Incrementally Learning Objects by Touch:
Online Discriminative and Generative Models
for Tactile-based Recognition

Harold Soh (haroldsoh@smart.mit.edu) and Yiannis Demiris (y.demiris@imperial.ac.uk)

Abstract

Human beings not only possess the remarkable ability to distinguish objects through tactile feedback but are further able to improve upon recognition competence through experience. In this work, we explore tactile-based object recognition with learners capable of incremental learning. Using the sparse online infinite Echo-State Gaussian process (OIESGP), we propose and compare two novel discriminative and generative tactile learners that produce probability distributions over objects during object grasping/palpation. To enable iterative improvement, our online methods incorporate training samples as they become available. We also describe incremental unsupervised learning mechanisms, based on novelty scores and extreme value theory, when teacher labels are not available. We present experimental results for both supervised and unsupervised learning tasks using the iCub humanoid, with tactile sensors on its five-fingered anthropomorphic hand, and 10 different object classes. Our classifiers perform comparably to state-of-the-art methods (C4.5 and SVM classifiers) and findings indicate that tactile signals are highly relevant for making accurate object classifications. We also show that accurate “early” classifications are possible using only 20-30% of the grasp sequence. For unsupervised learning, our methods generate high quality clusterings relative to the widely-used sequential k-means and self-organising map (SOM), and we present analyses into the differences between the approaches.

I. INTRODUCTION

Our sense of touch is considered our most pervasive sense and is exquisitely sensitive; a large number of nerve receptors distributed across our body allows us to finely examine and explore our environment. With touch sensing, we can detect, identify and manipulate objects not within our field-of-view. For example, we can easily feel to locate and grasp a pen on a desk behind us while simultaneously attending to another visual-based task, such as reading a paper. In addition, objects that have similar appearance may be more easily distinguished using tactile feedback (e.g., full and half-full bottles, ripe and unripe fruits).

The development of “artificial skin” — tactile sensors created from flexible semiconducting materials — may provide robots with similar, or even heightened, touch-sensing capabilities. A particularly interesting and important area of research is the recognition of objects based on tactile feedback. Identifying handled objects can enable better manipulation, for example, moving fingers to obtain a better grasp position. In recent years, studies have explored tactile-based object classification using offline machine learning methods, such as the C4.5 decision tree [1], [2]
Fig. 1. Tactile Learning using OIESGP Online Learning Experts. Both our discriminative and generative learners work directly on temporal data and are capable of creating new spatio-temporal experts “on-the-fly” as new objects are taught and refining models of familiar objects. A probability distribution over object classes/clusters is maintained and updated throughout the grasping action.

and a bag-of-words model [3]. There has, however, been far less work into incrementally learning tactile-based object recognition, i.e., learning over time as more data becomes available. Such an approach is essential for the development of autonomous learning, e.g., for robots operating in the real-world where both novel and familiar items would be encountered.

To fill this gap, we contribute and compare novel online generative and discriminative probabilistic tactile learners (Fig. 1), with experiments on the iCub humanoid platform [4]. This paper extends a shorter version [5] with a discriminative model, unsupervised learning, feature relevance detection and an in-depth analysis of the obtained results. Similar to recent work (e.g., [2], [3]), we aim to identify objects using tactile feedback during a grasping or palpation motion. But the primary difference is that our systems continuously improve internal models of familiar things and create models for novel objects on-the-fly. Moreover, compared to existing tactile classifiers [1], [2], [6], our models provide probability distributions over object classes and do not require the construction of an extensive dictionary (such as a bag-of-features [3][7]).

In addition to learning with taught samples, we provide mechanisms for unsupervised tactile recognition, which has thus far received little attention. Unsupervised tactile learning is relevant when new objects are encountered and a teacher is not present, e.g., during autonomous exploration. Without teacher labels, our methods discover structure in the data by clustering objects iteratively. We present novelty scores — based on likelihoods and extreme value theory — for creating clusters for newly discovered objects.

The core learning unit within our approach is a sparse Bayesian sub-model: the Online Infinite Echo-State Gaussian Process (OIESGP). Described in Section III, this online GP uses a recursive kernel to account for temporal correlations, and incorporates stochastic natural gradient descent (SNGD) for efficient, online hyperparameter learning. When combined with automatic relevance determination, the OIESGP learns the relevance of spatial and temporal features.

We conducted experiments using the iCub humanoid platform, which is equipped with two anthropomorphic
dexterous hands and tactile sensors on its fingertips (Fig. 2). This platform, along with the grasp controller, dataset objects (composed of 9 everyday items) and testing methodology are described in Section V.

Our experimental results, described in Section VI, reveal that our methods perform comparably to state-of-the-art optimised methods (incremental SVM and offline C4.5 and SVM classifiers). In addition, we discuss results from an online-learning experiment involving human subjects and plastic bottles of varying fullness, which suggest potential research avenues in active learning and sensor fusion. Section VII details our unsupervised learning experimental results; we compare our proposed discriminative and generative methods against each other and against popular competing algorithms, i.e., sequential k-means [8] and self-organising maps (SOM) [9].

Finally, ideas and other considerations for future work are discussed in Section VIII. In the next section, we first give an overview of related work in object classification using data obtained from tactile sensors.

II. RELATED WORK

Our focus in this work is on the identification of objects being grasped using tactile data. The development of the tactile sensors themselves constitutes a major research field and we refer interested readers to recent reviews [10], [11].

A related strand of research is in grasp stability estimation and analysis using tactile feedback. Recent work [12], [13] has evaluated the effectiveness of machine learning methods (AdaBoost, Support Vector Machines, hidden Markov Models) for stability prediction. In particular, the studies showed that sequential or temporal data was beneficial for generalisation, dynamic grasp execution and when working with deformable objects. Within the context of this paper, the closest related work was by Schill et al. [14]; a support vector machine (SVM) was used for online detection of grasp stability during execution. Unlike the methods described in this paper however, the SVM has to be trained offline. Although not explored here, our proposed methods can be applied towards grasp stability estimation. Similar to the algorithms previously explored, our approach makes use of temporal tactile data but with the added benefit of incremental training and prediction.

In the remainder of this section, we describe classification techniques applied along with the features used to determine the identity of handled objects. To organise our discussion, we group the reviewed works into discriminative or generative methods. In general, discriminative models tend to have lower asymptotic errors, whilst generative models may more quickly reach their lower bound [15].

A. Discriminative Tactile Classifiers

Given observation (or features) \( o \in \mathbb{R}^d \), we can directly model the conditional distribution \( p(c_i | o) \) over object class \( c_i \in C \), which yields a discriminative classifier. For example, Schöpfier, Ritter and Heidemann [6] used a local linear map (LLM) neural network trained on features\(^1\) obtained using a low-cost \( 16 \times 16 \) tactile sensor array mounted on a Unimation 6-DOF PUMA 200 robot arm. More recent work by Schöpfier [16] used a C4.5 decision

\(^1\)Features were dimensionally-reduced using principal components analysis.
tree trained on 51 tactile features (e.g., maximum taxel value), which achieved 92.5% accuracy score on 16 different objects.

