Auto-tagging music with a discriminative RBM

Razvan Pascanu

work done with

Michael Mandel
Hugo Larochelle
Yoshua Bengio

Outline

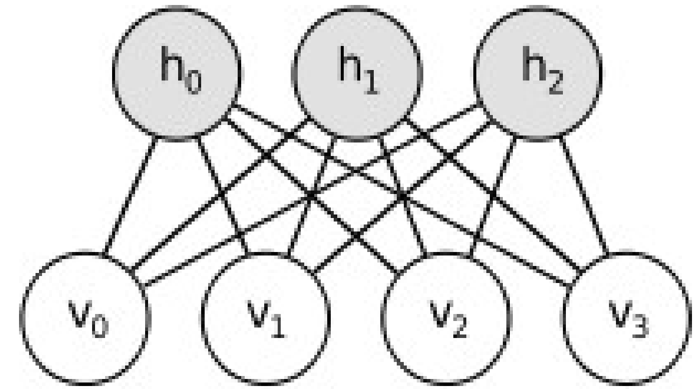
- Auto-tagging music
- Restricted Boltzmann Machines
- Tag smoothing
- Discriminative RBMs
- Multi-label Discriminative RBMs
- Data-sets
- Results
- Summary

Auto-tagging

- The increased amount of music easily available online requires better tools for searching and exploring
- Tags (short textual description given by users like rock, guitar, rhythmic, etc.) proved to be a popular solution
- Tags lead to the cold start problem for items that are new or niche, which can be solved by auto-tagging

Restricted Boltzmann Machines

- The RBMs is an energy based model where



$$p(v,h)=\frac{1}{Z}e^{-E(v,h)} \text{ given that}$$

$$E(v,h)=-h^T U v - c^T h - d^T v \ , \ Z=\sum_{v,h} e^{-E(v,h)}$$

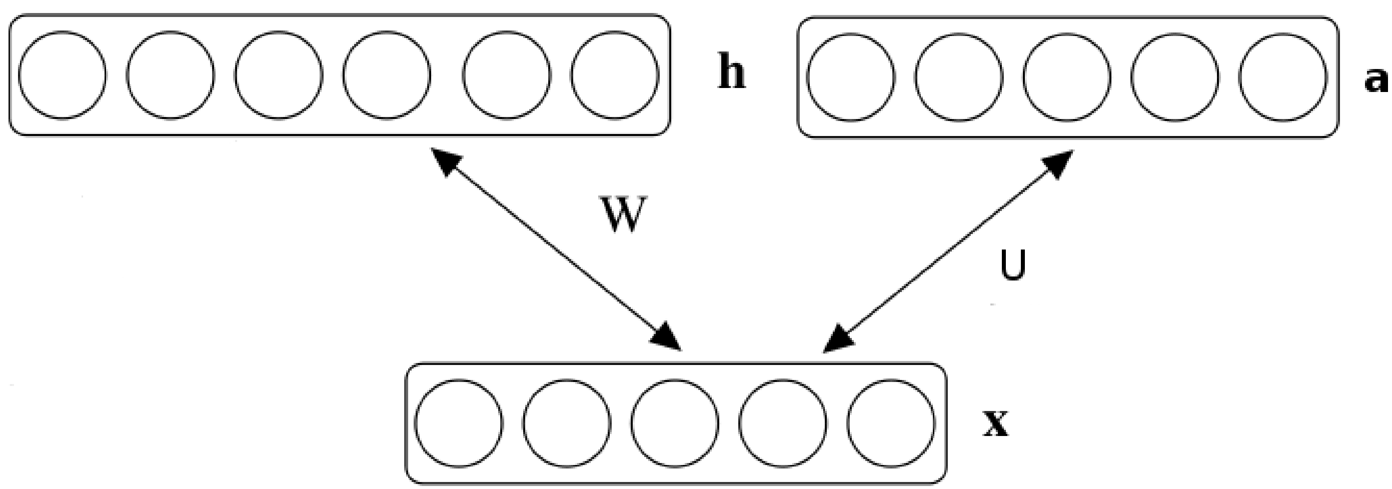
$$F(v)=-\log\sum_h e^{-E(v,h)}=-d^T v-\sum_i \log(1+e^{c_i+U_i v})$$

$$\frac{\partial}{\partial\theta}p(v_t)=-\boldsymbol{E}_{\boldsymbol{h}|\boldsymbol{v}_t}[\frac{\partial}{\partial\theta}E(v_t,h)]+\boldsymbol{E}_{\boldsymbol{v},\boldsymbol{h}}[\frac{\partial}{\partial\theta}E(v,h)]$$

