Spatial-Temporal Modeling of Interactive Image Interpretation

Jun Zhou[1], Li Cheng[2] and Walter F. Bischof[3]

[1]National ICT Australia
Canberra, Australia
jun.zhou@nicta.com.au

[2]Toyota Technological Institute
Chicago, USA
chengli@ieee.org

[3]University of Alberta
Edmonton, Canada
wfb@ualberta.ca

Short Title: Interactive Image Interpretation

Corresponding author:
Walter F. Bischof
Computing Science
University of Alberta
Edmonton, AB T6G 2E9
Canada

Abstract

We consider the problem of spatial-temporal modeling of interactive image interpretation. The interactive process is composed of a sequential prediction step and a change detection step. Combining the two steps leads to a semi-automatic predictor that can be applied to a time-series, yields good predictions, and requests new human input when a change point is detected. The model can effectively capture changes of image features and gradually adapts to them. We propose an online framework that naturally addresses these problems in a unified manner. Our empirical study with a synthetic data set and a road tracking dataset demonstrate the efficiency of the proposed approach.

Keywords: image interpretation, sequential prediction, online learning, adaptive tracking.

1 Introduction

Computer-aided image interpretation is important in many areas, including, for example, medical image interpretation (Harders and Székely, 2003), content-based image retrieval (Vasconcelos, 2004), and object recognition in remote-sensed images (e.g., Rochery et al., 2006; Hu et al., 2007; Zhou et al., 2007). Most research has focused on fully automatic methods. There is, however, still a large gap between the requirements of practical applications and what is currently being achieved by automatic methods in terms of completeness, correctness and reliability. Most systems require checking by experts before any final decision can be made. For this reason, many successful systems retain the 'human in the loop' in the sense that a human operator supervises the image interpretation process and the computer acts as an intelligent assistant. As Caelli et al. (2001) pointed out

> "There is a strategic need for technologies to assist humans in the interpretation, depicting and querying of images in domains where improvements in data acquisition and archiving techniques have lead to collection and storage of a large amount of images (for example, in remote sensing, mapping, surveillance and medical image domains)." (p. 197)

Caelli et al. (2001)'s approach was exemplified with a real-world application where the shape of buildings in remote-sensed images needed to be tracked. The tracking task was performed by a trainable and dynamic system, in which human experts provided training examples of shapes, for example, the boundaries of buildings. Then the system used hidden Markov models to encode the boundaries as a series of shape states associated with expected types of image features. Finally, the building boundaries were optimally recovered using maximum likelihood estimation.

Subsequently, many authors concentrated on developing real-world applications using a semi-automatic approach. This is still considered the preferred solution as it is robust, flexible, and gives the user control over the interpretation tasks (e.g., Everingham et al., 2003; Koike et al., 2001). Zhou et al. (2005) proposed a general framework for human-guided image interpretation consisting of five components, a human-computer interface, a user model, a set of computational algorithms, a knowledge transfer scheme, and a performance evaluation scheme. Of these five components, the human-computer interface and the computational algorithms have been studied extensively, but research on the knowledge transfer scheme, more specifically on the learning of computational algorithms from humans, has been very limited.

Expanding on the above idea, Zhou et al. (2006) proposed a human-computer interactive framework for road tracking in aerial images. The framework enables knowledge transfer from human to computer via human input and failure diagnosis. The human inputs provide the road tracker with historical and current road information, which in turn is used to initialize a Bayesian filtering process to estimate the states of the road tracker. The road location is then estimated by selecting the optimal candidate from a number of potential road locations. A knowledge accumulation mechanism is implemented to store historical data so that the tracking model can use temporal information to compute the best matches for the current spatial road profiles.

Many of the semi-automated image interpretation systems (e.g., Hu et al., 2004; Amo et al., 2006), use a static image interpretation model that is modified by human input. The static model is typically obtained by analyzing and implementing human knowledge about a task domain and task-related experiences. The model works without any change throughout the whole interpretation process. The human interacts with the system by initializing the interpretation

process, correcting errors, and terminating an ongoing process when an error happens. This approach is illustrated by Boykov and Jolly (2001)'s interactive image segmentation algorithm. The human provides foreground and background labels using a digital brush, and then unlabeled data are assigned to the corresponding class using graph cuts. Similarly, Wang et al. (2005) proposed an interface for object segmentation by providing only foreground labels. On this basis, optimal segmentation performance is achieved using mean-shift and min-cut methods.

The problem with static models is that they do not consider the spatial-temporal properties of image interpretation. Consider the following scenario: a user is working with a semi-automatic image interpretation system, in a time-series, in sequential manner, and on tasks such as classification or regression. Once in a while during the process, the system may decide that it does not work properly, e.g. it detects abrupt changes in the time-series, and it requests user input. Such changes are normal, given the spatial variation of image features. In the static model, even with a user input, the system cannot adapt to the changes because input cannot lead to a model update.

To solve this problem, a predictor is required that can adapt to changes. In this dynamic scheme, the overall robustness of the system is influenced by several factors. First, it is important to retain the "human-in-the-loop" in the sense that the predictor is able to request necessary guidance at change points. Second, the predictor should be able to gradually update itself, on the basis of both, human inputs and autonomous analyses. Third, the predictor should be capable of detecting changes correctly and timely. Fourth, it is desirable to have as few human interventions as possible, as they can be very costly.

