COMPARISON OF FEATURE-LEVEL AND KERNEL-LEVEL DATA FUSION METHODS IN MULTI-SENSORY FALL DETECTION

Che-Wei Huang, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory (SAIL),
University of Southern California, 3710 McClintock Avenue, Los Angeles, CA 90089, USA
cheweihu@usc.edu, shri@sipi.usc.edu

ABSTRACT

In this work, we studied the problem of fall detection using signals from tri-axial wearable sensors. In particular, we focused on the comparison of methods to combine signals from multiple tri-axial accelerometers which were attached to different body parts in order to recognize human activities. To improve the detection rate while maintaining a low false alarm rate, previous studies developed detection algorithms by cascading base algorithms and experimented on each sensory data separately. Rather than combining base algorithms, we explored the combination of multiple data sources. Based on the hypothesis that these sensor signals should provide complementary information to the characterization of human's physical activities, we benchmarked a feature level and a kernel-level fusions to learn the kernel that incorporates multiple sensors in the support vector classifier. The results show that given the same false alarm rate constraint, the detection rate improves when using signals from multiple sensors, compared to the baseline where no fusion was employed.

Index Terms— Fall Detection, Multi-Sensor Fusion, Healthcare

1. INTRODUCTION

The world is growing old [1] and one of the biggest concerns toward the elderly population, especially people who lead a independent living, is the fall. A fall could cause severe injury to aged people and puts heavy burden on the healthcare system. Recently, the tasks related to fall detection have drawn attention in the human activity recognition (HAR) community in order to improve human's life. In addition to fall detection, researcher also took a proactive approach to fall risk factor analysis and fall prevention mechanism [2].

A multitude of previous work [2, 3] have been dedicated to the task of fall detection, investigating various aspects depending on the devices and the data types. In general, the studies can be categorized into three classes: the vision based, the ambient sensing based and the wearable sensor based. The vision based and the ambient sensing based methods are typically spatially localized so it is important to have the information of whereabouts of the users in advance. The vision based approach based is also known for its high computational cost and the privacy issue. The wearable sensor based approaches are economical and preserve users' privacy, but are comparably more obtrusive since the users are required to put sensors on their body parts. Another common approach is to exploit information from a combination of methods aforementioned.

In general, fall detection systems based on a single sensor are lacking robustness and sufficient accuracy, and hence multi-sensory systems are gaining popularity. For instance, the work by Grassi et al. [4] demonstrated the effectiveness of fusing multiple sensors including a camera, an accelerometer and a microphone. Even within the category of wearable sensor based methods, different types of sensors could provide complementary information [5].

Recently, Koshmak et al. [3] have surveyed the work on the fusion of multiple sensors and identified some of open challenges. The major issue specifically pertain to the wearable sensors is the reliability of the wireless connection during data acquisition, which could cause delay and loss of signals. Another one is that a certain level of obtrusiveness is possible if users have to wear it for a period of time. Past work on fusing multiple wearable sensors mainly contributed to fusion of *heterogeneous* sensors; for example, using an accelerometer and a magnetometer [5]. However, homogeneous sensors attached to different body parts could also provide complementary information.

Our work belongs to the wearable sensor based approach, in which we considered the sensory signals captured by tri-axial accelerometers. Particularly, we studied the fusion of multiple accelerometers, attached to different body parts, at the feature-level and the kernel-level.

2. RELATED WORK

Gjoreski et al. [6] studied fall detection using four accelerometers attached to four body parts including the chest, waist, right ankle and right thigh, and experimented on different combinations of sensory signals for the tasks of posture recognition and fall detection. A random forest model was trained to perform posture recognition, followed by a rule based algorithm for fall detection. While their work is quite similar to ours in the use of multiple homogeneous sensors, there are differences we would like to stress. First of all, we focused on the comparison of fusion methods using information from wearable sensors, while they concentrated on finding the optimal placement of the sensors. Second, in our algorithm the fall detection is one-step and data driven. Li et al. [7] proposed a grammar-based fall detection framework using multiple acceleration and vibration monitors, focusing on finding the most critical placement for sensors. Many other investigations on the fusion of heterogeneous data sources are reviewed in the work of [3], but none of them has studied the comparison of different fusion techniques. As verified by Gjoreski et al., using more sensors would improve the performance. In this work, we compared two methods of fusing multiple homogeneous data sources.

