Sign Language Recognition with Support Vector Machines and Hidden Conditional Random Fields

Going from Fingerspelling to Natural Articulated Words

César Roberto de Souza and Ednaldo Brigante Pizzolato

Universidade Federal de São Carlos, São Carlos, Brasil
{cesar.souza, ednaldo}@dc.ufscar.br

Abstract. This paper describes the authors' experiments with Support Vector Machines and Hidden Conditional Random Fields on the classification of freely articulated sign words drawn from the Brazilian Sign Language (Libras). While our previous works focused specifically on fingerspelling recognition on tightly controlled environment conditions, in this work we perform the classification of natural signed words in an unconstrained background without the aid of gloves or wearable tracking devices. We show how our choice of feature vector, extracted from depth information and based on linguistic investigations, is rather effective for this task. Again we provide comparison results against Artificial Neural Networks and Hidden Markov Models, reporting statistically significant results favoring our choice of classifiers; and we validate our findings using the chance-corrected Cohen's Kappa statistic for contingency tables.

Keywords: Gesture Recognition, Sign Languages, Libras, Support Vector Machines, Hidden Conditional Random Fields, Neural Networks, Hidden Markov Models, Discriminative Models.

1 Introduction

Humans are social creatures; and therefore depend heavily on communication to perform daily tasks and achieve their goals. But in the event one of the main communication channels have been damaged or have been deemed unavailable, humans will inevitably find a way to restore this communication. In the case of deafness or speech impairment, the communication is often restored through Sign Languages.

However, one of the most immediate problems of the Sign Languages is that very few people outside the deaf community are actually able to speak them. Bridging this gap with an autonomous translator seems like a logical step towards a more inclusive society. Hence, this work aims to walk the first steps in this direction.

This paper enlists comparative results for the automatic recognition of a finite number of words drawn from the Brazilian Sign Language, heretofore Libras. Our system works in an unconstrained environment without the aid of gloves, markers or controlled lighting. At the heart of our system lies a two-layer architecture based on the automatic learning of discriminative models to work on different stages of the

recognition process, as suggested in [1]. However, unlike the aforementioned work, our layers are composed of Support Vector Machines (SVMs) and Hidden Conditional Random Fields (HCRFs), as in [2]. Besides, our system works directly with naturally articulated words rather than fingerspelled ones. These words may also involve more than just one hand and even the user's facial expressions. We intend to present our system to the reader, along with comparison results against other approaches and also discuss the overall linguistic foundations behind our recognition strategy.

This paper is organized as follows. After this introduction, Section 2 gives a list of related works, raising some points of interest and discussions. Section 3 gives an overview of Sign Languages, giving special consideration to Libras. Section 4 presents the methods, models and tools used in this work. In Section 5 we detail our approach to Sign Language recognition. Section 6 then lists our experiments, detailing the dataset and strategies we have used. Section 7 presents and discusses our findings, while Section 8 concludes this work.

2 Related Works

Sign Language recognition is closely related to gesture recognition. Following a comprehensive survey on this topic given in [3], one can say gesture recognition methods have been traditionally divided into two main categories: Device-based [4] and vision-based [5,6,7,8]. Device-based approaches often constrain the users to wear a tracking device, such as a tracking glove, resulting in less natural interaction. Vision based approaches, on the other hand, free the user from wearing potentially movement-limiting and otherwise expensive devices. In this paper, we will deal only with vision-based approaches.

Gestures can also be either static or dynamic. Static gestures, often called poses, are still configurations performed by the user, passive to be registered in a single still image. Dynamic gestures, in turn, vary on time, and have to be captured as a sequence of still images, such as image streams. Often, gestures have both elements, such as in the case of sign languages [3]. In this paper we will be covering both gesture types.

Several works have also been published aiming the recognition of specific Sign Languages. As examples we have systems targeting the American Sign Language (ASL) [9], the German Sign Language (*Deutsche Gebärdensprache*, DGS) [6], the British Sign Language (BSL) [10] and the Australian Sign Language (Auslan) [11]. We also have works focusing the same language as ours, such as the works done by Pizzolato *et al.* [1] and colleagues [2], who addressed only fingerspelling; and the works by Dias *et al.* [7], who focused specifically on the movement aspects of Libras.

The work by Dias *et al.* provided a convenient mathematical formulation of the movement recognition problem, presenting a solution using Self-Organizing Maps (SOM) networks and Vector Quantization. Interestingly enough, Carneiro *et al.* [12] also used the SOM model as a preprocessing step to classify signs from the Libras manual alphabet, both reporting high classification rates.

Moreover, other papers have already explored HCRFs [13] and other variants for gesture recognition. Morency *et al.* used Latent Dynamic-CRFs (LD-CRFs) [14] to perform gesture recognition in continuous image streams, with excellent results.

Elmezain *et al*. [8] also studied CRFs, HCRFs and LD-CRFs in the recognition of alphabet characters and numbers drawn in mid-air using hand trajectories, obtaining 91.52%, 95.28% and 98.05% for each model, respectively.