We propose a spatial-temporal model for addressing this problem from a machine learning and feature extraction point of view. The machine learning

part is based on online learning from human inputs. The approach is intuitive and flexible, and the learning rate of the proposed online learning algorithm is adaptive. We also discuss related issues, such as dealing with non-stationary distributions, the issue of unbalanced data, and working with an ensemble of models. Our work is motivated by transductive support vector machines (TSVM; Vapnik, 1998) for semi-supervised learning, and it is closely related to work on online learning (e.g., Littlestone and Warmuth, 1994; Kivinen et al., 2004) and to work on the detection of abrupt changes in time-series (Basseville and Nikiforov, 1993). As discussed in more detail below, the feature extraction part is based on domain knowledge and human perceptual processes.

This work extends that of Zhou et al. (2007) who developed an online learning approach to integrate human knowledge with computational models for novelty detection in road tracking. The input road profiles were used as training examples to generate road predictors. In this sense, this approach is closer to the idea of Caelli et al. (2001) than that of Zhou et al. (2006). As the human-computer interaction continues, multiple road predictors can be learned and generate a more robust road tracker.

Closely related problems on online learning and change point detection have been independently studied in the machine learning community (e.g., Littlestone and Warmuth, 1994; Littlestone, 1988; Vovk, 1995; Kivinen et al., 2004; Basseville and Nikiforov, 1993; Kifer et al., 2004; Ho, 2005), where the focus is on theoretical investigations rather than practical applications. What is new about our approach is that we try to incorporate current developments in online learning and change point detection techniques into practical applications such as image interpretation.

Finally, we must also consider feature extraction, another major issue that affects the performance of the image interpretation system. It has been men-

tioned by Caelli et al. (2001) that, in order to achieve a good interpretation performance, one needs to rely on combinations of features. This idea has been supported by the recent progress in image classification and object recognition (e.g., Varma and Ray, 2007). Nilsback and Zisserman (2006) used combinations of color features, scale-invariant feature transforms (SIFTs; Lowe, 1999) and texture features to classify flowers. In the PASCAL Visual Object Classes Challenge (Everingham et al., 2008), researchers used up to 30 possible combination of point samplings, spatial pyramids and descriptors to categorize 20 classes of images collected from the Internet. In specific image interpretation tasks, for example in road tracking, feature selection is more strongly affected by domain knowledge. In the following sections, we will also illustrate how road interpretation systems perform with different features and their combinations.

The paper is organized as follows. We start by formulating the problem, which leads to the proposed online learning approach. A number of related issues are discussed, and by addressing them, we explain the details of the proposed method. The applicability of the proposed approach is demonstrated with experiments on a synthetic simulation (interactive classification) and on a real-world problem (interactive road tracking). Finally, we also discuss the influence of feature selection and combination, as well as related issues.

2 Proposed Approach

Our goal is to develop a system that learns image interpretation tasks by observing the actions of human experts and uses these observations to learn classification rules. The newly acquired classification rules are used to classify further examples until an example is encountered that cannot be dealt with. At this point, control should be handed back to the human expert who then deals with these new data, initiating a new learning-prediction session. In the following

sections, we first present a formal problem description, and then we motivate and describe our method. Further technical details of our approach can be found in (Zhou et al., 2008).

2.1 Problem Formulation

Let x denote an instance and let y be a label. An example contains either an instance-label pair (x, y) or only an instance x. An interactive application takes a time-series $\mathcal{T}$ as input and operates in sessions, $\mathcal{T} = (\cdots, S_i, \cdots)$. Each session

$$S = \Big((x_{j+1}, y_{j+1}), \cdots, (x_{j+m}, y_{j+m}), x_{j+m+1}, \cdots, x_{j+n}\Big)$$

corresponds to one disjoint segment of the time-series, which starts with a few examples with corresponding human inputs $(x_{j+1}, y_{j+1}), \cdots, (x_{j+m}, y_{j+m})$, then produces predictions for a sequence of unlabeled examples $x_{j+m+1}, \cdots, x_{j+n}$, and ends with an abrupt change. This change, which is detected by the session predictor h, triggers a request for new human input for the next few examples, leading to a new session. An example of performing interactive classification on an synthetic time-series is illustrated in Fig. 1 top. The prediction problem could be one of classification, regression, ranking, or others, depending entirely on the interactive application at hand.

To begin with, let us consider a special case where the time-series contains exactly one session, i.e. $\mathcal{T} = \Big((x_1, y_1), \cdots, (x_m, y_m), x_{m+1}, \cdots, x_n\Big)$. This is closely related to the typical setting of semi-supervised learning[1], where a large number of unlabeled examples is expected to help elucidate prediction, together with a few labeled examples. As advocated by (Vapnik, 1998) in the TSVM, this

[1] In case of classification (or regression), it is exactly a semi-supervised classification (or regression) problem.

can be achieved by exploiting the so-called cluster assumption: the prediction hyperplane should maintain a large margin over the dataset including both the labeled and unlabeled examples, and minimize a regularized risk function on the *entire* set of examples

$$\min_f \quad \frac{1}{2}\|f\|^2 + \lambda_l \sum_{i=1}^{m} L_l(x_i, y_i, f) + \lambda_u \sum_{j=m+1}^{n} L_u(x_j, f) \tag{1}$$

where $\lambda_l, \lambda_u > 0$ and f is a parameter vector to predict a label. In the case of binary classification, $h(x, f) = +1$ if $f(x) > 0$ and $h(x, f) = -1$ otherwise. We obtain a TSVM by letting $L_l(x_i, y_i, f) = (\rho_l - y_i f(x_i))_+$ and $L_u(x_j, f) = (\rho_u - |f(x_j)|)_+$, where the margins $\rho_l, \rho_u > 0$ and $(\cdot)_+ = \max\{0, \cdot\}$. In the following, we assume $\rho \triangleq \rho_l = \rho_u$. The induced optimization problem cannot be solved easily because the second loss term $(\rho - |f(x_j)|)_+$ is a non-convex function. A lot of effort has been devoted to minimizing non-convex objective functions (using, e.g. deterministic annealing or the concave-convex procedure (**?**)). Returning to our situation, things are even worse because there are multiple sessions, and, in addition to making predictions, abrupt changes have to be detected since we do not know a priori when a session should end.

