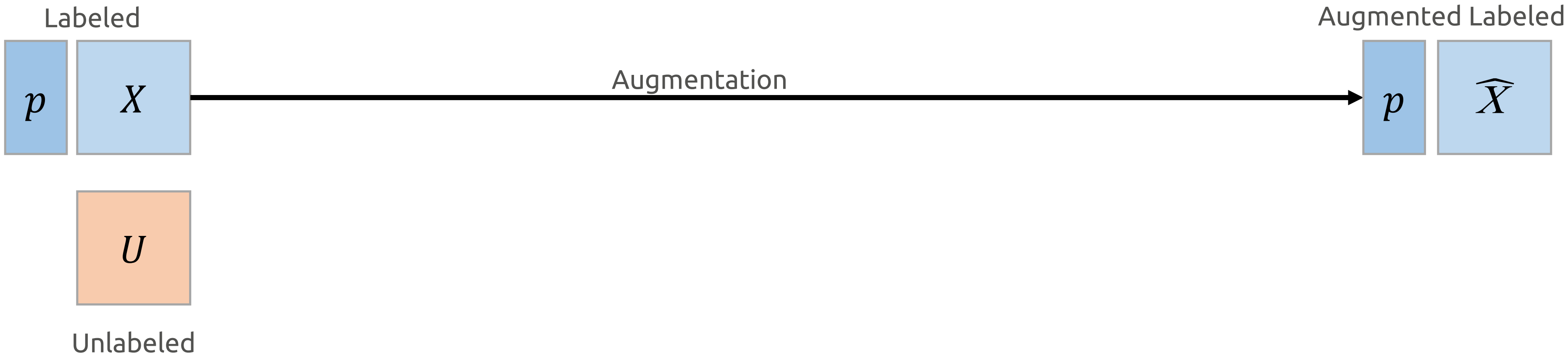
where $\lambda \in [0, 1]$ is a random number

