Learning with labeled and unlabeled data

Author: Matthias Seeger

Presented by:

Miquel Perelló-Nieto[1] and Raúl Santos-Rodríguez[1]

[1]University of Bristol, UK

Email: [1]{Miquel.PerelloNieto, enrsr}@bristol.ac.uk

December 8, 2017

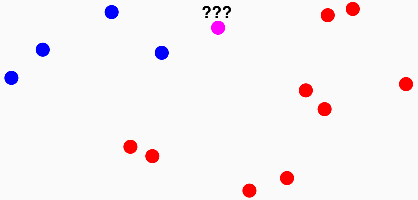
Index

Index

Motivation

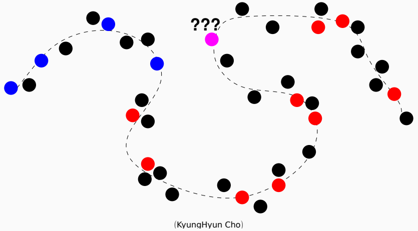
- We are overwhelmed with data
- Best learning methods need labels
- Labels are expensive (thus scarce)

Example

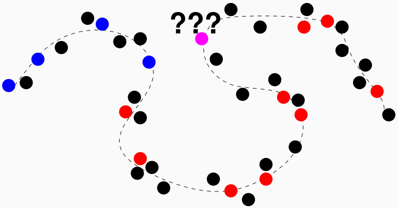


(KyungHyun Cho)

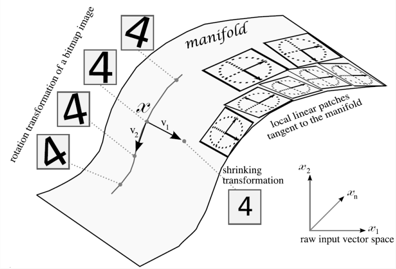
Example



Example



(KyungHyun Cho)



(Bengio, 2009)

Notation

- Learning as *drastically compress* data
- Assume a *hidden inherent simplicity* of relationships via latent variables
- A *model family* is a $P(\mathcal{A}|\mathcal{B}, \theta)$
  - $\mathcal{A}$ and $\mathcal{B}$ are disjoint sets of variables
  - $\theta \in \Theta$ is a latent variable
- *Divide and conquer*
  - A model for $\mathcal{A}|\mathcal{B}$ described as a *mixture* of models $\mathcal{A}|\{\mathcal{B}, k\}$

University of BRISTOL

Notation

- Learning as *drastically compress* data
- Assume a *hidden inherent simplicity* of relationships via latent variables
- A *model family* is a $P(\mathcal{A}|\mathcal{B}, \theta)$
  - $\mathcal{A}$ and $\mathcal{B}$ are disjoint sets of variables
  - $\theta \in \Theta$ is a latent variable
- *Divide and conquer*
  - A model for $\mathcal{A}|\mathcal{B}$ described as a *mixture* of models $\mathcal{A}|\{\mathcal{B}, k\}$

Notation

- Learning as *drastically compress* data
- Assume a *hidden inherent simplicity* of relationships via latent variables
- A *model family* is a $P(\boldsymbol{\mathcal{A}}|\boldsymbol{\mathcal{B}}, \boldsymbol{\theta})$
  - $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are disjoint sets of variables
  - $\boldsymbol{\theta} \in \Theta$ is a latent variable
- *Divide and conquer*
  - A model for $\mathcal{A}|\mathcal{B}$ described as a *mixture* of models $\mathcal{A}|\{\mathcal{B}, k\}$

Notation

- Learning as *drastically compress* data
- Assume a *hidden inherent simplicity* of relationships via latent variables
- A *model family* is a $P(\boldsymbol{\mathcal{A}}|\boldsymbol{\mathcal{B}}, \boldsymbol{\theta})$
  - $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are disjoint sets of variables
  - $\boldsymbol{\theta} \in \Theta$ is a latent variable
- *Divide and conquer*
  - A model for $\boldsymbol{\mathcal{A}}|\boldsymbol{\mathcal{B}}$ described as a *mixture* of models $\boldsymbol{\mathcal{A}}|\{\boldsymbol{\mathcal{B}}, k\}$