We consider an online learning framework, where, at time t, given the current parameter f_t and an example (x_t, y_t) (or x_t), we update the parameter f_{t+1} by minimizing a regularized risk function on the *current* example

$$f_{t+1} \;=\; \operatorname*{argmin}_f \;\; \frac{1}{2}\|f - f_t\|^2 + \eta_t L_l(x_t, y_t, f), \tag{2}$$

when (x_t, y_t) is presented. When only x_t is presented, this becomes

$$f_{t+1} \;=\; \operatorname*{argmin}_f \;\; \frac{1}{2}\|f - f_t\|^2 + \eta_t L_u(x_t, f). \tag{3}$$

In both cases, the new parameter f_{t+1} is expected to be reasonably close to the previous f_t (the first term), while incurring a small loss on the current example (the second term). η_t is a trade-off parameter that balances between the two objectives and is usually fixed a priori, i.e. $\eta = \eta_t,\ \forall t$.

There are distinct advantages to choosing this online learning framework: First, it is computationally more efficient, and second, an online algorithm can be elegantly extended to track a slowly drifting target over time in one segment, as will be shown later. Moreover, as shown next, by incorporating the change detection component, we end up working with a convex objective function.

This framework is very flexible in the sense that various loss functions can be deployed for different applications. To illustrate this point, we present three types of loss functions:

Binary classification loss: We use binary hinge loss (Schölkopf and Smola, 2002), which gives $L_l(x_t, y_t, f) = (\rho - y_t f(x_t))_+$ and $L_u(x_t, f) = (\rho - |f(x_t)|)_+$. This loss is used in applications such as interactive road tracking and video segmentation.

Regression loss: Using insensitive loss (Schölkopf and Smola, 2002), we have $L_l(x_t, y_t, f) = (|f(x_t) - y_t| - \rho)_+$ and $L_u(x_t, f) = (|f(x_t) - x_t| - \rho)_+$.

Ranking loss: We use ordinal regression hinge loss (Chu and Keerthi, 2005) for ranking problems. Each instance x_t is associated with a vector $y \in \{-1, +1\}^r$ as follows: If the rank of an instance is k, we set the first k components of y to $+1$ and the rest of the components to -1. Now $f(x_t)$ is r-dimensional with the k-th component being $f(x_t, k)$. Then $L_l(x_t, y_t, f) = \sum_{k=1}^{r} (\rho - y_{t,k} f(x_t, k))_+$ and $L_u(x_t, f) = \sum_{k=1}^{r} (\rho - |f(x_t, k)|)_+$. This loss can be used in weblog-based recommendation systems.

2.2 Change Detection

We use a moving-average method (e.g., Basseville and Nikiforov, 1993) to detect the change points in a classification problem. This is executed by maintaining the recent values of $\{x_a : a \in \mathcal{A}_c\}$ in a FIFO queue of fixed size $|\mathcal{A}|$ for class c. A change is detected if the distance between $\sum_a x_a/|\mathcal{A}|$ and x_t exceeds a threshold $\delta > 0$, when x_t is predicted as belonging to class c. This method can be easily extended to regression and ranking problems.

In addition, the cluster assumption also suggests that encountering a severely in-separable example (i.e. it is too close to the prediction hyperplane in the case of classification and ranking) indicates the beginning of a new session. For a real $\epsilon > 0$, the predictor decides whether to request human input as follows: For classification and ranking problems, a change point is detected if $|f(x_t)| \leq \epsilon$ with $0 < \epsilon < \rho$, while, for regression, the rule becomes $|f(x_t) - x_t| \geq \epsilon$ where $0 < \rho < \epsilon$.

Having incorporated change detection in this manner, we now deal with a convex minimization problem. The reason is illustrated in Fig. 2 on a classification problem, which was originally a non-convex problem due to the term $|f(x)|$, which peaks at zero. We use an ϵ-ball centered around the peak point, and the predictor stops asking for labels, whenever the value of $f(x)$ falls into the ϵ-ball, turning (3) into a convex minimization problem with the feasible regions being entirely outside the ϵ-ball. This holds similarly for regression and ranking problems. In addition, one can show that it is safe to proceed by just considering whether $f_t(x_t)$ is outside the ϵ-ball, which is much easier than to compute $f(x_t)$ (see Zhou et al., 2008).

2.3 Dealing with Non-stationary Distributions

So far we have considered a stationary scenario where, in each session of the time-series, the examples are drawn from the same distribution. In practice, however, the examples in each session might drift slowly. This can be accommodated by extending (2) and (3) as follows:

$$f_{t+1} = \underset{f}{\operatorname{argmin}} \ \frac{1}{2}\|f - f_t\|^2 \ + \left(\frac{\lambda}{2}\|f\|^2 + \mathrm{c}_l \mathrm{L}_l(x_t, y_t, f)\right), \tag{4}$$

and

$$f_{t+1} = \underset{f}{\operatorname{argmin}} \ \frac{1}{2}\|f - f_t\|^2 \ + \left(\frac{\lambda}{2}\|f\|^2 + \mathrm{c}_u \mathrm{L}_u(x_t, f)\right), \tag{5}$$

where η_t is incorporated into λ and c_l (or c_u). This leads to a geometrical decay of previous estimates with $f_t = (1-\tau)f_{t-1} - c(1-\tau)\partial_f \mathrm{L}$, where $\tau = \frac{\lambda}{1+\lambda}$ and L is either $\mathrm{L}_l(x_t, y_t, f)$ or $\mathrm{L}_u(x_t, f)$.

