A Pattern Discovery Approach to Retail Fraud Detection

Prasad Gabbur, Sharath Pankanti, Quanfu Fan, and Hoang Trinh
Exploratory Computer Vision Group, IBM Research
19 Skyline Drive, Hawthorne, NY 10532
pgabbur@email.arizona.edu, {sharat, qfan, trinhh}@us.ibm.com

ABSTRACT

A major source of revenue shrink in retail stores is the intentional or unintentional failure of proper checking out of items by the cashier. More recently, a few automated surveillance systems have been developed to monitor cashier lanes and detect non-compliant activities such as fake item checkouts or scans done with the intention of deriving monetary benefit. These systems use data from surveillance video cameras and transaction logs (TLog) recorded at the Point-of-Sale (POS). In this paper, we present a pattern discovery based approach to detect fraudulent events at the POS. Our approach is based on mining time-ordered text streams, representing retail transactions, formed from a combination of visually detected checkout related activities called *primitives* and barcodes from TLog data. Patterns representing single item checkouts, i.e. anchored around a single barcode, are discovered from these text streams using an efficient pattern discovery technique called *Teiresias*. The discovered patterns are used to build models for true and fake item scans by retaining or discarding the anchoring barcodes in those patterns respectively. A pattern matching and classification scheme is designed to robustly detect non-compliant cashier activities in the presence of noise in either the TLog or the video data. Different weighting schemes for quantifying the relative importance of the discovered patterns are explored: *Frequency*, *Support Vector Machine (SVM)* and *Frequency+SVM*. Using a large scale dataset recorded from retail stores, our approach discovers semantically meaningful cashier scan patterns. Our experiments also suggest that different weighting schemes result in varied false and true positive performances on the task of fake scan detection.

Categories and Subject Descriptors: I.5.4 [Pattern Recognition]: Applications—*computer vision, text processing*

General Terms: Security

Keywords: Retail, Security, Sweethearting, Pattern discovery, Teiresias

KDD'11 August 21–24, 2011, San Diego, California, USA.

1. INTRODUCTION

A major source of revenue shrink in retail stores is the intentional or unintentional failure of proper checking out of items by the cashier. It is estimated to cause billions of dollars in revenue losses each year [8]. In the case of an intentional failure, it is referred to as *sweethearting* between a cashier and a customer as both intend to derive monetary benefit from it. More recently a few automated surveillance systems have been developed to monitor cashier lanes and detect non-compliant activities. These systems use data from surveillance video cameras and transaction logs (TLog) recorded at the Point-of-Sale (POS). Video-based [8, 1, 2] systems visually monitor the activities of a cashier around the Point-of-Sale to detect item checkouts and verify them using TLog data. Being able to detect as many non-compliant events as possible while keeping the number of false alarms low is key to the successful deployment of these systems. It is a challenging problem to optimize the two conflicting objectives due to variations and noise within the input data streams.

Instead of using the two streams (video and TLog) of data separately, we posit that much can be learned about the nature of an item scan performed by a cashier by combining them into a single stream. This is because most item checkouts are normal (no fraud) and a barcode is registered in the TLog. By analyzing visual information around the registered barcode events, it is possible to model variations in the cashier's activities for checking out an item. This is helpful in detecting non-compliant cashier activities more robustly in the presence of noise in either the TLog or the video data.

Video-based systems usually make use of complex analytics to detect and monitor cashier activities [7]. Many of these approaches make adhoc assumptions such as only a constrained set of activities being performed by a cashier during retail transactions. Real-world data presents a number of challenges to such approaches due to those assumptions being violated very often. Computer vision systems are based on extracting visual features from input video streams. Recently, there has been some work in transforming videos into a sequence of discrete features [11] and applying text-based data mining algorithms on the resulting streams. This has opened up new possibilities for looking at video data in a different light in order to infer useful knowledge. Text-based algorithms are simpler and faster than many sophisticated video analysis techniques. But their potential in addressing some of the vision challenges have not been explored. In this work, we propose a text-based approach to analyzing

1: A typical checkout involves the pickup (P), scan (S) and drop (D) of the item of interest. Each such action is detected as a *primitive* by a motion-based primitive detector (See Section 3.1).

videos represented as time-ordered discrete features called *primitives* in the context of a retail surveillance application.