We consider our work to be more closely related to the work done by [10], in which the authors considered a linguistic model for signed words. Their structural approach attempted to decompose the sign into *visemes* in the same way a spoken word can be decomposed into *phonemes*. The next sections should make it clear how we took a similar approach by decomposing the Libras sign into its appropriate component units.

3 Sign Languages and the Brazilian Sign Language

Languages based on visual signs have arisen in the same manner as all spoken languages. Contrary to popular belief, those languages are not mimics. Most often they are also not a sign version of the spoken languages, such as English or Portuguese. They are fully qualified languages, with their own grammar, lexicons and semantic rules to be obeyed.

Furthermore, Sign Languages are not universal – and thus signers from one country or community should not be expected to be able to talk with signers from other communities unless they know and agree to sign in the same language. Hence Brazil's official Sign Language is the Libras, recognized as such since early 2002.

The recognition of the Libras as the official language for the country was perceived as a victory for the deaf communities in Brazil. This eventually led to its regulation in 2005, making Libras classes mandatory in teacher formation curses – such as for obtaining a Pedagogy degree in Higher Education. The deaf also acquired the right to the interpreter in court and in education, further increasing the importance of the interpreter in the Brazilian society. The availability of tools for the automatic recognition of Sign Language could thus be of major interest to help those professionals.

The Libras also have been studied under linguistics grounds. One of the first and still most comprehensive descriptions of the structural organization of the Libras was given by Brito [15]. Her effort at characterizing the elements of the Libras sign has been one of the major guides in designing this system. Brito had identified 46 fundamental Hand Configurations (HC) in the Libras. She also identified other parameters such as the Articulation Points (AP), the Movement types (M), the Hand Palm Orientation (HO), the Contact Region (CR) and the Non-Manual Components (NM) used in the language. If one could identify all possible elements in the aforementioned parameter sets, then a sign could possibly be denoted as a tuple

$$S = \langle HC, AP, M, HO, CR, NM \rangle.$$

One must also give special consideration to the NM set. Non-manual information often plays a crucial role in determining the true meaning of a signal in Libras. Facial information, for instance, can be used to resolve ambiguities, further qualify the sign being performed and even completely negate or change the meaning of a sign. This information thus cannot be simply ignored by an automatic recognition system.

4 Methods and Tools

4.1 Artificial Neural Networks

As the name implies, at their creation Artificial Neural Networks (ANNs) had a strong biologic inspiration. However, despite their biological origins, ANNs can be regarded as simple functions $f: \mathbb{R}^n \rightarrow \mathbb{Y}$ mapping a given input $\boldsymbol{x} \in \mathbb{R}^n$ to a corresponding output $\boldsymbol{y} \in \mathbb{Y}$. The output vectors $\boldsymbol{y} = \langle y_1, \ldots, y_m \rangle$ are also restricted to a specific subset of $\mathbb{R}^n$. Each $y_i \in \boldsymbol{y}$ is restricted to a particular range according to the choice of activation function for the output neurons. In the case of sigmoid activation functions, this range is $[0; 1]$; in case of bipolar sigmoid functions, it is $[-1; +1]$.

Since ANNs can be seen as standard mathematical functions, the learning problem can also be cast as a standard optimization problem, in which one would like to minimize a divergence, in some sense, between the network outputs $\hat{\boldsymbol{y}}$ and the desired answers $\boldsymbol{y}$. One possible way to achieve it is through the minimization of the error gradient; and a promising method for this is given by the Resilient Back-propagation algorithm (Rprop) [16,17].

The Rprop algorithm is one of the fastest methods for gradient learning restricted solely to first-order information. Its basic operational principle is to eliminate the (possibly bad) influence of the gradient magnitude in the optimization step. Unlike other gradient based methods, such as Gradient Descent, in which the step size is always proportional to the gradient vector, Rprop takes into account only the direction of the gradient, making it a local adaptive learning algorithm. Because Rprop relies only in first-order information, it is not required to compute (and hence store) the Hessian matrix of second derivatives for the learning problem, making it especially suitable for high dimensional problems.

One of the biggest challenges in learning ANNs is the presence of multiple local minima and the relatively large number of hyper parameters which have to be carefully adjusted in order to ensure a good generalization. In despite of this, they have been finding much renewed interested from the machine learning and artificial intelligence community thanks to recent advances in learning algorithms for deep layer architectures, in a new approach which is now referred as the deep learning paradigm [18].