2.4 Kernels

To make use of the powerful kernel methods, the proposed framework can be lifted to reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ by letting $f \in \mathcal{H}$, with the defining kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying the reproducing property, $\langle f, k(x, \cdot)\rangle)_{\mathcal{H}} = f(x)$. The representer theorem (Schölkopf and Smola, 2002) guarantees that f can be expressed uniquely as $f_t = (1-\tau)f_{t-1} + \alpha_t k(x_t, \cdot)$ (see Cheng et al., 2006).

2.5 The Method

We can now present our method. For ease of exploration, we focus only on the problem of binary classification. With proper modifications, it can be easily

adapted to regression, ranking and other problems. Recall that, at time t, we are presented with an example x_t and possibly with a ground-truth label y_t, and we want to update the parameter f_t to f_{t+1} by incorporating the new example.

On the one hand, when (x_t, y_t) is presented, f is updated by solving the optimization problem (4) with $\mathrm{L}_l(x_t, y_t, f) = (\rho - y_t f(x_t))_+$. After simple derivations, we have $f_{t+1} = (1-\tau)f_t + \alpha_t k(x_t, \cdot)$, where

$$\alpha_t = \begin{cases} \hat{\alpha}_t & \text{if } y_t\hat{\alpha}_t \in [0, (1-\tau)c_l] \\ 0 & \text{if } y_t\hat{\alpha}_t < 0 \\ y_t(1-\tau)c_l & \text{if } y_t\hat{\alpha}_t > (1-\tau)c_l, \end{cases} \tag{6}$$

with

$$\hat{\alpha}_t \;=\; \frac{\rho - (1-\tau)y_t f_t(x_t)}{y_t k(x_t, x_t)}.$$

On the other hand, when only x_t is presented, f is updated by solving the optimization problem (5) with $\mathrm{L}_u(x_t, f) = (\rho - |f(x_t)|)_+$. As a result, the value of α_t is hinged on $f_t(x_t)$ and has two cases:

$$\alpha_t = \begin{cases} \alpha_t^+ & \text{if } f_t(x_t) \geq \epsilon \\ \alpha_t^- & \text{if } f_t(x_t) \leq -\epsilon. \end{cases} \tag{7}$$

In the first case, lettting

$$\hat{\alpha}_t^+ = \frac{\rho - (1-\tau)f_t(x_t)}{k(x_t, x_t)}, \tag{8}$$

we have

$$\alpha_t^+ = \begin{cases} \hat{\alpha}_t^+ & \text{if } \hat{\alpha}_t^+ \in [0, (1-\tau)c_u] \\ (1-\tau)c_u & \text{if } \hat{\alpha}_t^+ > (1-\tau)c_u \\ 0 & \text{if } \hat{\alpha}_t^+ < 0. \end{cases} \tag{9}$$

Similarly, in the second case, letting

$$\hat{\alpha}_t^- = \frac{-\rho - (1-\tau) f_t(x_t)}{k(x_t, x_t)}, \tag{10}$$

we have

$$\alpha_t^- = \begin{cases} \hat{\alpha}_t^- & \text{if } \hat{\alpha}_t^- \in [-(1-\tau)c_u, 0] \\ -(1-\tau)c_u & \text{if } \hat{\alpha}_t^- < -(1-\tau)c_u \\ 0 & \text{if } \hat{\alpha}_t^- > 0. \end{cases} \tag{11}$$

2.6 Dealing with Unbalanced Data

For practical interactive classification applications, the number of examples are often unbalanced across different categories. In road tracking, for example, the number of positive examples (road examples) is much smaller than the number of negative ones (off-road examples). Our framework can be extended to deal with this issue. Consider a binary classification case, and let ρ^+ and ρ^- be the margins for the positive and negative sides of the separating hyperplane, respectively. When receiving (x_t, y_t), the loss is associated with proper margin conditioning on whether y_t is positive or negative. Similarly, when presented only with x_t, it is conditioned upon $f_t(x_t)$ as $L_u(x_t, f) = (\rho^+ - f(x_t))_+$ if $f_t(x_t) > 0$, and $L_u(x_t, f) = (\rho^- + f(x_t))_+$ otherwise.

2.7 An Ensemble of Predictors

Over time, an interactive system collects a number of predictors from past sessions. Intuitively this ensemble can help to improve predictions for new sessions. While sophisticated algorithms with guaranteed theoretical bounds exist for ensemble methods (Dietterich, 2000), we use a simple strategy: Recent predictors are maintained in a queue of bounded size, and when an abrupt change is detected, we first search in the queue for an optimal predictor that can still perform well on the change point and switch to this predictor to continue with automatic predictions rather than resorting to human input. The intuition is simple. The models learned in the past might turn out to be useful for the current scenario. Later this strategy is applied in the road tracking application, leading to an overall improved performance.

3 Experiments and Performance Evaluation

We applied the proposed approach to the problem of interactive classification of a synthetic time-series and to a real-world road tracking task. The comparison approach is a nearest-neighbor moving average (NNMA) algorithm. In each session, the NNMA assigns to the current example a label according to the closest match among the human inputs, and change points are detected using the same moving average method. The parameters were tuned individually for good performance.

