Improving Control of Dexterous Hand Prostheses Using Adaptive Learning

Tatiana Tommasi, Francesco Orabona, Claudio Castellini, and Barbara Caputo

***Abstract*—At the time of this writing, the main means of control for polyarticulated self-powered hand prostheses is surface electromyography (sEMG). In the clinical setting, data collected from two electrodes are used to guide the hand movements selecting among a finite number of postures. Machine learning has been applied in the past to the sEMG signal (not in the clinical setting) with interesting results, which provide more insight on how these data could be used to improve prosthetic functionality. Researchers have mainly concentrated so far on increasing the accuracy of sEMG classification and/or regression, but, in general, a finer control implies a longer training period. A desirable characteristic would be to shorten the time needed by a patient to learn how to use the prosthesis. To this aim, we propose here a general method to reuse past experience, in the form of models synthesized from previous subjects, to boost the adaptivity of the prosthesis. Extensive tests on databases recorded from healthy subjects in controlled and noncontrolled conditions reveal that the method significantly improves the results over the baseline nonadaptive case. This promising approach might be employed to pretrain a prosthesis before shipping it to a patient, leading to a shorter training phase.**

***Index Terms*—Electromyography, human–computer interfaces, learning and adaptive systems, prosthetics.**

I. Introduction

In the prosthetics/rehabilitation robotics community, it is generally understood nowadays [1]–[3] that advanced hand prostheses are in dire need of accurate and reliable control schemas to make them easy to use by the patient. Together with excessive weight and low reliability, *lack of control* (inconsistency between the desired and performed movements) is the main reason why 30–50% of upper limb amputees do not use their prosthesis regularly [4], although the exact factors leading to abandonment of a prosthesis seem to be dependent on the age and status of each subject, still remaining to be thoroughly investigated [5].

Fig. 1. Dexterous hand prostheses. (From left to right) RSL Steeper's *BeBionic* (reproduced from www.bebionic.com), Vincent Systems' *Vincent hand* (www.handprothese.de), and Touch Bionics's *i-LIMB Ultra* (www.touchbionics.com).

The force-controlled and polyarticulated hand prosetheses that are currently being used in the clinical setting are not yet comparable with nonprosthetic mechanical hands but enjoy a high level of dexterity. They have five fingers and can potentially achieve an infinite number of configurations. They include, i.e., the *BeBionic* hand by RSL Steeper, Vincent Systems' *Vincent hand*, and the *i-LIMB* by Touch Bionics (see Fig. 1). However, control by the patient is poor, and it is still enforced using two surface electromyography (sEMG) electrodes and complex sequences of muscle contraction impulses; this is essentially an old scheme used since the 1960s [6]–[8]. The patient must get acquainted and proficient with this "language" if (s)he wants to achieve a minimum degree of control over the prosthesis.

To overcome this drawback, a more "natural" form of control has been individuated and studied for two decades; namely, sEMG has been revamped by the application of machine learning techniques. More electrodes (typically 5+) and complex statistical classification/regression techniques (e.g., support vector machines [9], linear discriminants [10], [11], and neural networks [12]) allow, at least in principle, to more easily detect what the patient wants to do. The word "natural" here is still quite a misnomer, as it refers to the choice among a finite number of predefined hand configurations, but this kind of control is still much more natural than before, as each posture is achieved by configuring one's muscle remnants as they would be if the missing limb were still there. Recent results on amputees indicate that even long-term patients can generate rather precise residual activity to the extent that there is essentially no statistically significant difference in the classification/regression accuracy attained by transradial amputees and intact subjects [9], [13].

In this paper, we concentrate upon a specific aspect of hand prostheses control, namely, we try to reduce the *training time*, i.e., the time required for adaptation of the prosthesis itself to the patient. Anatomical similarity among humans intuitively

Manuscript received December 22, 2011; revised June 28, 2012; accepted October 19, 2012. Date of publication November 26, 2012; date of current version February 1, 2013. This paper was recommended for publication by Associate Editor T. Asfour and Editor D. Fox upon evaluation of the reviewers' comments. This work was supported in part by the Swiss National Science Foundation *Sinergia* project Ninapro (Non-Invasive Adaptive Prosthetics). The work of T. Tommasi was supported by the EMMA Hasler Project.

T. Tommasi is with the Idiap Research Institute, Martigny 1920, Switzerland, and also with the École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: tatiana.tommasi@idiap.ch).

F. Orabona is with the Toyota Technological Institute at Chicago, Chicago, IL 60637 USA (e-mail: francesco@orabona.com).

C. Castellini is with the Robotics and Mechatronics Center, German Aerospace Research Center Oberpfaffenhofen, Wessling 82234, Germany (e-mail: claudio.castellini@dlr.de).

B. Caputo is with the Idiap Research Institute, Martigny 1920, Switzerland (e-mail: barbara.caputo@idiap.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2012.2226386

suggests that good statistical models built in the past might be proficiently reused when training a prosthesis for a new patient. This idea cannot be naively enforced with standard learning techniques, as shown at least in [14], where cross-subject analysis (i.e., using a model trained on a subject to do prediction on a new subject) is performed with poor results. We present here a more refined approach to the problem exploiting *adaptive learning* in order to boost the training phase of a hand prosthesis by reusing previous experience.

We build on our own previous work [15], which proposed a principled method *to choose one* from among multiple pretrained models on known subjects as a source for adaptation, and to evaluate the right degree of closeness to the target task for a new subject. This approach was based on an estimate of the model generalization ability through the leave-one-out error that was minimized solving a *nonconvex* optimization task. Here, we improve the original method in two key aspects: 1) we constrain the new model, to be close *to a linear combination* of pretrained models stored in the memory of the prosthesis; and 2) the learning process to define from whom and how much to adapt is now defined through a *convex* optimization problem, avoiding local minima issues. This leads to a bootstrapping in the control abilities of the new subject, who can now acquire control of the device faster than what would be achieved without adaptation.

We test our method on two databases. The first is the one already described in [14] and [15], consisting of sEMG, posture, and force signals gathered from ten intact subjects in various (controlled and noncontrolled) laboratory situations. The second is the NinaPro database [16], which is a publicly available database that contains kinematic and sEMG data from the upper limbs of 27 intact subjects, while performing a total of 52 hand postures. The benefits are apparent and the goal is to be able to ship a pretrained prosthesis to which the patient could very quickly adapt, get comfortable with quickly, and be able to use in daily life activities.

This paper is organized as follows. After reviewing related work, in Sections II and III, we present our method. Section IV describes the databases used, while Section V shows and discusses the results. Finally, Section VI contains conclusions and ideas for future work.

A. Related Work

1) Using Surface Electromyography for Hand Prostheses: sEMG detects muscle unit activation potentials, which typically present a quasi-linear relation with the force exerted by the muscle to which the electrode is applied. In the more specific case of hand prostheses, several electrodes are applied to the forearm (or stump), while the subject reaches specific hand configurations (postures) and/or grabs a force sensor. The raw sEMG signal is then preprocessed (filtered, rectified, and subsampled); features are subsequently extracted from it and fed, together with force values and labels denoting the postures, to a (usually supervised) machine learning method. Hand postures can be classified accordingly, and the force applied is predicted using a regression scheme. The two processes can happen simultaneously [17]. Up to 12 hand postures [13] have been classified with acceptable accuracy, and there are strong hints [9], [13], [18]–[20] that with data from transradial amputees, it may be possible to achieve similar performance.

Comprehensive surveys can be found in [2], [3], and [21], and the most recent results at the time of this writing are probably those exposed in [22]–[24] and [25]. The use of sEMG has been widely explored, and a number of possible features have been extracted and tested with many machine learning methods.

2) Adaptive Learning: One of the main assumptions of machine learning is that the training data on which any method is learned and the test data on which it is verified are drawn from the same distribution. However, in real problems, this is not always the case, and adaptive learning is used to overcome the distribution mismatch. In general, the goal of transfer learning [26] and domain adaptation [27] is to reuse information gathered on some source task when solving a new target problem, and they, respectively, address two aspects of this problem. Transfer learning focuses mostly on binary tasks and on the use of helpful information across different categories (classes with different labels). Domain adaptation considers the possibility of exploiting common information among slightly different tasks when the set of labels is the same. By applying domain adaptation, data collected in different domains can be used together (source + target), or it is possible to leverage on pretrained models built on rich training sets (source) when facing the same problem in a new domain with few available samples (target).

Over the past few years, various techniques for domain adaptation and transfer learning have gained attention in natural language processing [28], [29], computer vision [30]–[32], and sentiment classification [33], [34]. Many adaptive methods have been compared and benchmarked in [29]; however, most of them are computationally inefficient because it is necessary to retrain each time over old source and new target data. An approach that does not use retraining, based on support vector machines (SVMs), has been proposed in [35], but the authors do not address the possibility that the known source model may be too different from the new target one because of high variability between the two domains.

3) Adaptive Learning on Surface Electromyography Data: Adaptive learning can be used to augment the prostheses control and, in particular, to shorten the training time or aid the collection of training data. One interesting attempt in this direction can be found in [36], where two adaptive methods (one supervised and the other unsupervised) are shown to dramatically outperform a nonadaptive approach.

The solution of adapting from data collected on different subjects is adopted in [37]. Decoupling between subject-dependent and motion-dependent components is enforced on a limited dataset, and an improvement over the baseline method is shown. In [38], samples from multiple source subjects are combined with the target subject samples. When learning the final classifier on the whole set of data, a weighting factor is added to evaluate the real relevance of each source with respect to the target task. The sensitivity of the method to this parameter is evaluated empirically, but how to choose it is left as an open problem.