The performance of each proposed systems varies under different experimental conditions. To have a fair comparison, we followed closely to the recent work by Cheng et al. [8] on the public data we used. Cheng et al. has developed a cascade-AdaBoost-SVM classifier for the task of fall detection and tested the algorithm on each accelerometer separately. In contrast to their work, we leveraged on the combination of information from multiple sources, rather than combination of algorithms, to improve the detection rate.

3. DATA SET AND FEATURE EXTRACTION

3.1. Data Set

We used the *Localization Data for Person Activity Data Set* on the UCI's machine learning repository [9] for the experimentation on the fall detection. In this data set, five volunteers were asked to perform a series of activities, which is called the scenario. The scenario, though continuous, can be broadly divided into three stages including two rapid falls (*walking to falling to lying*) and then one slow fall (*sitting to falling to sitting down*). Every volunteer performed the scenario five times so there were 25 recordings of the scenario in the data set. Every recording of the scenario lasted about 3-5 minutes.

During performing the scenario, volunteers wore four triaxial accelerometers on their chest, belt, right ankle and left ankle. As a result, every recording is in fact a set of four 3-dimensional multivariate time series. The accelerometers kept radio contact with a sensor mounted on the wall and calculated its location based on the time difference of arrivals and the arrival angle, which indirectly incurred time-varying delays. Three of the four accelerometers had the sampling frequency at 10 Hz so the (x, y, z) coordinates of the sensors were recorded roughly every 0.1 second with minor time delays due to wireless connection. However, there was one sensor with the sampling frequency at 5 Hz. Moreover, while most of the time the sensors and the wireless connection worked regularly well, there was certain number of short time intervals when the sensors malfunctioned or the wireless connection was unstable and the sensors failed for several seconds. It is therefore necessary to fix the mismatch between samples, due to different sample frequencies and loss of samples, before fusing information from four accelerometers.

3.2. Feature Extraction

In a wearable sensor based system, common features for the fall detection include the (x, y, z) coordinates of the tri-axial accelerometers, the signal magnitude vector (SMV), the signal magnitude (SMA) and so on. The (x, y, z) coordinates are recorded according to the sampling frequency and denoted by $x[n]$, $y[n]$ and $z[n]$ for the x-, y- and z- coordinates, respectively, at time n. When rapid falls occur, it is assumed to have drastic changes in the values of the raw (x, y, z) coordinates, and thus the raw coordinates should contain the information for detecting when a fall takes place. From top three panels of Fig. 1, it is clear that the assumption is valid while we can also see that there exist lots of drastic changes, seemingly due to noise, in the coordinate values when the actual activity is not a fall.

The SMV defined by

$$\mathrm{SMV[n]} = \sqrt{\mathrm{x[n]^2 + y[n]^2 + z[n]^2}}$$

measures the intensity of the tri-axial coordinates. Because the locations calculated by the accelerometers are noise-corrupted, the SMA tries to minimize the effect of noise by performing integration over a window of time frame, defined as

$$\mathrm{SMA[n]} = \frac{1}{\mathrm{N}} \left(\sum_{\mathrm{i=n-N+1}}^{\mathrm{n}} |\mathrm{x[i]}| + |\mathrm{y[i]}| + |\mathrm{z[i]}| \right)$$

where N is an integer denoting the size of the window to integrate over.

In addition to the five low-level features, $x[n]$, $y[n]$, $z[n]$, SMV$[n]$ and SMA$[n]$, we created the delay embedded vectors $\mathbf{f}[n]$ [10]:

$$\mathbf{f}[n] = \begin{bmatrix} \mathbf{x}_n \\ \mathbf{x}_{n-1} \\ \vdots \\ \mathbf{x}_{n-N+1} \end{bmatrix}$$

where $\mathbf{x}_n = [x[n], y[n], z[n], \mathrm{SMV[n]}, \mathrm{SMA[n]}]^{\mathrm{T}}$. The delay embedded vectors with a fixed size of memory N take into account the temporal information in human activities since a low-level feature vector alone would be too short and temporally localized to encode such dynamics.