The performance of an interactive system is usually evaluated objectively based on two criteria: accuracy and efficiency. The accuracy criterion considers tracking errors (i.e. those with large deviation from corresponding manual labels), while the efficiency criterion deals with the time saving for the human operator (e.g. by measuring how many mouse clicks the operator has made in

a fixed set of road maps for interactive road tracking). These lead to a 2-D accuracy-efficiency plot (see Fig. 1 middle) where the horizontal axis measures accuracy and the vertical axis shows efficiency. After proper normalization, the results fall in a $[0, 1]$ bounding box. Similar to ROC curves, the performance of an ideal system would fall in the top-right area of the box. We use this method to evaluate related systems throughout the experiments.

4 Interactive Classification in Synthetic Time-series

We present two experiments on a synthetic time-series. Fig. 1 top presents an example time-series, which contains 1-D examples sampled with uniform probabilities from two classes (marked with circle and asterisk signs). To mimic a real-world situation, each class-conditioned distribution, a 1-D Gaussian distribution, is allowed to drift slowly over the time. Five disjoint subsequences are sampled in this way, each using a distinct drifting pattern, to model abrupt changes. With this type of time-series, a semi-automatic system is expected to predict with good accuracy and to make minimum queries for inputs.

Fig. 1 middle shows a comparison of the NNMA method and the proposed approach on the accuracy-efficiency plot, where the results are averaged over five time-series. The results indicate that the proposed approach delivers better performance when the class-conditioned distributions drift over time, as is typical in real-world settings.

To get an idea of the robustness of the proposed approach, we conducted a second experiment, where the algorithms were evaluated on time-series of different levels of difficulty: At the easiest level, the means of the two class-conditional distributions are well separated, while at the most difficult level,

the two means are very close to each other, with the standard deviations fixed throughout this experiment. In Fig. 1 bottom, from left to right, the problems become easier to deal with as the gap between the sample means of both positive and negative classes grows. The vertical axis measures the distance to the ideal performance at the top-left corner (1,1) of the accuracy-efficiency plot. A small value therefore indicates better performance. Fig. 1 bottom displays the results, where each value along the curves is computed by averaging over ten time-series. As one can see, our proposed method gives overall a better performance than NNMA.

5 Interactive Road Tracking

Interactive road tracking (IRT) refers to a semi-automatic image understanding system to assist a cartographer annotating road segments in aerial photographs. Given an aerial photograph containing a number of roads, the IRT system assists the cartographer to sequentially identify and extract road segments (including, for example, transnational highways, intrastate highways, and roads for local transportation). As shown in Fig. 3 top, road-tracking is not a trivial task because road features vary considerably due to changes in road material, occlusions of the road, and lack of contrast with off-road areas. It is extremely difficult for a fully automatic system to annotate the road segments with reasonable accuracy. Much research has been devoted to road tracking in aerial photographs (e.g. Merlet and Zerubia, 1996; Geman and Jedynak, 1996; Yuille and Coughlan, 2000; Lacoste et al., 2005), but these attempts have been devoted to automatic systems. People have gradually realized that the human cartographer can and should provide help in these systems (e.g. Caelli et al., 2001; Geman and Jedynak, 1996). In recent years, a number of semi-automatic systems have been proposed (e.g. Rochery et al., 2006; Hu et al., 2007; Zhou

et al., 2007).

In our system, a session comprises an *input* phase and a *detection and prediction* phase. The input phase contains a series of labeled examples along a road segment, where each example (also called an on-road profile) records the location and direction of the local road segment, together with features that characterize the local texture. We also collect a set of off-road profiles by randomly sampling nearby regions. In the detection and prediction phase, the system searches ahead and returns a set of candidate profiles based on the location and direction of the current road segment. Then the online predictor is used to select on-road profiles. The location and direction of the next example is decided by a weighted average of these profiles, where the weights are proportional to their distance from the separating hyperplane. In cases where too few on-road profiles exist, or where a good portion of the candidate profiles is within the ϵ-ball, the session ends and further input is requested from the user. Fig. 3 bottom shows an example consisting of two sessions. The first one starts in the top-right corner. White line segments indicate the locations of human inputs while white dots indicate road axis points detected by the system. When the road changes from dark to light (just before the freeway crossing), a human input is required to guide the tracker because the light road surface has not been experienced before.

We used the dataset from (Zhou et al., 2007). Our goal was to semi-automatically identify the 28 roads in one large photograph of the Marietta area in Florida. Eight participants were involved in this experiment. Each participant was asked to provide manual annotations of the road center axes, and we simulated the semi-automatic process using the recorded human data as virtual users. Over the eight participants, accuracy results were similar (0.97 for both methods, corresponding to 1.95 vs. 1.85 pixels deviation from the ground-

truth road center axes, for the proposed method and NNMA, respectively), but our proposed approach was able to work with substantially fewer human interactions (efficiency score 0.70 for our proposed method; 0.64 for the NNMA method).

5.1 Features and Feature Combinations

We implemented four features for characterizing road profiles, intensity (I), gradient (G), direction (D) and saliency (S). Road profiles of each feature were extracted in directions parallel (PA) and perpendicular (PE) to the road direction. The rationale for using the intensity feature is straight-forward. The intensity of the road surface is assumed to change little along the road because road segments are normally built with one material while there are typically strong intensity differences between on-road and off-road areas as well as within off-road areas. The contrast between roads and other areas suggests that one should find two parallel edges, with an intensity gradient stronger across these edges than along the road. To compute image gradients, we used a Sobel operator. To model the response to the road direction, we used Gabor filters, computed in eight directions uniformly distributed in the interval $[0\ \pi)$. The overall Gabor response at each pixel was computed as the maximum response of the Gabor filters in all directions. Saliency was computed following Itti et al. (1998)'s method, without using color information. This method computes a saliency map that highlights those regions that are likely to attract visual attention. The motivation for using saliency was the idea that roads look rather different from neighboring areas. Examples of the different feature maps for one road segment are show in Fig. 4.