The C4.5 tree was also used by Chitta et al. [1], [2] to estimate the internal state of bottles with varying fullness grasped using the PR2 gripper equipped with a capacitive sensor with 22 cells. The classifier used the frequency response of the manipulated objects, achieving an accuracy score of 93.9% in recognising the different liquid containers.

Opting for a nearest neighbour approach, Drimus et al. [17] performed classification of 10 objects (both rigid and deformable objects) using the k-Nearest Neighbour algorithm. Each object was represented by a variable length vector composed of the first two moments of each “tactile frame”. Distances between the two vectors were computed using Dynamic Time Warping and the closest \( k \) neighbours voted on the object’s class. The best accuracy obtained using this method on the 10 different objects (with 10 repeated grasps per object) was 92%.

B. Generative Tactile Classifiers

In contrast to the discriminative approach, the generative approach models the joint distribution \( p(o, c_i) = p(o|c_i)p(c_i) \) and through application of Bayes rule, obtains the class distribution \( p(c_i|o) \propto p(o|c_i)p(c_i) \). The primary task in generative modelling is in creating the likelihood distribution \( p(o|c_i) \) since \( p(c_i) \) can usually be estimated easily from the obtained data or assumed uniform.

One class of generative models make use of shape representations — which can be generated via a number of methods (e.g., [18], [19], [20]) — to obtain an appropriate likelihood function, \( p(o|c_i) \). For example, Pezzementi, Reyda and Hager [21] constructed a 2-D tactile mosaic (based on occupancy grid mapping) that was then used to estimate the object class via particle and histogram filters. The latter estimation method achieved perfect object recognition accuracy on a constructed test set (raised letters) using a DigiTacts sensor. Work by Meier et al. [22] demonstrated that compact 3-D representations of unknown objects could be obtained by tactile sensors. In particular, Kalman filters were used to build a probabilistic point cloud (stored as a kd-tree).

As an alternative to shape models described above, it is possible to construct recognition models that do not explicitly encode the object’s form. For example, Schneider et al. [3] extracted a bag-of-features from low-resolution tactile intensity images. A vocabulary of codewords was obtained using k-means clustering on the obtained images; distances were measured using a linear weighted combination of the pixel-by-pixel Euclidean distance (with vertical shift correction) and the width of the fingers. Finally, a histogram of the codewords for each object was generated. This method achieved a recognition rate of 84.6% with 10 test grasps on each of the 16 objects tested.

Schneider’s work was recently extended by Pezzementi et al. [7] to incorporate descriptors from the image processing community and to use Gaussian Mixture Models (GMM) for codebook creation. This improved method was capable of handling varying object poses and achieved higher accuracies exceeding 95%. Another bag-of-words model was proposed by Gorges et al. [23] which used an anthropomorphic robot hand. Classifications were performed using a combination of tactile and finger configurations (so called haptic key features), which were dimensionally reduced using PCA. Instead of using k-means or GMM, the codebook was obtained using a SOM.
The average accuracy of this approach was 75% (individual object accuracies varied greatly, ranging from 50% to 100%).

C. Moving Forward: Incremental and Unsupervised Learning

Whether discriminative or generative, machine-learning classifiers generally achieved high accuracies, reflecting the informativeness of tactile signals. However, one common attribute of the aforementioned methods is that they are inherently offline; models are "frozen" after training and the grasp or palpation has to be completed before a classification can be made.

In contrast, humans are both able to learn incrementally from experience and determine the type or properties of handled objects during the manipulation process. Moreover, we are generally able to decide if an object is "novel" and can group objects by touch, even without object identities or labels. In the following sections, we describe tactile recognition models that are similarly capable, starting with our underlying core method.

III. Online Infinite Echo-State Gaussian Process

The Online Infinite Echo-State Gaussian Process (OIESGP) is an online learner for multi-variate time series. In brief, it combines a sparse online Gaussian Process (SOGP) [24] with a recursive automatic relevance determination (ReARD) kernel [5] and stochastic natural gradient descent (SNGD) [25] for iterative hyperparameter adaptation.

The composition of these elements enables the method to operate in an online manner (processing a sample/mini-batches at a time) on sequential data, which facilitates use in on-demand real-world applications such as robotics. By storing only a limited set of basis vectors, the OIESGP has a fixed maximum computational and storage budget that can be constrained based on available resources. This method was recently demonstrated in [5] to achieve lower errors than state-of-the-art online-learning methods, i.e., locally-weighted-process regression (LWPR) [26] and online echo-state networks. In the following subsections, we summarise the three prominent aspects of the OIESGP: the sparse online GP approximations, the ReARD kernel and finally, the SNGD optimiser.

A. Sparse Online Gaussian Process

The Sparse Online Gaussian Process (SOGP) proposed by Csató and Opper [24] is an application of Bayesian Online Learning to GPs. The SOGP is grouped under the larger umbrella of deterministic training conditional (DTC) approximations, where the basis vectors are more generally referred to as inducing inputs [27].

Denote $\mathbf{x}_{t+1}$ as the input and $y_{t+1}$ as the observed output signal at time $t + 1$. We incrementally update the GP at time $t$ given $(\mathbf{x}_{t+1}, y_{t+1})$ by performing a Bayesian update to yield a posterior:

$$\hat{p}(\mathbf{f}|y_{t+1}) = \frac{P(y_{t+1}|\mathbf{f}(\mathbf{x}_{t+1}))p_t(\mathbf{f})}{\langle P(y_{t+1}|\mathbf{f}(\mathbf{x}_{t+1}))p_t(\mathbf{f}) \rangle_t}.$$  (1)

where we have neglected conditioning on the inputs for notational simplicity. In the case of GP regression, this update is exact. In the general case where this update is not tractable, we project the process onto the closest GP $q$.
by minimising the Kullback-Leibler divergence, $KL(\hat{p}_t||q)$, which is equivalent to matching the first two moments of $\hat{p}$ and $q$:

$$m_t(x) = \alpha_t^T k(x)$$  \hspace{1cm} (2)

$$k_t(x, x') = k(x, x') + k(x)^T C_t k(x')$$  \hspace{1cm} (3)

where $k(\cdot, \cdot)$ is the kernel or covariance function and $k(x) = [k(x, x_i)]_{i=1}^{t-1}$. The vector $\alpha$ and matrix $C$ are updated using:

$$\alpha_{t+1} = \alpha_t + w_1 (C_t k_{t+1} + e_{t+1})$$  \hspace{1cm} (4)

$$C_{t+1} = C_t + w_2 (C_t k_{t+1} + e_{t+1}) (C_t k_{t+1} + e_{t+1})^T$$  \hspace{1cm} (5)

where $k_{t+1} = [k(x_1, x_{t+1}), \ldots, k(x_i, x_{t+1})]$, $e_{t+1}$ is the $t + 1^{th}$ unit vector and the scalar coefficients $w_1$ and $w_2$ are given by:

$$w_1 = \partial f'_t ln(P(y_{t+1} | f(x_{t+1})))_t$$  \hspace{1cm} (6)

$$w_2 = \partial^2 f'_t ln(P(y_{t+1} | f(x_{t+1})))_t$$  \hspace{1cm} (7)