Restricted Boltzmann Machines

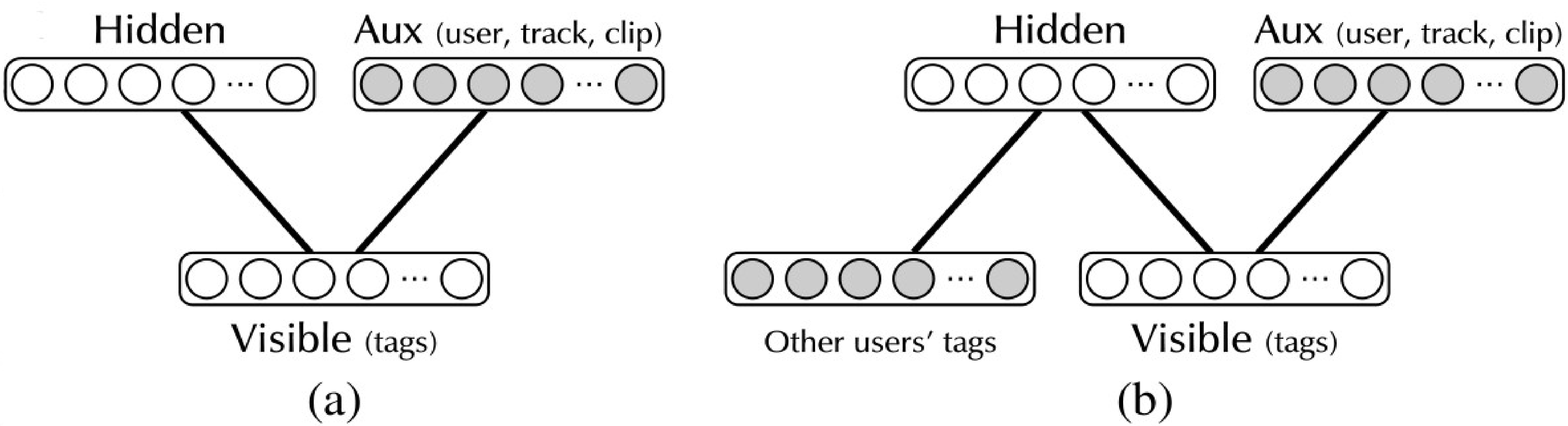
- RBMs can be conditioned on other variables (**a** in this case) by changing the energy function to (Taylor et al. 2007):

$$E(a,x,h) = -a^T U x - h^T W x - c^T h - d^T x$$



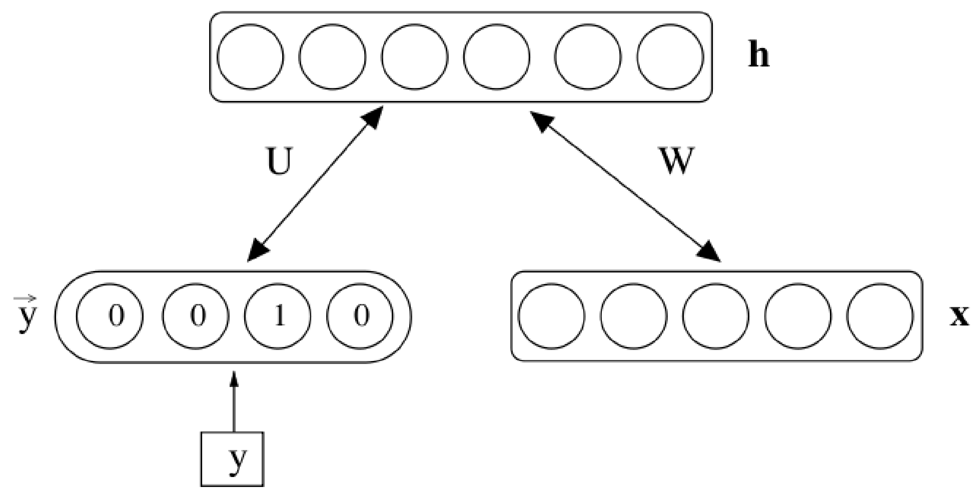
Tag smoothing

- Idea : extend the set of tags attached to a clip based on already provided tags