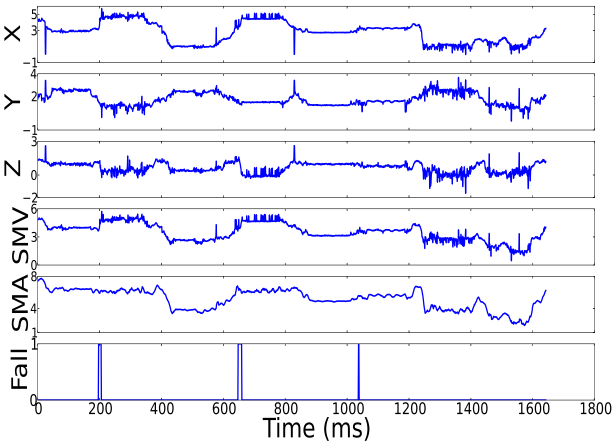

Fig. 1. Low-level features from the sensor on the chest of first volunteer. The top three panels are the (x, y, z) coordinates, and the fourth and the fifth are the SMV and SMA, respectively. The last panel indicates the time duration when a fall is in process.

4. METHODOLOGY

4.1. Baseline

Given that the delay embedded feature vectors $\mathbf{f}$ already incorporate temporal information, we formulated the fall detection problem into a binary classification problem using SVM as the baseline. For each recording there were three falls and the onset of the falls was labeled as positive, while the other data points were negative. Therefore, there were 75 positive samples from each accelerometer, while the number of negative samples varied from 35K to 42K, due to the mismatch among the accelerometers. Because of the huge difference between the number of samples from each class, we employed a cost-sensitive approach [11] to deal with the class imbalance issue.

4.2. Fusion and Detection Algorithms

4.2.1. Match the Frequencies

Before we could jointly analyze the sensory signals, we had to solve the synchronization issue due to different sampling frequencies and the loss of data points in the data acquisition phase. This pre-processing was the key step to enable the following joint analysis.

First of all, we removed the minor time-varying delays due to time difference arrivals by rounding the time stamps to the nearest tenth second. Second, by viewing the loss of data points as a change of the sampling frequency, we only needed to match the frequencies in the mixed-frequency time series data. Mixed-frequency data are common in the area of econometric studies [12]; for example, the gross domestic product is calculated quarterly whereas many leading economic indicators are monthly released. The most common empirical solutions to match the frequencies are the aggregation and the interpolation methods. The standard aggregation method averages over the high-frequency data to down-sample, while the interpolation method, though rarely used, is to up-sample the low-frequency data by interpolation. Both methods did not work for our problem. For the former, a rapid fall could be short in time duration so to down-sample by aggregating the sensory signals would incur the risk of losing the key information for fall detection. The latter was not applicable because the fact that time efficiency is important in the application of fall detection so we can't afford to delay the decision to wait for interpolating loss of data points which lasted several seconds. Instead, we employed a simple approach by replicating the previous value to up-sample the low frequency data, and match the number of samples in the raw (x, y, z) coordinate sequences from four tri-axial accelerometers. In an online situation to cope with the unreliable connection and the difference between sensor models, it may be the most computational efficient one.

4.2.2. Feature-Level Fusion

After matching the sampling frequencies of the sensory signals from the four accelerometers, we prepared the delay embedded feature vectors $\mathbf{f}^l[n]$, $\mathbf{f}^r[n]$, $\mathbf{f}^b[n]$ and $\mathbf{f}^c[n]$ using the low-level features from the accelerometer at left ankle, right ankle, belt and chest, respectively. A straight-forward method to combine the information captured by each accelerometer was to concatenate four delay embedded feature vectors $\mathbf{f}^l[n]$, $\mathbf{f}^r[n]$, $\mathbf{f}^b[n]$ and $\mathbf{f}^c[n]$ into a higher-dimensional one: $\mathbf{f}^{\text{cas}}[n]$. We then trained a SVC with the RBF kernel on the concatenated feature vectors $\mathbf{f}^{cas}[n]$ for the task of fall detection with the assumption that this feature-level fusion would leverage the hypothesized complementary information provided by each accelerometer and improve the detection rate while maintaining the same false alarm rate. Likewise, there was still the issue of class imbalance, so the same cost-sensitive class weights C_k for the soft margin regularization coefficient were specified during the training.