To maintain sparsity, the number of the data points retained — the basis vectors (BV), $b \in B$ — are limited using a scoring function that computes the “novelty” of $x_{t+1}$:

$$\gamma(x_{t+1}) = k(x_{t+1}, x_{t+1}) - k_{B,t+1}^T K_{B,t}^{-1} k_{B,t+1}$$  \hspace{1cm} (8)

where $k_{B,t+1} = [k(b_i, x_{t+1})]_{b_i \in B}$ and $K_{B,t}^{-1} = [k(b_i, b_j)]_{b_i, b_j \in B}$. If $\gamma(x_{t+1})$ is below some constant threshold, $\epsilon_\gamma$, then approximate update using (4) and (5) is performed with the only change being that we use:

$$\hat{e}_{t+1} = K_{B,t}^{-1} k_{t+1}$$  \hspace{1cm} (9)

instead of the unit vector $e_{t+1}$. This update does not increase the size $B$ but does absorb states which are not included. The inverse of $K_B$ can be computed iteratively using the Sherman-Morrison formula, i.e., $K_{B,t+1}^{-1} = K_{B,t}^{-1} + \gamma_{t+1}^{-1} (\hat{e}_{t+1} - e_{t+1}) (\hat{e}_{t+1} - e_{t+1})^T$. To delete a basis vector, we score each $b \in B$:

$$\epsilon_i = \frac{|\alpha_{t+1}(i)|}{K_{B,t+1}^{-1}(i, i) + C_{t+1}(i, i)}$$  \hspace{1cm} (10)

which is truncated loss between the approximated and updated GPs, and remove the lowest scoring BV.

Making predictions at an unknown test point $x_*$ is straightforward with the mean of the predictive distribution given by:

$$\mu_* = k_{B,t}(x_*)^T \alpha_t$$  \hspace{1cm} (11)

and variance:

$$\sigma_*^2 = k_t(x_*, x_*) + k_{B,t}(x_*)^T C_t k_{B,t}(x_*)$$  \hspace{1cm} (12)
The computational costs associated with model updates and predictions are on the order of $O(s_B^2)$ where $s_B$ is the maximum pre-defined size of $B$. In effect, $s_B$ determines the trade-off between accuracy and computational cost; larger basis vector sets allow the SOGP to approach the accuracy of the full GP with higher cost per iteration.

B. Recursive Kernel with Automatic Relevance Determination

To account for the temporal nature of many real-world signals, such as those produced by tactile sensors, we make use of recursive kernels [28] — a recently-proposed class of kernels that share an intimate relationship with recurrent neural networks (RNNs) and were principally derived as a means to extend RNNs to infinite size. Note that recursive kernels are denoted with the symbol $\kappa$ to differentiate them from standard kernels and although theoretically applicable to time-series of infinite length, we restrict the recursion depth using a parameter $\tau$ in practice.

The OIESGP employs a recursive kernel based on automatic relevance detection (ARD) [29], [30]:

$$
\kappa_t^{\text{ARD}}(x, x') = \exp \left( -\frac{1}{2} (x_t - x'_t)^T M (x_t - x'_t) \right) \exp \left( \frac{\kappa_{t-1}^\text{ARD}(x, x') - 1}{\sigma_\rho^2} \right)
$$

where $M = \text{diag}(l)^{-2}$, a diagonal $d \times d$ matrix where $l = [l_i]_{i=1}^d$. The kernel construction and proof of positive semi-definiteness are given in [30]. Unlike the standard squared exponential, this recursive ARD (ReARD) kernel is anisotropic; varying the $l_i$’s for different inputs allows us to control the impact that the inputs have on the predictions; responsiveness to input dimension $k$ is inversely related to $l_k$.

The parameter $\sigma_\rho$ weights the previous recursion kernel value. In other words, it functions as a temporal lengthscale; if $\sigma_\rho$ is large, the past recursive kernel value will decrease in relevancy. This temporal lengthscale is related to the spectral radius $\rho$ in echo-state networks (ESNs) [31] and affects kernel stability [28]. If the inverse lengthscale $\sigma_\rho^{-1} < 1$, the kernel is stable and iterative applications of the function will converge to a fixed point. The closer $\sigma_\rho^{-1}$ is to 1, the slower this rate of decay will be. The spectral radius of the ESN performs the same role: the larger $\rho$ is, the slower the rate of “memory fade”.

Finally, to simplify hyper-parameter optimisation, we unified our parameters into a single function and modified (13) above to include the noise and signal variance:

$$
\kappa(x, x') = \sigma_s^2 \kappa_t^{\text{ARD}}(x, x') + \sigma_n^2 \delta_{x, x'}
$$

where $\sigma_s^2$ is the signal variance, $\sigma_n^2$ is the noise variance, and $\delta_{x, x'}$ is the Kronecker delta which is one iff $x = x'$ and zero otherwise.

C. Stochastic Natural Gradient Descent

Applying the ReARD kernel requires us to specify its hyperparameters $\theta = [l, \sigma_\rho, \sigma_s, \sigma_n]$. In this work, we optimise $\theta$ with regard to the leave-one-out likelihood by following the natural gradient of the loss function:

$$
\mathcal{L}(\theta) = -\int \log p(y_t|x_t, \theta)p(x_t, y_t)dx_tdy_t
$$

where, again for notational convenience, we have dropped the dependence on the training data seen thus far. In our case,

$$\log p(y_t | x_t, \theta) = \log \mathcal{N}(y_t - \mu_t, \sigma_t^2)$$

$$= -\frac{1}{2} \log \sigma_t^2 - \frac{(y_t - \mu_t)^2}{2\sigma_t^2} - \frac{1}{2} \log 2\pi$$

(16)

where $\mu_t$ and $\sigma_t^2$ are given by (11) and (12) respectively. The direction of steepest descent is given by the natural gradient [25]:

$$G^{-1}_\theta \nabla_\theta \mathcal{L}$$

(18)

where $G^{-1}_\theta$ is the inverse Riemannian metric tensor (the covariance of the gradients [32]) and $\nabla_\theta \mathcal{L}$ is the standard gradient. For the space of probability distributions represented by the hyperparameters, $G_\theta$ is the Fisher Information matrix:

$$\mathcal{I}_\theta = \left[ E \left[ \frac{\partial \log p(y_t | x_t, \theta)}{\partial \theta_i} \frac{\partial \log p(y_t | x_t, \theta)}{\partial \theta_j} \right] \right]_{i,j=1}$$

(19)

In this work, we approximate the natural gradient over $s_g$ iterations using an averaging approach with the Sherman-Morrison formula:

$$\tilde{\mathcal{I}}^{-1}_\theta = \frac{\epsilon_t}{1 - \epsilon_t} \tilde{\mathcal{I}}^{-1}_\theta - \frac{\epsilon_t}{1 - \epsilon_t} \left( \tilde{\mathcal{I}}^{-1}_\theta \nabla_\theta \mathcal{L} \nabla_\theta \mathcal{L}^\top \tilde{\mathcal{I}}^{-1}_\theta \right)$$

(20)

where $\epsilon_t = 1/s_g$. A similar approach was used in [33] for multi-layer perceptrons but here, we average over the sampling window and update the hyperparameters progressively:

$$\theta_{j+1} = \theta_j + \eta \tilde{\mathcal{I}}^{-1}_\theta \nabla_\theta \mathcal{L}$$

(21)

where $\eta$ is the step size or learning rate. We use the straightforward approach of setting $\eta = 1/j$. Once the hyperparameters are updated, the GP posterior is out-of-date, forcing a re-computation of $K, C$ and $Q$ which we perform directly using the basis vector set $B$.