Restricted Boltzmann Machines

- RBMs can model the joint distribution between x and y by changing the energy function to :



$$E(y, x, h) = -h^T U y - h^T W x - c^T h - d^T x$$

$$L_{gen}(D_{train}) = -\sum_{i=1}^{|D_{train}|} \log p(x_i, y_i)$$

Discriminative RBMs

- Idea : minimize the discriminative log-likelihood instead of the generative log likelihood

$$L_{disc}(D_{train}) = -\sum_{i=1}^{|D_{train}|} \log p(y_i|x_i)$$

- When y can take only a few values (as in normal classification tasks), the gradients can be computed exactly (see Larochelle et al 2008)
- Hybrid models can be obtained by summing the two costs as:

$$L_{hybrid} = L_{disc} + \alpha L_{gen}$$

Multi-label Discriminative RBMs

- Tags are not mutually exclusive, making exact computation of the gradient intractable

$$\frac{\partial}{\partial \theta} p(y_t|x_t) = -\boldsymbol{E}_{\boldsymbol{h}|\boldsymbol{y}_t,\boldsymbol{x}_t}\left[\frac{\partial}{\partial \theta} E(x_t, y_t, h)\right] + \boldsymbol{E}_{\boldsymbol{y},\boldsymbol{h}|\boldsymbol{x}_t}\left[\frac{\partial}{\partial \theta} E(x_t, y, h)\right]$$

- We approximate the second expectation using Contrastive Divergence, mean field Contrastive Divergence, and loopy belief propagation. We also compare a similar computation that maximizes the pseudo-likelihood.

Approximations

- Contrastive Divergence proposes to replace the expectation $E_{y,h|x_t}$ by a point estimate at a sample obtained by running a Gibbs sampling initialized at y for K iterations.

- Mean-Field Contrastive Divergence is just a non-stochastic alternative where samples are replaced by expectations.

Approximation example : Contrastive Divergence

Algorithm 1 Discriminative RBM training update using Contrastive Divergence.

Input: training pair $(\mathbf{y}, \mathbf{x})$, number of iterations K and learning rate λ
Positive phase
$\mathbf{y}^0 \leftarrow \mathbf{y}, \widehat{\mathbf{h}}^0 \leftarrow \mathrm{sigm}(c + W\mathbf{x} + U\mathbf{y}^0)$

# Negative phase (we are doing CD-K here)
for K iterations **do**
$\quad \mathbf{h}^k \sim p(\mathbf{h}|\mathbf{y}^k, \mathbf{x})$
$\quad \mathbf{y}^{k+1} \sim p(\mathbf{y}|\mathbf{h}^k)$
$\quad \widehat{\mathbf{h}}^{k+1} \leftarrow \mathrm{sigm}(c + W\mathbf{x} + U\mathbf{y}^{k+1})$
end for

# Update
for $\theta \in \Theta$ **do**
$\quad \theta \leftarrow \theta - \lambda \left(\frac{\partial}{\partial \theta} E(\mathbf{y}^0, \mathbf{x}, \widehat{\mathbf{h}}^0) - \frac{\partial}{\partial \theta} E(\mathbf{y}^K, \mathbf{x}, \widehat{\mathbf{h}}^K) \right)$
end for

Approximations

- Loopy belief propagation is a popular algorithm for approximating the associated marginals required by the expectation :