4.2 Support Vector Machines

In face of the problems often found with other learning models, such as the presence of multiple local minima and the curse of dimensionality, the SVM had been specifically conceived to avoid such issues. Although some may argue that the SVM does not have any edge over the curse of dimensionality [19], their practical importance cannot be diminished. These models have shown great performance in many real-world problems [5,20,19], including problems of high dimensionality [21] and of large-scale [22]. In the linear case, the SVM decision is given by a simple hyperplane decision function on the form

$$h(\boldsymbol{x}) = sgn(\boldsymbol{\theta} \cdot \boldsymbol{x} + b) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 0 \quad (1)$$

in which we decide for class ω_1 if a point $\boldsymbol{x}$ lies on one side of the hyperplane defined by the parameter vector $\boldsymbol{\theta}$ and threshold b; or class ω_2 if it lies on the other. The SVM decision function is usually given in its dual form, in terms of Lagrange multipliers $\boldsymbol{\alpha}$, selected support vectors $\boldsymbol{z}$ and output labels y as

$$h(\boldsymbol{x}) = sgn\left(\sum_{i \in SV} y_i \boldsymbol{\alpha_i} \boldsymbol{z_i} \cdot \boldsymbol{x} + b\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 0\,. \quad (2)$$

In order to generalize this model to non-linear decision surfaces, one can introduce a nonlinear transformation $\varphi(\cdot)$: $\mathbb{R}^n \rightarrow \mathcal{F}$ such that, when applied to the input vectors $\boldsymbol{x_i} \in \mathbb{R}^n$, creates a projection in a high-dimensionality feature space $\mathcal{F}$. Using the kernel trick, one can replace inner products in eq. (3) with a Mercer's kernel of the form $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{z}) \rangle$. Since φ does not have to be explicitly computed, the feature space $\mathcal{F}$ can have an arbitrarily high dimensionality.

The learning procedure for such model can be done using an approximate version of the Structural Risk Minimization principle [23]. In this work, all training has been performed using Platt's Sequential Minimal Optimization (SMO) algorithm [20,24].

Multiclass classification approaches.

An immediate problem arising from the SVM's original hyperplane formulation is that it is not very obvious how to make the model applicable to more than two classes. Several approaches have been proposed to overcome this limitation, two of them being known as the 1-vs-1 and 1-vs-all strategies for multiple class classification.

For a decision problem over c classes, 1-vs-all requires the creation of c classifiers, each trained to distinguish one class from the others. The decision then is taken in a winner-takes-all approach. However there is no clear indication this approach results in an optimum decision. In the 1-vs-1 strategy, the problem is divided into $c(c-1)/2$ sub-problems considering only two classes at a time. At the decision phase, each machine casts a vote for one of the classes and the label with highest number of votes wins. This leaves the problem of evaluating an increased number of machines every time a new instance is classified – which could easily become troublesome or prohibitive in time sensitive applications.

The Decision Directed Acyclic Graph (DDAG), first proposed in [20], provides the fast evaluation times of the 1-vs-all strategy while at the same time offering strong generalization bounds on the generalization error. The DDAG also keeps the original hyperplane decision formulation by sequentially cutting the decision space until a decision is found. For a decision problem over c classes, only $(c-1)$ machines need be evaluated [20]. The performance of this approach improves significantly when using linear machines since each SVM evaluation is reduced to a single vector multiplication.

4.3 Hidden Markov Models

Hidden Markov Models (HMMs) attempt to model the joint probability distribution of a sequence observations $\boldsymbol{x}$ and their relationship with time through a sequence of hidden states $\boldsymbol{y}$. A HMM is described by a tuple $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ in which $\boldsymbol{A}$ denotes a matrix of possible state transition probabilities, $\boldsymbol{B}$ is a vector of probability distributions governing the observations and $\boldsymbol{\pi}$ is a vector of initial states probabilities. In the literature, HMMs are often described alongside three associated canonical problems: evaluation, learning and decoding. Although we will not be discussing those in detail, a very comprehensive explanation is due to Rabiner [25].

Exploring the fact that an HMM is able to provide the likelihood for a given sequence $\boldsymbol{x}$, it is possible to create a classifier by creating a model λ_i for each sequence label $\omega_i \in \Omega$. Treating each model λ_i as a density model conditioned to an associated class label ω_i, one can apply the Bayes' rule to obtain the a posteriori probability and then decide for the class with *maximum a posteriori.*

4.4 Hidden Conditional Random Fields

The HCRF can be seen as a latent-variable extension of the Conditional Random Field (CRF), first proposed by [26]. The HCRF attempts to model $p(\omega|\boldsymbol{x})$, the probability of a class label given a sequence, without incorporating an specific model for $p(\boldsymbol{x})$. This is in direct contrast with their generative counterpart given by generative classifiers based on sets of hidden Markov models (HMMs), which model $p(\boldsymbol{x}|\omega)$ individually for each class label and attempt to convert those to the posterior probabilities $p(\omega|\boldsymbol{x})$ either using Maximum Likelihood or Maximum a Posteriori estimates with the aid of Bayes' rule.

A general and comprehensive definition of a CRF can be found in [27], in which the authors define a CRF based on the partitioning of a factor graph. As such, consider a factor graph G partitioned in a set of clique templates $\mathcal{C} = \{C_1, C_2, \dots, C_P\}$. Each clique template C_p should specify a set of sufficient statistics $\{f_{pk}(\boldsymbol{x_p}, \boldsymbol{y_p})\}$ and parameters $\boldsymbol{\theta_p} \in \Re^{K(p)}$, in which $\boldsymbol{x}$ is the sequence of observations and $\boldsymbol{y}$ the sequence of hidden states associated with the observations $\boldsymbol{x}$. Then, a general model for a CRF can then be written as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x_c}, \boldsymbol{y_c}; \theta_p) \tag{3}$$

in which $\Psi_c(\boldsymbol{x_c}, \boldsymbol{y_c}; \theta_p) = exp\{\sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\boldsymbol{x_c}, \boldsymbol{y_c})\}$ and $Z(\boldsymbol{x})$ is the partition function used to keep results as probabilities.