To compute the gradients, we use the derivatives for $\kappa_t^{\text{ARD}}$, which are also recursive in nature:

$$\frac{\partial \kappa_t^{\text{ARD}}}{\partial l_i} = \kappa_t^{\text{ARD}} \left[ \frac{1}{\sigma^2} \frac{\partial \kappa_t^{\text{ARD}}}{\partial l_i} + \frac{\beta_i}{l_i^3} \right]$$

(22)

$$\frac{\partial \kappa_t^{\text{ARD}}}{\partial \sigma} = \kappa_t^{\text{ARD}} \left[ \frac{-2(\kappa_t^{\text{ARD}} - 1)}{\sigma^3} \frac{\partial \kappa_t^{\text{ARD}}}{\partial l_i} + \frac{1}{\sigma^2} \frac{\partial \kappa_t^{\text{ARD}}}{\partial \sigma} \right]$$

(23)

where $\beta_i = (x_{t,i} - x'_{t,i})^2$ with base cases:

$$\frac{\partial \kappa_t^{\text{ARD}}}{\partial l_i} = \frac{\beta_i}{l_i^3} \kappa_t^{\text{ARD}}$$

and

$$\frac{\partial \kappa_t^{\text{ARD}}}{\partial \sigma} = 0$$

IV. GENERATIVE AND DISCRIMINATIVE TACTILE LEARNERS WITH OIESGP SUB-MODELS

In this study, we developed both generative and discriminative methods composed of OIESGP sub-models, $m_i \in \mathcal{M}$, representing each object class (or cluster) $c_i$. At each time-step $t$, the methods are provided with an observation $o_t$ and are expected to provide a probability distribution $p(c_t)$. When teacher labels are available, our
models can be trained in the standard supervised fashion. For unsupervised learning, our methods generate new clusters when needed using probability-based novelty scores to detect when an "distinct" sample is observed. To simplify exposition, we use the following acronyms: Discriminative Supervised (DS), Generative Supervised (GS), Discriminative Unsupervised (DU) and Generative Unsupervised (GU).

A. Discriminative Tactile Learning

Recall that in the discriminative approach, we model the conditional distribution \( p(c_i|o) \) directly. Here, we have used one-versus-all classifier scheme whereby one classifier is trained for each object.

Unlike regression, the target values \( h_{i,t} \in \{-1, +1\} \) are positive for the correct class and negative otherwise, rendering the Gaussian likelihood inappropriate. As such, we applied the probit likelihood [34]. Each observation in a given sequence is associated with the appropriate target, which yields the following likelihood function:

\[
q_t(c_i|o_t) = \Phi \left( \frac{h_{i,t}\mu_{i,t}}{\sigma_{i,t}} \right)
\]

where

\[
\Phi(z) = \int_{-\infty}^{z} N(z|0,1)dz
\]

is a cumulative density function of a standard normal distribution, \( \mu_{i,t} \) and \( \sigma_{i,t} \) are the predicted mean and standard deviations using model \( m_i \). A normalised probability class distribution at a given time \( t \) can be obtained via:

\[
p_t(c_i|o_t) = \frac{q_t(c_i|o_t)}{\sum_j^{M_i} q_t(c_j|o_t)}.
\]

Following the standard winner-takes-all approach, the recognised object is the one with the highest probability throughout the sequence, i.e.,

\[
\arg \max_i p_t(c_i|o_t) \text{ for all } t = 1, 2, \ldots, T.
\]

1) Supervised Training (DS): To train a DS model, the appropriate sub-model \( m_i \) is provided with a positive (correct) sequence and updated as described in the section III-A. However, negative samples present a problem when dealing with temporal sequences in a multi-class setting; the same observation may be generated from different classes and the number of classes imply a positive-negative imbalance. We experimented with four different training schemes for dealing with negative samples: providing negative samples to all models, none of the models (only positive training), and the closest (as measured by probability) or wrongly selected model. Positive and closest-model training yielded the best scores and as such, we focus on the results obtained via the latter scheme for the remainder of this paper.

2) Unsupervised Learning (DU): To construct an online unsupervised model, we need ascertain which cluster a given sample should be assigned to or if it is necessary to create a new cluster.

Fortunately, one-versus-all discriminative models with probabilistic outputs lend themselves easily to unsupervised learning. Samples are assigned to the "winning" cluster, i.e., the sub-model with highest probability \( s = \arg \max_i p_t(c_i|o) \). In cases where the un-normalised probability \( q_t(c_s|o) \) is too low — the observation is not
probable given the best cluster [35] — a new cluster is generated and updated with the observation. Naturally, "low" is application and data-dependent. Furthermore, given our Bayesian sub-models, maximum uncertainty is attained when $q_t(c_s|\mathbf{o}) = 0.5$, i.e., the observed input is "far" from previously observed positive and negative samples. Taking this into consideration and that each grasped object results in sequential observations, we define a novelty score:

$$\nu_D = 1 - T^{-1} \sum_{t=1}^{T} \max(0, 2q_t(c_s|\mathbf{o}) - 1)$$  (28)

In this work, we control cluster creation using a user-defined threshold parameter, $\gamma_D \in [0, 1]$, which is compared against $\nu_D$. The lower $\gamma_D$, the higher the propensity for cluster creation; at the extreme of zero, a cluster will be created for each object and at the extreme of one, only one cluster will be generated.

Summarising, we begin the clustering process with one cluster generated using the first sample. As learning proceeds, each sample is assigned to the most probable cluster unless a novel sample is observed ($\nu_D > \gamma_D$), in which case the sample is placed in a new cluster.