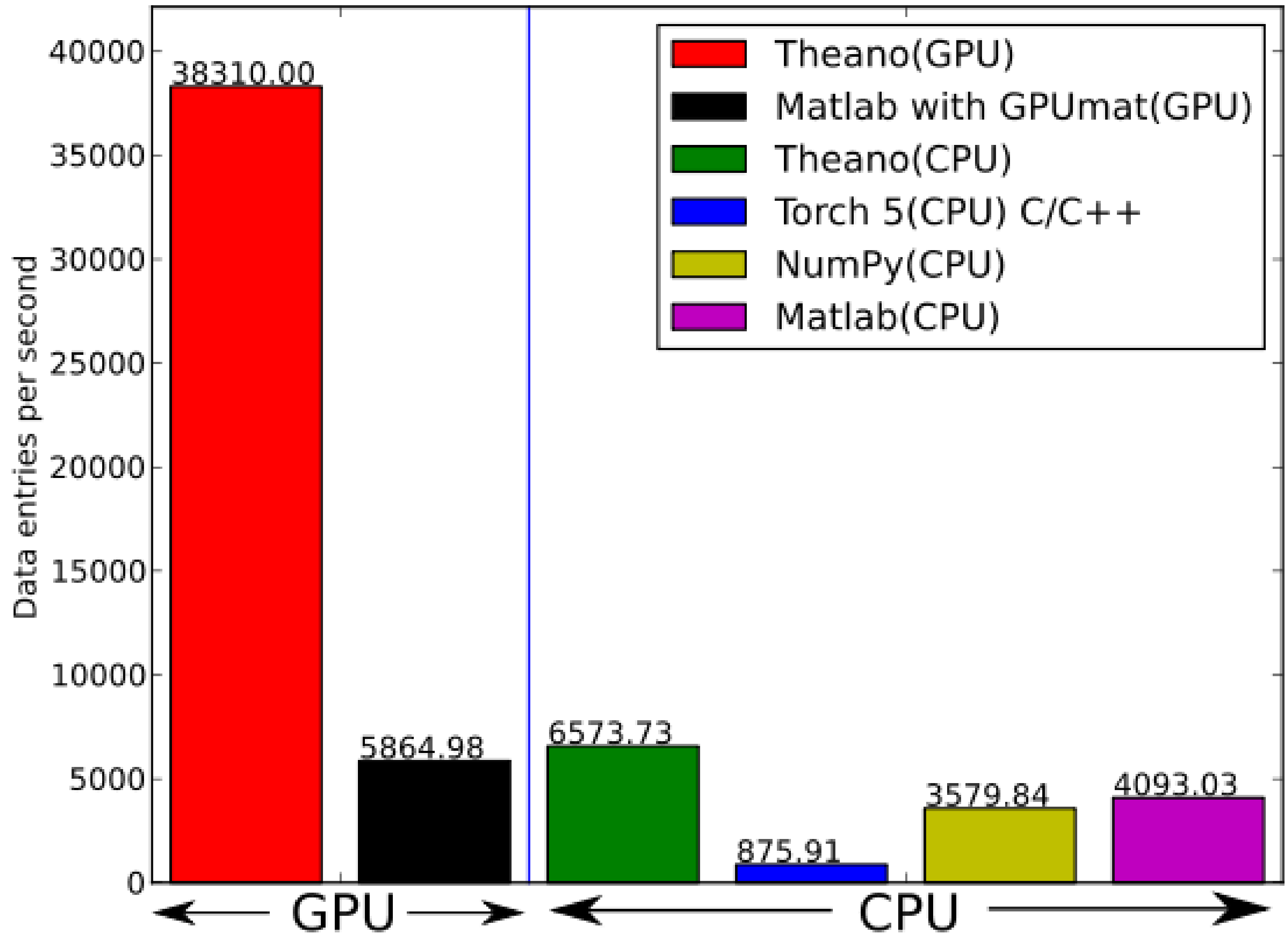
$$p(y_j = 1|x),\, p(h_k = 1|x),\, p(y_j = 1, h_k = 1|x)$$

- The final approximation replaces the log-likelihood by a pseudo-likelihood objective that allows computing the gradient exactly:

$$\log PL(\mathbf{y} \mid \mathbf{x}) = \sum_j \log p(y_j \mid \mathbf{y}_{\setminus j}, \mathbf{x}) = \sum_j \log p(\mathbf{y} \mid \mathbf{x}) - \log\left(p(\mathbf{y} \mid \mathbf{x}) + p(\tilde{\mathbf{y}}_j \mid \mathbf{x})\right)$$

Tools

- Theano – home grown python library for numerical computations with a focus on machine learning
- http://deeplearning.net/software/theano
- Deep Learning Tutorials – exemplification of how Theano can be used to implement deep learning architectures
- http://deeplearning.net/tutorial


Theano(GPU)
Matlab with GPUmat(GPU)
Theano(CPU)
Torch 5(CPU) C/C++
NumPy(CPU)
Matlab(CPU)
Data entries per second
40000
35000
30000
25000
20000
15000
10000
5000
0
38310.00
5864.98
6573.73
875.91
3579.84
4093.03
GPU
CPU