A retail transaction involves the checkout of a set of items the customer intends to buy. Ideally, each checkout event can be thought of as comprising three sub-events or primitives: Pickup (P), Scan (S), and Drop (D) in that temporal order, as illustrated in Fig. 1. A visual compliance approach [7] is designed to detect these sub-events in a video and group them into PSD triplets corresponding to visual scans of items. A visual scan with an associated barcode confirms a valid checkout, otherwise it is a potential candidate for a fraud or operational error. We refer to a valid checkout as a *true* or *visual scan* and a fraud or operational error as a *fake scan*.

Usually the output of the primitive and event detection modules is not a single PSD triplet for every checkout. Some of the sub-events may be repeated more than once or may not even be detected due to various reasons. One common situation arises when a cashier has difficulty scanning the item in a single attempt and ends up doing multiple scans resulting in repeated S primitives for a single item. Therefore the primitive sequence output by the detection system is usually noisy with repetitions and/or no occurrences of one or more of the P, S and D primitives. This poses difficulties to any approach looking for the presence of a PSD triplet corresponding to an item checkout. A more robust method accounting for missing and repeated primitives is needed.

In what follows, a paradigm for visual compliance based on pattern discovery and matching is proposed. A brief description of the proposed approach is given in Section 2. The details of the various steps involved including pattern discovery and matching, primitive classification and true/fake scan detection are given in Sections 3 and 4. Section 5 describes a few weighting schemes to quantify the relative importance of the discovered patterns for classification. Experimental details and results are given in Section 6 followed by a brief discussion of the results in Section 7 and conclusions in Section 8.

2. OUR APPROACH

Various patterns of item checkouts are discovered using an efficient pattern discovery algorithm called *Teiresias* [12]. The input to the pattern discovery algorithm are text strings formed from temporally ordering the output of the primitive detectors and the TLog data. By treating the barcode as another primitive, the alphabet set from which the text strings are formed has a cardinality of four: $\{P, S, D, B\}$. The output of the pattern discovery algorithm is a set of patterns within the input strings that are repeated at least a minimum number of times across those strings. Only those patterns that have a single barcode B primitive within them and with particular configurations of the other three primitives around it are selected and the others are discarded. Practically most item checkouts in retail transactions are genuine and fake checkouts are very rare. With this assumption, the selected patterns are representative of single item checkouts and model the variations in the process of checking out an item represented as a combination of discrete primitives. Conceptually, a fake scan or a fraudulent item checkout is similar in essence to a genuine checkout except that a barcode is not registered in the TLog. So a model representing fake scans is derived from the model for true scans by discarding the barcode primitives from the pattern set. For the sake of convenience, we term the models corresponding to true and fake scans as *positive* and *negative* models respectively.

The positive and negative models are used to detect the presence of any fake scans within a new transaction with the help of a pattern matching and classification scheme. Each primitive within the transaction stream is assigned a vote by each of the patterns in the positive and negative groups when they match a certain neighborhood of that primitive. Following a Bayesian approach, these votes are transformed into posterior likelihoods of the primitive being a part of a true or a fake scan. A classification based on maximum posterior likelihood (above a certain minimum confidence level) is done to assign the primitive to the most likely class. Finally a grouping of the classified primitives based on their class labels and temporal adjacency leads to a segmentation of the transaction stream into true and fake scans.

3. CHECKOUT PATTERN DISCOVERY

More details of the primitive detection and pattern discovery steps are given in the following subsections.