B. Generative Tactile Learning

Recall that in the generative framework, we model the joint probability distribution. Since our data is inherently temporal in nature, we model $p_t(o_t|c_i)$ and $p_t(c_i)$ where the latter represents the current belief over the classes. Inference is performed using a Bayes filter to update a probability distribution over the model classes as observations arrive:

$$p_t(c_i|\mathbf{o}_t, \mathcal{M}) = \frac{p_t(o_t|m_i)p_{t-1}(c_i)}{\sum_j p(o_t|m_i)p_{t-1}(c_j)}$$  (29)

where we use a standard (independent) observational model:

$$p(o_t|m_i) = \prod_k N(o_{k,t}|\mu_{i,k,t}, \sigma_{i,k,t})$$  (30)

When the palpation motion is initiated, the initial prior is set to be uniformly distributed across the object classes and the class distribution is continuously updated as sensory data is received. To compare accuracies with the other methods, the class with the highest probability at the end of the grasping action is selected as the recognised object.

1) Supervised Training (GS): With teacher labels, training the generative classifier GS simply corresponds to updating the correct model with example sequences. To reduce computational and storage costs, we use the same basis vector set for all sub-models for a given class and perform the SOGP updates given in Section III-A.

2) Unsupervised Learning (GU): As with the DU model, samples are assigned to the cluster with highest probability. However, deciding on when to create a new cluster is non-trivial because placing a threshold on the Gaussian likelihood is difficult; the likelihood values are often small, even for "close" or likely observations.

To perform unsupervised learning with GU, we use a novelty score based on recent results in Extreme Value Theory (EVT) for multivariate Gaussian distributions [36]. EVT is a statistical description of the extrema of generative distributions; given a set of i.i.d random variables $\mathcal{X} = \{x_i\}_{i=1}^m$ generated from the same distribution $p_1(x)$, the
extreme value distribution (EVD) is the cumulative distribution function $H^+(x_{\text{max}} \leq x)$ where $x_{\text{max}} = \max(X)$. The EVD of a multivariate Gaussian is available in closed form [36]:

$$G^e(y) = 1 - \exp \left[ - \left( \frac{y}{c_m} \right)^{\alpha_m} \right]$$

(31)

where $\alpha_m$ and $c_m$ are estimated Weibull parameters (See Section 6 of [36] for more details). Thus, given our predictions $\mu$ and uncertainties $\Sigma$, we can define the novelty score of a given observation $o$ as:

$$\nu_G(o, \mu) = 1 - G^e(\mathcal{N}(o|\mu, \Sigma))$$

(32)

$$= \exp \left[ - \left( \frac{1}{C_n c_m} e^{-\frac{(o-\mu)^T \Sigma^{-1} (o-\mu)}{2}} \right)^{\alpha_m} \right]$$

(33)

where $C_n = (2\pi)^{\frac{n}{2}}||\Sigma||^{\frac{1}{2}}$. Similar to the DU model, we define a threshold $\gamma_U \in [0, 1]$, which controls the cluster creation tendency.

V. Experiments on the iCub Platform

Using the iCub humanoid platform, we collected a tactile dataset and performed experiments for online supervised and unsupervised learning. In the following subsections, we give specifics on the test objects, grasping procedure, initialisation parameters and our testing methodologies.

A. Test Objects

In this experiment, we used nine different everyday objects and one baseline where the grasp was performed with no object, totalling ten classes (shown in Fig. 2). These objects comprise plastic bottles with differing amounts of liquid, two soft toys, soda-cans (also of varying fullness), a bottle of lotion and a hardcover book. For each class, we created a dataset of twenty samples using our grasp controller and recorded data for the pressing portion of the grasp, yielding a total of 200 samples. The objects were slightly perturbed (translation and rotation) between grasps. As can be seen in Fig. 3, each grasped object generates a distinctive spatio-temporal “signature”. Our classifiers are based on the notion that each of these signatures can be learned and represented by an “expert model” (Fig. 1). To encourage reproducibility and future development, we have made this dataset freely available online [37].

B. Grasping with the iCub Hand

Previous work on tactile classification has focussed mainly on using grippers (e.g., the two-fingered gripper on PR2 robot, [1]) and industrial-style robotic arms [6], [3]. For this study, we used the iCub humanoid robotic platform (Fig. 2) which has two anthropomorphic dexterous hands with 5 fingers (as shown in Fig. 4). Each fingertip is wrapped with 12 capacitive pressure sensors under a layer of soft silicone foam. When pressure is applied to the fingertips, the silicone foam is compressed, changing the capacitance. An embedded board samples all the sensors sequentially, generating an output values ranging from 0 to 255. In this work, the data capture rate was 10Hz.

We devised a grasp controller that executed a three-step action: first, it fixed the hand position and orientation and then fully opened the hand digits, yielding the pre-shape (as shown in Fig. 5a). After we placed an object in a
Fig. 2. iCub with Objects: Plastic bottles (full, half-full, empty), Soda cans (empty, half-full), Teddy-bear, Monkey soft-toy, Lotion bottle and Hardcover Book.

graspable position, the controller closed the digits with low velocity (20 deg/s for each degree of freedom) along a pre-defined trajectory, as illustrated in Fig. 5b, until the tactile sensor readings breached a critical threshold or the trajectory was completed. It was necessary to vary this threshold from 15-20 to compensate for sensor drift. Finally, the controller would press on the object by moving each digit further along the same trajectory but with a higher velocity (40 deg/s) until the motion was blocked or the trajectory motion was finished. Motion blocks were detected by checking that motor encoder values remained unchanged for 0.5 seconds.

C. Model Parameterisation and Set-up

Our models process the encoder (measuring each of the 9 DOFs) and tactile sensor data from each finger. Instead of using all twelve sensors directly, the sensor data was reduced by computing the first three moments (mean, standard deviation and skewness) as well as the maximum and minimum reading for the twelve sensors on each finger; note that this data reduction is “spatial” and not temporal as we do not compute statistics across the time-steps. This resulted in an observation/feature vector \( o_t \in \mathbb{R}^{34} \) (25 tactile features and 9 encoder values) for each time step \( t \). We also experimented with reduced data streams with only the tactile or encoder features.

Both methods were initialised with single model after the first object was encountered and a model class was created whenever a new object is taught to the classifier. Each model was set to use 50 basis vectors with initial lengthscales \( l = [10.0]^D \) for the generative model and \( l = [1.0]^D \) for the discriminative model. The remaining hyper-parameters initialised with \( \sigma_f = 1.0, \sigma_n = 0.1, \tau = 15 \) and \( \rho = 1.01 \) for both classifiers. Hyper-parameter adaptation was enabled with the sampling window \( s_g \) set at 30 and stopping criteria \( ||\Delta \theta||^2 < c_g = 10^{-4} \) over 10 iterations. The same parameters were used for both the supervised and unsupervised learning experiments.
Fig. 3. Sample tactile sensor spatio-temporal blocks for each of the 10 classes where each object generates an individual “signature”. In each plot, the left horizontal axis represents time and each vertical 5x5 slice represents the 25 tactile features (mean, standard deviation, skewness, maximum and minimum for each finger). Note that these features change across time (horizontally-stacked slices) as the fingers press on each object. Our classifiers are based on the notion that each of these signatures can be learned and represented by an OIESGP expert model.