4.2.3. Kernel-Level Fusion

In kernel methods, including SVM, the choice of the kernel function plays an essential role in the generalization performances. A poor kernel may fail to identify the similarity between data points. Therefore, many studies have developed learning algorithms for the optimal kernel in kernel methods. Among them, Lanckriet et al. [13] proposed that such a kernel is in the form of a linear combination of some base kernels, such as the RBF or linear kernels; the kernel combination coefficients and the SVM parameters together can be learned through the training phase. This approach is referred to as the multiple kernel learning (MKL), and is still an active research topic in the machine learning community. Since then, several approaches have been proposed for efficient learning of MKL, including methods based on the sequential minimization optimization (SMO) [14], on the semi-infinite linear programming (SILP) [15, 16], on the sub-gradient [17] and on the level-method [18]. The objective function in all of the above methods consists of a simplex constraint on the kernel combination weights and thus leads to a sparse selection of base kernels, also called the L_1MKL.

$$
\begin{aligned}
\arg\min_{\alpha,b,\xi} \quad & \frac{1}{2}\sum_m \frac{1}{d_m}(\alpha \odot \mathbf{y})^T \mathbf{K}_m (\alpha \odot \mathbf{y}) \\
& + C\left[C_1 \sum_{\{i|y_i=1\}} \xi_i + C_0 \sum_{\{i|y_i=0\}} \xi_i\right] \\
\text{subject to} \quad & y_i\left(\sum_m \sum_{j=1}^{n} \alpha_j y_j \mathbf{k}_m(\mathbf{x}_j, \mathbf{x}_i) + b\right) \geq 1 - \xi_i \\
& \sum_m d_m = 1, d_m \geq 0, \xi_i \geq 0. \qquad (1)
\end{aligned}
$$

The formulation of L_1MKL[17] in Eq.(1) is similar to the single-kernel SVM, except that the kernel matrix and the kernel function are now indexed by subscripts to denote each base kernel, and the kernel matrix is replaced by a linear combination of multiple base kernel matrices.

Despite the improvement since [13], Cortes et al. [19] and Kloft et al. [20] have shown that a simple average of the base kernels can outperform the L_1MKL on some real-world applications. The sparse selection of the base kernels due to the simplex constraint on the kernel combination weights was conjectured to be omit some useful information during the modeling. They therefore introduced the L_2-based MKL and L_p-based MKL, respectively, for a non-sparse selection of base kernels. Recently, Xu et al. [21] has shown that the traditional L_1MKL can be viewed as a hard margin MKL, which selects the combination of a subset of base kernels that minimize the objective function and throw away any other information. They proposed a soft margin MKL in analogy to the soft-margin SVM, that makes SVM robust by introducing the slack variables.

Until now, we only used a single kernel in the SVC. Our observation was that because of the physical limitation, a human's belt and ankle would have different ranges of variations. For example, the sensor at a human's belt would not leave the center of mass of the body too far away but the sensor at the right ankle could have a large variation in the X-Y plane. The hyper-parameter γ in SVM controls the variance of the kernel, and tries to capture the variation of the training data. It was natural to hypothesize that to model the data from each source a γ for each of them would result in a better model. Therefore, to take into consideration the difference in distributions of variation between accelerometers, we further employed the MKL framework for the task of fall detection. Since the MKL formulation is a sum of weighted kernels, each trained on each accelerometer, this fusion is at the decision level. We employed the soft margin MKL [21] to combine the information provided by multiple accelerometers in this study, instead of the traditional L_1MKL because we hoped to utilize all of the information and avoid cases where in MKL model may lead to, say extremely, just one of the sensors.