University of BRISTOL

Supervised vs Unsupervised learning

- Supervised learning
  - Learn a model $P(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \boldsymbol{X}$ are *input points* and $t \in T$ are *targets*
  - from *labeled* data $\{(\boldsymbol{x}_i, t_i) | i = 1, \dots, n\}$ drawn i.i.d from $P(\boldsymbol{x}, t)$
    - in *classification* $T \in \{1, \dots, C\}$
    - in *regression* $T \in \mathcal{R}$
- Unsupervised learning
  - Learn a model $P(\boldsymbol{x})$ from data $\{\boldsymbol{x}_i | i = 1, \dots, n\}$
  - E.g. clustering, anomaly detection, dimensionality reduction

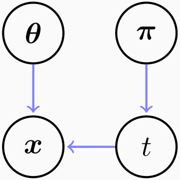
Supervised vs Unsupervised learning

- Supervised learning
  - Learn a model $P(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \boldsymbol{X}$ are *input points* and $t \in T$ are *targets*
  - from *labeled* data $\{(\boldsymbol{x}_i, t_i) | i = 1, \ldots, n\}$ drawn i.i.d from $P(\boldsymbol{x}, t)$
    - in *classification* $T \in \{1, \ldots, C\}$
    - in *regression* $T \in \mathcal{R}$
- Unsupervised learning
  - Learn a model $P(\boldsymbol{x})$ from data $\{\boldsymbol{x}_i | i = 1, \ldots, n\}$
  - E.g. clustering, anomaly detection, dimensionality reduction

Semi-supervised learning

- Learn a *predictor* $\hat{t}(\boldsymbol{x})$
- With small generalization error $P_{\boldsymbol{x},t}\{\hat{t}(\boldsymbol{x}) \neq t\}$
- Given $D = (D_l, D_u)$
  - *labeled* samples $D_l = \{\boldsymbol{x}_i, t)|i = 1, ..., n\} = (\boldsymbol{X}_l, T_l)$
  - *unlabeled* samples $D_u = \{\boldsymbol{x}_i)|i = n + 1, ..., n + m\} = (\boldsymbol{X}_u, T_u)$
    - Where $T_u = (t_{n+1}, ..., t_{n+m})$ are missing labels
  - and *prior knowledge*

Generative method



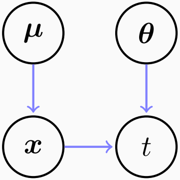
- Model $P(\boldsymbol{x}|t)$ with model families $\{P(\boldsymbol{x}|t, \boldsymbol{\theta})\}$
- and class priors $P(t)$ by $\pi_t = P(t|\boldsymbol{\pi})$
- Then by Bayes' formula

$$P(t|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\pi}) = \frac{\pi_t P(\boldsymbol{x}|t, \boldsymbol{\theta})}{\sum_{t=1}^{C} \pi_t P(\boldsymbol{x}|t, \boldsymbol{\theta})} \tag{1}$$

- Given D_l and D_u maximize the *joint log likelihood*

$$\sum_{i=1}^{n} \log \pi_t P(\boldsymbol{x}_i|t_i, \boldsymbol{\theta}) + \sum_{i=n+1}^{n+m} \log \sum_{t=1}^{C} \pi_t P(\boldsymbol{x}_i|t, \boldsymbol{\theta}) \tag{2}$$

Discriminative method



- Model $P(t|\boldsymbol{x})$ with model families $\{P(t|\boldsymbol{x}, \boldsymbol{\theta})\}$
- Can not use D_u as $P(\boldsymbol{x})$ is not modeled

Index

Simple baselines

1. Discard D_u and train only with D_l
2. Evaluate in artificial data where we know the true D_u
3. Drop known labels from D_l to generate D_u

unsupervised followed by supervised learning

1. Cluster and discriminate
   - Cluster the data into k clusters (eg. Gaussian Mixture Model)
   - Compute distances for all points to the k clusters
   - Use distances as the new features and train discriminative model (eg. k-nearest neighbor)
2. *Separator variable* k (details in page 23)
   - Assume x and t are conditionally independent given k
   - Train mixture model $P(\boldsymbol{x}, k)$
   - Fix $P(k)$ and $P(\boldsymbol{x}|k)$ and train $P(t|k)$
3. Mixture of experts
   - Learn a probabilistic partitioning $P(k|\boldsymbol{x}, \boldsymbol{\theta})$
   - Train expert k to learn $P(t|\boldsymbol{x}, k, \boldsymbol{\tau})$ weighted $P(k|\boldsymbol{x}_i, \boldsymbol{\theta})$