Unlike HMMs, CRFs assume the label sequence $\boldsymbol{y}$ to be known during training. One possible solution to this problem is to handle $\boldsymbol{y}$ as latent variables. By adding a variable ω to designate class labels, and setting $\boldsymbol{y}$ to be hidden, one arrives at the HCRF formulation given by

$$p(\omega|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \sum_{\boldsymbol{y}} \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x_c}, \boldsymbol{y_c}, \omega_c;\ \theta_p) \tag{4}$$

which can be computed by the same exact algorithms used to compute $Z(\boldsymbol{x})$ in the CRF case. Given a training dataset of N input observation sequences $\boldsymbol{x}^{(i)}$ and corresponding class labels $\omega^{(i)}$, the model can be estimated by taking the gradient of the log-likelihood function

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} log\, Z(\omega^{(i)}, \boldsymbol{x}^{(i)}) - \sum_{i=1}^{N} log\, Z(\boldsymbol{x}^{(i)}) \tag{5}$$

and using any off-the-shelf optimizer such as L-BFGS or Conjugate Gradient to estimate $\boldsymbol{\theta}$. In this work we will be using the Resilient Backpropagation (Rprop) algorithm, first proposed to be used in ANNs, but also applicable to arbitrary optimization problems. The work by Mahajan *et al.* has shown Rprop to be one of the best algorithms for estimating HCRF models [28].

5 A Two-Layered Approach For Sign Recognition

Our approach for recognizing sign words from Sign Languages is clearly inspired by the field of speech recognition, which has enjoyed an extensive and ever-increasing literature over the years. In the same way speech recognition methods often build language models over phoneme classifiers, here we have a first layer, aimed to detect static gestures such as hand shapes from Brito's set of hand configurations; and a second layer, aimed to classify sequences of static gestures plus temporal, trajectory and facial information into words from a finite lexicon.

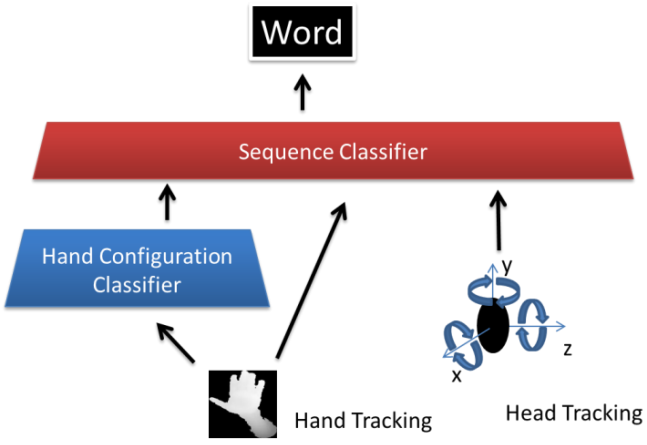

Fig. 1. The two layered architecture used in this work.

Considering Brito's work, along this paper we will consider the feature vector

$$\boldsymbol{x_t} = \langle h_c, h_{rx}, h_{ry}, h_\theta, f_\theta \rangle$$

$$\begin{array}{ll} h_c \in \mathbb{N}, & 1 \leq h_c \leq 46 \\ h_x, h_y \in \mathbb{R} & -1 \leq h_x, h_y \leq 1 \\ h_\theta, f_\theta \in \mathbb{R} & -\pi \leq h_\theta, f_\theta \leq \pi \end{array} \quad (6)$$

in which h_c denotes the hand configuration detected by the hand configuration classifier; h_{rx} and h_{ry} are the relative positions of the hand compared to the center of the head; and h_θ and f_θ are the angular orientation of the hand and face, respectively. All relative positions are normalized to the unit interval, and angular information is given in radians. For clarity, we will refer to the set of 46 possible configurations as $\mathbb{H}$. To maintain consistency with the following sections, the feature vector will be named $\boldsymbol{x_t}$, and the individual element at the i-th position will be indicated by $\boldsymbol{x_{t_i}}$.

5.1 Static gesture recognition layer

In order to estimate the location of the hands and the face of the user in a stream of continuous images, we use combined information gathered through depth and intensity sensors. To estimate the position of the face and hands, we combine the standard Haar object detection algorithm of Viola and Jones [29] and the Camshift [30] object tracker (with a few modifications for increased stability and robustness) with a Dynamic Virtual Wall Algorithm [31] for depth image segmentation.

Although we will not be discussing the segmentation algorithm in detail, our approach is centered on quickly detecting a tracking failure and quickly recovering from this error. This technique is able to work even when facing noisy and uncontrolled environments. As we shall see shortly, since the subsequent processing layers are able to cope with isolated frame errors, this approach works very well.

