Learning with labeled and unlabeled data

Author:
Matthias Seeger, Institute for Adaptive Neural Computation, University of Edinburgh

Presented by:
William Elazmeh, Ottawa-Carleton Institute for Computer Science, Canada

Outline

- **Supervised vs. unsupervised learning**
- **Supervised learning aided by additional unlabeled data**
- Paradigms for supervised classification
  - **Sampling**
  - **Diagnostic**
  - Regularization depending on input distribution
- Baseline Methods:
  - Unsupervised learning, then cluster assignment
  - **Expectation-maximization techniques**
  - Expectation-maximization with separator
  - Expectation-maximization on diagnostic models

Outline (continued..!)

- Literature review
  - Early work
  - Expectation-maximization on a joint density model
  - **Co-training (paper 2: Understanding the behavior of Co-training)**
  - Adaptive regularization
  - The Fisher kernel
  - Restricted Bayes Optimal Classification
  - Transduction

Outline (continued..!)

- Related Problems
  - Active learning
  - Coaching, learning how to learn
  - Transfer of knowledge from a related task
- Caveats and trade offs
  - Labels as missing data
  - Diagnostic versus generative methods
  - The sampling assumption

The Problem

- compress data without loss of information
- Occam's razor: *hidden inherent simplicity of relationships*
- Knowledge of the *latent* variables reduces the complexity of describing the *observables*
- A *model* family is a conditional probability distribution $P(A|B, \theta)$, where
  - A and B are disjoint sets of variables
  - $\theta \in \Theta$ is a latent variable associated with the model family $\{P(A|B, \theta)|\theta \in \Theta\}$
  - The elements $A|B$ are indexed by the values of θ

The Problem Divided

- Introduce a clustering variable k of a finite range
- $A|B$ can be described by $A|B, k$
- An alternative option is to use functional relationships to describe a functional model where the objective is to separate structure from noise models

Supervised learning

- examples $x \in X$
- Labels $t \in T$
- An unknown probabilistic relationship $P(x, t)$
- Learn from data $\{(x_i, t_i)|i = 1, \cdots, n\}$
- (x_i, t_i) are drawn independently from $P(x, t)$
- Classification or pattern recognition (T is finite)
- Regression ($T \in R$)

Unsupervised learning

- Follows a well-defined goal
- Minimize the generalization error in classification
- Minimize the expected loss in regression
- No definitive criteria but ”interesting structures”
- Samples $\{x_i|i = 1, \cdots, m\}$ are drawn independently from $P(x)$
- Perform a density estimation
- Principal Component Analysis (a latent variable u, noise over $x|u$ in 2-d)
- Factor Analysis (relational over the prior $P(\theta)$)
- Mixture Models (a latent variable is a grouping variable from a finite set)

Unsupervised learning aided by additional unlabeled data

- Classification problem $P(x, t)$ with unlabeled data
- labeling x from $P(x)$ is expensive according t $P(t|x)$
- Given an unknown probabilistic relationship $P(x, t)$ between data points x and class labels $t \in T = \{1, \cdots, c\}$
- Predict t from x, i.e. find a predictor $\hat{t} = \hat{t}(x)$ such that the generation error of $\hat{t}$, $P_{x,t}\{\hat{t}(x) \neq t\}$ is small (close to Bayes error)

Unsupervised learning aided by additional unlabeled data (continued..!)

- An algorithm computes $\hat{t}$ from:
  - labeled sample $D_l = \{(x_i, t_i) | i = 1, \cdots, n\}$ where (x_i, t_i) are drawn independently from $P(x, t)$
  - unlabeled sample $D_u = \{x_i | i = n + 1, \cdots, m\}$ where x_i are drawn independently from the marginal distribution $P(x) = \Sigma_{t=1}^{c} P(x, t)$
  - Prior knowledge about the unknown relationship
- D_u is empty, then supervised learning
- Interesting case, $n = |D_l|$ is small and $m = |D_u| \gg n$

The sampling paradigm (generative methods)

- Model the class distributions $P(x|t)$ using model family $P(x|t, \theta)$
- Class priors $P(t)$ are modeled by $\pi_t = P(t|\pi)$
- This is called a joint density model because we model $P(x, t)$ by $\pi_t P(x|t, \theta)$
- For a fixed $\hat{\theta}$ and $\hat{\pi}$, estimate $P(t|x)$ by Bayes formula:

$$P(t|x, \hat{\theta}, \hat{\pi}) = \frac{\hat{\pi}_t P(x|t, \hat{\theta})}{\Sigma_{t'=1}^{c} \hat{\pi}_t' P(X|t', \hat{\theta})}$$

- We can obtain the predictive Bayesian predictive distribution $P(x|t, D_l)$ by averaging $P(x|t, \theta, \pi)$ over the posterior $P(\theta, \pi|D_l)$
- We have labeled and unlabeled examples, we maximize the joint log likelihood of both D_l and D_u:

$$\sum_{i=1}^{n} log \pi_{t_i} P(x_i|t_i, \theta) + \sum_{i=n+1}^{n+m} log \sum_{t=1}^{c} \pi_t P(x_i|t, \theta)$$

The diagnostic paradigm (diagnostic methods)

- Model conditional distribution $P(x|t)$ directly using $\{P(t|x,\theta)\}$ to get a complete sampling of data
- Also, model $P(x)$ using $P(x|\mu)$
- We are interested in updating θ only or in predicting t on unseen points
- θ and μ are a-priori independent, $P(\theta,\mu) = P(\theta)P(\mu)$
- The likelihood factor is:

$$P(D_l, D_u|\theta,\mu) = P(T_l|X_l,\theta)P(X_l, D_u|\mu)$$

- which implies:
  - $P(\theta|D_u, D_l)$ is proportional to $P(T_l|X_l,\theta)P(\theta)$ thus $P(\theta|D_u, D_l) = P(\theta|D_l)$
  - θ and μ are a-posteriori independent
  - $P(\theta|D_l,\mu) = P(\theta|D_l)$

The expectation-maximization algorithm

- Used for learning in the presence of unobservable variables
- We need to know the general form of the probability distribution governing these variables
- The EM algorithm can be used to:
  - Train Bayesian belief networks
  - Train radial basis function networks
  - Unsupervised clustering algorithm
  - Basis for forward-backward algorithm for learning Partially Observable Markov Models

The EM algorithm

- Data D is generated by a probability distribution of k normal distributions
- Simplify $k = 2$, each data point is generated by:
  - randomly select one of the k normal distributions
  - generate a single random data point x_i according to the selected distribution
- A special case where step 1 has a uniform probability and the k normal distributions have the same variance σ^2 (known!)
- The learning outputs the hypothesis $h = (\mu_1, \cdots, \mu_k)$
- Find the maximum likelihood hypothesis of the means to maximize $P(D|h)$
- For a single normal distribution
  - The sum of squared errors is minimized by the sample mean: $\mu_{ML} = \frac{1}{n} \Sigma_{i=1}^{n} x_i$

The EM algorithm (continued..!)

- For a mixture of k different normal distributions: hidden variables! then we have data of the form (x_i, z_{i1}, z_{i2}) where z_i indicates which distribution the data point was generated from
- The EM algorithm searches for the maximum likelihood hypothesis by
  - repeatedly re-estimating the expected values of the hidden variables z_{ij}

    $$E[z_{ij}] = \frac{P(x = x_i|\mu = \mu_i)}{\Sigma_{n=1}^{2} P(x = x_i|\mu = \mu_i)} = \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\Sigma_{n=1}^{2}\, e^{-\frac{1}{2\sigma^2}(x_i-\mu_n)^2}}$$

  - recalculating the maximum likelihood using these expected values

    $$\mu_j = \frac{1}{n}\sum_{i=1}^{n} E[z_{ij}]x_i$$

Problems with EM algorithm

- Can get stuck in a local optima (reasonable high marginal likelihood)
- On some models containing structural choices, the M step is intractably hard
- A standard fix is simulated annealing (run a sequence of EM algorithms on data and use its solution to initialize the next one) to find a reasonable deep optimum

Co-Training algorithm

- Addresses the problem where strong structural prior knowledge is present
- A robust variant of the EM algorithm to compute a MAP approximation to Bayesian inference if we assume compatibility of target concept and the input is a conditional prior
- Differences between EM and Co-training:
    - Feature split
    - Labeling unlabeled data (EM does them all in each round!)
    - EM uses all unlabeled example, while Co-training is incremental

Co-Training experiments

- Create 2 class problem from 4 data sets
- First 2 data sets provide +ves
- Second 2 datasets provide -ves
- Words in 1st and 3rd datasets are from the same vocabulary
- Words in 2nd and 4th datasets are from the different vocabulary
- true class-conditional independence
- redundancy between features
- run random test/split for co-training
- EM and naive Bayes use 6 labeled, 1000 unlabeled, and 976 tests

Experimental algorithms

- Co-training using feature split
- Co-Training EM is an iterative algorithm that uses feature split. First, train A with A-set from labeled data, then A probabilistically labels the unlabeled examples. The train B with B-set which uses the labeled examples (originally and those produced by A) and relabels the unlabeled examples
- EM algorithm
- Self-training is an incremental algorithm without using feature split. Initially, it builds a classifier from the labeled examples, then converts most confidently predicted examples into labels of training examples and reuses them for next iteration until all examples are labeled.

Experimental Results

Dataset with an independent feature split

Method	Labeling	Feature Split	Error
co-training	incremental	uses	3.7%
co-EM	iterative	uses	3.3%
EM	iterative	ignores	8.9%
self-training	incremental	ignores	5.8%

Dataset with random feature split

Method	Labeling	Feature Split	Error
training	incremental	uses	5.5%
co-EM	iterative	uses	5.1%
EM	iterative	ignores	8.9%
self-training	incremental	ignores	5.8%

Conclusions

- Co-training performs better than EM when feature set independence is a valid assumption
- EM uses naive Bayes classifier to assign class probabilities for unlabeled examples. These are poorly estimated because in text data word independence is violated. Co-training makes limited use of the underlying assumptions of independence.
- EM is likelihood-based and is not specific to the classification task and suffers when the natural clustering of unlabeled examples does not correspond to class-based cluster.
- Co-training is more discriminant, it adds examples to its labeled set to help the classification