Tracking performance of each feature is shown in Table 1. Several observations are important. First, reliance on perpendicular features is better than re-

I-PE	I-PA	G-PE	G-PA	D-PE	D-PA	S-PE	S-PA	E	A
1.0								0.775	2.05
	1.0							0.737	2.09
		1.0						0.690	2.51
			1.0					0.633	2.70
				1.0				0.773	1.83
					1.0			0.755	1.79
						1.0		0.520	2.49
							1.0	0.480	2.38
1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.794	2.13

Table 1: Performance for individual features and combinations of features. I, G, D, S denote intensity, gradient, direction, and saliency, respectively. PE and PA denote perpendicular or parallel to road direction. E denotes efficiency. A denotes accuracy, which is the average deviation from the true road centers (in pixels). The last row reports performance for a combination of features I-PE and I-PA with weights 1.0 and 0.2, respectively.

liance on parallel features, indicating that the variations across roads are better suited for road detection than variations along the road. Second, intensity and direction have similar efficiencies, both much higher than the other two features. This is due to the fact that, in many areas, roads do not have adequate contrast to the neighborhood, violating the assumption that roads can be characterized by parallel edges and higher saliency. Third, considering the accuracy-efficiency trade-off, the direction feature outperforms all other features, suggesting that, if a single feature were to be used in road tracking, the Gabor filter response would be the best choice.

We also tried combinations of features, i.e. concatenated individual features with different weights. The best performance was achieved when only intensity was used and when the perpendicular feature was assigned a higher weight than the parallel one (see Table 1).

5.2 Discussion

What we set out to develop was a system that learns to track roads by observing an expert (at the beginning of a session), continuing autonomously for the remainder of the session, and returning control to the human expert as soon as there is too much discrepancy between hypothesized road segments and measured image characteristics. The system was trained with a set of general features for characterizing local road image patterns. After relatively short training sessions, certain features (intensity and direction) turn out to be much more useful for detection road segments than others (parallel edges and saliency). This is of course the well-known and well-investigated feature selection problem. What is new in our approach is that features and spatial patterns are learned in a dynamic sequential learning process and in the presence of gradually changing features.

With relatively short training, our system learned that roads can be characterized by several characteristics: 1) Road are smooth. Changes of the road surface should thus be small, and the decay factor in the temporal modeling should be small. 2) The presence of parallel edges normally presented in the road leads to gradient information being useful for road detection. 3) The contrast between roads and its neighboring areas suggests that one can perform machine learning in a binary classification setting.

Our system learns simple image interpretations in a setting where the system has to select optimal classification features in a dynamic context and with changing feature characteristics. This is very different from the classic view of image analysis that relies on a fixed, albeit well-chosen set of features for classifying images or image fragments. It is interesting to note that this is a view that Terry Caelli has proposed a very long time ago (e.g., Caelli, 1988; Caelli et al., 1987). Similarly, we show that our kernel-based learning algorithm per-

forms as an information screener that selects and adapts to useful road features as support vectors. In addition, the decay factor simulates a forgetting process: Outdated support vectors are given weights that decrease over time so that the system can adapt to the most recent appearance of roads.

6 Conclusions

We have presented a novel approach to sequential prediction and change detection, which often arise in interactive applications. We devised an online-learning algorithm that naturally unifies the problems of prediction and change detection into a single framework. We applied the proposed approach to an interactive classification of a synthetic time-series, and we also experimented with a real-world road tracking task with different feature settings. We were able to show that our proposed approach is very competitive with other approaches. In our future work, we are looking into applying our framework to other interactive, complex image interpretation problems.

Acknowledgments

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. This work is supported by the IST Program of the European Community, under the Pascal Network of Excellence, IST-2002-506778. WFB was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Amo, M., Martínez, F., and Torre, M. (2006). Road extraction from aerial images using a region competition algorithm. *IEEE Transactions on Image Processing*, 15(5):1192–1201.

Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes: theory and application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Boykov, Y. Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of the International Conference on Computer Vision*, pages 105–112.

Caelli, T. (1988). An adaptive computational model of texture segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):9–17.

Caelli, T., McCabe, A., and Briscoe, G. (2001). Shape tracking and production using hidden Markov models. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):197–221.

Caelli, T., Rentschler, I., and Scheidler, W. (1987). Visual pattern recognition in humans I: Evidence for adaptive filtering. *Biological Cybernetics*, 57:233–240.

Cheng, L., Vishwanathan, S. V. N., Schuurmans, D., Wang, S., and Caelli, T. (2006). Implicit online learning with kernels. In Schölkopf, B., Platt, J., and Hofmann, T., editors, *Advances in Neural Information Processing Systems 19*, Cambridge MA. MIT Press.

Chu, W. and Keerthi, S. S. (2005). New approaches to support vector ordinal regression. In *Proceedings of the International Conference on Machine Learning*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15, London, UK. Springer-Verlag.

Everingham, M., Thomas, B., and Troscianko, T. (2003). Wearable mobility aid for low vision using scene classification in a markov random field model framework. *International Journal of Human-Computer Interaction*, 15(2):231–244.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2008). The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008.

Geman, D. and Jedynak, B. (1996). An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14.

Harders, A. and Székely, G. (2003). Enhancing human-computer interaction in medical segmentation. *Proceedings of the IEEE*, 91(9):1430–1442.