Continuing the processing flow, after the hands have been located, a bank of SVMs disposed in a Large-Margin DDAG is used to classify the hand image into one of the possible 46 hand configurations from $\mathbb{H}$. The initial experiments and results shown in [2] have been proven particularly useful to adequately learn the models used in this layer, particularly due the hyperparameter heuristics we had explored earlier.

5.2 Dynamic gesture recognition layer

Our second processing layer takes the output of the first layer, combined with trajectory, spatial and facial information and creates the feature vectors shown in (eq. 6). Considering each feature vector as an individual observation $\boldsymbol{x_t}$ belonging to a sequence of observations $\boldsymbol{x}$, the goal of this layer is to estimate the word label ω which is most likely associated with a given $\boldsymbol{x}$.

To create and learn the dynamic gesture models of this layer we considered feature functions of both discrete and continuous nature. We initialize our HCRF models with probabilities taken from corresponding HMMs, which have been crafted to use inde-

pendent, mixed joint-distributions of both discrete and continuous variables. This independent formulation could be seen as the application of the naïve Bayes assumption to our feature vectors. The emission distributions for the observation sequences could then be expressed in the form

$$p(\boldsymbol{x_t}) = \prod_{i=1}^{5} p_i\left(\boldsymbol{x_{t_i}}\right) \tag{7}$$

in which p_1 is a discrete distribution and $p_{2\ldots5}$ are assumed approximately normal with unknown mean and variance. We note there may be some concerns since a normal distribution is being assumed for variables which are of circular nature. However, the importance of this imprecision can be diminished when we consider that face and hands movements are limited by the joints of the body and, in case of the face, cannot possibly wrap. Choosing a Normal distribution also makes it easier to draw our linear-chain HCRF features as

$$\begin{aligned}
f_{\omega'}^{(Label)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\omega=\omega'\}} && \forall \omega' \in \Omega \\
f_{i,j}^{(Tr)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\boldsymbol{y_t}=i\}}\mathbf{1}_{\{\boldsymbol{y_{t-1}}=j\}} && \forall i, j \in \mathrm{S}_\omega \\
f_{i,o,d}^{(Em)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\boldsymbol{y_t}=i\}}\mathbf{1}_{\{(\boldsymbol{x_t})_d=o\}} && \forall i \in \mathrm{S}_\omega,\ \forall o \in \mathbb{H},\ \boldsymbol{x_{t_d}} \in \mathbb{N} \\
f_{\mathrm{i,d}}^{(Occ)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\boldsymbol{y_t}=i\}} && \forall i \in \mathrm{S}_\omega,\ \boldsymbol{x_{t_d}} \in \mathbb{R} \\
f_{\mathrm{i,d}}^{(M1)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\boldsymbol{y_t}=i\}}\left(\boldsymbol{x_{t_d}}\right) && \forall i \in \mathrm{S}_\omega,\ \boldsymbol{x_{t_d}} \in \mathbb{R} \\
f_{i,d}^{(M2)}(\omega, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \boldsymbol{x_t}) &= \mathbf{1}_{\{\boldsymbol{y_t}=i\}}\left(\boldsymbol{x_{t_d}}\right)^2 && \forall i \in \mathrm{S}_\omega,\ \boldsymbol{x_{t_d}} \in \mathbb{R}
\end{aligned} \tag{8}$$

in which Ω is the set of all possible class labels in our classification problem, S_ω is the number of states assumed for sequences of class ω. The label features $f_{\omega'}^{(Label)}$ trigger when a sequence belongs to class ω'. Transition features $f_{i,j}^{(Tr)}$ trigger whenever there is a transition from state i to state j. Emission features $f_{i,o,d}^{(Em)}$ trigger when a discrete symbol o occurs in the d-th position of the observation vector while inside state i. Occupancy features $f_{\mathrm{i,d}}^{(Occ)}$ trigger whenever state i is reached, while the first and second moment features $f_{\mathrm{i,d}}^{(M1)}$ and $f_{i,d}^{(M2)}$ perform the sum and sum of squares of the observation features at positions d when the state is i.

Using this set of feature functions, an HMM classifier created after each class label ω with prior probabilities α_ω , transition matrices $\boldsymbol{A}_\omega$ and emission densities $\boldsymbol{B}_\omega$ can be viewed as a HCRF with the corresponding components given as