3.1 Primitive Detection

To detect primitives: Pickup (P), Scan (S), and Drop (D) from the retail video stream, we implement an efficient vision technique proposed in [14]. The approach is based on the observation that each primitive P, S, D follows a specific motion pattern - each primitive is an in/out process in which the cashier's hand enters and exits a region of interest (ROI). To detect hand motion, a hand color model is first adaptively learned by continuously collecting hand pixel examples. Based on this model, each pixel in the motion map computed by frame differencing is classified as hand pixel or not, to form a hand motion map for each video frame. This map explicitly captures the hand motion robustly in the presence of motion noise from belt movement, background changes, and customer interactions. A hierarchical finite state machine (FSM) is then applied to deterministically check whether or not the hand motion follows the predefined in/out pattern corresponding to each primitive. After the primitives P, S, D are detected, these are combined with the barcode primitives (B) from the TLog data to form a temporally ordered text string of $\{P, S, D, B\}$, which is used as input string to the pattern discovery algorithm.

3.2 TEIRESIAS

Teiresias [12] is a widely used algorithm for discovering repeated patterns across finite alphabet streams. It was originally developed for discovering rigid patterns in biological sequences, eg., motifs in amino acid sequences. However, it has been applied to problems in other domains such as spam filtering [13] and intrusion detection [18] in UNIX processes. Teiresias discovers all unique $< L, W >$ *maximal* patterns having a certain minimum support K within a given set of alphabetic sequences. A $< L, W >$ pattern is defined to be one in which every sub-pattern of length W or more has at most $(W - L)$ wildcard characters, the others are alphabets. The term *maximal* implies that the discovered patterns cannot be made more specific without reducing their support sets within the input streams. The computational complexity of the algorithm is seen to scale quasi-linearly with the size of the output patterns, hence making it a very efficient pattern discovery tool.

3.3 Patterns in retail transactions

The Teiresias pattern discovery algorithm is used to discover patterns that are representative of genuine item scans within retail transactions. In our system, a transaction is a stream of P, S, D and B primitives, where the ordering is determined by their occurrence in time. Since a typical item checkout involves picking, scanning and then dropping it, some representative patterns that correspond to this process would be $PSBD$, $PBSD$ or $PSDB$, allowing for time delays in primitive detection or barcode registration. In reality, there are a number of other patterns that are possible due to noisy primitive streams rendered by the primitive detector and TLog data. Some typical examples include:

1. Multiple scan: An item is scanned more than once before a successful barcode registration. Representative patterns in this case might be $PSDSBD$, $PSDSDSBD$, and many more.

2. Late barcode: A successful scan but a delay in the arrival of barcode might result in patterns such as $PSDB$, $PSDPSB$, etc. Such patterns might also result from items which have to be manually registered at the POS without scanning. Examples include organic produce such as vegetables and fruits.

3. Parallel scan: Following a successful scan, an item is dropped after the next item in the pickup queue is picked up. This can be represented using $PSPBD$, $PSBPD$ and so on.

Variations in the patterns representing item scans are captured by mining primitive streams, resulting from transactions, for patterns that repeat across transactions. Most of the item checkouts in transactions are true scans with a barcode being registered for each scan. Therefore patterns that are repeated across transactions and have a single barcode (B) primitive within them can be thought of as corresponding to true scans. We seek such patterns by first discovering unique maximal $< L, W >$ patterns with support at least K from a set of transactions, employing the Teiresias algorithm. This process is repeated for a few different $L = W$ values to capture representative patterns of different lengths, i.e., with different levels of noise embedded in them. Setting $L = W$ ensures that the resulting patterns do not have any wildcard characters so that exact pattern matching can be done using those patterns. For the experiments in this work, the $L = W$ values range from 3 to 10. We choose a conservative value for K (2) so that even patterns that are repeated only once are captured. From the resulting set, only those patterns that match the regular expression $P[PSD] * B[PSD] * D$ are retained. These are all the single barcode (B) patterns that start with a pickup (P) and end in a drop (D) primitive with any possible number and arrangement of P, S and D primitives between them.