Data-sets

- 10 second clips were tagged among other things in terms of genre, emotion, instruments and overall production
- First dataset was obtained using Amazon.com's Mechanical Turk service and resulted in collecting 15500 (user,clip,tag) triplets from 210 unique users for 925 clips taken from 185 songs
- Second dataset was collected from the MajorMinor music labelling game. The set contains 80000 (user,clip,tag) triplets with 2600 unique clips, 650 unique users and 1000 unique tags

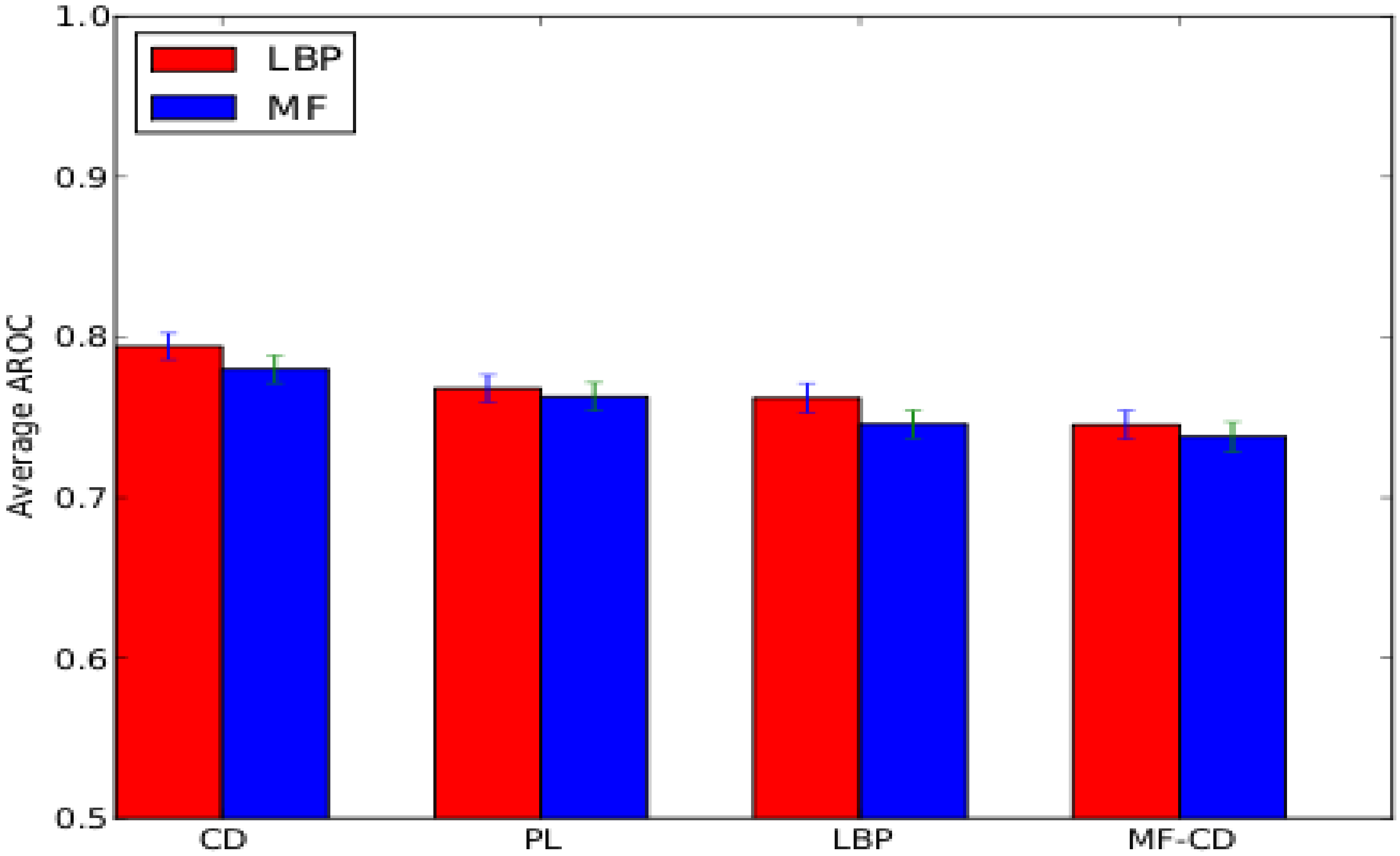
Data-sets

- The third dataset was collected from Last.fm's website and contains about 7 million (user,track,tag) triplets from 84000 unique users, 1 million unique tracks. We used a subset of only 1.5 million (user,track,tag) triplets
- We used timbral and rhythmic features to describe the audio
- The timbral features are the mean and rasterized full covariance of the clip's mel frequency cepstral coefficients

