COMPUTATIONALLY EFFICIENT MULTI-LABEL CLASSIFICATION BY LEAST-SQUARES PROBABILISTIC CLASSIFIER

Hyun Ha Nam, Hirotaka Hachiya, and Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology, Tokyo 152-8552, Japan
hyunha@sg.cs.titech.ac.jp, *hachiya@sg.cs.titech.ac.jp*, and *sugi@cs.titech.ac.jp*

ABSTRACT

Multi-label classification allows a sample to belong to multiple classes simultaneously, which is often the case in real-world applications such as audio tagging, image annotation, video search, and text mining. In such a multi-label scenario, taking into account correlation between multiple labels can boost the classification accuracy. However, this in turn makes classifier training more challenging because handling multiple labels tends to induce a high-dimensional optimization problem. In this paper, we propose a highly scalable multi-label classifier based on a computationally efficient classification algorithm called the least-squares probabilistic classifier. Through experiments, we show the usefulness of our proposed method.

Index Terms— Multi-Label Classification, Least-Squares Probabilistic Classifier, Freesound, $\Psi\ \Psi^\top$

1. INTRODUCTION

Multi-label classification [1] is an important class of classification problems, where a single sample can belong to multiple classes at the same time. For example, in web-based audio repositories such as the *Freesound* [2], each audio signal is associated with various tags such as 'acoustic', 'drum', and 'vocal', and tag prediction for new sounds can be formulated as a multi-label classification problem. Image annotation tasks such as the *PASCAL Visual Object Classes* [3] also involve multi-label classification because each image contains various objects such as 'airplane', 'sky', and 'person'.

A naive way to solve a multi-label problem is to take the *one-vs-rest* approach, i.e., multiple binary classification problems are solved separately. However, this approach cannot take the correlation between labels into account, and thus no advantage of multiple labels can be enjoyed.

In order to systematically utilize the correlation between labels, *multi-task learning* is useful [4]. The basic idea of multi-task learning is, rather than solving multiple learning tasks separately, solving them simultaneously by sharing some common information behind the tasks may improve the classification accuracy. If prediction of a single label in multi-label classification is regarded as a task in the multi-task formulation, multi-label classification can be systematically solved by a multi-task learning method.

A popular approach to multi-task learning is to impose solutions of related tasks to be similar to each other, by which related tasks implicitly share training samples effectively [5, 6]. However, handling multiple tasks tends to induce a high-dimensional optimization problem because multiple classifiers need to be trained at the same time.

To cope with this computational problem, a method called the *multi-task least-squares probabilistic classifier* (MT-LSPC) was proposed [7]. MT-LSPC is a multi-task extention of LSPC [8] that gives a non-parametric estimator of the class-posterior probability *analytically*. A notable advantage of LSPC is that its analytic solution can be computed efficiently just by solving a system of linear equations[1]. MT-LSPC effectively combines multiple LSPCs and still gives an analytic estimator of the class-posterior probabilities that can be computed by solving a linear system. However, naively solving the linear system is computationally expensive in the multi-task scenario because a large number of parameters for multiple classifiers are involved. The key idea of MT-LSPC is that the essential number of parameters to be optimized is reduced to the number of training samples, by which solutions for multiple classifiers can be computed efficiently.

However, MT-LSPC does not scale well in multi-label scenarios because the essential number of parameters to be optimized is equivalent to the total number of parameters for multiple classifiers. Furthermore, MT-LSPC assumes that all tasks are equally similar to each other, which is not necessarily true in multi-label classification. In this paper, we give a novel extension of LSPC called *multi-label LSPC* (ML-LSPC) that can overcome the limitations of MT-LSPC. Our key idea is that the system of linear equations we need to solve has useful block structure and we utilize this to efficiently solve the linear system. Through experiments, we illustrate the usefulness of the proposed approach.

HHN is supported by the GCOE program, HH is supported by the FIRST program, and MS is supported by the FIRST program.

[1] *Kernel logistic regression* can also be used for estimating the class-posterior probability in a non-parametric manner. However, it involves a non-linear optimization problem that is usually solved by a computationally expensive method such as (quasi-)Newton methods [9, 10].