D. Testing Methodology

Our testing methodology for the supervised and unsupervised learning scenarios differed, particularly with regards to the performance metrics. To compare the captured metrics, we conducted 50 repeated tests with shuffled samples and significance testing was performed using the Mann-Whitney U-Test.
Fig. 4. The iCub Hand has 19 joints and 9 DoFs; this induces a coupling between certain joints as indicated by the colour codes and labelings. In our experiments, we use the tactile (capacitive) sensors on each fingertip. Each fingertip incorporates a flexible printed circuit board underneath an electrically conductive silicone and provides 12 pressure measurements.

Fig. 5. The Grasping controller would first initialise the iCub hand into a “pre-shape” as shown in Fig. 5a. It would then close the fingers onto the object via the motion depicted in Fig. 5b, pressing onto the object to obtain tactile sensory input.
Online Discriminative and Generative Classification Accuracies of Grasped Objects

Fig. 6. Classification Accuracy as Trials Progressed. To summarise and visualise the results, we grouped each subsequent set of 19 objects (a “trial set”) and computed the accuracy scores over this set. The generative line plot is offset by 0.5 for improved visualisation. Both the discriminative and generative models improved classification performance as the trials progressed. The best results were obtained when both tactile and encoder features were used; by the fourth trial set, both methods achieved perfect accuracy. However, using only the tactile features also resulted in strong performance, particularly with the generative classifier. The models trained only on the encoder features attained comparatively poor performance, suggesting that the tactile data was more informative for object identification.

1) Supervised Learning: Supervised learning was conducted in two phases: in the first phase, a sample from each object class was presented to the classifier once. Strictly speaking, this phase was not necessary since new models can be created as the trials progressed but this simplified comparison with offline methods and the human subjects who were informed of the objects were in advance. In the second phase, an object was chosen randomly (stratified sampling) and presented for classification. After making a prediction, the algorithm was trained with the actual class label. Training and testing was conducted for the first 80% of the data, and only testing for the final 20%. We used accuracy — the proportion of correct classifications — as a comparison metric.

2) Unsupervised Learning: For unsupervised learning, the algorithms also underwent two phases. In the first “learning” phase, the algorithm was run against all the samples to obtain the cluster models. Then, to evaluate the cogency of the cluster models, we re-assigned all the samples to the generated models.

To compare the clusterings obtained by the different methods, we used three standard scores: purity, the adjusted Rand index (ARI) [38], [39] and the normalised mutual information (NMI) [40]. In brief, purity indicates cluster homogeneity and is high if a majority in each cluster are the same object. ARI is similar to purity but penalises clustering errors, i.e., false positives and false negatives. Both purity and ARI can only be used to compare clusterings with the same number of clusters. Indeed, both scores reach their maximum when each sample is allocated its own cluster. The information theoretic NMI incorporates the trade-off between cluster quantity and quality, and thus allows for comparison between different-sized clusters.

VI. EMPIRICAL RESULTS: CLASSIFICATION

Our classification results are summarised in Tbl. II and Fig. 6. Both online classifiers performed similarly and improved with more samples seen, as indicated by the higher accuracies and smaller standard deviations across
the repeated tests. Comparing the progression of the online methods, the discriminative model appeared to achieve marginally better scores early on, but the differences are not statistically significant. On the other hand, the generative model attained higher accuracies as more samples are provided — this phenomena is most clearly seen when using only the encoder values (differences statistically significant with $p = 7.48 \times 10^{-4}$).

The best accuracies were obtained when both the tactile and joint angle features were used — both discriminative and generative classifiers attained 100% accuracy on the testing portion (final 20%) of the dataset. This perfect result suggests that our dataset may be oversampled and excellent performance can be achieved when training is limited to only 50% of the dataset. More importantly, we noted that the tactile information plays a crucial role in making a proper classification; using only the encoder features (joint angles) results in significantly lower accuracies ($p = 5.53 \times 10^{-5}$ and $p = 9.43 \times 10^{-7}$ for the discriminative and generative models respectively).

Looking closer at the classification errors, both methods made similar mistakes (confusion matrices shown in Fig. 7). To summarise, the two most misclassified objects were the baseline class (no object) and the Half-full Soda Can. The baseline class was most frequently mistaken for the Monkey soft-toy and the Half-full Soda-can confused with the Half-Full Plastic Bottle. These mistakes are unsurprising given the soft nature of the soft-toy, and the size and compression similarities between the half-full containers. When using only the encoder values, the cans and
plastic bottles were most frequently confused, as well as the lotion bottle and hardcover bottle, as these objects had similar widths and resulted in similar hand configurations.

A. Comparison to Partial-Online and Offline Methods

We first compared the results obtained by our classifiers against those obtained by the incremental SVM (iSVM) [41] with a squared-exponential (SE) kernel. Since iSVM does not directly work on temporal data, we created a feature vector similar to [2] by computing statistics across the temporal dimension (shown in Tbl. I). Thus, unlike our methods, classifications by the iSVM were obtained only after completing the entire grasping motion. As such, we term this a *partial-online* algorithm. The iSVM parameters (penalty term $C$ and kernel width) were optimised via a grid search to minimise the error over 80% of the data.

Table II shows that the iSVM attained a $\approx 4\%$ lower accuracy score of 0.954 (results statistically significant with $p < 10^{-18}$) compared to DU and GU. In Fig. 6, we observe lower accuracies during the first six time steps (differences statistically significant, $p < 10^{-5}$), indicating a slower learning rate.

Next, we compared our results against offline methods, i.e., the C4.5 decision tree (used with success in several research works on tactile classification [1], [2]) and SVM using the RapidMiner platform\(^1\). Similar to the iSVM, we optimised each algorithm using a grid-search. Specifically, we optimised the penalty term $C$ and the SE kernel width for the SVM and C4.5 parameters (i.e. leaf size, split criteria, minimal gain). For the SVM, we also performed feature selection (FS) using a genetic algorithm\(^3\). The best 5-fold cross-validation accuracies of the two methods (in Tbl. II) are comparable to those achieved by our online methods. However, our methods do not require heavy offline optimisation and produce predictions during the grasping motion.

To better understand the robustness of our methods, we injected additive zero-mean Gaussian noise with variance $\sigma^2_n = 100$ and 1000 (corresponding to low and high noise conditions) to the raw sensor signals (recall that the output for each taxel ranged from 0 to 255). Figure 8 illustrates that accuracy decreased as the noise increased; accuracy for the generative model fell from 96% to 67% (-29%). The performance drop was even more severe for the discriminative model, i.e., from 90% to 47% (-43%). Nevertheless, both GS and DS produced accuracies statistically better than the incremental SVM ($p < 10^{-6}$ for GS and $p < 10^{-2}$ for DS).

B. Early Classification

One benefit of our approach is the ability to predict object labels before the grasping motion is complete. To illustrate this point, Fig. 9 shows the “early” classification accuracies on the test set (using all features) during the grasp. The scores are shown versus sequence proportion since the actual temporal lengths varied from one grasp to the next. Even with 10% of the data, both classifiers achieved a high accuracy of $\approx 98\%$. Indeed, both models achieve perfect accuracy when using only 30% of the data, demonstrating that excellent object recognition is achievable without having to complete the entire motion.