Some of the patterns discovered by Teiresias are representative of different possible ways an item checkout occurs at the POS. Table 1 lists the first eight patterns based on their repeatability across transactions used for pattern discovery along with the number of transactions in which they occur. Upon visual inspection of the corresponding videos, the patterns $PBSD$ and $PSBD$ represent idealized item checkouts involving one each of the P, S, D and B primitives in them. $PDBSD$, $PBSPD$, and $PDBSPD$ are representative of overlaps between checkouts of two successive items. $PDBSD$ indicates that the drop of the previous item occurs in parallel or after the pickup of the next item, whose barcode is included in the pattern. Similarly, $PBSPD$ represents picking up the next item while or before dropping off the current item, whose barcode forms a part of the pattern. These are instances of parallel scanning. Parallel scanning of the current item with both the previous and the next one results in the pattern $PDBSPD$, with the B primitive corresponding to the current item. The other patterns in the table result from a missing S primitive in one or more of the cases described above. Similarly, we found that multiple scanning was represented by the patterns: $PDSDSBD$ and $PDSDBSD$, although they did not occur in as many transactions (19 and 53 respectively) as the patterns above.

3.4 Reducing pattern redundancy

The output of the Teiresias algorithm is a set of unique maximal patterns for a particular setting of the $< L, W >$ parameter values. We capture various levels of noise in the representation of a true scan by discovering patterns with different settings for $L = W$. This leads to some redundancy in the discovered pattern set in that a few patterns are repeatedly discovered for two different settings of $L = W$ and there are overlaps between some patterns. One example of an overlapping pair of patterns is PSB and SBD, which form sub-patterns of the canonical true scan pattern $PSBD$. This might be the result of slightly different support sets for the two patterns caused by a missing P or a D primitive in one of the scans within the input transactions. Repetitions are easily taken care of by retaining only the unique patterns. One possible way to account for overlaps is to find all pairs of patterns with overlaps among them and stitch them together into longer and longer patterns in an iterative process. But this has a high likelihood of reducing the support set of the resulting longer patterns relative to the component patterns if the component patterns did not come from a single parent pattern. This is not desirable because each of the overlapping patterns may uniquely represent a true scan with some missing primitives.

In order to address redundancy due to overlapping patterns, we do a pattern-transaction co-occurrence analysis, which is similar to Latent Semantic Analysis [6] used in the area of information retrieval. A pattern-transaction co-

occurrence matrix C is constructed, where each row in the matrix corresponds to a unique discovered pattern and each column to an input transaction used for pattern discovery. The $(i, j)$the entry of the matrix is a count of the number of times pattern i appears in the jth transaction. Usually C is a sparse matrix with a few positive entries scattered among zeros. Each row of this sparse matrix can be thought of as a representation of the corresponding pattern in a space of dimensionality equal to the number of transactions used for pattern discovery. In general, these dimensions are not uncorrelated. So a measure of similarity among patterns such as the correlation coefficient cannot be reliably computed in this space. Singular Value Decomposition is performed on the co-occurrence matrix C resulting in the following factorization of C

$$C = U\Sigma V^T \tag{1}$$

where U and V are orthogonal matrices of size $m * m$ and $n * n$, assuming C is of size $m * n$, i.e., there are a total of m patterns discovered from a set of n transactions. Σ is a rectangular matrix with only non-zero elements along its diagonal, which correspond to singular values. The columns of U and V are the eigenvectors of CC^T and C^TC respectively. Choosing only the first k largest singular values and the corresponding eigenvectors leads to the best rank k approximation of C in the least squares sense. Further, each row of matrix U is a representation of the corresponding pattern (in C) in an orthogonalized space.

Since the number of transactions n is usually much smaller than the number of patterns m, there are only n non-zero singular values in the SVD of C. Therefore only the first n columns of U comprise a full representation of the patterns in an orthogonal space while allowing a complete reconstruction of C. Correlation coefficient computed in this space is a more reliable measure of similarity between patterns based on their co-occurrence within the transactions. We use this measure for clustering patterns together that have a high correlation among them. A greedy agglomerative clustering scheme is adopted, where the patterns are first ordered in the decreasing order of their total number of occurrences. Then all the patterns that have a high correlation (> 0.5) with the most repeated pattern are put into one cluster. This is repeated with the next available most repeated pattern and so on until all patterns are exhausted. Each cluster is then represented by the most occurring pattern among its members. Only the cluster representatives are used for pattern matching.