2. PROBABILISTIC CLASSIFICATION BY LSPC

In this section, we review a probabilistic classification method called LSPC [8].

Suppose that we are given a set of training samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ drawn independently from a joint probability distribution with density $p(\boldsymbol{x}, y)$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ is a feature vector, D is the dimension of feature vector $\boldsymbol{x}$, $y_n \in \{1, \ldots, Y\}$ is a class label, and Y is the number of classes. The objective of LSPC is to learn the class-posterior probability $p(y|\boldsymbol{x})$ from the training samples. Based on the class-posterior probability, classification of a new sample $\boldsymbol{x}$ can be carried out by $\mathrm{argmax}_{y \in \{1,\ldots,Y\}} p(y|\boldsymbol{x})$, with confidence $p(y|\boldsymbol{x})$.

For each $y \in \{1, \ldots, Y\}$, $p(y|\boldsymbol{x})$ is modeled by

$$q(y|\boldsymbol{x}; \boldsymbol{\theta}_y) := \sum_{n=1}^N \theta_{y,n} K(\boldsymbol{x}, \boldsymbol{x}_n),$$

where $\boldsymbol{\theta}_y \in \mathbb{R}^N$ is the parameter vector and $K(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel basis function. The model is fitted to the true class-posterior probability under the squared loss:

$$J_y(\boldsymbol{\theta}_y) := \frac{1}{2} \int \left(q(y|\boldsymbol{x}; \boldsymbol{\theta}_y) - p(y|\boldsymbol{x})\right)^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

where $p(\boldsymbol{x})$ denotes the marginal density of feature vector $\boldsymbol{x}$. Expanding the squared term, we can express J_y as

$$\begin{aligned} J_y(\boldsymbol{\theta}_y) = \frac{1}{2} \int q(y|\boldsymbol{x}; \boldsymbol{\theta}_y)^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ - \int q(y|\boldsymbol{x}; \boldsymbol{\theta}_y) p(\boldsymbol{x}|y) p(y) \mathrm{d}\boldsymbol{x} + C, \end{aligned}$$

where $p(y|\boldsymbol{x}) = p(\boldsymbol{x}|y)p(y)/p(\boldsymbol{x})$ is used and C is a constant independent of $\boldsymbol{\theta}_y$.

Approximating the expectations over $\boldsymbol{x}$ and the class-prior probability $p(y)$ by samples, ignoring C, and including an ℓ_2-regularizer, we have the following training criterion:

$$\begin{aligned} \widehat{J}_y(\boldsymbol{\theta}_y) &:= \frac{1}{2N} \sum_{n=1}^N q(y|\boldsymbol{x}_n; \boldsymbol{\theta}_y)^2 \\ &\quad - \frac{1}{N} \sum_{n: y_n = y} q(y|\boldsymbol{x}_n; \boldsymbol{\theta}_y) + \frac{\lambda}{2} \|\boldsymbol{\theta}_y\|^2 \\ &= \frac{1}{2N} \boldsymbol{\theta}_y^\top \boldsymbol{K}^2 \boldsymbol{\theta}_y - \frac{1}{N} \boldsymbol{\theta}_y^\top \boldsymbol{K} \boldsymbol{\xi}_y + \frac{\lambda}{2} \|\boldsymbol{\theta}_y\|^2, \end{aligned}$$

where $\lambda > 0$ is the regularization parameter, and $\boldsymbol{K}$ is the $N \times N$ matrix and $\boldsymbol{\xi}_y$ is the N-dimensional vector defined by

$$K_{n,n'} := K(\boldsymbol{x}_n, \boldsymbol{x}_{n'}), \quad \xi_{y,n} := \begin{cases} 1 & (y_n = y), \\ 0 & (y_n \neq y). \end{cases}$$