Ho, S.-S. (2005). A martingale framework for concept change detection in time-varying data streams. In *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pages 321–327, New York, NY, USA. ACM.

Hu, J., Razdan, A., Femiani, J., Cui, M., and Wonka, P. (2007). Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4144–4157.

Hu, X., Zhang, Z., and Tao, C. (2004). A robust method for semi-automatic extraction of road centerlines using a piecewise parabolic model and least square template matching. *Photogrammetry Engineering & Remote Sensing*, 70(12):1393–1398.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Kifer, D., Ben-David, S., and Gehrke, J. (2004). Detecting change in data streams. In *VLDB'2004: Proceedings of the Thirtieth International Conference on Very Large Databases*, pages 180–191. VLDB Endowment.

Kivinen, J., Smola, A. J., and Williamson, R. C. (2004). Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8).

Koike, H., Sato, Y., and Kobayashi, Y. (2001). Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system. *ACM Transaction on Computer-Human Interaction*, 8(4):307–322.

Lacoste, C., Descombes, X., and Zerubia, J. (2005). Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1568–1579.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, 108(2):212–261.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157.

Merlet, N. and Zerubia, J. (1996). New prospects in line detection by dynamic-programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):426–431.

Nilsback, M. E. and Zisserman, A. (2006). A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:1447–1454.

Rochery, M., Jermyn, I. H., and Zerubia, J. B. (2006). Higher order active contours. *International Journal of Computer Vision*, 69(1):27–42.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley and Sons, New York.

Varma, M. and Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *Proceedings of the International Conference on Computer Vision*, pages 1–8.

Vasconcelos, N. (2004). On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Transactions on Information Theory*, 50(7):1482–1496.

Vovk, V. G. (1995). A game of prediction with expert advice. In *COLT '95: Proceedings of the Eigth Annual Conference on Computational Learning theory*, pages 51–60, New York, NY, USA. ACM.

Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F. (2005). Interactive video cutout. In *ACM SIGGRAPH Conference*, pages 585–594, New York, NY, USA. ACM.

Yuille, A. L. and Coughlan, J. M. (2000). Fundamental limits of Bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):160–173.

Zhou, J., Bischof, W. F., and Caelli, T. (2005). Human-computer interaction in map revision systems. In *CD-ROM Proceedings of the 11th International Conference on Human-Computer Interaction*, Las Vegas, Nevada.

Zhou, J., Bischof, W. F., and Caelli, T. (2006). Road tracking in aerial image based on human-computer interaction and Bayesian filtering. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(2):108–124.

Zhou, J., Cheng, L., and Bischof, W. F. (2007). Online learning with novelty detection in human-guided road tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):3967–3977.

Zhou, J., Cheng, L., and Bischof, W. F. (2008). Prediction and change detection in sequential data for interactive applications. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 805–810.

Figure Captions

Figure 1: An interactive classification task for a synthetic time-series. Top: An example of a synthetic time-series. Middle: A comparison of the proposed method and NNMA on the accuracy-efficiency plot. Bottom: A comparison of two methods over different levels of difficulty. See text for details.

Figure 2: The figure illustrates the effect of considering change detection and classification problems together. For a given x_t, the horizontal axis denotes $f(x_t)$ and the vertical axis is the instantaneous loss. In classification, the loss function contains both the solid and the dashed lines, leading to a non-convex problem. When dealing with both change detection and classification, only the solid lines outside the ϵ-ball are left, which gives a convex problem.

Figure 3: Top: A close-up view of an orthorectified aerial photograph of roads. Notice that occlusions, crossings, and changes in road surface materials lead to abrupt change in road appearance. Bottom: An example of interactive road tracking. See text for details.

Figure 4: Image feature maps for four different features. From left to right: intensity, gradient, direction, and saliency.

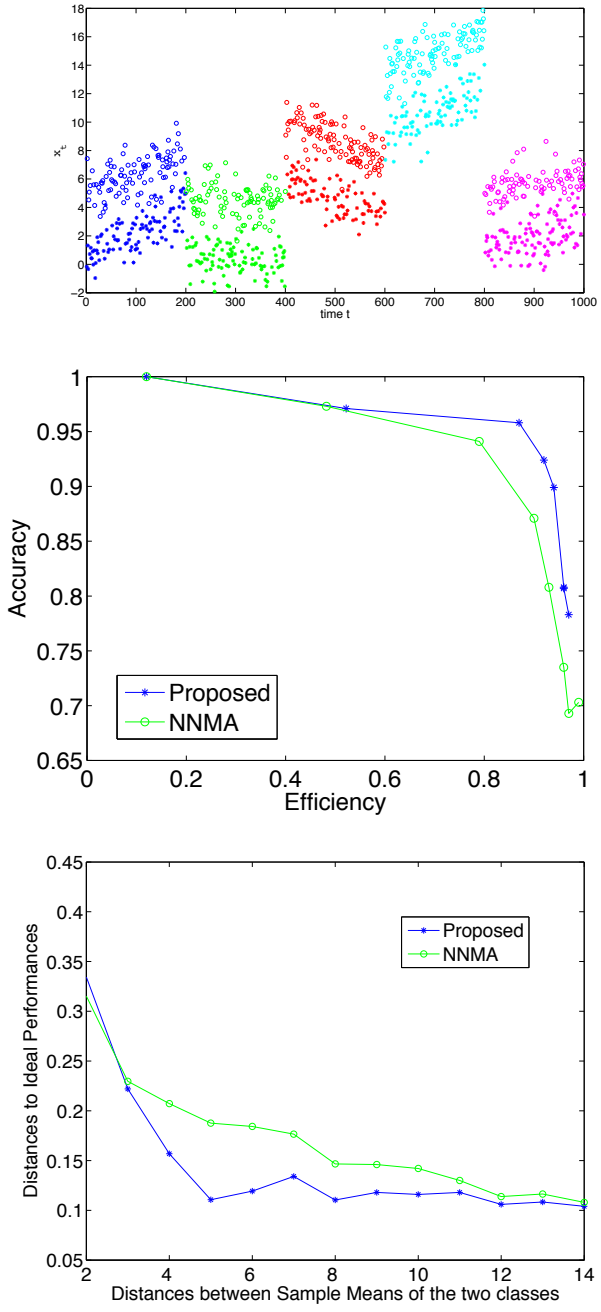
time t
Accuracy
Efficiency
Proposed
NNMA
Distances to Ideal Performances
Distances between Sample Means of the two classes
Proposed
NNMA