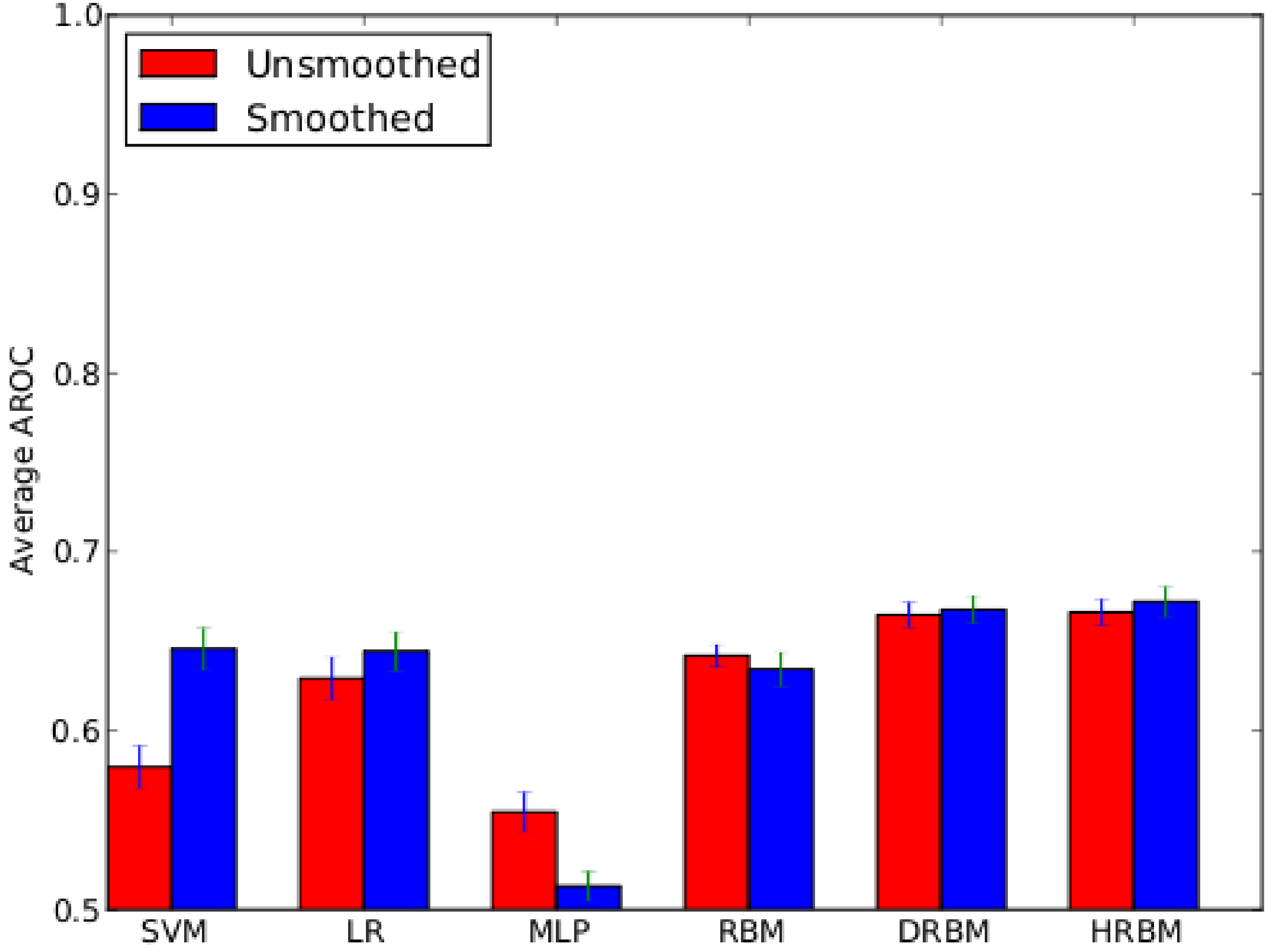
Data-sets

- The rhythmic feature are based on modulation spectra in four large frequency bands (closely related to the auto correlation in those bands)
- The metrics used to measure the performance is the Area under the ROC (Receiver operating characteristic) curve.
- A random ranking will achieve an AROC of 0.5, while a perfect ranking will give a score of 1.0

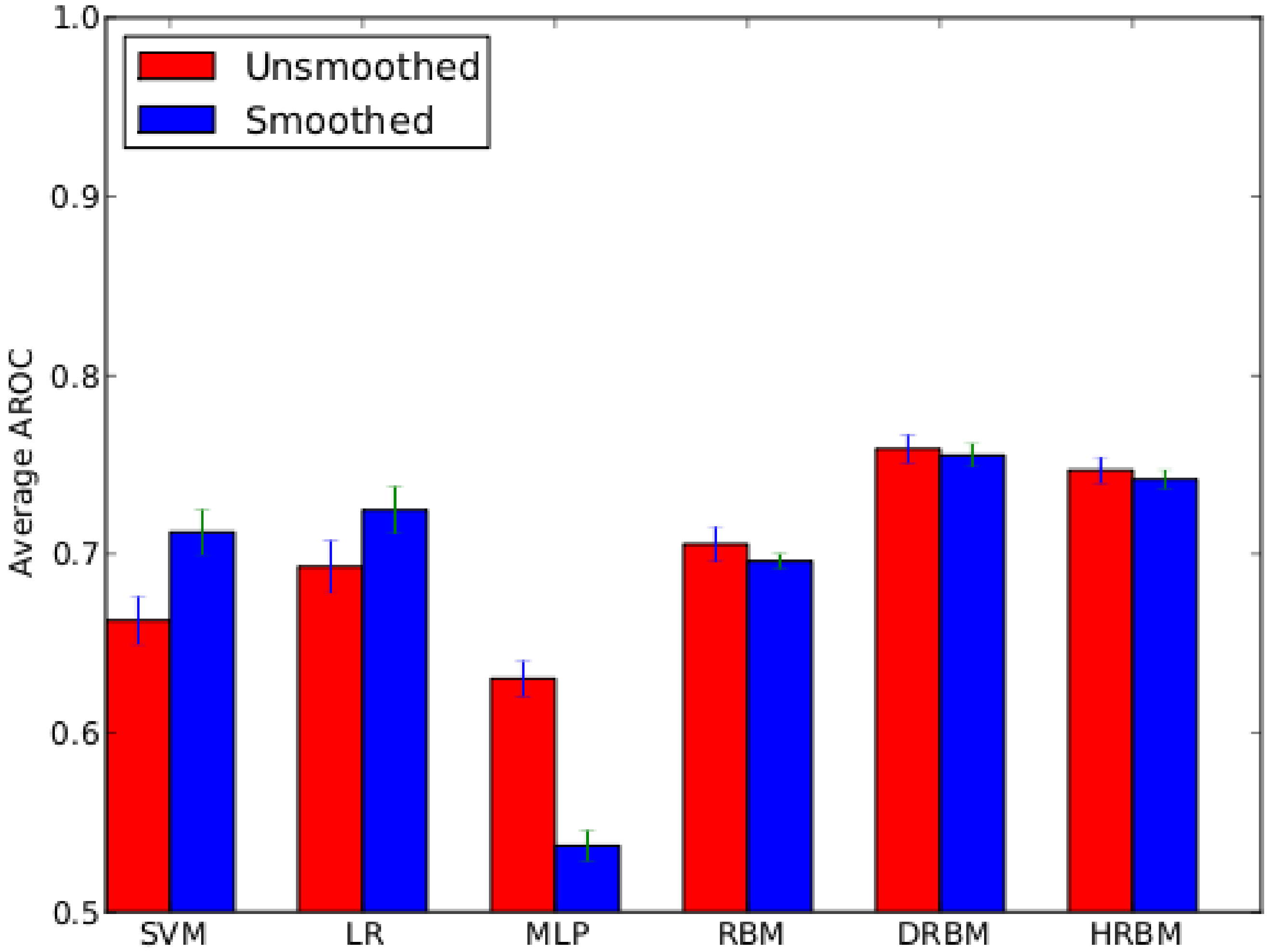
Results



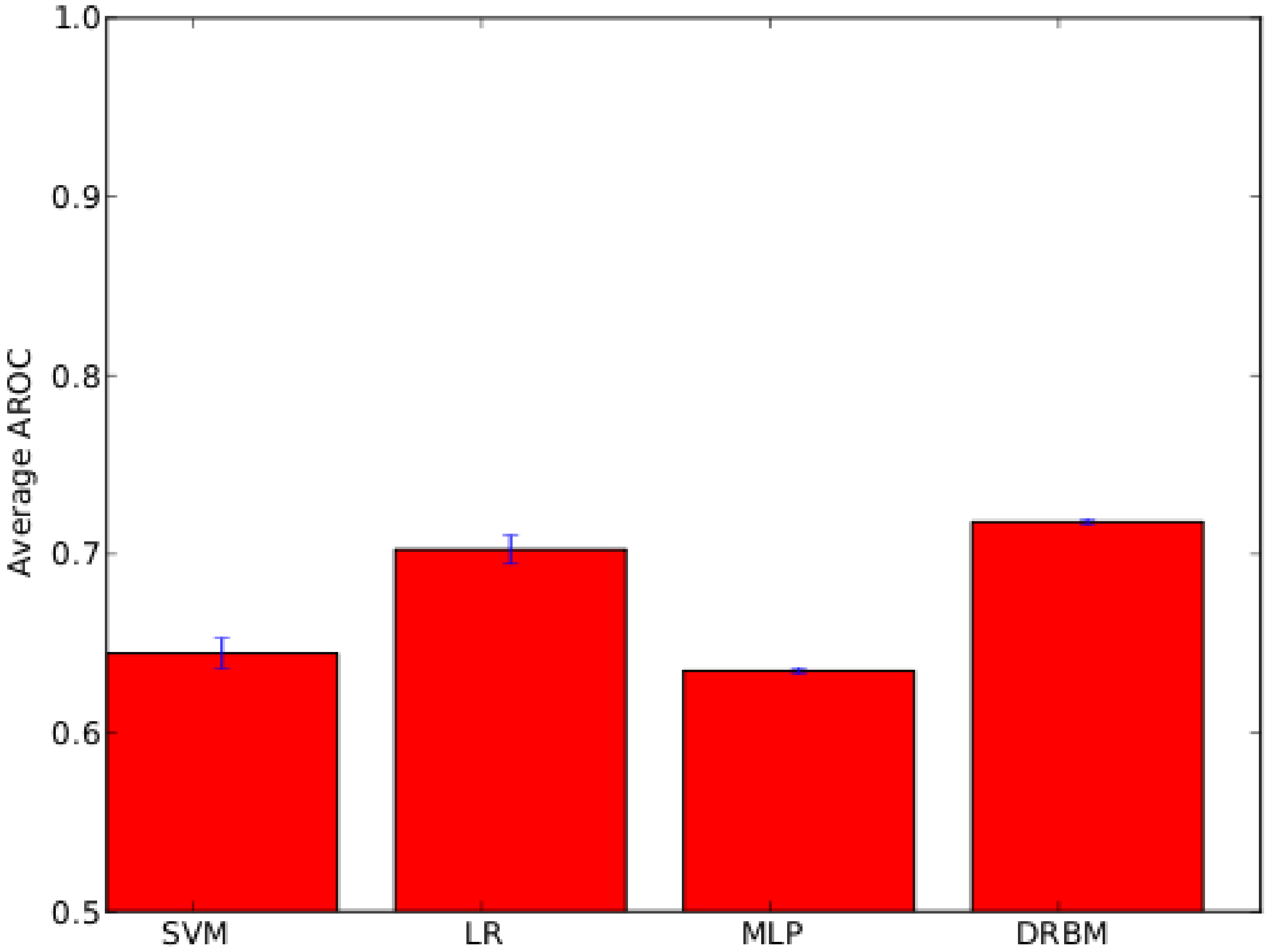
Results – Mechanical Turk data



Results – MajorMinor data



Results – Last.fm data



Summary

- We used RBMs to enhance a data-set by smoothing the already existing tags
- We further extended the concept of discriminative RBMs to multi-label problems, by approximating the gradient
- We tried four different approximations, contrastive divergence, mean-field contrastive divergence, loopy belief propagation and pseudo-likelihood

Thank You !

References

- Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527– 1554.
- G. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. In NIPS 19, pages 1345–1352. MIT Press, Cambridge, MA, 2007.
- H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In Andrew McCallum and Sam Roweis, editors, Proc. ICML, pages 536–543. Omnipress, 2008