University of BRISTOL

unsupervised followed by supervised learning

1. Cluster and discriminate
   - Cluster the data into k clusters (eg. Gaussian Mixture Model)
   - Compute distances for all points to the k clusters
   - Use distances as the new features and train discriminative model (eg. k-nearest neighbor)
2. *Separator variable* k (details in page 23)
   - Assume $\boldsymbol{x}$ and t are conditionally independent given k
   - Train mixture model $P(\boldsymbol{x}, k)$
   - Fix $P(k)$ and $P(\boldsymbol{x}|k)$ and train $P(t|k)$
3. Mixture of experts
   - Learn a probabilistic partitioning $P(k|\boldsymbol{x}, \boldsymbol{\theta})$
   - Train expert k to learn $P(t|\boldsymbol{x}, k, \boldsymbol{\tau})$ weighted $P(k|\boldsymbol{x}_i, \boldsymbol{\theta})$

University of BRISTOL

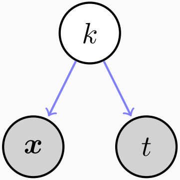
unsupervised followed by supervised learning

1. Cluster and discriminate
   - Cluster the data into k clusters (eg. Gaussian Mixture Model)
   - Compute distances for all points to the k clusters
   - Use distances as the new features and train discriminative model (eg. k-nearest neighbor)
2. *Separator variable* k (details in page 23)
   - Assume $\boldsymbol{x}$ and t are conditionally independent given k
   - Train mixture model $P(\boldsymbol{x}, k)$
   - Fix $P(k)$ and $P(\boldsymbol{x}|k)$ and train $P(t|k)$
3. Mixture of experts
   - Learn a probabilistic partitioning $P(k|\boldsymbol{x}, \boldsymbol{\theta})$
   - Train expert k to learn $P(t|\boldsymbol{x}, k, \boldsymbol{\tau})$ weighted $P(k|\boldsymbol{x}_i, \boldsymbol{\theta})$

Expectation-maximization overview

- Learning of latent variables
- Need of prior knowledge
- Two iterative steps
    1. E-step:
        - Assume the model is correct (fix the model)
        - Estimate latent variables or labels
    2. M-step:
        - Assume the estimates are correct (fix the latent variables)
        - Modify the model to maximize a performance metric towards the estimates

Expectation-maximization with separator variable k



- Introduce a *separator variable* k
- joint log likelihood

$$\sum_{i=1}^{n} \log \sum_{k} \beta_{t_i,k} \pi_k P(\boldsymbol{x}_i | k, \boldsymbol{\theta}) + \sum_{i=n+1}^{n+m} \log \sum_{k} \pi_k P(\boldsymbol{x}_i | k, \boldsymbol{\theta}) \quad (3)$$

Index

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis θ on X is *compatible* with distribution $P(x)$ if:
  - hypotheses $\theta^{(1)}$,$\theta^{(2)}$ on $X^{(1)}$,$X^{(2)}$ if for any $x = (x^{(1)}, x^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\theta^{(1)}$,$\theta^{(2)}$ on $x^{(1)}, x^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\theta^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\theta^{(1)}$,$\theta^{(2)}$ on $x^{(1)}, x^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\theta^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\theta^{(1)}$,$\theta^{(2)}$ on $x^{(1)}, x^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\theta^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\boldsymbol{\theta}^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\boldsymbol{\theta}^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\boldsymbol{\theta}^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\boldsymbol{\theta}^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Co-Training algorithms [Blum and Mitchell, 1998]

- *Compatibility* between different views of an example $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \boldsymbol{X} = \boldsymbol{X}^{(1)} \times \boldsymbol{X}^{(2)}$
- Hypothesis $\boldsymbol{\theta}$ on $\boldsymbol{X}$ is *compatible* with distribution $P(\boldsymbol{x})$ if:
  - hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{X}^{(1)}$,$\boldsymbol{X}^{(2)}$ if for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ predict the same labels
- It assumes that the views are conditionally independent
- Steps:
  1. Start with set $D_w = D_l$ and learn two hypotheses $\boldsymbol{\theta}^{(1)}$,$\boldsymbol{\theta}^{(2)}$ on $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$
  2. Increase set D_w with some samples from D_u and the predicted targets from one hypothesis $\boldsymbol{\theta}^{(j)}$
  3. Train the other hypothesis with the augmented D_w
  4. Alternate the hypotheses as *student* and *teacher*

Restricted Bayes Optimal Classification [Tong and Koller, 2000]

- Usual discriminators use a *loss function* $L(h(\boldsymbol{x}), t)$ and *regularization functional* $\mathcal{R}(h)$
- We want a hypothesis which minimizes the tradeoff

$$E_{P_{emp}}[L(h(\boldsymbol{x}), t)] + \lambda\mathcal{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\boldsymbol{x}_i), t_i) + \lambda\mathcal{R}(h) \quad (4)$$