Image

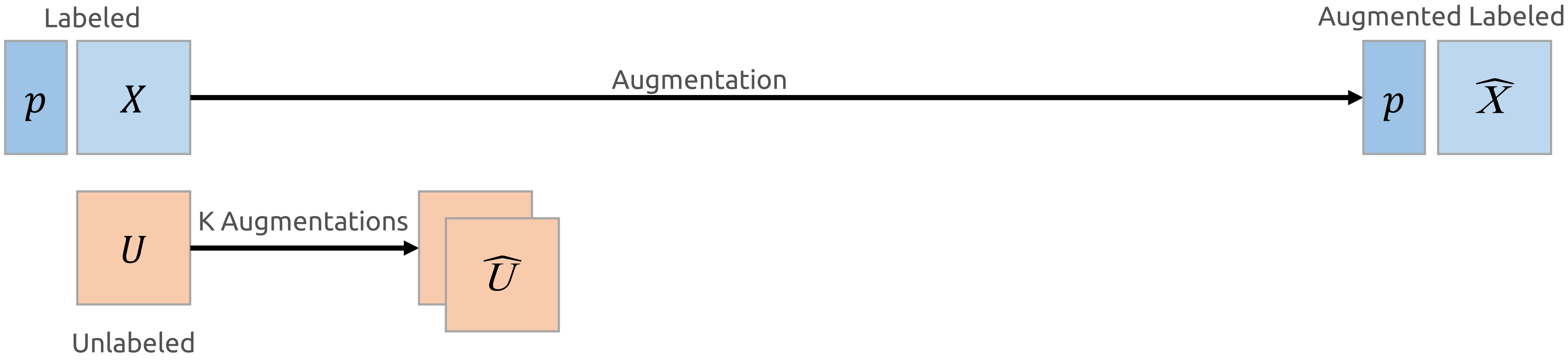
Label

[1.0, 0.0]	[0.0, 1.0]	[0.7, 0.3]
cat dog	cat dog	cat dog

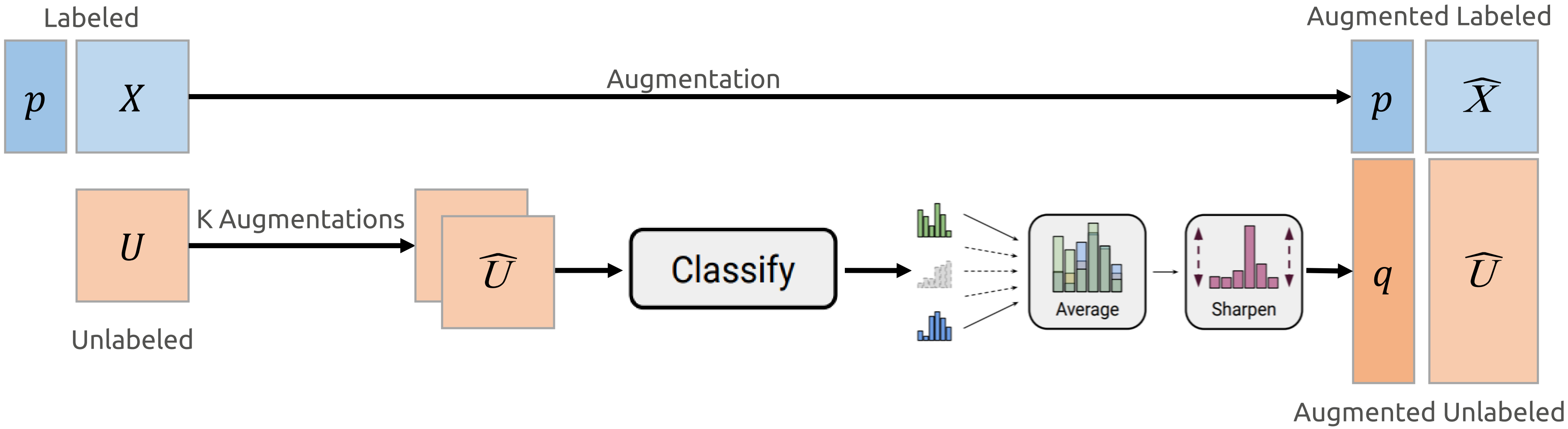
MixMatch Diagram



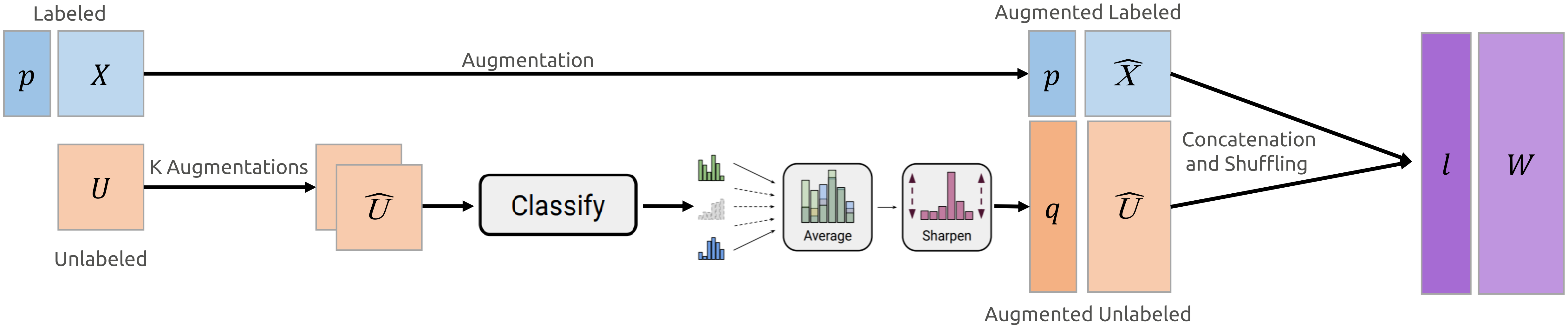
MixMatch Diagram



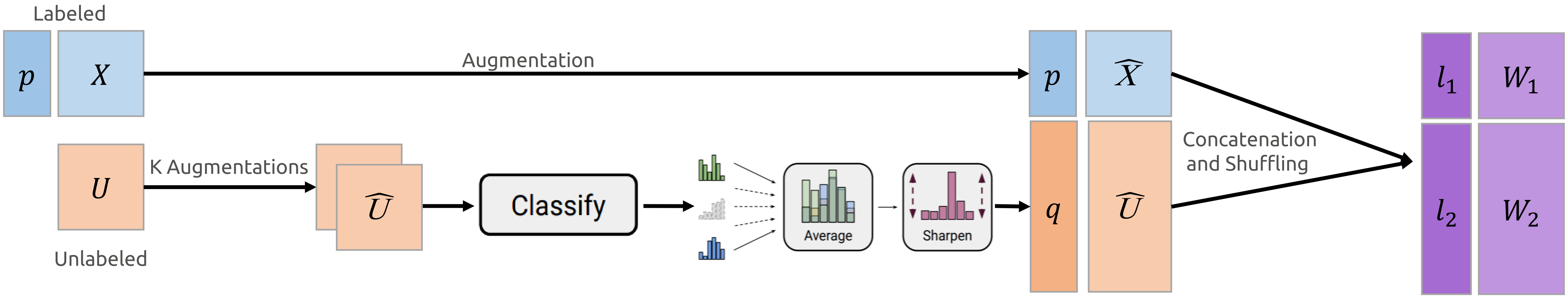
MixMatch Diagram



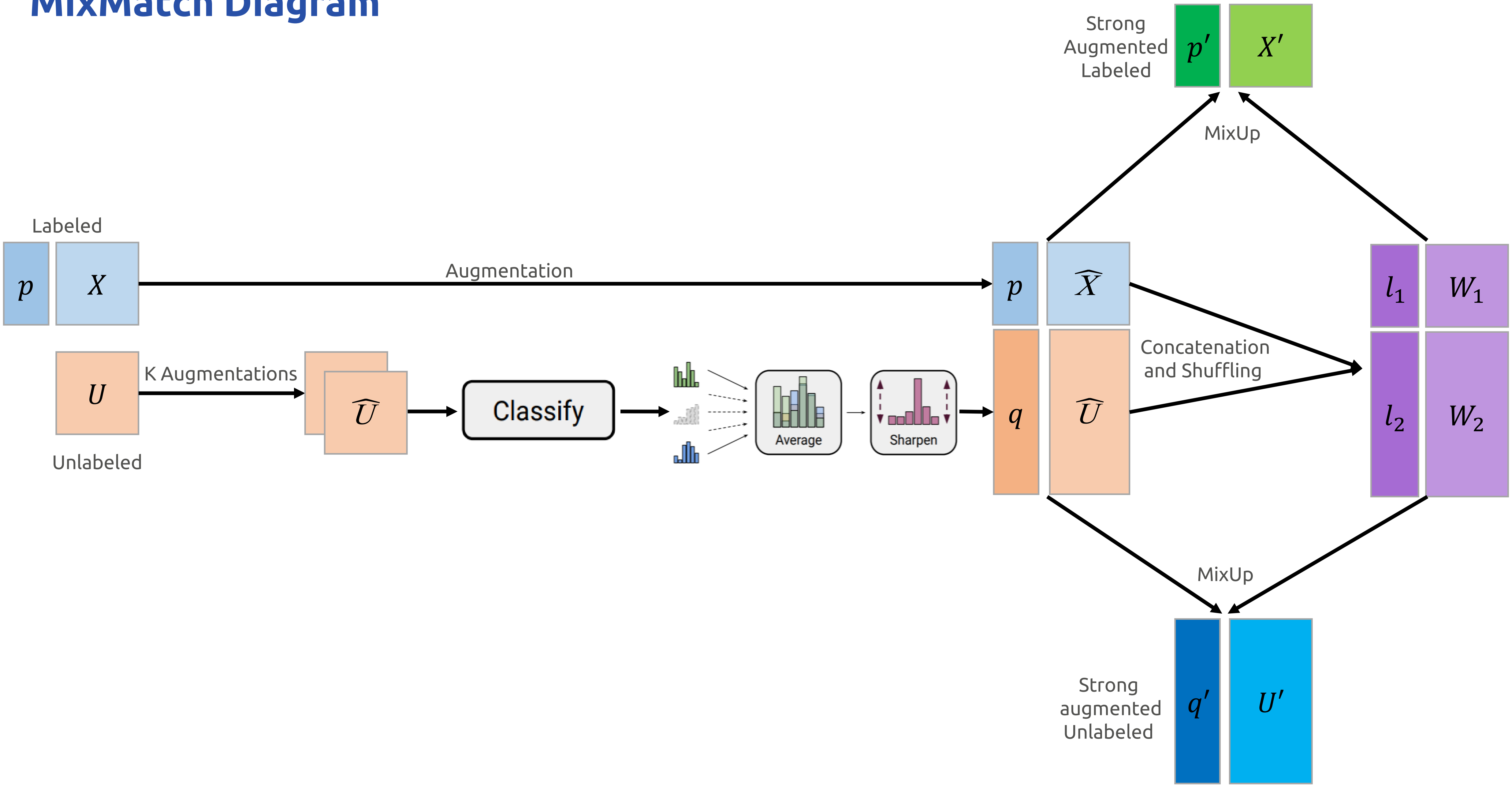
MixMatch Diagram



MixMatch Diagram



MixMatch Diagram



MixMatch

U′ and *X′* are then used in computing separate labeled and unlabeled loss terms.

More formally, the combined loss *L* for semi-supervised learning is defined as

$$\mathcal{X}', \mathcal{U}' = \mathrm{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \mathrm{H}(p, \mathrm{p}_{\mathrm{model}}(y \mid x; \theta)) + \lambda_{\mathcal{U}} \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \mathrm{p}_{\mathrm{model}}(y \mid u; \theta)\|_2^2$$

where *H(p, q)* is the cross-entropy between distributions *p* and *q*, and *T*, *K*, α, and *U* are hyperparameters.

Algorithm 1 MixMatch takes a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ (resp. $\mathcal{U}'$) of processed labeled examples (resp. unlabeled with guessed labels).

1: **Input:** Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
2: **for** $b = 1$ **to** B **do**
3:     $\hat{x}_b = \mathrm{Augment}(x_b)$ *// Apply data augmentation to x_b*
4:     **for** $k = 1$ **to** K **do**
5:         $\hat{u}_{b,k} = \mathrm{Augment}(u_b)$ *// Apply k^{th} round of data augmentation to u_b*
6:     **end for**
7:     $\bar{q}_b = \frac{1}{K} \sum_k \mathrm{p}_{\mathrm{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of u_b*
8:     $q_b = \mathrm{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*
9: **end for**
10: $\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ *// Augmented labeled examples and their labels*
11: $\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ *// Augmented unlabeled examples, guessed labels*
12: $\mathcal{W} = \mathrm{Shuffle}\big(\mathrm{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*
13: $\mathcal{X}' = \big(\mathrm{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ *// Apply MixUp to labeled data and entries from $\mathcal{W}$*
14: $\mathcal{U}' = \big(\mathrm{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ *// Apply MixUp to unlabeled data and the rest of $\mathcal{W}$*
15: **return** $\mathcal{X}', \mathcal{U}'$