---

\(^1\) More information at www.rapid-i.com.

\(^3\) We also attempted to use principal component analysis (PCA) for feature reduction but this yielded suboptimal results.
Fig. 8. Classification performance under low-noise ($\sigma_n^2 = 100$) and high-noise ($\sigma_n^2 = 1000$) conditions. A fall in accuracy was observed (as expected), but performance remained superior to the incremental SVM.

Fig. 9. Median classification accuracies at different stages of the pressing motion. High accuracies are achievable even before the motion is complete — perfect identifications are made using only the first 20% and 30% of the sequence for the discriminative and generative classifiers respectively.
TABLE I

**Features used for partial-online and offline methods** (computed for each finger separately).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-25</td>
<td>First 3 moments, max and min of the mean tactile data</td>
</tr>
<tr>
<td>26-50</td>
<td>First 3 moments, max and min of the sd. tactile data</td>
</tr>
<tr>
<td>51-75</td>
<td>First 3 moments, max and min of the tactile data skewness</td>
</tr>
<tr>
<td>76-165</td>
<td>Encoder data during maximum and minimum tactile readings</td>
</tr>
</tbody>
</table>

TABLE II

**Object classification for the OIESGP-based classifiers, partial-online and offline methods.**

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Algorithm</th>
<th>Score-Type</th>
<th>All Features</th>
<th>Tactile Features</th>
<th>Encoder Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online</td>
<td>Discriminative (DS)</td>
<td>All</td>
<td>0.991 (0.026)</td>
<td>0.985 (0.025)</td>
<td>0.907 (0.066)</td>
</tr>
<tr>
<td>(OIESGP)</td>
<td>Discriminative (DS)</td>
<td>Final 20%</td>
<td>1.000 (0.000)</td>
<td>0.992 (0.012)</td>
<td>0.936 (0.029)</td>
</tr>
<tr>
<td>Generative (GS)</td>
<td>All</td>
<td>0.991 (0.027)</td>
<td>0.989 (0.030)</td>
<td>0.921 (0.096)</td>
<td></td>
</tr>
<tr>
<td>Generative (GS)</td>
<td>Final 20%</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>0.970 (0.024)</td>
<td></td>
</tr>
<tr>
<td>Partial-Online</td>
<td>iSVM</td>
<td>All</td>
<td>0.952 (0.009)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>iSVM</td>
<td>Final 20%</td>
<td>0.993 (0.018)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Offline</td>
<td>C4.5</td>
<td>5-fold</td>
<td>0.985 (0.012)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>5-fold</td>
<td>0.995 (0.001)</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

C. **Feature Relevance**

By analysing the optimised lengthscales of the discriminative classification model, we can gain insights into the relevance of features. We observed an average increase in the lengthscales to 1.8, suggesting the initial values were too low. The minimum lengthscales — indicative of importance — were different across the objects, indicating that feature relevancy was object-specific. The three most relevant features (with minimum lengthscales) were the tactile features corresponding to the middle finger and thumb, as well as the middle-finger proximal flexion/extension joint angle. Interestingly, the C4.5 decision tree method picked a similar feature set corresponding to the mean tactile reading for the thumb, middle and ring fingers and the standard deviation of the thumb tactile reading.

D. **Computational Costs**

Our computational test-bed was an Apple Macbook laptop with a 2.6Ghz Intel Core-i7 processor and 16GB of RAM. For the discriminative classifier, each prediction iteration (for each observation) required an average of 0.0093s. Training was more expensive at 0.02s given the need to compute derivatives for hyperparameter optimisation. Prediction and training times were marginally higher for the generative model at 0.0097s and 0.0265s respectively. Nevertheless, both models were capable of updating their internal states in less time than required to execute the grasping motion. In short, the training and prediction can be made in real-time given reasonable limits.
Fig. 10. Human online classification experiment with the three plastic bottles (of varying fullness). Participants were provided true-label feedback after each sample, giving them the opportunity to learn from mistakes. (10b) Human Classification Accuracy as trials progressed (with medians shown in the boxes) show an improvement in accuracy between trials 1-10 and 11-20.

E. Comparison to Human Subjects

From preliminary tests, we discovered that humans easily classified the (easily deformed) soda cans, the soft toys (through texture), the lotion bottle and the hard-cover book. However, they had difficulty discriminating between the plastic bottles. Similar findings were reported by Chitta et al. [2], where human subjects achieved a classification accuracy score of 75.2% when asked to discriminate between full and empty (both open and closed) plastic bottles — note however that in that study, the humans trained before-hand (until they were confident about their abilities) and were not offered the correct labels during the testing stage.

We invited 15 human subjects (ages 21-38 years, mean 28.5 years, 13 males) to participate in an experiment similar to the classification task described in Section V but limited to the three plastic bottles (Fig. 10a). In the first phase, each participant was allowed to grasp each of the items once and told the object’s class; they were not allowed to lift or otherwise move the bottles. In the second phase, they were asked to classify a randomly selected bottle (30 trials with stratified sampling) hidden in covered box. After each attempt, the participants were provided feedback (the true-label), giving them the opportunity to learn from mistakes.

The average human score was 80.4% (sd. 12.4%), lower than that achieved by iCub with OIESGP classifiers. From the confusion matrix (Tbl. III ), we observed that the empty and half-full bottles were more difficult to tell apart; our participants frequently misclassified the empty bottle as half-full, particularly during earlier trials. The one participant who achieved perfect accuracy revealed to us after the experiment that he used a combination of temperature and bottle deformation to help him with the task. Since our methods can easily accommodate other sensor-streams, future work may fuse sensor data to better classify items.

To determine if the subjects improved as the trials progressed, we segmented the data into three portions, i.e.,
TABLE III
CONFUSION MATRIX FOR FULL, HALF-FULL AND EMPTY PLASTIC BOTTLES (HUMAN STUDY)

<table>
<thead>
<tr>
<th>True Label</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Empty</td>
</tr>
<tr>
<td>Empty</td>
<td>111</td>
</tr>
<tr>
<td>Half-Full</td>
<td>17</td>
</tr>
<tr>
<td>Full</td>
<td>0</td>
</tr>
</tbody>
</table>

Cluster Evaluation Scores

![Graph showing Purity, ARI, and NMI scores against novelty threshold](image)

Fig. 11. Purity and ARI scores fell as the novelty threshold was increased. The NMI penalises the large number of effective clusters spawned by the discriminative model and hence, we observe a rise from approximately 0.5 to 0.9 (at $\gamma_D = 0.3$), followed by a fall back to the initial score.

the first ten, second ten and final ten trials. As can be seen in Fig. 10b, the median accuracy scores rose from 70% to 90% from the first to the second segments and remained relatively stable after. The difference in the first and second segments were statistically significant at the 95% level ($p = 0.033$), in favour of the hypothesis that human beings learnt continuously from feedback. We also noted that the human subjects repeated their grasps in order to make a decision (especially when they were unsure). As future work, we plan to investigate using the model’s uncertainty estimates in a similar “active-touch” strategy.