In [15], we proposed an approach that exploited previously trained models on known subjects as a starting point when learning on a new one. This method chooses automatically the best prior knowledge to use and how much to rely on it, overcoming at the same time the problems present in [35] and [38]. Compared with the work in [36] that performs adaptation *during* the prediction task, our algorithm defines a way of boosting the performance in training, i.e., *before* beginning the prediction. We propose here to enlarge the approach in [15], even building over our work in [31] that shares the same basic mathematical framework. Specifically, we propose a novel multiclass adaptive learning method that is able to rely upon many prior knowledge models at the same time, with the aim of exploiting at the best all the available information.

II. Defining the Adaptive Model

In this section, we describe the mathematical framework at the basis of our adaptive learning method. We first introduce the basic notation (see Section II-A); then, we present our algorithm for online model adaptation from the best known subject (see Section II-B) and how to enlarge it to exploit multiple known subjects (see Sections II-C). We conclude by explaining how to extend the described approach in the multiclass setting (see Section II-D).

In the following, we denote with small and capital bold letters, respectively, column vectors and matrices, e.g., $\boldsymbol{a} = [a_1, a_2, \ldots, a_N]^T \in \mathbb{R}^N$ and $\boldsymbol{A} \in \mathbb{R}^{G \times N}$ with A_{ji} corresponding to the (j, i) element. The subscripts indicate specific rows and columns. When only one subscripted index is present, it represents the column index, e.g., $\boldsymbol{A}_i$ is the ith column of the matrix $\boldsymbol{A}$.

A. Background

Assume $\boldsymbol{x}_i \in \mathbb{R}^m$ is an input vector and $y_i \in \mathbb{R}$ is its associated output. Given a set $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ of samples drawn from an unknown probability distribution, we want to find a function $f(\boldsymbol{x})$ such that it determines the best corresponding y for any future sample $\boldsymbol{x}$. This is a general framework that includes both regression and classification. The problem can be solved in various ways. Here, we will use kernel methods and, in particular, least-squares support vector machines (LS-SVM) [39]. In LS-SVM, the function $f(\boldsymbol{x})$ is built as a linear model $\mathbf{w} \cdot \phi(\boldsymbol{x}) + b$, where $\phi(\cdot)$ is a nonlinear function mapping input samples to a high-dimensional (possibly infinite-dimensional) Hilbert space called *feature space*. Rather than being directly specified, the feature space is usually induced by a *kernel function* $K(\boldsymbol{x}, \boldsymbol{x}')$, which evaluates the inner product of two samples in the feature space itself, i.e., $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$. A common kernel function is the Gaussian kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|x - x'\|^2) \qquad (1)$$

that will be used in all our experiments.

The parameters of the linear model, i.e., $\mathbf{w}$ and b, are found by minimizing a regularized least-squares loss function [39]. This approach is similar to the well-known formulation of SVMs, the difference being that the loss function is the square loss that does not induce sparse solutions.

This formulation can be easily generalized to the multiclass classification, where we have $g = 1, \ldots, G$ different classes. Consider one model for each class, $\mathbf{w}_g$ and b_g, that discriminates one class against the others (i.e., one-versus-all). Hence, the model g is trained on the binary problem to distinguish class g, which is considered positive, versus all the others, which are considered negative. The predicted class for sample i is then defined as $\text{argmax}_g\{\mathbf{w}_g \cdot \phi(\boldsymbol{x}_i) + b_g\}$.

A key concept that we will use is the one of leave-one-out predictions [40]. Denote by $\tilde{y}_i,\ i = 1, \ldots, N$, the prediction on sample i when it is removed from the training set and by $\ell(y, \tilde{y})$ a generic loss function that measures the loss of predicting $\tilde{y}$ when the true label is y. We have that $\frac{1}{N}\sum_{i=1}^N \ell(y_i, \tilde{y}_i)$ is an almost unbiased estimator of the classifier generalization error [41], which is measured using ℓ.

LS-SVMs make it possible to write the leave-one-out predictions in closed form and with a negligible additional computational cost [40]. This property is useful to find the best parameters for learning [e.g., γ in (1)], and it will be used in our adaptation method. Note that we use the same general formulation to solve both regression and classification problems.

B. Model Adaptation From the Best Subject

Let us assume we have K pretrained models stored in memory, which are trained offline on data acquired on K different subjects. When the prosthetic hand starts to be used by subject $K + 1$, the system begins to acquire new data. Given the differences among the subjects' arms as well as the placement of the electrodes, these new data will belong to a new probability distribution, which is, in general, different from the K previously modeled and stored. Still, as all subjects perform the same grasp types, it is reasonable to expect that the new distribution will be *close* to at least one of those already modeled; then, it should be possible to use one of the pretrained models as a *starting point* for training on the new data. We expect that, by doing so, learning should be faster than using the new data alone. To solve this problem, we generalize the adaptation approach proposed in [35] for SVMs. The basic idea is to slightly change the regularization term of the SVM cost function so that the solution will be close to the pretrained one. The optimization problem is

$$\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w} - \hat{\mathbf{w}}\|^2 + C\sum_{i=1}^N \xi_i$$
$$\text{subject to } \xi_i \geq 0, \;\; y_i(\mathbf{w} \cdot \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \qquad (2)$$

where $\hat{\mathbf{w}}$ is a pretrained model, and C is a parameter to tradeoff the errors and the regularization. In order to tune the closeness of $\mathbf{w}$ to $\hat{\mathbf{w}}$, we introduce a scaling factor β, weighing the pretrained model. In addition, we use the square loss and, therefore, resort to the LS-SVM formulation. In this way, the leave-one-out predictions can be evaluated in closed form, enabling automatic tuning of β. The optimization problem reads now like this [15]

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}-\beta\hat{\mathbf{w}}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2$$
$$\text{subject to } y_i = \mathbf{w}\cdot\phi(\boldsymbol{x}_i) + b + \xi_i \tag{3}$$

and the corresponding Lagrangian problem is

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}-\beta\hat{\mathbf{w}}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N} a_i\{\mathbf{w}\cdot\phi(\boldsymbol{x}_i) + b + \xi_i - y_i\} \tag{4}$$

where $\boldsymbol{a}\in\mathbb{R}^N$ is the vector of Lagrange multipliers. The optimality conditions can be expressed as

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \beta\hat{\mathbf{w}} + \sum_{i=1}^{N} a_i\phi(\boldsymbol{x}_i) \tag{5}$$

$$\frac{\partial\mathcal{L}}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} a_i = 0 \tag{6}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = 0 \Longrightarrow a_i = C\xi_i \tag{7}$$

$$\frac{\partial\mathcal{L}}{\partial a_i} = 0 \Longrightarrow \mathbf{w}\cdot\phi(\boldsymbol{x}_i) + b + \xi_i - y_i = 0. \tag{8}$$

From (5), it is clear that the adapted model is given by the sum of the pretrained model $\hat{\mathbf{w}}$ (weighted by β) and a new model $\mathbf{w}$ obtained from the new samples. Note that when β is 0, we recover the original LS-SVM formulation without any adaptation to previous data. Using (5) and (7) to eliminate $\mathbf{w}$ and $\boldsymbol{\xi}$ from (8), we find that

$$\sum_{j=1}^{N} a_j\phi(\boldsymbol{x}_j)\cdot\phi(\boldsymbol{x}_i) + b + \frac{a_i}{C} = y_i - \beta\hat{\mathbf{w}}\cdot\phi(\boldsymbol{x}_i). \tag{9}$$

Denoting with $\boldsymbol{K}$ the kernel matrix, i.e., $\boldsymbol{K}_{ji} = \boldsymbol{K}(\boldsymbol{x}_j,\boldsymbol{x}_i) = \phi(\boldsymbol{x}_j)\cdot\phi(\boldsymbol{x}_i)$, the obtained system of linear equations can be written more concisely in matrix form as

$$\begin{bmatrix}\mathbf{K}+\frac{1}{C}\mathbf{I} & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{a}\\ b\end{bmatrix} = \begin{bmatrix}\boldsymbol{y}-\beta\hat{\boldsymbol{y}}\\ 0\end{bmatrix} \tag{10}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the vectors containing, respectively, the label samples and the prediction of the previous model, i.e., $\boldsymbol{y} = [y_1,\ldots,y_N]^T$, $\hat{\boldsymbol{y}} = [\hat{\mathbf{w}}\cdot\phi(\boldsymbol{x}_1),\ldots,\hat{\mathbf{w}}\cdot\phi(\boldsymbol{x}_N)]^T$. Thus, the model parameters can be calculated with

$$\begin{bmatrix}\boldsymbol{a}\\ b\end{bmatrix} = \boldsymbol{P}\begin{bmatrix}\boldsymbol{y}-\beta\hat{\boldsymbol{y}}\\ 0\end{bmatrix} \tag{11}$$

where $\boldsymbol{P} = \boldsymbol{M}^{-1}$, and $\boldsymbol{M}$ is the first matrix on the left in (10).

We now show that for (3), it is possible to write the leave-one-out predictions in a closed formula (see the proof in the Appendix). Let $[\boldsymbol{a}'^T, b']^T = \boldsymbol{P}[\boldsymbol{y}^T, 0]^T$ and $[\boldsymbol{a}''^T, b'']^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^T, 0]^T$ with $\boldsymbol{a} = \boldsymbol{a}' + \beta\boldsymbol{a}''$. Then, we have the following.