5. EXPERIMENTS

In the baseline, we treated signals from each accelerometer separately as previous studies did [8]. A support vector classifier (SVC) with the RBF kernel was employed. We did five-fold cross validation on the delay embedded feature vectors using the LIBSVM software [22]. The hyper-parameters C and γ were tuned by grid search in the range of $C = 10^{-1}, 10^0, 10^1, 10^2, 10^3$ and $\gamma = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0$. A cost-sensitive approach was used to adjust the soft margin based on the class size to deal with the class imbalance issue. Also because of the large number of negative samples, a single measure of accuracy rate was not enough. Therefore, the performance was evaluated by three measures, the detection rate (DR), the false alarm rate (FAR) and the accuracy rate (AR) defined by

$$
\begin{aligned}
\text{DR} &= \frac{\text{TP}}{p} \times 100\% \\
\text{FAR} &= \frac{\text{FP}}{q} \times 100\% \\
\text{AR} &= \frac{\text{TP} + \text{TN}}{p+q} \times 100\%
\end{aligned}
$$

where p and q are the number of positive and negative samples, respectively, and TP, TN and FP are the number of true positive, true negative and false positive results, respectively. After the initial grid search, we finer tuned the hyper-parameters to reach the same false alarm rate at 5% for each classifiers. The performance of the baseline is summarized in the Table 1.

	Accelerometer Position			
	Left ankle	Right ankle	Belt	Chest
AR	94.84%	94.88%	94.92%	94.95%
DR	24.0%	30.67%	57.3%	73.3%
FAR	5.0%	5.0%	5.0%	5.0%

Table 1. The performance of the fall detection using the support vector classifier on the sensory signals from each sensor at different body parts.

In terms of the accuracy rate, the SVC performed almost the same with the sensory signals from each tri-axial accelerometer. However, given the false alarm rate set to 5%, the detection rate varied from one accelerometer to another. From the summary, we can see that the height of the accelerometer, i.e. the z-coordinate plays an important role in detecting the falls because the detection rate monotonically decreases as the height of the accelerometer decreases. At the height of the ankles, the z-coordinate offer next to zero contribution to the detection, but these two accelerometers at the ankles apparently were still able to recognize one fourth to one third of the falls.

It is to be noted, the baseline result using SVC is similar to the SVM baseline reported in Cheng et al. [8], but not exact. The reason is that they randomly selected only 100 negative samples from each recording, 2.5K in total, for the experiments and left the rest unused. We tried to repeat their experiments and found out that the result kept changing at each randomization, and some randomizations outperformed their result and some did not. Hence we took all of the negative samples, 35K to 42K depending on sensors, into our experiment. The trend of their result and ours are similar where chest sensor captured the most useful information, belt sensor next and the sensors at ankles the worst. The detection rates are similar with a minor degradation except on the belt accelerometer.

For the feature-level fusion, we trained an SVC with RBF kernel on the concatenated features $\mathbf{f}^{\text{cas}}[n]$. A five-fold cross validation was performed as well. The improvement based on the feature fusion is significant; both accuracy rate and detection rate increase, compared to the baseline. There is only a modest improvement on the accuracy but there is a 9.37% (absolute) improvement on the detection rate. For the kernel-level fusion, we prepared each base kernel based on the delay embedded features from each accelerometer. The kernel type was the RBF kernel. Then using the hinge loss soft margin MKL [21] we trained a MKL-SVC with five-fold cross validation. It turns out that the overall performance ends up in the middle of the four results in the baseline. The kernel combination weights range from 0.2 to 0.28, therefore the result is close to an average of the baseline. The summary is in the Table 2.

	Fusion type		
	Feature-level	Kernel-level	Best in [8]
AR	**95.03%**	94.61%	**98.23%**
DR	**82.67%**	34.67%	**88%**
FAR	5.0%	5.0%	**1.27%**

Table 2. The performances based on the feature-level and the kernel-level fusion. The best in [8] is the cascade-AdaBoost-SVM classifier using the chest accelerometer.

6. DISCUSSION

First of all, the result on the feature-level fusion seems promising. The hypothesis that each accelerometer contains complementary information to one another is validated from the gain of the performance. The frequency match-up approach is very simple, easy to implement and suitable for an online situation. During the training phase, the complementary information from each accelerometer is enhanced while the redundant and noisy information is suppressed. The model complexity is low, only as expensive as a SVM. However, the MKL result was disappointing. The kernel-level fusion can only take the result given by each kernel proportional to the corresponding kernel combination weight. The training requires multiple kernel matrices and and their combination weights, which requires more memory and computational power.