VII. EMPIRICAL RESULTS: CLUSTERING

Recall that in unsupervised learning, the methods are required to cluster the data without teacher labels. Fig. 11 summarises our results, showing how the clustering quality (measured by purity, ARI and NMI) changed as the novelty threshold was varied from 0.1 to 0.95.

We observe a general falling trend across the scores as the threshold was increased, with the notable exception of the discriminative model’s NMI. The reason for this exception is illustrated in Fig. 12. As discussed in Section
Fig. 12. The average number of clusters and effective clusters generated decreased as the novelty threshold was increased. Note that the relationship is non-linear. The discriminative model kept active the same number of clusters that it generated — 200 when the threshold was at 0.1 and only one at 0.95. The generative model generated many initial clusters (up to 50 at low novelty thresholds) but kept effective clusters a much smaller number (only up to 18).

Fig. 13. NMI scores for all four methods increased with the number of effective clusters, up to 15 clusters. After the respective maxima, NMI fell slowly (within the window of 20 clusters). Both the generative and discriminative model achieved higher NMI scores relative to sequential k-means and SOM ($p < 10^{-4}$ from 11 effective clusters onwards).
TABLE IV

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Purity</th>
<th>ARI</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrimative (DU)</td>
<td>0.835 (0.021)</td>
<td>0.807 (0.022)</td>
<td>0.914 (0.011)</td>
</tr>
<tr>
<td>Generative (GU)</td>
<td>0.824 (0.025)</td>
<td>0.774 (0.042)</td>
<td>0.903 (0.018)</td>
</tr>
<tr>
<td>Sequential k-means</td>
<td>0.735 (0.080)</td>
<td>0.700 (0.092)</td>
<td>0.857 (0.040)</td>
</tr>
<tr>
<td>Sequential SOM</td>
<td>0.777 (0.056)</td>
<td>0.718 (0.068)</td>
<td>0.858 (0.036)</td>
</tr>
</tbody>
</table>

IV, the novelty thresholds ($\gamma_D$ and $\gamma_G$) control the cluster creation process; increasing the threshold decreases the propensity for clusters to be generated. Fig. 12 clearly shows this effect on both the number of clusters generated during training and the number of active clusters actually used during testing (effective clusters). The discriminative model kept effective the same number of clusters that it generated — 200 when the threshold was at 0.1 and only one at the other extreme of 0.95. The generative model also generated many initial clusters (up to 50) at low novelty thresholds, but the number of effective clusters was low (up to 18). Unlike purity and ARI, the NMI penalises the large number of effective clusters spawned by the discriminative model and hence, we observe a rise from approximately 0.5 to 0.9 (at $\gamma_D = 0.3$), followed by a fall back to 0.5.

Note also that the shape of the performance scores track that of the number of effective clusters. For GU, there is a sharp decrease in model scores as the novelty threshold is increased from 0.3 to 0.4. The scores also start decreasing for DU in the same regime, albeit more gradually. Both accuracy and computational cost depends on the number of clusters generated, implying the importance of selecting an appropriate novelty threshold. Here, we have empirically determined a reasonable threshold of 0.3 – 0.35, which generates 10 to 11 effective clusters; automatically selecting this threshold makes for interesting future work.

A. Comparison to Online Clustering Algorithms

Fig. 13 compares the NMI scores between our models and sequential k-means and SOM. Like the iSVM, these methods also used the features shown in Tbl. I. At a low number of effective clusters, all four methods perform similarly but the differences between our methods and the competing methods grew as the number of effective clusters increased. In particular, our methods achieve better NMI scores from 11 effective clusters onwards ($p < 10^{-4}$). Table IV shows more precisely the three scores obtained by the discriminative and generative models, k-means and SOM for 10 effective clusters. Our methods attain higher purity, ARI and NMI scores, with the discriminative model achieving the best scores (statistically significant with $p < 10^{-10}$ compared to k-means and SOM).
VIII. Summary and Conclusions

This work presented online discriminative and generative classifiers — composed of flexible spatio-temporal OIESGP sub-models — for distinguishing objects by touch.

In contrast to existing methods, our classifiers provide online classification via a probability distribution over object classes at each time-step. When applied to the tactile and encoder signals provided by the iCub anthropomorphic hand, our methods achieved high accuracies on a variety of objects. The best scores were obtained when using a combination of tactile and encoder (joint angle) signals, indicating that both feature sets are relevant for accurate object identification. On the unsupervised learning task, our algorithms generated more cogent clusterings relative to the popular sequential k-means and SOM methods. Comparing the two approaches, the generative model achieved better classifications, but the discriminative model produced better clusters.

Touch-based recognition remains an interesting and challenging area for future work. In particular, we note several unresolved issues and avenues for future work. First, it would be interesting to determine what other spatial features (other than the simple moments used here) can be extracted in real-time from the sensor stream — in particular, adapting image-based features used in the vision community appears a promising route. From an algorithmic perspective, the OIESGP models used were optimised using stochastic natural gradient descent which, while effective, may fall prey to local minima. Future work may investigate “global” optimisation methods such as evolutionary algorithms (e.g., [42]).

Unlike the classification methods, the unsupervised algorithms (DU and GU) can be used for data exploration on data collected from tactile sensors. The clusters may also serve as a codebook (e.g., [43]) or as lower-level features in deep-learning classifiers [44] or in hierarchical methods [45]. Unsupervised learning is also particularly relevant for autonomous robots operating in unstructured environments. For example, our novelty-based methods enable robots to detect newly-discovered objects on which to direct their attention, allowing for real-time intelligent exploration and learning by-touch.

As future work, we noted a proper novelty threshold was required for good performance. Can the threshold be set automatically from the data? In addition, the generative model created many clusters that were not ultimately used. How best to prune these unused clusters over time and free up resources remains an open problem. It would also be interesting to consider a semi-supervised approach (merging elements from the methods presented) whereby both labelled and unlabelled data can be fully utilised.

Finally, our experiments with human subjects have provided two additional means of extending our work. The first is the fusion of additional sensory data such as temperature and vision to better classify objects (such as [46]). The second research thread is active classification, i.e., gathering additional sensory data when confidences are low. Moreover, the current system uses exploration with a single hand and grasp trajectory; it would be interesting to consider multiple actuators (e.g., grasping with two hands) and grasp controllers, using the uncertainties as guidance. In a similar vein, active learning can be used to minimise the amount of samples required for learning. We anticipate that research in these areas will bring us closer to autonomous systems that are better capable of tactile exploration.
and interaction in unconstrained environments.

ACKNOWLEDGMENT

The authors would like to thank members of the Personal Robotics Lab at Imperial College London for their help in the preparation of this manuscript. This work was supported in part by the EU FP7 project WYSIWYD under Grant 612139 and a SMART Scholar Grant.

REFERENCES