$$
\begin{aligned}
\lambda_{\omega'}^{(Label)} &= \log \alpha_{\omega} && \forall \omega' \in \Omega \\
\lambda_{i,j}^{(Tr)} &= \log A_{i,j} && \forall i,j \in S_{\omega} \\
\lambda_{i,o,d}^{(Em)} &= \log B_{\boldsymbol{i}}(o) && \begin{matrix} \forall i \in S_{\omega} \\ \forall o \in \mathbb{H} \\ \boldsymbol{x_{t_d}} \in \mathbb{N} \end{matrix} \\
\lambda_{i,d}^{(Occ)} &= -0.5\left(\log 2\pi\sigma^2 + \frac{{\mu_{i,d}}^2}{{\sigma_{i,d}}^2}\right) && \begin{matrix} \forall i \in S_{\omega} \\ \boldsymbol{x_{t_d}} \in \mathbb{R} \end{matrix} \\
\lambda_{i,d}^{(M1)} &= \frac{\mu_{i,d}}{{\sigma_{i,d}}^2} && \begin{matrix} \forall i \in S_{\omega} \\ \boldsymbol{x_{t_d}} \in \mathbb{R} \end{matrix} \\
\lambda_{i,d}^{(M2)} &= -\frac{1}{2{\sigma_{i,d}}^2} && \begin{matrix} \forall i \in S_{\omega} \\ \boldsymbol{x_{t_d}} \in \mathbb{R} \end{matrix}
\end{aligned} \tag{9}
$$

in which $\mu_{i,d}$ and ${\sigma_{i,d}}^2$ refer to the mean and variance for the emission density at state i for the observation vector element at position d in case this element has an assumed normal distribution. In case this element is discrete, $B_{\boldsymbol{i}}(o)$ denotes the probability mass function for the state i for the hand configuration symbol $o \in \mathbb{H}$.

6 Experiments

6.1 Datasets

In order to create and learn our classification models, we acquired and organized a gesture dataset containing both static and dynamic gestures. We collected samples from 21 distinct subjects using Microsoft's Kinect sensor, registering both color and depth information. We note, however, that any of the results arising from our experiments are not restricted to this particular choice sensor, as all image processing has been done at the depth and intensity representation level. Those also have been gathered at varying luminosity levels, with both natural and artificial light sources.

The data acquisition occurred in two phases. At the first phase, the subjects had been asked to perform each of the fundamental 46 hand configurations of the Libras, purposely varying the location and orientation of the hand while keeping the configuration fixed. We sampled a total of 300 frames for each class to serve as training data to our static classifiers, giving a total of 13,800 training instances. Another independent and mutually exclusive sample of the same size has been drawn to be used as a validation set in the intermediate static gesture classification step.

In the second phase, we asked subjects to perform 13 natural words from the Libras. Those words have been repeated multiple times for a single subject, accommodating small variations between different performances. This gave us a total of 939 sequences of frames and a total sum of 139,154 frames in the dynamic gesture data-

base. Furthermore, those sequences have been further divided into 10 mutually exclusive sets in a preparation for applying 10-fold cross-validation.

The words explored in this work are shown in Table 1. Those have been chosen due their particular difficulties: *Cabinet* involves the occlusion of the face, *Sorry* involves the tilting of the head; *I*, *Shoes*, *Like and Buy* require touching other body parts. *Sorry* and *Age* are performed with the same hand configuration but differs in spatial location and in a head tilting movement. *Cabinet* and *Car* are performed giving equal importance to both hands.

Table 1. Signed words contained in our dataset.

Armário (Cabinet)	Idade (Age)
Sapato (Shoes)	Carro (Car)
Desculpa (Sorry)	Comprar (To Buy)
Eu (I)	Gostar (To Like)
Dia (Day)	Nome (Name)
Tchau (Bye)	Querer (To Want)
Oi (Hi)	

6.2 Static gesture classifiers

For the first processing layer we have created a number of SVMs with varying kernel functions and multi-class decision strategies. We also created feed-forward activation neural networks with a varying number of hidden neurons for comparison purposes. All classification machines have been designed to learn directly on a rescaled depth image of the user's hand. Those images have been gathered in the segmentation step as described in Section 5, but rescaled to a uniform size of 32×32 and then converted into a feature vector of 1024 positions. Despite the extremely simple nature of those features, the raw performance of this layer in correctly recognizing each of the hand configurations in $\mathbb{H}$ will not be as important as will the regularity of the classifier in classifying similar gestures with similar labels. Here, the hand configuration labels of the linguistic model are being used only as a guide – as long as the gesture recognition layer can detect patterns in the class labels generated at this stage, an absolute accuracy will not be required.

6.3 Dynamic gesture classifiers

After creating our static classifiers, we tagged the entire dataset of signed words and formed the feature vectors containing both hand configuration labels and trajectory information. In each cross-validation run an HMM has been created for each of the word labels in our dynamic gesture dataset. Those HMMs have then been used to initialize our HCRFs to compose the second processing layer. All results for Cohen's Kappa (κ) were averaged using ten-fold cross-validation, with variance pooled from all validation runs. We also compared the performance of the system without using the static classifier information at all, relying solely on trajectory and orientation information to perform classification.

7 Results

The static classifiers have shown some interesting results. Table 2 enlists selected results from tested SVM configurations and the average number of support vector (SV) evaluations as a measure of sparseness and efficiency.

Table 2. Results for the hand configuration candidate machines (first processing layer).