- where $P_{emp}(\boldsymbol{x}, t) = n^{-1}\sum_i \delta((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i))$ is the *empirical distribution* of the data D_l and λ is a tradeoff parameter
- Steps:
  1. Estimate $P(\boldsymbol{x}, t)$ from data D_l, D_u called $\hat{P}(\boldsymbol{x}, t)$
  2. Minimize the following instead of Eq. 4

$$E_{\hat{P}}[L(h(\boldsymbol{x}), t] + \lambda\mathcal{R}(h)$$

Restricted Bayes Optimal Classification [Tong and Koller, 2000]

- Usual discriminators use a *loss function* $L(h(\boldsymbol{x}), t)$ and *regularization functional* $\mathcal{R}(h)$
- We want a hypothesis which minimizes the tradeoff

$$E_{P_{emp}}[L(h(\boldsymbol{x}), t)] + \lambda\mathcal{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\boldsymbol{x}_i), t_i) + \lambda\mathcal{R}(h) \quad (4)$$

- where $P_{emp}(\boldsymbol{x}, t) = n^{-1}\sum_i \delta((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i))$ is the *empirical distribution* of the data D_l and λ is a tradeoff parameter
- Steps:
  1. Estimate $P(\boldsymbol{x}, t)$ from data D_l, D_u called $\hat{P}(\boldsymbol{x}, t)$
  2. Minimize the following instead of Eq. 4

$$E_{\hat{P}}[L(h(\boldsymbol{x}), t] + \lambda\mathcal{R}(h)$$

Restricted Bayes Optimal Classification [Tong and Koller, 2000]

- Usual discriminators use a *loss function* $L(h(\boldsymbol{x}), t)$ and *regularization functional* $\mathcal{R}(h)$
- We want a hypothesis which minimizes the tradeoff

$$E_{P_{emp}}[L(h(\boldsymbol{x}), t)] + \lambda\mathcal{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\boldsymbol{x}_i), t_i) + \lambda\mathcal{R}(h) \quad (4)$$

- where $P_{emp}(\boldsymbol{x}, t) = n^{-1}\sum_i \delta((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i))$ is the *empirical distribution* of the data D_l and λ is a tradeoff parameter
- Steps:
  1. Estimate $P(x, t)$ from data D_l, D_u called $\hat{P}(x, t)$
  2. Minimize the following instead of Eq. 4

$$E_{\hat{P}}[L(h(\boldsymbol{x}), t] + \lambda\mathcal{R}(h)$$

Restricted Bayes Optimal Classification [Tong and Koller, 2000]

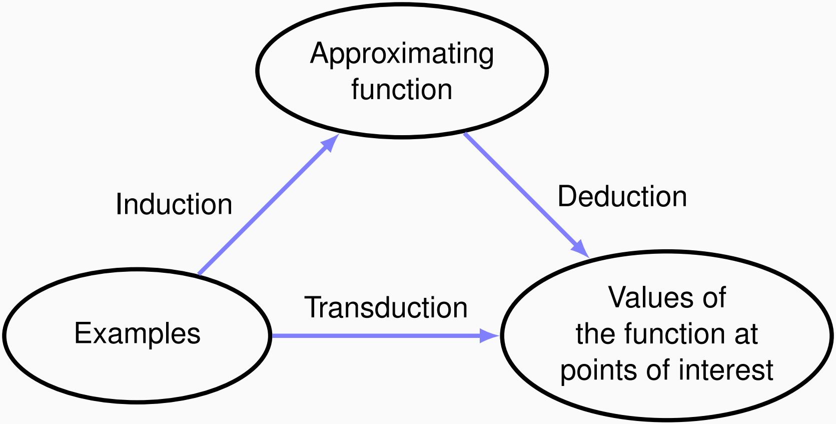
- Usual discriminators use a *loss function* $L(h(\boldsymbol{x}), t)$ and *regularization functional* $\mathcal{R}(h)$
- We want a hypothesis which minimizes the tradeoff

$$E_{P_{emp}}[L(h(\boldsymbol{x}), t)] + \lambda\mathcal{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\boldsymbol{x}_i), t_i) + \lambda\mathcal{R}(h) \quad (4)$$

- where $P_{emp}(\boldsymbol{x}, t) = n^{-1}\sum_i \delta((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i))$ is the *empirical distribution* of the data D_l and λ is a tradeoff parameter
- Steps:
  1. Estimate $P(\boldsymbol{x}, t)$ from data D_l, D_u called $\hat{P}(\boldsymbol{x}, t)$
  2. Minimize the following instead of Eq. 4