Taking the derivative of $\widehat{J}_y$ with respect to $\boldsymbol{\theta}_y$ and setting it to zero, we can obtain the minimizer $\widehat{\boldsymbol{\theta}}_y$ analytically as

$$\widehat{\boldsymbol{\theta}}_y = \left(\boldsymbol{K}^2 + \lambda N \boldsymbol{I}_N\right)^{-1} \boldsymbol{K} \boldsymbol{\xi}_y,$$

where $\boldsymbol{I}_N$ denotes the N-dimensional identity matrix. By rounding up a negative output to zero and normalization, the final solution is given as

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\max(0, q(y|\boldsymbol{x}; \widehat{\boldsymbol{\theta}}_y))}{\sum_{y'=1}^Y \max(0, q(y'|\boldsymbol{x}; \widehat{\boldsymbol{\theta}}_{y'}))}.$$

This method is called the *least-squares probabilistic classifier* (LSPC) [8].

Thanks to the above analytic-form solution, LSPC was demonstrated to be computationally much more efficient than kernel logistic regression [9, 10] (trained by the L-BFGS quasi-Newton method), whereas the classification accuracy is kept comparable [8].

3. MULTI-TASK CLASSIFICATION BY LSPC

By combining multiple LSPCs, a computationally efficient multi-task learning method can be developed. In this section, we review multi-task LSPC (MT-LSPC) [7].

Suppose that we are given a set of training samples $\{(\boldsymbol{x}_n, y_n, t_n)\}_{n=1}^N$, where $t_n \in \{1, \ldots, T\}$ denotes the task index. We assume that $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ are drawn independently from a joint probability distribution with density $p_{t_n}(\boldsymbol{x}, y)$. The objective of MT-LSPC is to learn the class-posterior probabilities $p_t(y|\boldsymbol{x})$ for $t \in \{1, \ldots, T\}$.

The idea of MT-LSPC follows the line of [5], i.e., solutions of all tasks are imposed to be close to each other in terms of the ℓ_2-norm. More specifically, for each $t \in \{1, \ldots, T\}$ and $y \in \{1, \ldots, Y\}$, $p_t(y|\boldsymbol{x})$ is modeled as

$$q(y|\boldsymbol{x}; \boldsymbol{\theta}_{t,y}) := \sum_{n=1}^N \theta_{t,y,n} K(\boldsymbol{x}, \boldsymbol{x}_n).$$

Then the simultaneous training criterion for all models $\{q(y|\boldsymbol{x}; \boldsymbol{\theta}_{t,y})\}_{t=1}^T$ is given by

$$\begin{aligned} \widehat{J}_y^{\mathrm{MT}}(\boldsymbol{\theta}_y) &:= \frac{1}{2N} \sum_{n=1}^N q(y|\boldsymbol{x}_n; \boldsymbol{\theta}_{t_n,y})^2 \\ &\quad - \frac{1}{N} \sum_{n: y_n = y} q(y|\boldsymbol{x}_n; \boldsymbol{\theta}_{t_n,y}) + \frac{\lambda}{2T} \sum_{t=1}^T \|\boldsymbol{\theta}_{t,y}\|^2 \\ &\quad + \frac{\gamma}{4T^2} \sum_{t,t'=1}^T \|\boldsymbol{\theta}_{t,y} - \boldsymbol{\theta}_{t',y}\|^2, \end{aligned}$$

where $\boldsymbol{\theta}_y = (\boldsymbol{\theta}_{1,y}^\top, \ldots, \boldsymbol{\theta}_{T,y}^\top)^\top \in \mathbb{R}^{NT}$, and $\gamma > 0$ is the multi-task parameter. Taking the derivative of $\widehat{J}_y^{\mathrm{MT}}$ with respect to $\boldsymbol{\theta}_y$ and setting it to zero, we have equation $\boldsymbol{A}\boldsymbol{\theta}_y = \boldsymbol{b}_y$ (the definition of $\boldsymbol{A}$ and $\boldsymbol{b}_y$ are omitted; see [7] for details).