Kernel function	**Multiclass Strategy**	**Kappa ±** (0.95 C.I.)	**Average Number of Vector Evaluations**
Linear	DDAG	0.2390 ± 0.0074	45
	Max-Wins	0.2404 ± 0.0074	1,035
Quadratic	DDAG	0.4737 ± 0.0085	8,069
	Max-Wins	0.4790 ± 0.0085	339,509
Gaussian	DDAG	0.3401 ± 0.0042	8,707
	Max-Wins	0.3417 ± 0.0042	375,372

The graph below also shows the performance of the ANNs as a function of the number of neurons in their hidden layer. The best result reported by a neural network occured at 1000 neurons, with $\hat{\kappa}_{ANN} = 0.301$ and $\widehat{var}(\hat{\kappa}_{ANN}) = 4.05 \times 10^{-3}$. We note that after this peak the networks seemed to start overfitting the training data.

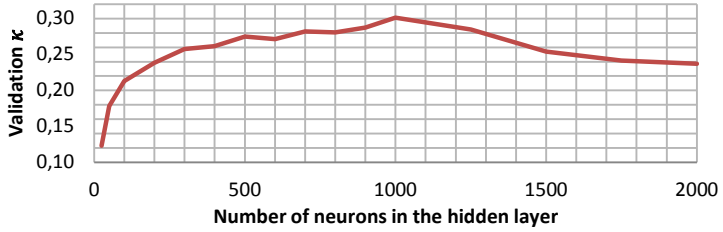

Fig. 2. Results for the hand configuration candidate networks (first processing layer)

Considering accuracy alone a quadratic SVM would be the clear choice to serve in the first classification layer. It performed statistically significantly better than all other models considered in this experiment. However, the performance cost in running such a machine can be huge. In contrast, the cost of computing a linear-SVM DDAG is much reduced. The DDAG based on linear SVMs has a constant evaluation rate, since its evaluation does not depend on the number of support vectors, but rather on the number of classes in the problem. Since a DDAG reduces the evaluation effort from computing 1035 decisions to only 45 constant-time decisions, we attain a much efficient, but weaker classifier, to serve as the first step in our dynamic gesture recognition system. We shall see shortly that the reduced performance will not be a problem due the probabilistic nature of the models on the second layer.

Table 3 below shows the results for the dynamic gesture experiments. The best combination of models was given by a SVM and a HCRF as the first and second processing layers in the system, respectively. However, we note that results using ANNs

were not significantly distinguishable from SVMs. The real difference occurred between the use of HMMs and HCRFs.

Table 3. Recognition results for the sequence classification models (second processing layer).

Static Gesture	Dynamic Gesture	Training	Validation
		Kappa ± (0.95 C.I.)	**Kappa** ± (0.95 C.I.)
SVM	HMM	0.8886 ± 0.0186	0.7951 ± 0.0717
SVM	HCRF	0.9466 ± 0.0131	0.8019 ± 0.0708
ANN	HMM	0.8610 ± 0.0205	0.7473 ± 0.0771
ANN	HCRF	0.9330 ± 0.0148	0.7727 ± 0.0743
None	HMM	0.6411 ± 0.0287	0.5308 ± 0.0898
None	HCRF	0.6638 ± 0.0283	0.5524 ± 0.0897

While this difference was not statistically visible in validation sets, the same could not be said of the training sets. It can be seen the models did not *overfit* – we obtained a statistically significant ($p < 0.01$) performance gain over training instances but no loss of generality over unobserved instances. Discriminative models were able to retain more knowledge without losing generalization when compared to HMMs.

On the other hand, when compared with models which did not use the hand configuration information at all, we achieved statistically significant results both for training *and* validation sets. As hinted before, the relatively small values for κ reported in Table 2 were no hurdle for the overall system performance. In fact, as in boosting mechanisms, where the combination of weak classifiers is able to produce one strong classifier, here the second processing layer is able to detect the patterns being output by the first layer, consolidating them into notable useful information when classifying new gestures. Thus the first processing layer effectively acts as a supervised feature extraction stage guided by linguistic information. Interestingly enough, the increased knowledge absorption by the discriminative models was mostly noticeable only in the presence of this first classification layer.

8 Conclusion

Here we have presented our approach to sign word recognition in Libras. By combining linear SVMs organized in Decision Directed Acyclic Graphs with Hidden Conditional Random Fields, we have shown how the use of discriminative models over generative ones helped improve the system's performance without causing a likely overfit. We have shown how the use of linguistic information has been helpful at designing such a gesture recognition system; and how our choice of simple features, based on a mixed vector with both discrete and continuous components have been suitable for this task. One can regard the first processing layer of our system as a guided feature extraction step rather than a definite classification stage. The use of a fast hand posture recognition layer based on DDAGs had been shown extremely useful when combined with trajectory and temporal information, achieving statistically significant results in comparison to models which did not use the presented technique.

Acknowledgements. We would like to express our thanks to FAPESP and CNPq. FAPESP sponsored the elaboration of this paper, and CNPq, the entire research. We would like to also express our gratitude towards our anonymous referees for sharing their valuable comments and opinions with us.