The comparison with Cheng et al. [8] is to provide an idea of how close the proposed method is to the state-of-the-art. However, there are a few things to address; first, the optimization directions are different. We were focused toward the combination of data sources while Cheng et al. focused on the combination of algorithms. Second, we used a larger number of negative samples compared to their experiments.

There are a few future directions we would like to pursue. First of all, we only used the RBF kernel in this study. We would like to know how does other kernels perform using the feature-level fusion. Second, despite the negative result on MKL framework, we are still interested in deploying one kernel for each dimension of the delay embedded features and implicitly performed the feature selection using L_1MKL. Further, we would like to know if we can reduce the number of accelerometers in the feature-level fusion while maintaining the performance, and if so, which combination works the best. The extension of the current work to a combination of multiple algorithms, such as in [8] is also interesting.

7. CONCLUSION

We explored the data fusion in multiple homogeneous wearable sensors for the task of fall detection. We addressed the synchronization issue by matching up the frequencies among sensors to enable the joint analysis and modeling. The result shows that at the feature-level fusion, the detection rate gains 9.37% and the accuracy rate gains less than 1% while maintaining the false alarm rate. We also showed that the kernel-level fusion of multiple accelerometer using MKL framework is not successful.

8. ACKNOWLEDGEMENT

This research is supported by NSF, NIH, DARPA and Google Inc.

9. REFERENCES

[1] Department of Economic Population Division and United Nations Social Affairs, "World population ageing," 200.

[2] Natthapon Pannurat, Surapa Thiemjarus, and Ekawit Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, 2014.

[3] Gregory Koshmak, Amy Loutfi, and Maria Linden, "Challenges and issues in multisensor fusion approach for fall detection: Review paper," *J. Sensors*, August 2015.

[4] M. Grassi, A. Lombardi, G. Rescio, and P. Malcovati et al., "A hardware-software framework for high-reliability people fall detection," in *IEEE Sensors*, 2008.

[5] Filipe Felisberto, Florentino Fdez.-Riverola, and Antnio Pereira, "A ubiquitous and low-cost solution for movement monitoring and accident detection based on sensor fusion," *Sensors*, vol. 14, no. 5, pp. 8961–8983, 2014.

[6] H. Gjoreski, M. Lustrek, and M. Gams, "Accelerometer placement for posture recognition and fall detection," in *Proceedings of the 7th International Conference on Intelligent Environments*, 2011.

[7] Q. Li and J. A. Stankovic, "Grammar-based posture and context cognitive detection for falls with different activity levels," in *Proceedings of the 2nd Conference on Wireless Health*, 2011.

[8] Wen-Chang Cheng and Ding-Mao Jhan, "Tri-axial accelerometer-based fall detection method using self-constructing cascade-adaboost-svm classifier," *IEEE Trans. Biomedical and Health Informatics*, March 2013.

[9] B. Kaluza, V. Mirchevska, E. Dovgan, M. Lustrek, and M. Gams, "An agent-based approach to care in independent living," pp. 177–186. 2010, Proc. 1st. Int. Joint Conf. Ambient Intell.

[10] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.*, vol. 45, pp. 712–716, Sep 1980.

[11] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," 1999.

[12] C. Foroni and M. Marcellino, "A survey of econometric methods for mixedfrequency data," Economics Working Papers ECO2013/02, European University Institute, 2013.

[13] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[14] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," 2004.

[15] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Dec. 2006.

[16] Alexander Zien and Cheng Soon Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[17] Alain Rakotomamonjy, Francis R. Bach, Stphane Canu, and Yves Grandvalet, "Simplemkl," 2008.

[18] Zenglin Xu, Rong Jin, Irwin King, and Michael Lyu, "An extended level method for efficient multiple kernel learning," in *Advances in Neural Information Processing Systems 21*. 2009.

[19] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009.

[20] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien, "lp-norm multiple kernel learning," *J. Mach. Learn. Res.*, 2011.

[21] Xinxing Xu, Ivor W. Tsang, and Dong Xu, "Soft margi multiple kernel learning," *IEEE Trans. Neural Networks Learning Systems*, May 2013.

[22] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.