Proposition 1: The prediction $\tilde{y}_i$, which is obtained on sample i when it is removed from the training set, is equal to

$$y_i - \frac{a_i'}{P_{ii}} + \beta\frac{a_i''}{P_{ii}}. \tag{12}$$

Notice that in the above formula, β is the only parameter; hence, it is possible to set it optimally in order to minimize the sum of the leave-one-out errors $\ell(y_i,\tilde{y}_i)$, while at the same time choosing the best pretrained model for adaptation. Moreover, $\boldsymbol{a}$ depends linearly on β; thus, it is straightforward to define the learning model that is fixed once β has been chosen.

The complexity of the algorithm is dominated by the evaluation of the matrix $\boldsymbol{P}$, which must anyway occur while training; thus, the computational complexity of evaluating the leave-one-out errors is negligible, if compared with the complexity of training. As a last remark, we underline that the pretrained model $\hat{\mathbf{w}}$ can be obtained by any training algorithm, as far as it can be expressed as a weighted sum of kernel functions. The framework is, therefore, very general.

C. Model Adaptation From Multiple Subjects

The approach described in the previous section has a main drawback. Although many prior knowledge models are available, it uses only one of them, which is selected as the most useful in terms of minimal leave-one-out errors. Even if the pretrained models are not equally informative, relying on more than one of them may be beneficial. To this goal, it is possible to define a new learning problem that considers the linear combination of all the known models [31]

$$\min_{\mathbf{w},b} \frac{1}{2}\left\|\mathbf{w} - \sum_{k=1}^{K}\beta^k\hat{\mathbf{w}}^k\right\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2$$
$$\text{subject to} \quad y_i = \mathbf{w}\cdot\phi(\boldsymbol{x}_i) + b + \xi_i. \tag{13}$$

The original single coefficient β has been substituted with a vector $\boldsymbol{\beta}$ containing as many elements as the number of prior models K. For this formulation, the optimal solution is

$$\mathbf{w} = \sum_{k=1}^{K}\beta^k\hat{\mathbf{w}}^k + \sum_{i=1}^{N} a_i\phi(\boldsymbol{x}_i). \tag{14}$$

Here, $\mathbf{w}$ is expressed as a weighted sum of the pretrained models scaled by the parameters β^k plus the new model built on the incoming training data [31]. The leave-one-out prediction of each sample i can again be written in closed form, similar to Proposition 1, as

$$\tilde{y}_i = y_i - \frac{a_i'}{P_{ii}} + \sum_{k=1}^{K}\beta^k\frac{a_i''^k}{P_{ii}} \tag{15}$$

where $[\boldsymbol{a}''^{kT}, b''^k]^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^{kT}, 0]^T$, and $\hat{\boldsymbol{y}}^k$ is the vector that contains the predictions of the kth previous model $[\hat{\mathbf{w}}^k\cdot\phi(\boldsymbol{x}_1),\ldots,\hat{\mathbf{w}}^k\cdot\phi(\boldsymbol{x}_N)]$. As before, the leave-one-out errors can be calculated and minimized to evaluate the best weights β^k.

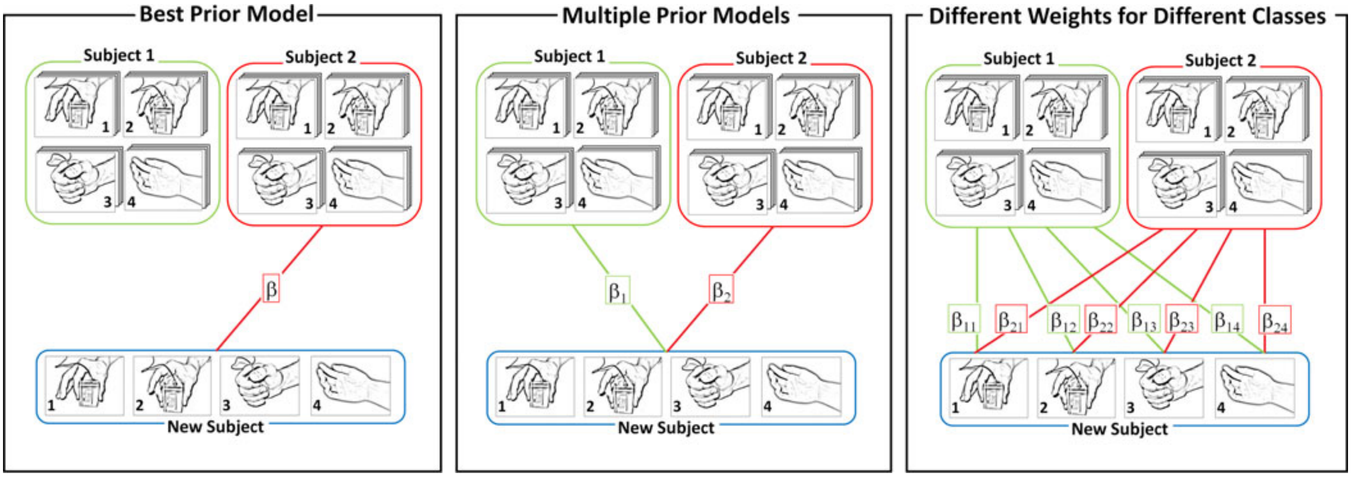

Fig. 2. Three methods adopted to leverage information from multiple known subjects when learning on a new one. For all the known subjects, many sEMG signal samples are available, while few sEMG signals are recorded from the new subject. (Left) Choosing only the best known subject and using its reweighted model as a starting point for learning. (Center) Considering a linear combination of the known subjects with equal weight for all the grasp models of each subject. (Right) Considering again a linear combination of all the known models but assign a different weight to each grasp model for each subject.

D. Multiclass Extensions

In case of classification problems, the methods discussed up until here are suitable for binary tasks but can be extended to the case of G classes using the one-versus-all formulation described in Section II-A. We define the matrix $\boldsymbol{Y} \in \mathbb{R}^{G\times N}$ composed of the columns $\boldsymbol{Y}_i$, where for each sample i, the vector $\boldsymbol{Y}_i$ has all the components equal to -1 except for the y_ith that is equal to 1. In the same way, define the matrix $\hat{\boldsymbol{Y}}$, which is composed of the columns $\hat{\boldsymbol{Y}}_i$ that contain the predictions generated by a known multiclass model on the sample i. For each sample i, we also obtain a vector of G leave-one-out predictions; we indicate it with $\tilde{\boldsymbol{Y}}_i$, and it is easy to show that it can be calculated as

$$\tilde{\boldsymbol{Y}}_i = \boldsymbol{Y}_i - \frac{\boldsymbol{A}'_i}{P_{ii}} + \beta\frac{\boldsymbol{A}''_i}{P_{ii}} \tag{16}$$

where

$$[\boldsymbol{A}', \boldsymbol{b}'] = [\boldsymbol{Y}, \boldsymbol{0}]\boldsymbol{P}^T \tag{17}$$

$$[\boldsymbol{A}'', \boldsymbol{b}''] = [\hat{\boldsymbol{Y}}, \boldsymbol{0}]\boldsymbol{P}^T. \tag{18}$$

Here, $\boldsymbol{A}', \boldsymbol{A}'' \in \mathbb{R}^{G\times N}$, and $\boldsymbol{b}, \boldsymbol{0} \in \mathbb{R}^G$. In case of multiple prior models, we use the superscript k to indicate each of them, and considering their linear combination, we get

$$\tilde{\boldsymbol{Y}}_i = \boldsymbol{Y}_i - \frac{\boldsymbol{A}'_i}{P_{ii}} + \sum_{k=1}^{K}\beta^k\frac{\boldsymbol{A}''^k_i}{P_{ii}} \tag{19}$$

with

$$[\boldsymbol{A}''^k, \boldsymbol{b}''^k] = [\hat{\boldsymbol{Y}}^k, \boldsymbol{0}]\boldsymbol{P}^T. \tag{20}$$

III. Learning How Much to Adapt

The adaptive learning methods described above look for the model parameters ($\mathbf{w}$,b) once the value of the weight β, or the corresponding vector $\boldsymbol{\beta}$, has been chosen. Searching the optimal β defines a separate learning problem that depends on the choice of the loss function ℓ. As a result, we have an indication of how much each of the pretrained models is reliable for adaptation. In the following, we define how to face this issue in the classification and regression cases; a general scheme of the proposed solutions is shown in Fig. 2.

A. Classification

For a binary classification problem, and in case of a single pretrained model, we can follow the approach proposed in [40] and find β by minimizing leave-out-out errors using the logistic loss function

$$\ell(y_i, \tilde{y}_i) = \frac{1}{1+\exp(-10(\tilde{y}_i - y_i))}. \tag{21}$$

Note that the resulting objective function would be nonconvex w.r.t. β. When moving to the choice of multiple weights for all the pretrained models, we can also overcome the nonconvexity issue described above, by minimizing the loss function proposed in [31]

$$\ell(y_i, \tilde{y}_i) = \max(1 - y_i\tilde{y}_i,\ 0). \tag{22}$$

This is a convex upper bound to the misclassification loss, and it also has a smoothing effect, similar to the logistic function in (21).

However, in our application, we have multiple pretrained models and G classes, corresponding to the different grasp types. Hence, it is necessary to define a loss function over vectors, which compose all these values to define a single estimate of the multiclass error.

1) Best Prior Model: A first solution could be to consider

$$\ell(\boldsymbol{Y}_i, \tilde{\boldsymbol{Y}}_i) = \frac{1}{1+\exp(-10(\max_{g\neq y_i}\{\tilde{Y}_{gi}\} - \tilde{Y}_{y_i i}))} \tag{23}$$

and to evaluate it separately for each of the $k \in \{1,\ldots,K\}$ pretrained models on the basis of (16), varying β with small steps in $[0, 1]$ (this is the approach used in [15]). The minimal result identifies both the best known subject for adaptation and, at the same time, the corresponding β. Still, this approach, as (21), is nonconvex; thus, reaching the global optimum is not computationally efficient. This solution is schematically depicted in Fig. 2 (left).

2) Multiple Prior Models: To consider multiple prior knowledge models, we propose using (19) in the convex multiclass loss [42]

$$\ell(\boldsymbol{Y}_i, \tilde{\boldsymbol{Y}}_i) = \max\{1 - \tilde{Y}_{y_i i} + \max_{g \neq y_i}\{\tilde{Y}_{gi}\},\ 0\}. \tag{24}$$

This loss is zero if the confidence value for the correct label is larger by at least 1 than the confidence values assigned to the rest of the labels. Otherwise, we suffer a loss that is linearly proportional to the difference between the confidence value of the correct label and the maximum among the confidence values of the other labels. The final objective function is

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell(\boldsymbol{Y}_i, \tilde{\boldsymbol{Y}}_i) \ \text{ subject to } \ \|\boldsymbol{\beta}\|_2 \leq 1,\ \ \beta^k \geq 0. \tag{25}$$

The condition of having $\boldsymbol{\beta}$ in the intersection of the unitary ball and the positive semiplane can be seen as a form of regularization, and it is a natural generalization of the original constraint $\beta \in [0, 1]$ used in [15]. This constraint is necessary to avoid overfitting problems that can happen when the number of known models is large compared with the number of training samples [31].

We implemented the optimization process using a simple projected subgradient descendent algorithm, where at each iteration, $\boldsymbol{\beta}$ is first projected onto the l_2-sphere, $\|\boldsymbol{\beta}\|_2 \leq 1$, and then onto the positive semiplane. The pseudocode is given in Algorithm 1, where in line 8, $\mathbf{1}\{\cdot\}$ denotes the indicator function. Fig. 2(center) describes this solution.

3) Different Weights for Different Classes: Until now, we considered techniques that assign a unique weight to each known subject. This means that the whole set of one-versus-all pretrained models for a subject are equally weighted. However, for example, when learning the model for the first class, it may be useful to give more weight in adaptation to the first subject than to the second, while it could be the opposite when learning the model for the second class, and so on. Hence, to have one more degree of freedom and decide the adaptation specifically for each class, we enlarge the set of weight parameters, introducing the matrix $\mathbf{B} \in \mathbb{R}^{K \times G}$, where each row $k \in \{1, \ldots, K\}$ contains the vector $\boldsymbol{\beta}_k^T$ with G elements, one for each class. This approach is described in Fig. 2 (right).

The optimization problem is analogous to the one described in (25), with a change in the constraints. Each class problem is now considered separately; therefore, we have G conditions, one for each of the columns $\mathbf{B}_g$ of the $\mathbf{B}$ matrix; we impose $\|\mathbf{B}_g\|_2 \leq 1$ and $B_{ji} \geq 0$.

B. Regression

Our goal in using regression is the prediction of the force applied by one subject in grasping, independent of the specific kind of grasp performed. Thus, now the output y_i for each corresponding input $\boldsymbol{x}_i$ is a continuous real value, rather than a discrete one as in classification.

Algorithm 1. Projected Sub-gradient Descent Algorithm

```
1:  $\boldsymbol{\beta} = [\beta_1 \ldots \beta^K] \leftarrow \mathbf{0}$
2:  $t \leftarrow 1$
3:  calculate $\boldsymbol{A}'$ according to (17)
4:  calculate $\boldsymbol{A}''^k$ according to (20)
5:  **repeat**
6:      $\tilde{\boldsymbol{Y}}_i \leftarrow \boldsymbol{Y}_i - \frac{\boldsymbol{A}'_i}{P_{ii}} + \sum_{k=1}^{K} \beta^k \frac{\boldsymbol{A}''^k_i}{P_{ii}}\ \forall i = 1, \ldots, N$
7:      $g_i^* \leftarrow \text{argmax}_{g \neq y_i}\{\tilde{Y}_{gi}\}\ ,\ \forall i = 1, \ldots, N$
8:      $d_i \leftarrow \mathbf{1}\{1 - \tilde{Y}_{y_i i} + \tilde{Y}_{g_i^* i} > 0\}\ ,\ \forall i = 1, \ldots, N$
9:      $\beta^k \leftarrow \beta^k - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i \frac{(A''^k_{g_i^* i} - A''^k_{y_i i})}{P_{ii}}\ ,\ \forall k = 1, \ldots, K$
10:     **if** $\|\boldsymbol{\beta}\|_2 > 1$ **then**
11:         $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}/\|\boldsymbol{\beta}\|_2$
12:     **end if**
13:     $\beta^k \leftarrow \max(\beta^k,\ 0),\ \forall k = 1, \ldots, K$
14:     $t \leftarrow t + 1$
15: **until convergence**
```

Output: $\boldsymbol{\beta}$

Similar to what we showed before, it is possible to learn the regression model relying on information from the closest known subject or on the combination of multiple pretrained models.

1) Best Prior Model: We can use the leave-one-out prediction in (12) to evaluate the square loss [mean square error (MSE)]

$$\ell(y_i, \tilde{y}_i) = (y_i - \tilde{y}_i)^2 = \left(\frac{a'_i}{P_{ii}} + \beta \frac{a''_i}{P_{ii}}\right)^2.$$

The choice of the square loss gives us, summing over i, a quadratic function in β, and the minimum is obtained using

$$\beta = \frac{\sum_{i=1}^{N} \frac{a'_i}{P_{ii}} \frac{a''_i}{P_{ii}}}{\sum_{i=1}^{N} \left(\frac{a''_i}{P_{ii}}\right)^2}. \tag{26}$$

We use the constraint $\beta \geq 0$, just imposing $\beta = 0$ every time it results negative. Hence, different from the classification case, here we do not need any optimization procedure; the optimal β is given by a closed formula. Once calculated the minimum value of the summed square loss values for each $k \in \{1, 2, \ldots K\}$, comparing all of them, we can identify the best known subject to use for adaptation when learning the regression model on a new subject.

2) Multiple Prior Models: To take advantage of all the available pretrained models, we can combine them linearly and search for a vector of weights as in classification. Hence, the

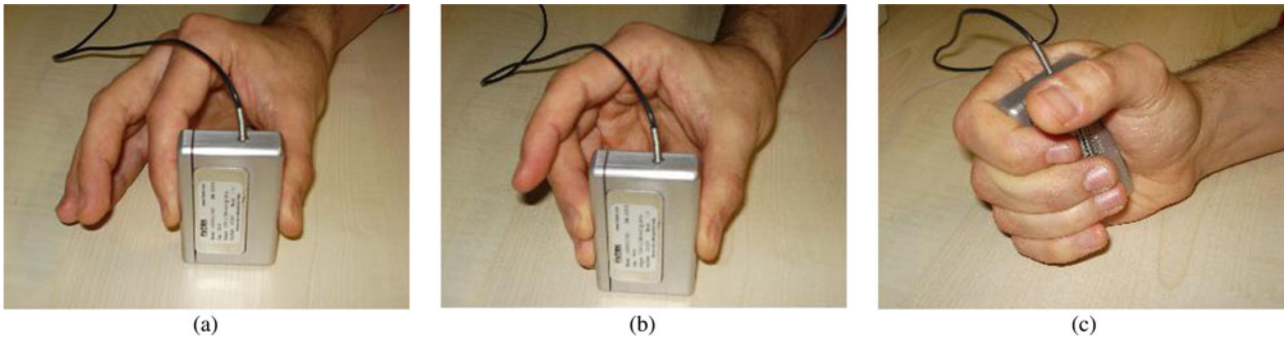

Fig. 3. Three different grasp types recorded in the hand posture and force signal dataset [14]. (a) Index precision grip. (b) Other fingers precision grip. (c) Power grasp. (Reproduced from [14].)

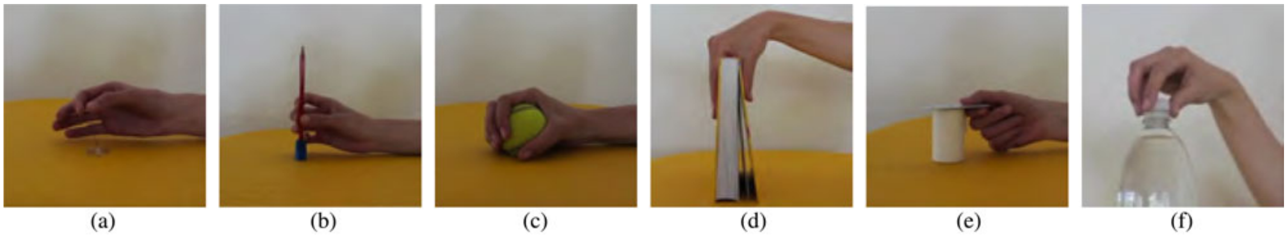

Fig. 4. Six different grasp types extracted from the Ninapro dataset [16]. (a) Tip pinch grasp. (b) Prismatic four fingers grasp. (c) Power grasp. (d) Parallel extension grasp. (e) Lateral grasp. (f) Opening a bottle with a tripod grasp. (Reproduced from [16].)

loss function ℓ can now be defined as

$$\ell(y_i, \tilde{y}_i) = \left(\frac{a'_i}{P_{ii}} + \sum_{k=1}^{K} \beta^k \frac{a''^k_i}{P_{ii}} \right)^2. \tag{27}$$

Adding also the condition $\|\boldsymbol{\beta}\|_2 \leq 1$, we can find the best $\boldsymbol{\beta}$ vector that minimizes the loss with a quadratically constrained quadratic program solver. In our experiments, we used CVX [43], which is a package for specifying and solving convex programs in MATLAB.

IV. Experimental Data

To test the effectiveness of our model adaption techniques, we use two datasets.

A. *Hand Posture and Force Signals [14]*

This database of sEMG/hand posture/force signals has already been presented in [14] and used in [14] and [15]. (The following description of the database is very concise; see the above cited paper(s) for more details.) The signals are collected from ten intact subjects (two women and eight men) using seven sEMG electrodes (Aurion ZeroWire wireless) placed on the dominant forearm according to the medical literature [44]. A FUTEK LMD500 force sensor [45] is used to measure the force applied by the subject's hand during the recording. Data are originally sampled at 2 kHz. Each subject starts from a rest condition (sEMG baseline activity) and then repeatedly grasps the force sensor using, in turn, three different grips, which are visible in Fig. 3. The subject either remains seated and relaxed while performing the grasps or is free to move (walk around, sit down, stand up, etc.). These phases are referred to as *Still-Arm (SA)* and *Free-Arm (FA)*, respectively. Each grasping action is repeated along 100 s of activity. The whole procedure is repeated twice. The root mean square of the signals along 1 s (for classification) and 0.2 s (for regression) is evaluated; subsampling at 25 Hz follows. Samples for which the applied force is lower than 20% of the mean force value obtained for each subject are labeled as "rest" class. After this preprocessing, we got around 15 000 samples per subject; each sample consists of a seven-element sEMG signal vector and one force value.

B. *Ninapro [16]*

This database has been presented in [16] and already used in [46]. It contains kinematic and sEMG data from the upper limbs of 27 intact subjects (seven women and 20 men), while performing 12 finger, nine wrist, 23 grasping and functional movements, plus eight isometric, isotonic hand configurations. Data are collected using ten surface sEMG electrodes (double-differential OttoBock MyoBock 13E200). Eight are placed just beneath the elbow at fixed distance from the radio-humeral joint, while two are on the flextor and extensor muscles. Each subject sits comfortably on an adjustable chair in front of a table and is instructed to perform ten repetitions of each movement by imitating a video, alternated with a rest phase. The sEMG electrodes are connected to a standard data acquisition card, sampling the signals at 100 Hz, and provide an RMS rectified version of the raw sEMG signal. (For a more detailed description of the dataset, see [16] and [46].) We focused only on the grasp and functional movements extracting six actions: tip pinch, prismatic four fingers, power, parallel extension, lateral, and opening a bottle with a tripod grasp (see Fig. 4). Each of them belongs to a different branch of a hierarchy containing all the dataset hand postures, and the first three grasps are the most similar to the ones considered in [14]. We randomly extracted two sets of 10 and 20 subjects from the dataset and performed classification experiments on the described seven class (six grasps plus rest) problem considering the mean absolute value of the sEMG signal as time-domain features [46]. We repeated the preprocessing and data split procedure described in [46] with an extra subsampling of the "rest" data to get a class-balanced setting.

V. Experimental Results

As already mentioned in Section II-B, our working assumption is to have K pretrained models stored in memory; new data come from subject $K+1$ and the system starts training to build the $(K+1)$th model. The performance is then evaluated using unseen data from subject $K+1$. To simulate this scenario and to have a reliable estimation of the performance, we use a leave-one-out approach. Out of the 10 (20) subjects for which we have data recordings, we train 9 (19) models offline. These correspond to the K stored models in memory, while data from the remaining subject are used for the adaptive learning of the $(K+1)$th model. This procedure is repeated 10 (20) times, using in turn all the recorded subjects for the adaptive learning of the model.

We name the proposed adaption methods:

1) *Best-Adapt*: adaptive learning starting from the best prior knowledge model (method originally presented in [15] and revised here in Section III-A1);
2) *Multi-Adapt*: adaptive learning starting from a linear combination of the known models (see Section III-A2);
3) *Multi-perclass-Adapt*: adaptive learning (for classification) starting from a linear combination of the known models with a different weight for each class (see Section III-A3).

To assess the performance of all these methods, we compare them with the following baseline approaches:

1) *No-Adapt:* This is plain LS-SVM using only the new data for training, as it would be in the standard scenario without adaption.
2) *Prior Average:* This consists of using only the pretrained models without updating them with the new training data. We consider their average performance.
3) *Prior Start*: This corresponds to the performance of the best model chosen by Best-Adapt at the first training step.
4) *Prior Test*: This is the result that can be obtained *a posteriori*, comparing all the prior knowledge models on the test set and choosing the best one.

As a measure of performance, for classification, we use the standard classification rate; for regression, the performance index is the correlation coefficient evaluated between the predicted force signal and the real one. Although we minimized the MSE in the regression learning process, the choice of the correlation coefficient is suggested by a practical consideration. When driving a prosthesis, or even a nonprosthetic mechanical hand, we are not interested in the absolute force values desired by the subject. Mechanical hands usually cannot apply as much force as human hands do, for obvious safety reasons, or e.g., in teleoperation scenarios, they could be able to apply *much more* force than a human hand can. As already done, e.g., in [9], [14], and [17], we are instead concerned with getting a signal which is *strongly correlated* with the subject's will. The significance of the comparisons between the methods is evaluated through the sign test [47].

To build the pretrained models, we used the standard SVM algorithm. All the parameters to be set during training (C and γ of the Gaussian kernel) were chosen by cross validation. Specifically, when the subject k^* is the new problem, this is excluded form the dataset, and the parameters are chosen over the remaining set $\mathcal{K} = \{1, \ldots, K \backslash k^*\}$, looking for the values that produce on average the best recognition rate or correlation coefficient by learning on each subject k in $\mathcal{K}$ and testing on $\{\mathcal{K} \backslash k^*, k\}$.

A. Hand Posture and Force Signals [14]

For the experiments running on the dataset described in [14], the training sequences are random subsets from the entire dataset of the new subject, i.e., they are taken without considering the order in which they were acquired. We considered 24 successive learning steps; for each of them, the number of available training samples increases by 30 elements, reaching a maximum of 720 samples. The test runs over all the remaining samples. We conducted three sets of experiments, considering different prior knowledge-new problem couples: SA–SA, FA–FA, and SA–FA. In the first two cases, we have consistent recording conditions among the source and the new target problem. The last case reproduces the more realistic scenario where the prior knowledge is built on data recorded on subjects in laboratory-controlled conditions, while the new subject moves freely. We both classify the grasp type and predict the force measured by the force sensor.

Fig. 5 (left) reports the classification rate obtained at each step when using SA–SA data. The plot shows that Multi-perclass-Adapt outperforms both the baselines No-Adapt, Priors, and all the other adaptive learning methods. The difference between Multi-perclass-Adapt and Best-Adapt shows an average advantage in recognition rate of around 2% ($p < 0.03$). The gain obtained by Multi-perclass-Adapt with respect to No-Adapt ($p < 0.003$) stabilizes around 5% for 500–720 training samples.

Analogous results are obtained when considering FA–FA data. Fig. 5 (center) reports the classification rate results in this setting. Multi-perclass-Adapt shows again the best performance, but now the advantage with respect to Best-Adapt is significant ($p < 0.03$) only for less than 100 training samples. Multi-perclass-Adapt outperforms No-Adapt ($p < 0.03$) with a gain of 4% in recognition rate for 500–720 samples.

Finally, Fig. 5 (right) shows the SA–FA results. Here, the statistical comparison among Multi-perclass-Adapt, Best-Adapt, and No-Adapt is the same as in the FA–FA case.

Analyzing Fig. 5 as a whole, we can state that all the proposed adaptive methods outperform learning from scratch with the best results obtained when exploiting a linear combination of pretrained models with a different weight for each known subject and each class (Multi-perclass-Adapt). Moreover, we notice that learning with adaption with 30 training samples

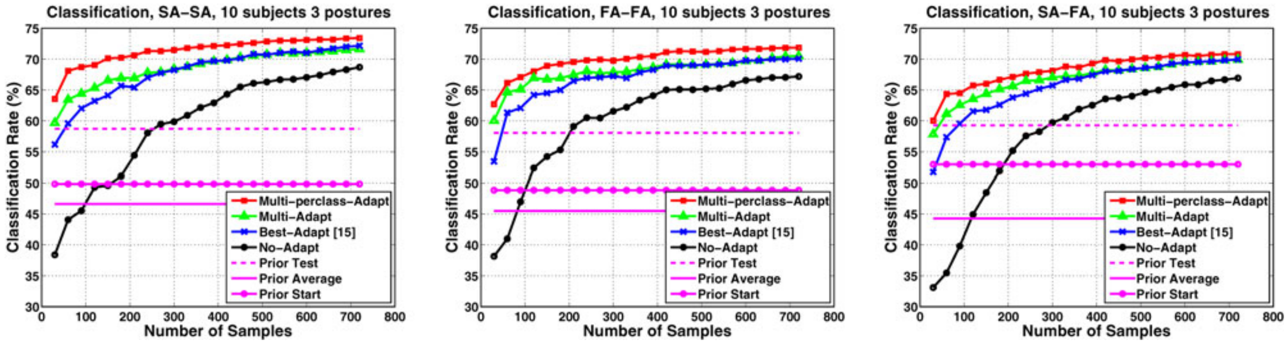

Fig. 5. Hand posture and force signals dataset [14]. Classification rate obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure specifies if the data used as source and target are registered in Still-Arm (SA) or Free-Arm (FA) setting.

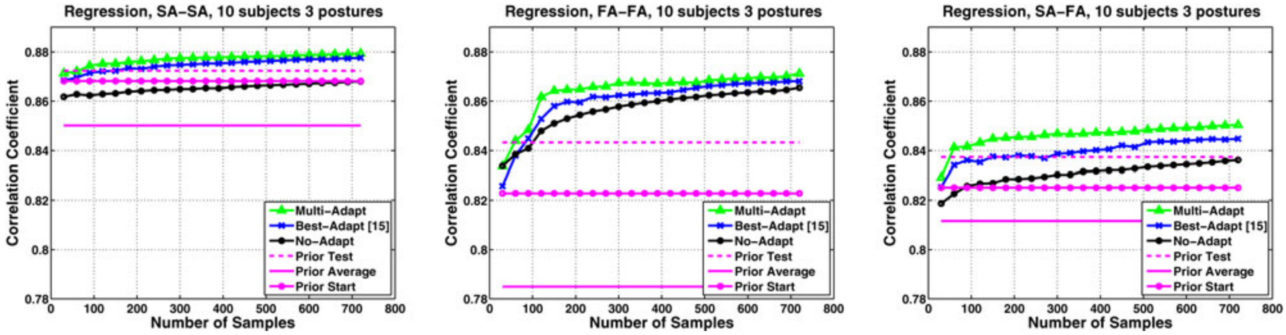

Fig. 6. Hand posture and force signals dataset [14]. Correlation coefficient obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure specifies if the data used as source and target are registered in Still-Arm (SA) or Free-Arm (FA) setting.

performs almost like No-Adapt with around 300 samples. Considering the acquisition time, this means that the adaptive methods are almost ten times faster than learning from scratch. Using the prior knowledge by itself appears to be a good choice if only very few training samples are available but loses its advantage when the dimension of the training set increases. Passing from SA–SA and FA–FA to SA–FA, we also notice that the results for Prior Average show a small drop (46.3%, 45.5%, and 44.3%) related to the change in domain between the data used for pretrained model and the one used for the new subject. The increasing difficulty of the task can also be evaluated by the progressive decrease in performance of Multi-perclass-Adapt at the very first step in the three cases: SA–SA 63.6%, FA–FA 62.7%, and SA–FA 60.0%.

The corresponding regression results are reported in Fig. 6. From the plot on the left, we notice that, in the SA–SA case, both the adaptive learning methods outperform No-Adapt ($p < 0.03$). However, here Multi-Adapt and Best-Adapt perform almost equally (no statistical significant difference).

Fig. 6 (center) shows that Best-Adapt is slightly worse than Multi-Adapt when passing to the FA–FA setting. Still, the two methods are statistically equivalent, and they show a significant gain with respect to No-Adapt only for more than 200 training samples ($p < 0.03$).

The problem becomes even harder in the SA–FA case [see Fig. 6 (right)]; here, Multi-Adapt outperforms No-Adapt only for more than 500 training samples ($p < 0.03$).

Globally, the increasing difficulty of the three regression tasks passing from left to right in Fig. 6 is demonstrated by the general drop in performance. Although we decided to show the correlation coefficient results, the corresponding MSE would lead to the same conclusions.

B. Ninapro [16]

We randomly shuffled the samples of the Ninapro dataset, and considered 36 learning steps adding 30 training samples each time until a maximum of 1080 data was achieved.

Fig. 9 (left) reports the obtained classification rate at each step when considering ten subjects for the six grasp postures plus rest. The plot shows that all the adaptive methods perform almost like No-Adapt; in particular, for less than 200 samples, there is no statistical difference between learning from scratch, learning with adaption, or using the prior knowledge directly (the fairest comparison is with Prior Average and Prior Start). It is important to remark that the "few sample" range grows together with the number of considered classes. The samples are selected randomly, and a minimum amount of data per class is needed to get meaningful classification results. Only Multi-perclass-Adapt outperforms No-Adapt ($p < 0.05$) with an average advantage of 2.5% in recognition rate for more than 200 samples.

Fig. 9 (right) shows the corresponding results in the case of 20 subjects. On average, No-Adapt and Prior Average perform almost equally to the previous case (with ten subjects), showing that the average learning capability per subject is almost stable in a fixed range. On the other hand, Prior Test and Prior Start present an increase in performance. The higher the number of available prior models, the higher the

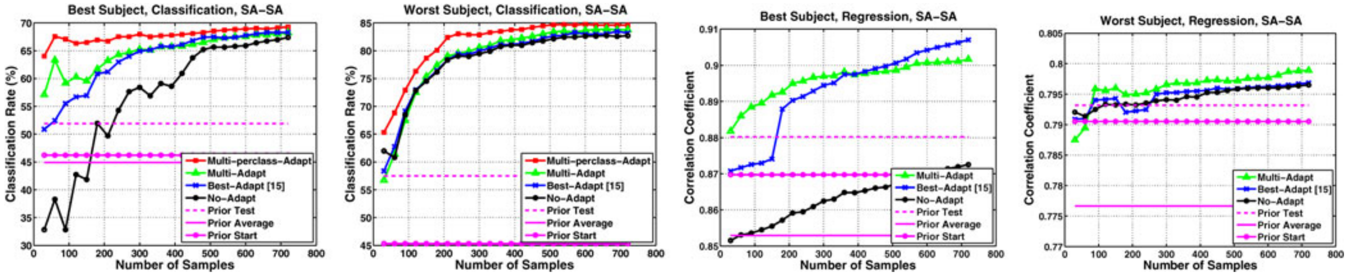

Fig. 7. Hand posture and force signals dataset [14]. Classification and regression in the SA–SA setting for the best and worst subjects. With best and worst, we mean the subjects for which the difference in performance between learning with adaption and learning from scratch is, respectively, maximum and minimum.

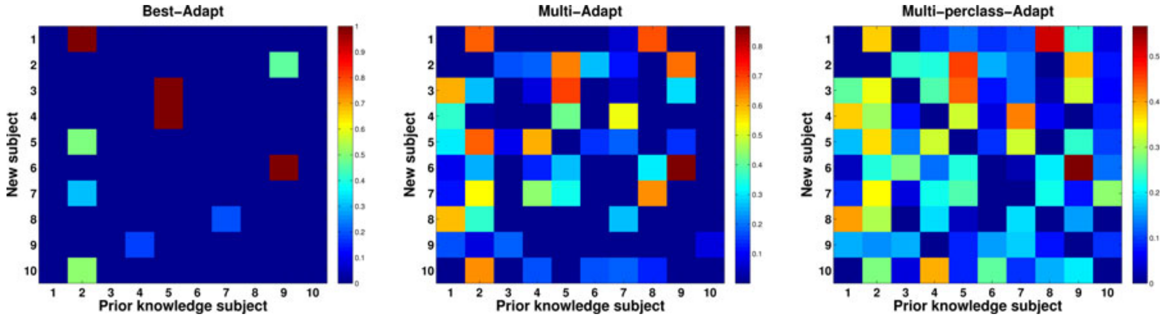

Fig. 8. Hand posture and force signals dataset [14]. Maps of the beta values for the three adaptive methods in classification SA–SA, obtained for 300 training samples. The title of each figure indicates the adaptive method that produced the corresponding beta weights, in particular for Multi-perclass-Adapt we are showing the average values over the four classes (three grasp postures plus rest). Rows 1 and 9 in all the matrices correspond, respectively, to the best and worst subjects in classification considered in Fig. 7, first and second plots from the left.

probability of finding useful information for the new problem. Moreover, here Multi-perclass-Adapt outperforms both Best-Adapt and No-Adapt ($p < 0.001$) with an average gain of 6% with respect to learning from scratch.

C. Discussion

As a general remark, we can state that the three proposed adaptive methods (i.e., Multi-perclass-adapt, Multi-Adapt, and Best-Adapt) improve the learning performance to different extents if prior knowledge contains useful information for the new task, and no harm if a good match between the data of the new subject and the old source subjects is found. To further support this statement, Fig. 7 shows the classification and regression results on SA–SA data, respectively, for the subjects that have the maximum (best) and the minimum (worst) difference in recognition and regression performance with adaptation compared to No-Adapt. The worst case subject represents the paradigmatic case of no previous models matching the current distribution; as a consequence, the parameter β ($\boldsymbol{\beta}$) is set automatically to a small value (to a vector of small norm). In this case, there is essentially no transfer of prior knowledge. More insight on this point is given by Fig. 8. Here, we are mapping the beta values for each adaptive model in a specific learning step (300 training samples) of the classification SA–SA experiment. Best-Adapt chooses only one prior model as reference, while Multi-Adapt can rely on more than one known subject. For Multi-perclass-Adapt, we show the average beta values over the four classes (three grasps plus rest). The results are consistent to each other: e.g., for subject 1 (first row in the matrices), all the adaptive methods choose subject 2 as very relevant, Multi-Adapt gives credit also to subject 8 and the same happens for Multi-perclass-Adapt that has more freedom in weighting each class and finds also subject 9 a bit useful. Subject 1 corresponds to the best subject, with the corresponding classification performance reported in Fig. 7 (first from the left). Subject 9 is instead the worst one (see Fig. 7 second from the left), and the ninth row of all the matrices of Fig. 8 actually indicates that all the beta values are small.

It is reasonable to claim that the overall performance of the adaptive methods would increase along with the number of stored models, since this would mean a larger probability of finding matching pretrained models. This is confirmed by the results on the Ninapro dataset. In the long run, a large database of sEMG signals and force measures, with subjects possibly categorized (per age, sex, body characteristics, etc.) would definitely help getting uniformly better performance.

We point out here that the direct use of prior knowledge on a new problem is only partially helpful without an appropriate way to 1) choose the best prior knowledge model, and 2) weigh and combine it with the new information. In fact, Prior Test shows that possibly tuning on the test, one prior knowledge model useful for the new problem could be found, but its usefulness declines with the number of available training samples. On the other hand, the Prior Average line corresponds to an attempt to directly use a flat combination of all the pretrained models on a new subject. The obtained performance shows that this is not a good solution.

Let us also briefly discuss the choice of the learning parameter C. Here, we followed the standard approach in the community

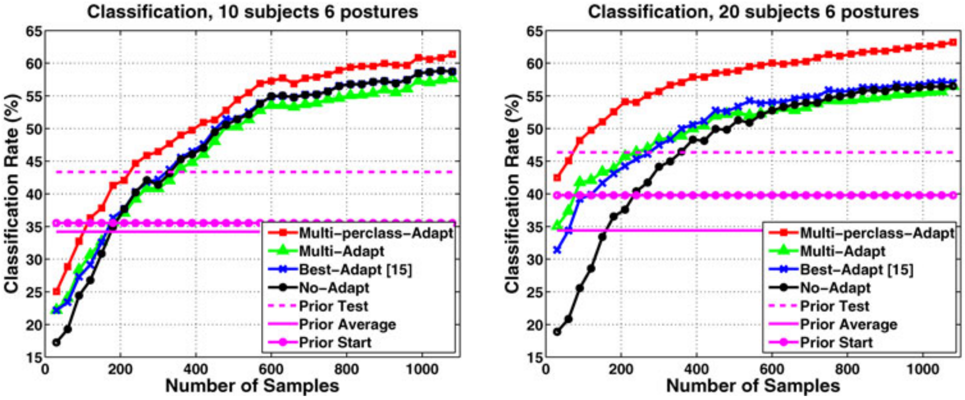

Fig. 9. Ninapro dataset [16]. Classification rate obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure indicates the number of subjects and hand postures considered.

and kept the parameter C fixed using the best value obtained from cross validation on the known subjects. Still, one might argue that the best way to define it is to optimize it by using the available training samples of the target subject, separately for each learning approach. For the proposed adaptive methods, this would imply defining C together with β, leading to a nonconvex problem and a great increase in computational complexity.

VI. Conclusion

The results presented in this paper clearly show that machine-learning-based classification and regression applied to surface EMG can be improved by means of reusing previous knowledge. In particular, we start from SVM models previously built by training on a pool of human subjects *to decrease the training time* of an LS-SVM to new subjects.

All the proposed adaptive methods show a significant gain in recognition rate for grasp-type classification and in correlation coefficient for regression when predicting the applied force, with respect to learning from scratch on the new subject. We note that the classification error/regression accuracy values obtained in our experiments are in many cases below the best results shown in the competing literature (an almost comprehensive table appears in [3, p. 725]), but the point here is to perform the comparison with nonadaptive baselines.

A comprehensive analysis of the practical applicability of our methods on real patients is out of scope here; hopefully, however, our results show that the presented method can be used in any (sEMG classification/regression) scenario.

The overall idea is that a prosthesis could be embedded with additional, preexisting knowledge before being shipped out to a new patient. This needs to be done once and for all and, most likely, for a large pool of healthy subjects and/or amputees of diverse condition, age and type of operation, and degree of muscle remnant fitness. The fact that the free-arm condition consistently benefits as well from the proposed technique—essentially to the same extent as the controlled one—is a very promising result, hinting that one could potentially pretrain a prosthesis *in a laboratory* and then ship it out and still give a significant benefit to the patient with respect to the learning-from-scratch case.

The databases we used consist of intact subjects only, but it is believed that transradial amputees can generate similarly accurate signals ([19] is the most recent result on this topic); therefore, this seems not to be a major objection to the applicability of the method. The project NinaPro (http://www.idiap.ch/project/ninapro/) is currently concerned with collecting such a large database of mixed subjects. If confirmed on data acquired from amputees, the current result could pave the way to a significantly higher acceptance of myoprostheses in the clinical setting. As future work, it would also be interesting to enlarge the presented approaches to more specific ongoing learning conditions on the new subject, covering the hypothesis of an increased number of hand postures.

Appendix
Closed Formula for the Leave-One-Out Prediction

We show here that, following the same steps presented in [40], it is possible to demonstrate Proposition 1, obtaining the closed formula for the leave-one-out prediction in (12). We start from

$$\boldsymbol{M} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \beta\hat{\boldsymbol{y}} \\ 0 \end{bmatrix} \tag{28}$$

and we decompose $\boldsymbol{M}$ into block representation, isolating the first row and column as follows:

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} = \begin{bmatrix} m_{11} & \mathbf{m}_1^T \\ \mathbf{m}_1 & \mathbf{M}_{(-1)} \end{bmatrix}. \tag{29}$$

Let $\boldsymbol{a}_{(-i)}$ and $b_{(-i)}$ represent the parameters of LS-SVM during the ith iteration of the leave-one-out cross-validation procedure. In the first iteration, where the first training sample is excluded, we have

$$\begin{bmatrix} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{bmatrix} = \mathbf{P}_{(-1)}(\boldsymbol{y}_{(-1)} - \beta\hat{\boldsymbol{y}}_{(-1)}) \tag{30}$$

where $\mathbf{P}_{(-1)} = \mathbf{M}_{(-1)}^{-1}$, $\boldsymbol{y}_{(-1)} = [y_2, \ldots, y_N, 0]^T$, and $\hat{\boldsymbol{y}}_{(-1)} = [\mathbf{w}' \cdot \phi(\boldsymbol{x}_2), \ldots, \mathbf{w}' \cdot \phi(\boldsymbol{x}_N), 0]^T$. The leave-one-out prediction for the first training sample is then given by

$$\tilde{y}_1 = \mathbf{m}_1^T \begin{bmatrix} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{bmatrix} + \beta\mathbf{w}' \cdot \phi(\boldsymbol{x}_1) \tag{31}$$

$$= \mathbf{m}_1^T \mathbf{P}_{(-1)}(\boldsymbol{y}_{(-1)} - \beta\hat{\boldsymbol{y}}_{(-1)}) + \beta\mathbf{w}' \cdot \phi(\boldsymbol{x}_1). \tag{32}$$

Considering the last N equations in the system in (28), it is clear that $[\mathbf{m}_1 \mathbf{M}_{(-1)}][\boldsymbol{a}^T, b]^T = (\boldsymbol{y}_{(-1)} - \beta \hat{\boldsymbol{y}}_{(-1)})$, and therefore

$$\begin{aligned}\tilde{y}_1 &= \mathbf{m}_1^T \mathbf{P}_{(-1)} [\mathbf{m}_1 \mathbf{M}_{(-1)}][a_1, \ldots, a_N, b]^T + \beta \mathbf{w}' \cdot \phi(\boldsymbol{x}_1) \\ &= \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1 a_1 + \mathbf{m}_1^T [a_2, \ldots, a_N, b]^T + \beta \mathbf{w}' \cdot \phi(\boldsymbol{x}_1).\end{aligned} \tag{33}$$

Noting from the first equation in the system in (28) that $y_1 - \beta \mathbf{w}' \cdot \phi(\boldsymbol{x}_1) = m_{11} a_1 + \mathbf{m}_1^T [a_2, \ldots, a_N, b]^T$, we have

$$\tilde{y}_1 = y_1 - a_1 (m_{11} - \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1). \tag{34}$$

Finally, using $\mathbf{P} = \mathbf{M}^{-1}$ and applying the block matrix inversion lemma, we get

$$\mathbf{P} = \begin{bmatrix} \mu^{-1} & -\mu^{-1} \mathbf{m}_1 \mathbf{P}_{(-1)} \\ \mathbf{P}_{(-1)} + \mu^{-1} \mathbf{P}_{(-1)} \mathbf{m}_1^T \mathbf{m}_1 \mathbf{P}_{(-1)} & -\mu^{-1} \mathbf{P}_{(-1)} \mathbf{m}_1^T \end{bmatrix}$$

where $\mu = m_{11} - \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1$, and noting that the system of linear equations (28) is insensitive to permutations of the ordering of the equations and of the unknowns, we have

$$\tilde{y}_i = y_i - \frac{a_i}{P_{ii}}. \tag{35}$$

Let $\boldsymbol{a} = \boldsymbol{a}' + \beta \boldsymbol{a}''$, $[\boldsymbol{a}'^T, b']^T = \boldsymbol{P}[\boldsymbol{y}^T, 0]^T$, and $[\boldsymbol{a}''^T, b'']^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^T, 0]^T$; from the above equation, we get

$$\tilde{y}_i = y_i - \frac{a_i'}{P_{ii}} + \beta \frac{a_i''}{P_{ii}}. \tag{36}$$

Acknowledgment

The authors would like to thank A. Gijsberts for his help in cleaning the code for the experiments and with the details on the hand posture hierarchy over the Ninapro dataset. The database presented in [14] and used in this study was collected and refined in 2009 at the Robotics, Brain and Cognitive Sciences Department, Italian Institute of Technology, Genova, Italy, mainly by A. E. Fiorilla. The authors would also like to thank A. E. Fiorilla and G. Sandini, also of the Italian Institute of Technology, for making the database available to them.

References

[1] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Crit. Rev. Biomed. Eng.*, vol. 30, no. 4–6, pp. 459–485, 2002.

[2] S. Micera, J. Carpaneto, and S. Raspopovic, "Control of hand prostheses using peripheral information," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 48–68, Oct. 2010. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5598519&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5598519

[3] B. Peerdeman, D. Boere, H. Witteveen, R. Huis in 't Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 719–738, 2011.

[4] D. J. Atkins, D. C. Y. Heard, and W. H. Donovan, "Epidemiologic overview of individuals with upper-limb loss and their reported research priorities," *J. Prosthet. Orthot.*, vol. 8, no. 1, pp. 2–11, 1996.

[5] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: A survey of the last 25 years," *Prosthet. Orthot. Int.*, vol. 31, no. 3, pp. 236–257, 2007.

[6] A. H. Bottomley, "Myoelectric control of powered prostheses," *J. Bone Joint Surg. Br.*, vol. 47-B, no. 3, pp. 411–415, 1965.

[7] D. S. Childress, "A myoelectric three-state controller using rate sensitivity," in *Proc. 8th Int. Conf. Med. Biol. Eng.*, Chicago, IL, 1969, pp. 4–5.

[8] L. Philipson, D. S. Childress, and J. Strysik, "Digital approaches to myoelectric state control of prostheses," *Bull. Prosthet. Res.*, vol. 18, no. 2, pp. 3–11, 1981.

[9] C. Castellini, E. Gruppioni, A. Davalli, and G. Sandini, "Fine detection of grasp force and posture by amputees via surface electromyography," *J. Physiol. (Paris)*, vol. 103, no. 3–5, pp. 255–262, 2009.

[10] F. R. Finley and R. W. Wirta, "Myocoder computer study of electromyographic patterns," *Arch. Phys. Med.*, vol. 48, pp. 20–24, 1967.

[11] R. W. Wirta, D. R. Taylor, and F. R. Finley, "Pattern-recognition arm prosthesis: A historical perspective—A final report," *Bull. Prosthet. Res.*, vol. Fall, pp. 8–35, 1978.

[12] C. Huang and B. Li, "A neural network-based surface electromyography motion pattern classifier for the control of prostheses," in *Proc. Int. Conf. Eng. Med. Biol. Soc.*, Oct./Nov. 1997, vol. 3, pp. 1275–1277.

[13] F. V. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 5, pp. 1427–1434, May 2009.

[14] C. Castellini, A. E. Fiorilla, and G. Sandini, "Multi-subject/daily-life activity EMG-based control of mechanical hands," *J. Neuroeng. Rehabil.*, vol. 6, no. 41, Nov. 2009. Available: http://www.jneuroengrehab.com/content/6/1/41

[15] F. Orabona, C. Castellini, B. Caputo, E. Fiorilla, and G. Sandini, "Model adaptation with least-squares SVM for hand prosthetics," in *Proc. Int. Conf. Robot. Autom.*, 2009, pp. 2897–2903.

[16] M. Atzori, A. Gijsberts, S. Heynen, A.-G. M. Hager, O. Deriaz, P. V. der Smagt, C. Castellini, B. Caputo, and H. Mller, "Building the ninapro database: A resource for the biorobotics community," in *Proc. IEEE Int. Conf. Biomed. Robot. Biomechatron.*, Jun. 2012, pp. 1258–1265.

[17] C. Castellini and P. van der Smagt, "Surface EMG in advanced hand prosthetics," *Biolog. Cybern.*, vol. 100, no. 1, pp. 35–47, 2009.

[18] F. C. P. Sebelius, B. N. Rosén, and G. N. Lundborg, "Refined myoelectric control in below-elbow amputees using artificial neural networks and a data glove," *J. Hand Surg.*, vol. 30A, no. 4, pp. 780–789, 2005.

[19] C. Cipriani, C. Antfolk, M. Controzzi, G. Lundborg, B. Rosen, M. Carrozza, and F. Sebelius, "Online myoelectric control of a dexterous hand prosthesis by transradial amputees," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 3, pp. 260–270, Jun. 2011.

[20] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *J. Rehabil. Res. Devel.*, vol. 48, no. 6, pp. 643–660, 2011.

[21] P. Parker, K. Englehart, and B. Hudgins, "Myoelectric signal processing for control of powered limb prostheses," *J. Electromyograph. Kinesiol.*, vol. 16, pp. 541–548, 2006.

[22] R. Merletti, M. Aventaggiato, A. Botter, A. Holobar, H. Marateb, and T. Vieira, "Advances in surface EMG: Recent progress in detection and processing techniques," *Crit. Rev. Biomed. Eng.*, vol. 38, no. 4, pp. 305–345, 2011.

[23] R. Merletti, A. Botter, C. Cescon, M. Minetto, and T. Vieira, "Advances in surface EMG: Recent progress in clinical research applications," *Crit. Rev. Biomed. Eng.*, vol. 38, no. 4, pp. 347–379, 2011.

[24] R. Merletti, A. Botter, A. Troiano, E. Merlo, and M. Minetto, "Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art," *Clin. Biomechan.*, vol. 24, pp. 122–134, 2009.

[25] T. Lorrain, N. Jiang, and D. Farina, "Influence of the training set on the accuracy of surface EMG classification in dynamic contractions for the control of multifunction prostheses," *J. Neuroeng. Rehabil.*, vol. 8, no. 25, May 2011. Available: http://www.jneuroengrehab.com/content/8/1/25

[26] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[27] H. Daumé III and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Intell. Res.*, vol. 26, pp. 101–126, 2006.

[28] S. B. David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1, pp. 151–175, May 2010.

[29] H. Daumé III, "Frustratingly easy domain adaptation," presented at the Conf. Assoc. Comput. Linguist., Prague, Czech Republic, 2007.

[30] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[31] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3081–3088.

[32] T. Tommasi and B. Caputo, "The more you know, the less you learn: From knowledge transfer to one-shot learning of object categories," in *Proc. Br. Mach. Vis. Conf.*, 2009, pp. 80-1–80-11.

[33] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 407–412.

[34] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. Conf. Assoc. Comput. Linguist.*, 2007, pp. 187–205.

[35] J. Yang, R. Yan, and A. G. Hauptmann, "Adapting SVM classifiers to data with shifted distributions," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops*, 2007, pp. 69–76.

[36] J. W. Sensinger, B. A. Lock, and T. A. Kuiken, "Adaptive pattern recognition of myoelectric signals: Exploration of conceptual framework and practical algorithms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 3, pp. 270–278, Jun. 2009.

[37] T. Matsubara, S. Hyon, and J. Morimoto, "Learning and adaptation of a stylistic myoelectric interface: EMG-based robotic control with individual user differences," in *Proc. IEEE Int. Conf. Robot. Biomimet.*, Phuket Island, Thailand, 2011, pp. 390–395.

[38] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 717–725.

[39] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[40] G. C. Cawley and N. L. C. Talbot, "Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters," *J. Mach. Learn. Res.*, vol. 8, pp. 841–861, May 2007.

[41] A. Luntz and V. Brailovsky, "On estimation of characters obtained in statistical procedure of recognition (in Russian)," *Technichesk. Kibernet.*, vol. 3, 1969. Available: http://www.citeulike.org/user/kchang10/article/9640940

[42] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Mar. 2002.

[43] M. Grant and S. Boyd. (2011, Apr.). CVX: MATLAB Software for Disciplined Convex Programming, Version 1.21. [Online]. Available: http://cvxr.com/cvx

[44] F. P. Kendall, E. K. McCreary, P. G. Provance, M. M. Rodgers, and W. Romani, *Muscles: Testing and Function, with Posture and Pain*. Philadelphia, PA: Williams & Wilkins, 2005.

[45] Futek LMD500 Medical Load Cell (Hand). [Online]. Available: http://www.futek.com/product.aspx?stock=FSH00125

[46] I. Kuzborskij, A. Gijsberts, and B. Caputo, "On the challenge of classifying 52 hand movements from surface electromyography," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012.

[47] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference (Statistics: A Series of Textbooks and Monographs)*. London, U.K.: Chapman & Hall, 2003.

Tatiana Tommasi received the M.Sc. degree in physics and the Dipl. degree in medical physics specialization from the University of Rome, La Sapienza, Rome, Italy, in 2004 and 2008, respectively. She is currently working toward the Ph.D. degree in electrical engineering with the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

She is also a Research Assistant with the Idiap Research Institute, Martigny, Switzerland. Her research interests include machine learning and computer vision with a particular focus on knowledge transfer and object categorization using multimodal information.

Francesco Orabona received the M.S. degree in electronic engineering from the University of Naples "Federico II," Naples, Italy, in 2003, and the Ph.D. degree in bioengineering and bioelectronics from the University of Genoa, Genoa, Italy, in 2007.

He is currently a Research Assistant Professor with the Toyota Technological Institute at Chicago, Chicago, IL. His research interests are in the area of theoretically motivated and efficient learning algorithms, with emphasis on online learning, kernel methods, and computer vision. He is a co-author of more than 40 peer-reviewed papers.

Claudio Castellini received the Laurea degree in biomedical engineering from the University of Genova, Genova, Italy, in 1998 and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 2005.

He was a Postdoctoral Researcher with the Advanced Robotics Laboratory, University of Genova for four and a half years, working on machine learning applied to human sensorimotor data. Since 2009, he has been a Researcher with the German Aerospace Center, Oberpfaffenhofen, Germany, in the Bionics group. He is a co-author of more than 35 peer-reviewed papers.

Barbara Caputo received the Ph.D. degree in computer science from the Royal Institute of Technology, Stockholm, Sweden, in 2004.

She has been a Senior Research Scientist with the Idiap Research Institute, Martigny, Switzerland, since 2006, where she leads the Cognitive Visual Systems group. Her main research interests include in computer vision, machine learning, and robotics, with which she has been active since 1999. She has edited four books and is a co-author of more than 70 peer-reviewed papers.