Figure 1:

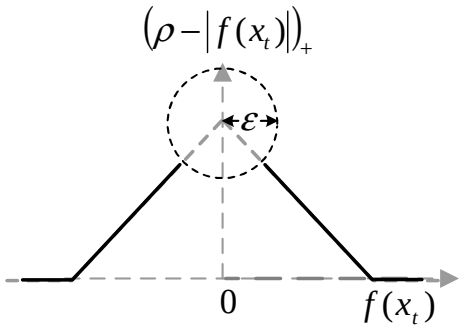

Figure 2:

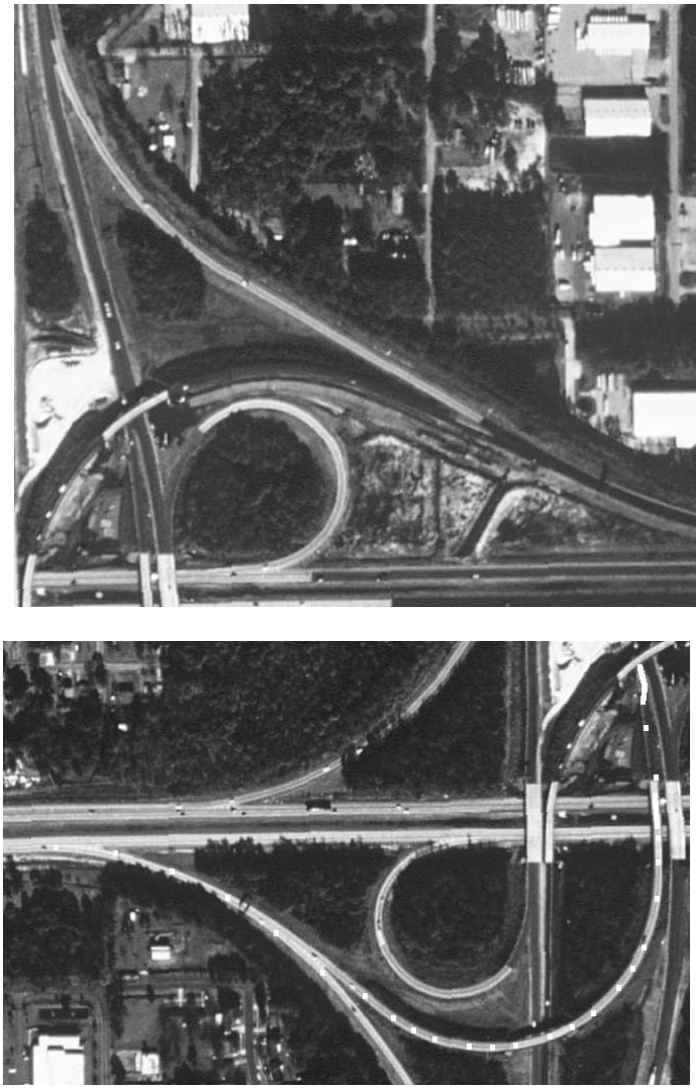

Figure 3:

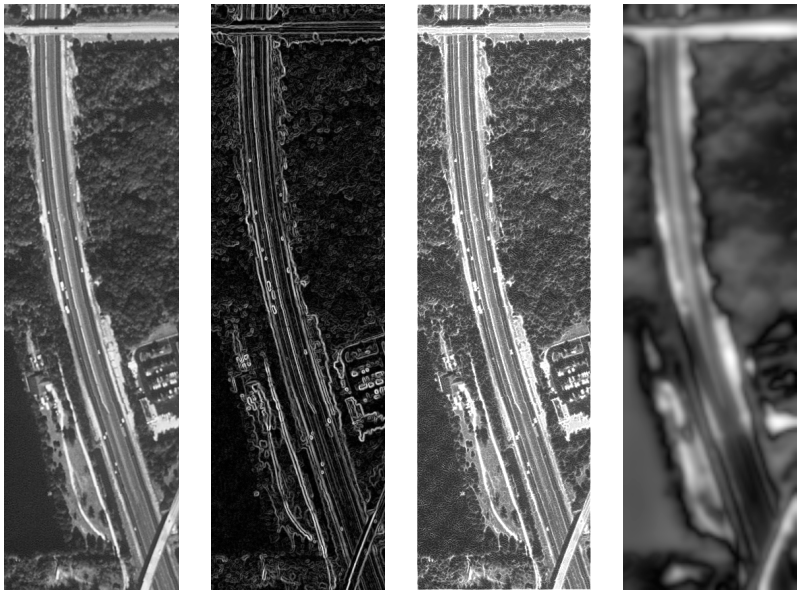

Figure 4: