RBMs&LR for Discrimination

Yunfei WANG

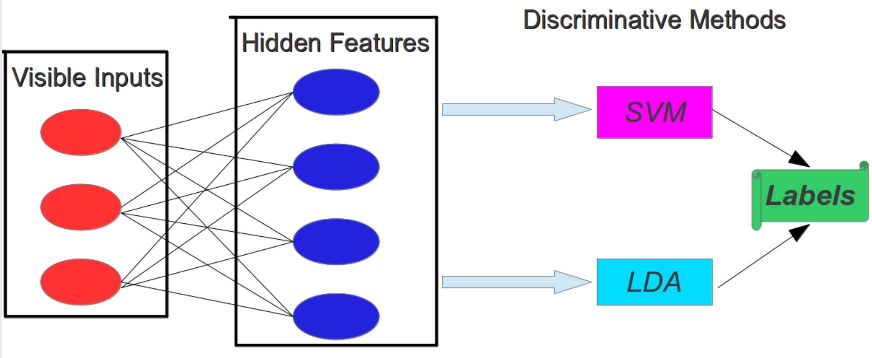
[1]School of Computer Science & Technology
Huazhong University of Science & Technology

May 20, 2013

Table of contents

Using Hidden Features directly

Hidden Features+Discriminative Methods

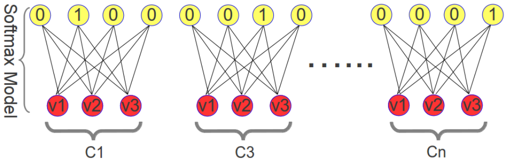


Train RBM for each class

Log probability for visible vector of RBM trained on class c:

$$\log p(v|c) = -F_c(v) - \log Z_c \tag{1}$$

where $F_c(v)$ is free energy of visible vector, Z_c is partition function of RBM for class c which is different for each class-specific RBM.

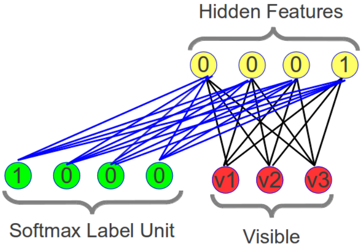


Predict class from free energy of all class-specific RBMs:

$$\log p(c|v) = \frac{\exp(-F_c(v) - \log \hat{Z}_c)}{\sum_d \exp(-F_d(v) - \log \hat{Z}_d)} \tag{2}$$

Train a joint density model

A joint density model using a single RBM with two sets of visible units:



The probability of picking the cth class:

$$\log p(c|v) = \frac{\exp(-F_c(v))}{\sum_d \exp(-F_d(v))} \tag{3}$$

Partition function is same for all classes.Apparently,the one with lowest free energy is chosen as the most likely class.

Free Energy for visible vector

The energy of each pair of visible vector v and hidden vector h:

$$E(v,h) = -v^T W h - a^T v - b^T h \tag{4}$$

We have the following equation:

$$\begin{aligned}
\exp(-F(v)) &= \sum_h \exp(-E(v,h)) \\
&= \exp(a^T v) \sum_h \exp(v^T W h + b^T h) \\
&= \exp(a^T v) \prod_j \sum_{h_j \in \{0,1\}} \exp(\sum_i W_{ij} h_j + b_j h_j) \\
&= \exp(a^T v) \prod_j (1 + \exp(\sum_i v_i W_{ij} + b_j))
\end{aligned} \tag{5}$$

Free Energy of v:$F(v) = -\sum_i v_i a_i - \sum_j \log(1 + \exp(\sum_i v_i W_{ij} + b_j))$.

Binary Case

Given two classes C_1 and C_2,transform linear decision function $y(x) = w^T x + b$ to model class-belonging probabilities $P(C_i|x)$:

$$y(x) = \log(\frac{P(C_1|x)}{P(C_2|x)}) \Longrightarrow P(C_1|x) = \frac{1}{1+e^{-y(x)}} = \frac{1}{1+e^{-(w^T x+b)}} \tag{6}$$

Assume training data $x_1, \cdots, x_n$ form a random sample from a sequence of n Bernoulli trails,the likelihood of observation $y_1, \cdots, y_n$ is:

$$L = \prod_{j=1}^{n} P(C_1|x_j)^{y_j} (1 - P(C_1|x_j))^{1-y_j} \tag{7}$$

Add extra regularisation term to avoid overfitting and instability:

$$\log L = \sum_{j=1}^{n} [y_j P(C_1|x_j) + (1-y_j)(1 - P(C_1|x_j))] - \lambda \|w\| \tag{8}$$

Multiclass Situation

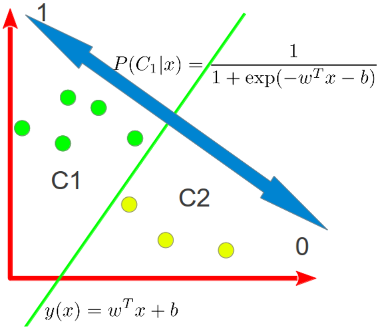
For multinominal LR with k classes,the decision function takes the following form:

$$P(C_i|x) = \frac{exp(w_i^T x)}{\sum_j \exp(w_j^T x)} \tag{9}$$

where $\exp(w_i^T x)$ is proportional to $P(C_i|X)$ under binary case.

Optimize weight vectors w_j:One-against-All or Pairwise.

Insight into posterior probability and geometric location:



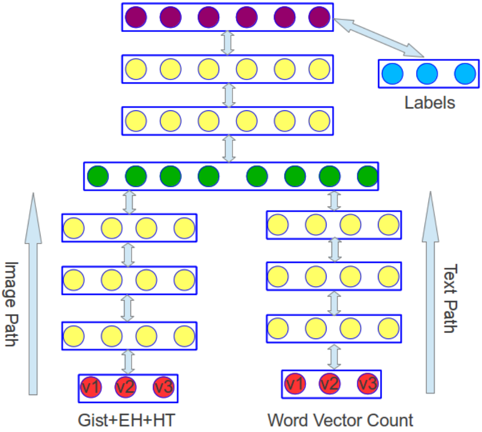


Figure: Framework of Processing