In principle, equation $\boldsymbol{A}\boldsymbol{\theta}_y = \boldsymbol{b}_y$ can be solved analytically as $\boldsymbol{\theta}_y = \boldsymbol{A}^{-1}\boldsymbol{b}_y$. However, this requires $\mathcal{O}(N^3T^3)$ time, which is prohibitive for large N and T. On the other hand,

through the dual formulation, the essential number of parameters was shown to be N, because only N training samples are available [7]. Taking this into account, we can reduce the size of the matrix to be inverted from $NT \times NT$ to $N \times N$. By that, the MT-LSPC solution can be computed in $\mathcal{O}(N^3)$ time, which is independent of T (see [7] for details).

4. PROPOSED METHOD: MULTI-LABEL LSPC

In this section, we describe our proposed method called *multi-label LSPC* (ML-LSPC).

Let us consider the following multi-label problem. Suppose that we are given a set of training samples $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$, where $\boldsymbol{y}_n \in \{1, \ldots, Y\}^T$ is a class-label vector and T is the dimension of class-label vector $\boldsymbol{y}$. $\boldsymbol{x}$ is assumed to be drawn independently from $p(\boldsymbol{x})$, and the t-th element y_t of $\boldsymbol{y} = (y_1, \ldots, y_T)^\top$ is assumed to be drawn from $p_t(y|\boldsymbol{x})$. The objective of ML-LSPC is to learn the class-posterior probabilities $p_t(y|\boldsymbol{x})$ for $t \in \{1, \ldots, T\}$.

A notable difference between the multi-task and multi-label formulations is that the number of training samples is N in the multi-task formulation (see Sec. 3), whereas the essential number of training samples in the multi-label formulation is NT. Thus, if we naively apply MT-LSPC to the multi-label problem, the computational complexity is still $\mathcal{O}(N^3T^3)$. Furthermore, the assumption behind MT-LSPC that all tasks are equally similar to each other is not necessarily appropriate in multi-label classification. Below, we describe a new method that can overcome the limitations of MT-LSPC.

Our basic idea for multi-label learning follows the same line as [6], i.e., similar labels should have similar solutions. Let $W_{t,t'} \geq 0$ be a similarity between label y_t and $y_{t'}$ (we assume $W_{t,t'} = W_{t',t}$); large $W_{t,t'}$ means that y_t and $y_{t'}$ are similar. Then our training criterion is defined as follows.

$$\widehat{J}_y^{\mathrm{ML}}(\boldsymbol{\theta}_y) := \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{2N}\sum_{n=1}^{N} q(y|\boldsymbol{x}_n;\boldsymbol{\theta}_{t,y})^2 - \frac{1}{N}\sum_{n:y_{t,n}=y} q(y|\boldsymbol{x}_n;\boldsymbol{\theta}_{t,y}) + \frac{\lambda}{2}\|\boldsymbol{\theta}_{t,y}\|^2\right) + \frac{\gamma}{4T^2}\sum_{t,t'=1}^{T} W_{t,t'}\|\boldsymbol{\theta}_{t,y} - \boldsymbol{\theta}_{t',y}\|^2.$$

After a few lines of calculation, we can show that $\widehat{J}_y^{\mathrm{ML}}$ is compactly expressed as

$$\widehat{J}_y^{\mathrm{ML}}(\boldsymbol{\theta}_y) = \frac{1}{2}\boldsymbol{\theta}_y^\top \boldsymbol{H}\boldsymbol{\theta}_y - \boldsymbol{\theta}_y^\top \boldsymbol{h}_y,$$

where $\boldsymbol{H} = \frac{1}{NT}\boldsymbol{I}_N \otimes \boldsymbol{K}^2 + \frac{\lambda}{T}\boldsymbol{I}_{NT} + \frac{\gamma}{T}\boldsymbol{L} \otimes \boldsymbol{I}_N$, $\otimes$ denotes the Kronecker product, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, $\boldsymbol{D}$ is the T-dimensional diagonal matrix with diagonal elements $\sum_{t=1}^T W_{1,t}, \ldots, \sum_{t=1}^T W_{T,t}$, and $\boldsymbol{h}_y = \frac{1}{NT}((\boldsymbol{K}\boldsymbol{\xi}_{1,y})^\top, \ldots, (\boldsymbol{K}\boldsymbol{\xi}_{T,y})^\top)^\top$. Taking the derivative of $\widehat{J}_y^{\mathrm{ML}}$ with respect to $\boldsymbol{\theta}_y$ and setting it to zero, we obtain

$$\boldsymbol{H}\boldsymbol{\theta}_y = \boldsymbol{h}_y. \tag{1}$$

Directly solving Eq.(1) takes $\mathcal{O}(N^3T^3)$ time, which is prohibitive when N and T are large. Here, we take into account the block structure of $\boldsymbol{H}$, and propose to solve the equation numerically by the conjugate gradient method. Specifically, we can compute the matrix-vector product $\boldsymbol{H}\boldsymbol{\theta}_y$ as

$$\boldsymbol{H}\boldsymbol{\theta}_y = \begin{pmatrix} \frac{1}{NT}\boldsymbol{K}^2\boldsymbol{\theta}_{1,y} + \frac{\lambda}{T}\boldsymbol{\theta}_{1,y} + \frac{\gamma}{T}\sum_{t=1}^T L_{1,t}\boldsymbol{\theta}_{t,y} \\ \vdots \\ \frac{1}{NT}\boldsymbol{K}^2\boldsymbol{\theta}_{T,y} + \frac{\lambda}{T}\boldsymbol{\theta}_{T,y} + \frac{\gamma}{T}\sum_{t=1}^T L_{T,t}\boldsymbol{\theta}_{t,y} \end{pmatrix}.$$

This takes only $\mathcal{O}(NT(N+T))$ time, whereas naive computation of $\boldsymbol{H}\boldsymbol{\theta}_y$ takes $\mathcal{O}(N^3 + N^2T^2)$ time including the computation of $\boldsymbol{K}^2$. As will be shown in the next section, this significantly contributes to reducing the computation time.

5. EXPERIMENTS

In this section, we first experimentally illustrate the behavior of the proposed ML-LSPC using an artificial data set, and then apply ML-LSPC to a real-world audio tagging task.

Illustrative Example: Let the feature dimension be $D = 20$, and we consider T binary classification tasks. Training samples of the t-th task is created as follows: $\boldsymbol{x}_n = (x_{1,n}, \ldots, x_{D,n})^\top$ is independently drawn from the standard normal distribution and $y_{t,n}$ is determined by linear decision boundary $\cos(2\pi t/T)x_{1,n} + \sin(2\pi t/T)x_{2,n}$ (i.e., the decision boundaries are rotated in the subspace spanned by the first two dimensions). We set the number of training samples to $N = 300$ and the number of test samples to $N = 1,000$. The label similarity $W_{t,t'}$ is set to $\max(0, \rho_{t,t'})$, where $\rho_{t,t'}$ is the Pearson correlation coefficient between $\{y_{t,n}\}_{n=1}^N$ and $\{y_{t',n}\}_{n=1}^N$. We use the Gaussian kernel as $K(\boldsymbol{x}, \boldsymbol{x}')$, and determine all tuning parameters (i.e., the Gaussian width, the regularization parameter λ, and the multi-task parameter γ) by 5-fold cross-validation in terms of the misclassification rate. We run the experiments 50 times with different random seeds, and evaluate the average misclassification rate and computation time as functions of the number of tasks.

The left graph in Fig. 1 plots the misclassification rate of plain LSPC (i.e., each task is solved separately) and ML-LSPC, showing that the classification performance tends to be enhanced as the number of tasks increases. The right graph in Fig. 1 plots the computation time of ML-LSPC when Eq.(1) is naively solved (we used the left-division function '*mldivide*' in MATLAB®) or when the proposed optimization method is used (we used the conjugate gradient function '*pcg*' in MATLAB®). This shows that the proposed optimization method is computationally much more efficient than the naive implementation.

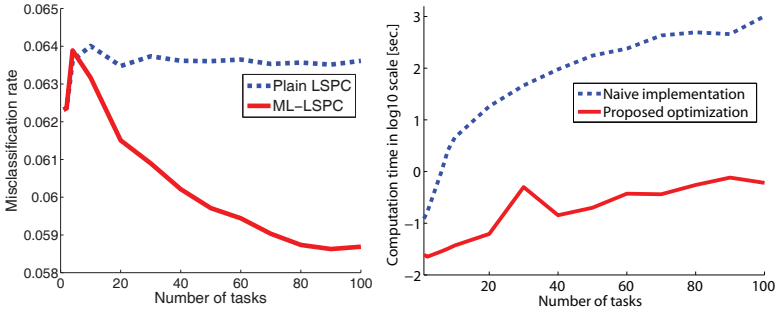

Fig. 1. Illustrative example. Misclassification rate (left) and computation time (right) as functions of the number of tasks.

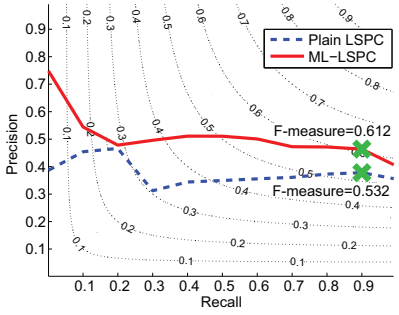

Fig. 2. Precision-recall curves for Freesound tagging task.

Audio Tagging: The *Freesound Project* is a collaborative database of Creative Commons licensed sounds [2]. Each sound file was annotated using a vocabulary about the genre, instrumentation, emotion, style, and rhythm. We downloaded $1,276$ audio files from the Freesound web page, and extracted the first 13 *mel-frequency cepstral coefficients* (MFCCs) [11] and their first and second instantaneous derivatives computed in a half-overlapping sliding window with width 23[msec]. Thus, each audio file was expressed as a set of 39-dimensional feature vectors. We then created a vector quantization codebook of size $D = 2,048$ using about $600,000$ feature vectors obtained from all the $1,276$ audio files, and extracted the normalized code histogram as $\boldsymbol{x}$.

For experiments, we picked 361 samples from the $1,276$ samples that contain the pre-specified 15 tags such as 'noise', 'percussion', 'drum', and 'rhythmic'. Then, we randomly chose $N = 20$ samples for training, and used the remaining 341 samples for performance evaluation. Because the presence and absence of tags were highly imbalanced, we decided to evaluate the performance by the F-measure (we also used the F-measure for cross-validation).

Fig. 2 depicts the precision-recall curves and the F-measures for plain LSPC and ML-LSPC, showing that ML-LSPC overall compares favorably with plain LSPC.

6. CONCLUSIONS

Multi-label classification is useful in various real-world problems such as audio tagging, image annotation, video search, and text mining. However, because the essential number of training samples for T-dimensional label vectors of size N is NT, naive implementation of multi-label classification is computationally expensive when N and T are large. To overcome this computational bottleneck, we proposed to combine a computationally efficient classifier called LSPC [8] with a standard multi-task learning technique [6]. Our key idea was that the system of linear equations we need to solve has useful block structure, and we utilized that structure to improve the computational efficiency. Through experiments, we showed that the proposed method, ML-LSPC, is promising.

7. REFERENCES

[1] G. Tsoumakas and I. Katakis, "Multi Label Classification: An Overview," *International Journal of Data Warehouse and Mining*, vol. 3, pp. 1–13, 2007.

[2] The Freesound Project, "Freesound," 2010.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2010.

[4] R. Caruana, L. Pratt, and S. Thrun, "Multitask learning," in *Machine Learning*, 1997, pp. 41–75.

[5] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2004, pp. 109–117, ACM.

[6] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Conic Programming for Multitask Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 957–968, 2010.

[7] J. Simm, M. Sugiyama, and T. Kato, "Computationally efficient multi-task learning with least-squares probabilistic classifiers," *IPSJ Transactions on Computer Vision and Applications*, vol. 3, pp. 1–8, 2011.

[8] M. Sugiyama, "Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 10, pp. 2690–2701, 2010.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer-Verlag, 2001.

[10] T. P. Minka, "A comparison of numerical optimizers for logistic regression," *Microsoft Research, Tech. Rep.*, 2007.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic Annotation and Retrieval of Music and Sound Effects," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, Feb. 2008.