References

1. Pizzolato, E., Anjo, M., Pedroso, G.: Automatic recognition of finger spelling for LIBRAS based on a two-layer architecture. In : Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, pp.969-973 (2010)
2. Souza, C., Anjo, M., Pizzolato, E.: Fingerspelling Recognition with Support Vector Machines and Hidden Conditional Random Fields. In : Proceedings of the 13th Ibero-American Conference on Artificial Intelligence (IBERAMIA 2012), Cartagena de Indias, Colombia (2012)
3. Mitra, S., Acharya, T.: Gesture recognition: A survey. IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 37(3), 311-324 (2007)
4. Chen, X., Xiang, Li, Y., Lantz, V., Wang, K., Yang, J.: A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41(6), 1064-1076 (November 2011)
5. Yang, H.-D., Sclaroff, S., Lee, S.-W.: Sign Language Spotting with a Threshold Model Based on Conditional Random Fields. IEEE Trans. Pattern Anal. Mach. Intell. 31(7), 1264-1277 (July 2009)
6. Bauer, B., Kraiss, K.-F.: Video-based sign recognition using self-organizing subunits. In : Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2, pp.434-437 (2002)
7. Dias, D., Madeo, R., Rocha, T., Bíscaro, H., Peres, S.: Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In : Proceedings of the 2009 international joint conference on Neural Networks, Atlanta, Georgia, USA, pp.2355-2362 (2009)
8. Elmezain, M., Al-Hamadi, A., Michaelis, B.: Discriminative Models-Based Hand Gesture Recognition. In : International Conference on Machine Vision, Los Alamitos, CA, USA, pp.123-127 (2009)
9. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P.: American Sign Language Recognition with the Kinect. In : Proceedings of the 13th international conference on multimodal interfaces, Alicante, Spain, pp.279-286 (2011)
10. Bowden, R., Windridge, D., Kadir, T., Zisserman, A., Brady, M.: A Linguistic Feature Vector for the Visual Interpretation of Sign Language. In Pajdla, T., Matas, J., eds. : European Conference on Computer Vision, pp.391-401 (2004)
11. Holden, E.-J., Lee, G., Owens, R.: Australian sign language recognition. Machine Vision and Applications 16(5), 312-320 (2005)
12. Carneiro, A., Cortez, P., Costa, R.: Reconhecimento de Gestos da LIBRAS com Classificadores Neurais a partir dos Momentos Invariantes de Hu. In : Interaction 09 - South America, São Paulo, pp.190-195 (2009)
13. Wang, S., Quattoni, A., Morency, L.-P., Demirdjian, D.: Hidden Conditional Random Fields for Gesture Recognition. In : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, Washington, DC, USA, pp.1521-1527 (2006)

14. Morency, L.-P., Quattoni, A., Darrell, T.: Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In : IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07, pp.1-8 (2007)
15. Ferreira-Brito, L.: Por uma gramática de Línguas de Sinais 2nd edn. Tempo Brasileiro, Rio de Janeiro (2010)
16. Igel, C., Hüsken, M.: Improving the Rprop Learning Algorithm. In : Symposium A Quarterly Journal In Modern Foreign Literatures, pp.115-121 (2000)
17. Riedmiller, M.: RProp - Description and Implementation Details. Technical Report, University of Karlsruhe, Karlsruhe (1994)
18. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. In : IEEE Transactions on Audio, Speech, and Language Processing (2012)
19. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition 2200935th edn. Springer (2009)
20. Platt, J., Cristianini, N., Shawe-taylor, J.: Large Margin DAGs for Multiclass Classification. In : Advances in Neural Information Processing Systems, pp.547-553 (2000)
21. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features Machine Learning. In Nédellec, C., Rouveirol, C., eds. : Machine Learning: ECML-98 1398. Springer Berlin / Heidelberg, Berlin/Heidelberg (1998) 137-142 Lecture Notes in Computer Science.
22. Joachims, T.: Making large-scale support vector machine learning practical. In : Advances in kernel methods. MIT Press, Cambridge, USA (1999) 169-184
23. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods 1st edn. Cambridge University Press, Cambridge, UK (2000)
24. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Comput. 13(3), 637-649 (March 2001)
25. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. In Waibel, A., Lee, K.-F., eds. : Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990) 267-296
26. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In : Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, pp.282-289 (2001)
27. Sutton, C., McCallum, A.: Introduction to Statistical Relational Learning. In Taskar, L., ed. : An Introduction to Conditional Random Fields for Relational Learning. MIT Press (2007)
28. Mahajan, M., Gunawardana, A., Acero, A.: Training algorithms for hidden conditional random fields. In : Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.273-276 (2006)
29. Viola, P., Jones, M.: Robust Real-time Object Detection. In : International Journal of Computer Vision (2001)
30. Bradski, G.: Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal(Q2) (1998)
31. Anjo, M., Pizzolato, E., Feuerstack, S.: A Real-Time System to Recognize Static Hand Gestures of Brazilian Sign Language (Libras) alphabet using Kinect. In : Proceedings of IHC 2012, the 6th Latin American Conference on Human-Computer Interaction, Cuiabá, Mato Grosso, Brazil (2012)