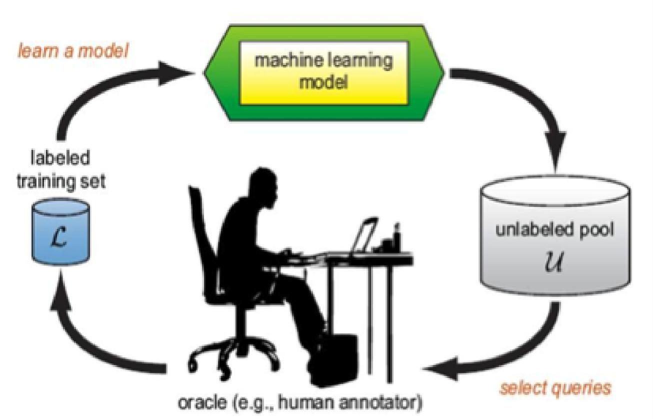
$$E_{\hat{P}}[L(h(\boldsymbol{x}), t] + \lambda\mathcal{R}(h)$$

Transduction [Vapnik and Kotz, 1982]



Index

Active learning



Coaching [Tibshirani and Hinton, 1998]

- $\mathbf{x}, t, z \sim P(\mathbf{x}, t, z)$
- but z is expensive or difficult to collect.

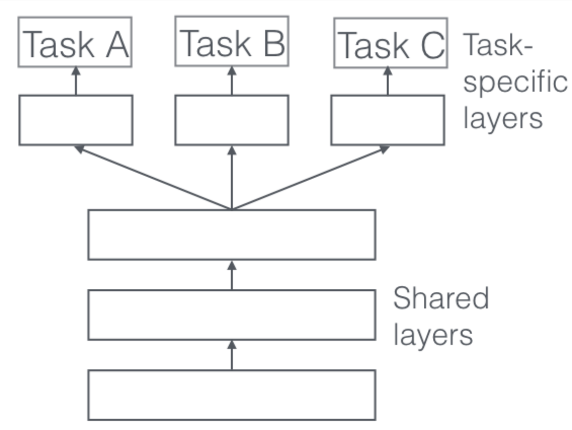
$$P(t|\mathbf{x}) = \int P(t|\mathbf{x}, z)P(z|\mathbf{x})dz$$

Coaching [Tibshirani and Hinton, 1998]

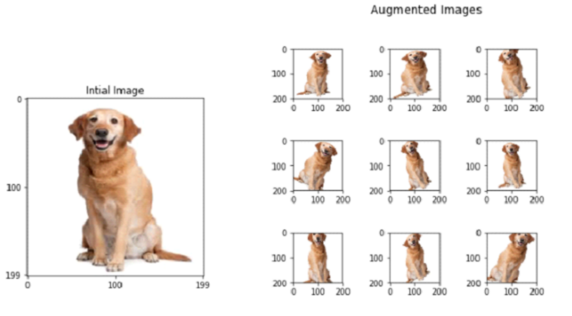
- $\mathbf{x}, t, z \sim P(\mathbf{x}, t, z)$
- but z is expensive or difficult to collect.

$$P(t|\mathbf{x}) = \int P(t|\mathbf{x}, z)P(z|\mathbf{x})dz$$

Multitask Learning



Data Augmentation



Index

Caveats and tradeoffs

- Labels as missing data?
- The sampling assumption: is iid realistic?

Caveats and tradeoffs

- Labels as missing data?
- The sampling assumption: is iid realistic?

References I

Blum, A. and Mitchell, T. (1998).
Combining labeled and unlabeled data with co-training.
In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Seeger, M. (2002).
Learning with labeled and unlabeled data.
Technical report.

Tibshirani, R. and Hinton, G. (1998).
Coaching variables for regression and classification.
Statistics and Computing, 8(1):25–33.

Tong, S. and Koller, D. (2000).
Restricted bayes optimal classifiers.
In *AAAI/IAAI*, pages 658–664.

References II

Vapnik, V. N. and Kotz, S. (1982).
Estimation of dependences based on empirical data, volume 40.
Springer-Verlag New York.

Learning with labeled and unlabeled data

Author: Matthias Seeger

Presented by:

Miquel Perelló-Nieto[1] and Raúl Santos-Rodríguez[1]

[1]University of Bristol, UK

Email: [1]{Miquel.PerelloNieto, enrsr}@bristol.ac.uk

December 8, 2017