4. PATTERN MATCHING AND CLASSIFICATION

The patterns derived from the above procedure define a model for true scans, which we refer to as the *positive* model for the sake of convenience. It can be hypothesized that fake scans are similar to true scans but without the registration of a barcode. With this assumption, a corresponding *negative* model is constructed for fake scans by eliminating the B primitive from the true scan patterns. Two tables are built, one corresponding to true scans (positive) and the other corresponding to fake scans (negative). The two tables contain the same set of patterns except for the barcode primitive being deleted from each pattern in the negative set. Given a primitive stream from a new transaction, each primitive in the stream is assigned a true or fake scan label based on a pattern matching and classification procedure using the two pattern sets. Each pattern in the positive set contributes a positive vote to every primitive of any corresponding substring it matches within the transaction stream. Similarly negative votes are accumulated for each primitive based on how many patterns in the negative set exactly match a substring containing the primitive. These votes are used for classifying the primitives as described below.

4.1 Classif cation

Each primitive p in the input transaction stream is assigned a true (T) or a fake (F) scan label depending on the number of votes it has accumulated for the two categories. In order to do this classification, a Bayesian framework is adopted. Let S_T and S_F denote the sets of all true and fake scan patterns respectively. Then the likelihood of the primitive p being a part of a true scan can be written as

$$P(T|p) \propto P(T, p) \tag{2}$$

Similarly, the likelihood of p belonging to a fake scan is given by

$$P(F|p) \propto P(F, p) \tag{3}$$

We define the joint likelihoods $P(T, p)$ and $P(F, p)$ to be the probabilities of the *context* of p belonging to the set S_T and S_F respectively.

$$P(T, p) \equiv P(p - context \in S_T) \tag{4}$$

$$P(F, p) \equiv P(p - context \in S_F) \tag{5}$$

where $p - context$ is used to denote the context or neighborhood of p. Each of the above likelihoods are proportional to the number of patterns in S_T and S_F respectively that match a sub-string containing p. If we denote the positive and negative votes accumulated by p with $Votes_T(p)$ and $Votes_F(p)$ respectively, then

$$P(p - context \in S_T) \propto Votes_T(p) \tag{6}$$

$$P(p - context \in S_F) \propto Votes_F(p) \tag{7}$$

Appropriate normalization of the cumulative votes of the primitive p for the two categories yields the posterior probabilities $P(T|p)$ and $P(F|p)$. The primitive is assigned to the class with the maximum posterior likelihood given it is above a certain threshold value.

4.2 True and fake scan detection

Classification of a transaction's primitive stream results in a class label (T or F) for those primitives whose posterior likelihoods for the respective class are above the chosen threshold. Other primitives remain unclassified. This usually results in continuous stretches of T's and F's with unclassified primitives in between them. One continuous stretch of T or F labels need not necessarily correspond to a single true or fake scan respectively. A segmentation like procedure groups subsets of adjacent primitives that are assigned T labels into true scans. This is achieved by choosing a B primitive within each such subset as the anchor for the corresponding true scan. Other primitives such as P, S and D that form a part of the true scan are searched for locally, i.e., the nearest ones to the anchoring B primitive within a small window. The size of the window is limited to the set of non-B primitives around the anchor, which have been classified as belonging to the class T. Only one P, S and D primitive is assigned to a true scan.

Among the available P's and S's within a window, preference is given to the ones that occur closest to B but earlier in the stream. On the other hand, a D that is closest to B but occurs later in the stream is preferably assigned to the true scan. In the absence of a P, S or D primitive within the neighborhood window of an anchoring B, a new such primitive is created and assigned to the true scan. Similar grouping is performed to detect fake scans using S primitives labeled as belonging to class F as anchors and searching for the nearest P and D primitives. In order to keep the number of false positives manageable, we assume that at least one each of P, S and D primitives should be present to provide enough evidence for a fake scan and that two successive fake scans be at least 3 seconds apart.

5. WEIGHTING VOTES

Within a transaction stream, a primitive is assigned a vote by each of the patterns matching a sub-string containing the primitive. A voting scheme where each pattern contributes the same vote to the matching primitives might not be optimal. Our goal is to be able to reliably distinguish true scans from fake scans based on the discovered patterns. Not all the patterns contain the same amount of information to distinguish between the two classes. There may still be some redundancies left among patterns filtered out by co-occurrence analysis. Therefore it is not guaranteed that each pattern contributes independent information relative to others in the set. Further some of the patterns may even be noisy, eg., resulting from missing or multiple primitive detections for single pickup, scan or drop events during transactions. A weighting scheme where patterns that carry independent information and are the most important for classification are weighted more relative to others is desirable. We experiment with two weighting schemes that assign different relative votes to the patterns.

5.1 Frequency weights

This is a simple weighting scheme where each pattern is assigned a weight inversely proportional to the number of times it occurs across all the transactions used for pattern discovery. This is mostly an empirical weighting scheme driven by experimentation. We tried different heuristic schemes that were based on the frequency of patterns and the size of their clusters in the co-occurrence analysis (Section 3.4). Among them, this simple weighting scheme led to the best results in terms of the number of falsely detected fake scans relative to the total number of detected true scans.

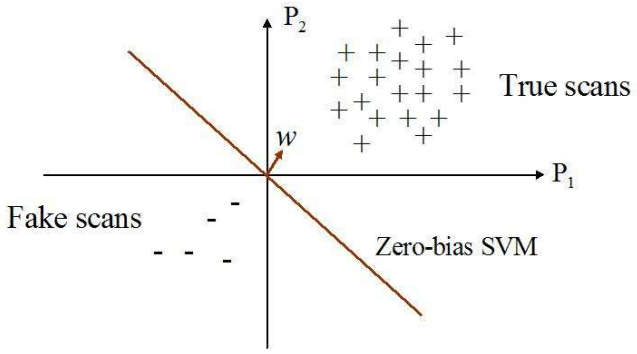
A possible reason might be that smaller patterns tend to occur more frequently than longer ones among our discovered patterns. Using a uniform voting scheme, the smaller patterns have a higher influence in the classification decision of a majority of primitives because smaller patterns match more sub-strings than longer ones. Further each pattern in the negative (F) class is one primitive shorter than its counterpart in the positive class. All these factors contribute to an increased number of short noisy primitive segments that are assigned to the negative (F) class. With the inverse frequency based voting scheme, longer (less frequent) patterns influence the classification decision of many primitives leading to longer, less noisy segments of continuous T or F labels. However, this scheme causes many genuine fake scans to be missed, based on our experiments (See Section 6). In order to circumvent this problem, we designed a weighting scheme where the weights were derived from a classifier trained to distinguish true from fake scans using ground truth data.

5.2 Support Vector Machine (SVM) weights

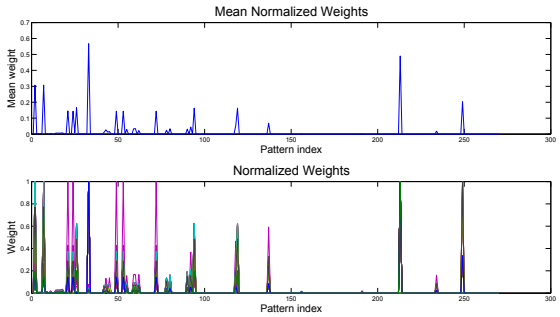
Support Vector Machine (SVM) [15, 3] is used in a variety of classification problems because of a number of its favorable properties including superior generalization ability. A two-class SVM estimates a hyperplane separating the training points in a feature space such that the margin between the hyperplane and the two classes is maximized. Assuming a linear kernel SVM, the weights are indicative of how useful the corresponding features are for the purpose of classification. This has been used for feature selection [5, 9]. We exploit this property of SVMs to learn the relative importance of the discovered patterns for classifying a primitive into true or fake scan based on the matches to its context. In order to achieve this, we sample a set of training points belonging to true and fake scan classes and map them onto a feature space with dimensionality equal to the number of patterns. The training points are a set of primitive strings centered around the B primitive for a true scan and the S primitive for a fake scan.

The B primitive locations of true scans are randomly sampled from the set of transaction streams used for training leaving out those transactions which have a fake scan in them. Note that we hand labeled the locations of the fake scans in our transaction streams by a visual inspection of the corresponding videos. There are a total of 13 fake scans in our data consisting of 1660 transactions. The number of randomly sampled true scan points from training data is in the order of a few hundreds. There is a large imbalance in the number of training points for the two classes. Training on such data leads to a SVM that is biased towards the class with the larger size [10]. A number of approaches have been proposed for SVM training in the presence of a large imbalance in data for the two classes. These methods either undersample the majority class [17], oversample the minority class [4] or adjust the SVM margin to reduce the bias towards majority class [16, 10].

Our use of SVM is to learn a weight for each of the feature dimensions, corresponding to a pattern, that quantifies the importance of that pattern for classification. It is desirable to have a non-negative weight for each pattern so that they can be easily interpreted. In order to avoid the data imbalance effects and to ensure positive weights, we learn a zero-bias SVM with a particular mapping of the true and fake scans into a feature space. The mapping is such that the true scan points always lie in the positive hyper-quadrant and the fake scan points in the negative hyper-quadrant. A hyper-quadrant refers to the extension of a quadrant in a 2D space to a higher dimensional space. So the positive hyper-quadrant is the set of all points with only positive coordinates and similarly for the negative hyper-quadrant. The sampled true scan points are matched with the set of patterns in the positive set to obtain a feature vector for each of those points. Only the dimensions corresponding to the patterns with exact matches are set to suitable non-zero positive values and the others are zero. Similarly the feature vectors for fake scan points are obtained by pattern matching with the negative set of patterns. However the values for the dimensions with exact matches are set to suitable negative values in this case. Fig. 2 illustrates the mapping and



2: A zero-bias SVM for classifying true scan points from fake scans. The mapping of points is such that true scan samples always lie in the positive quadrant and the fake scan samples in the negative quadrant. There are very few fake scan samples relative to the number of true scan samples. The SVM weights are all positive and quantify the relative importance of the feature dimensions (P_1 and P_2) for classification.



3: (Below) SVM weights after normalization (maximum value 1) for each random split of the available data. The learned weights for the different splits are superimposed. (Above) Averaged weights across all the random splits. The weights imply relative importance of patterns for the task of classification. Clearly, some patterns are more important than others.

the SVM geometry in a 2D space corresponding to patterns P_1 and P_2.

The available data within a training set (3 lanes) (See Section 6) is split into two equal parts randomly (approx. 250 true scans and 2-3 fake scans) with one half used for SVM training and the other half held out for testing the learned classifier. SVMs are trained with 100 different such random splits and the weights are averaged to obtain the final weights. Considering one training division of all the data, Fig. 3 shows the normalized weights after SVM training using each of the 100 random splits of the data and also the averaged weights across those splits.

The mapping into the feature space can be done in many different ways. Our experiments favored an asymmetric mapping scheme between the true and fake scan training points. In this scheme, the positive value assigned to a match between a true scan and a pattern is proportional to the frequency of the pattern. On the other hand the same negative value is assigned to all the fake scan point and pattern matches regardless of the pattern. Such a mapping causes true scan points that match to only a few of the patterns occurring less frequently to be moved more towards the origin in the feature space relative to others. Since the SVM hyperplane passes through the origin it seems that points matching to less frequent patterns are most likely to be the support vectors, which influence the final weights. This has a tendency to favor lesser number of false positives in our experiments. Table 1 shows the SVM weights obtained by averaging the learned normalized mean SVM weights across all the divisions of the data from 6 lanes into 3 training and 3 held-out lanes (See Section 6.1). Interestingly, the two idealized checkout patterns ($PBSD$ and $PSBD$) have the most significant average SVM weights among the above, implying their usefulness for classification.

5.3 Combining frequency and SVM weights

The SVM-based weights emphasize those patterns that are useful for distinguishing true and fake scan points within the training data. Interestingly, these weights also lead to an increased number of genuine fake scan detections (true positives) within the held-out data relative to using the frequency weights (Section 6). However the false positive performance of the classifier using SVM weights is worse compared to that using the frequency weights. This suggests an interesting possibility of combining the two weights such that the combined weights help achieve a better false positive performance relative to the SVM weights and better true positive performance relative to the frequency weights. The combined weights are obtained by summing the SVM and frequency weights, thereby reinforcing only those frequently occurring short patterns that are useful for classification. For convenience, these weights are referred to as *(Frequency+SVM)* weights.

6. EXPERIMENTS

6.1 Data

Video data recorded from 6 lanes of a retail store during one business day (approx. 16 hrs) of transactions was used for all the experiments reported in this work. During this period, 18 different cashiers checked out a total of about 32,969 registered items within 1660 transactions. The video data was transformed into discretized and time ordered primitives and merged with the barcode events from the TLog data. Primitive streams corresponding to transactions on 3 of the lanes were chosen for discovering patterns as described in Section 3.3. The same set of patterns were used for all the experiments reported here. The entire data was divided into two parts (3 lanes each). These two sets are referred to as *training* and *held-out* lanes respectively. All such divisions ($\binom{6}{3} = 20$) were considered and the results were averaged across the divisions. Note that the frequency weights remain the same for all the data divisions resulting in the same true and fake scan detections unlike the other two weighting schemes. Only the held-out statistics change across those divisions. Similarly for the unweighted Teiresias and all patterns (Section 6.2). The data from all the lanes was manually evaluated by six human subjects for the presence of sweethearting activities or operational errors leading to the failure of barcode registration for the corresponding items. A total of 13 such events were found.

Pattern	#Repeats across transactions	Average SVM weight
PBD	1584	0.005
$PBSD$	1302	0.593
$PDBD$	782	0.009
$PBPD$	774	0.010
$PBSPD$	734	0.127
$PDBSD$	629	0.127
$PSBD$	590	0.597
$PDBSPD$	585	0.014

1: The first eight most repeated patterns across all transactions used for pattern discovery by Teiresias. Each of those pattern's number of repeats (transactions in which they occur) and the learned average SVM weight are reported in the second and third columns respectively. $PBSD$ and $PSBD$ represent idealized item checkouts and also have the most significant average SVM weights. Other patterns represent parallel scans between successive items and scans with a missing S primitive. See text for more details.

6.2 All patterns

Our model for true scans is a set of discovered primitive patterns with different combinations of P, S and D primitives anchored around a single B primitive. The discovered set of patterns capture possible variations of the P, S and D primitives within a true scan based on a finite amount of training data. It is a natural question to ask whether the discovered set of patterns capture all such possible variations. In order to address this we designed an experiment considering all possible combinations of the P, S and D primitives around a single B primitive to model a true scan. The number of such patterns is finite since we consider patterns within a certain range of lengths (m to n) and is given by

$$\#Patterns = \sum_{i=m}^{n} i(3)^{i-1} \tag{8}$$

Considering patterns with lengths in the range 3 to 10 (see Section 3.3) results in a total of 280476 patterns. This is a large number relative to the set of 270 patterns discovered by Teiresias followed by reduction with co-occurrence analysis. Therefore a *brute force* approach considering all possible patterns is computationally expensive. However it helps quantify how much of the variation in true scans is not captured by the patterns discovered from training data.

6.3 Performance measures

We evaluate the true and fake scan detection performances using various patterns and weighting schemes on held out data using two measures: true positive recall rate and false positive detection rate. True positive recall rate for a set of lanes is defined as the total number of detected genuine fake scans relative to the total number of them found by human evaluators. The false positive rate across a set of lanes is defined as the total number of detected fake scans as a fraction of the total number of detected true scans on those lanes. The modified definitions above allow us to see the differences between different methods given the rarity of genuine fake scans in the data and the variations in the number of true scan detections across methods.

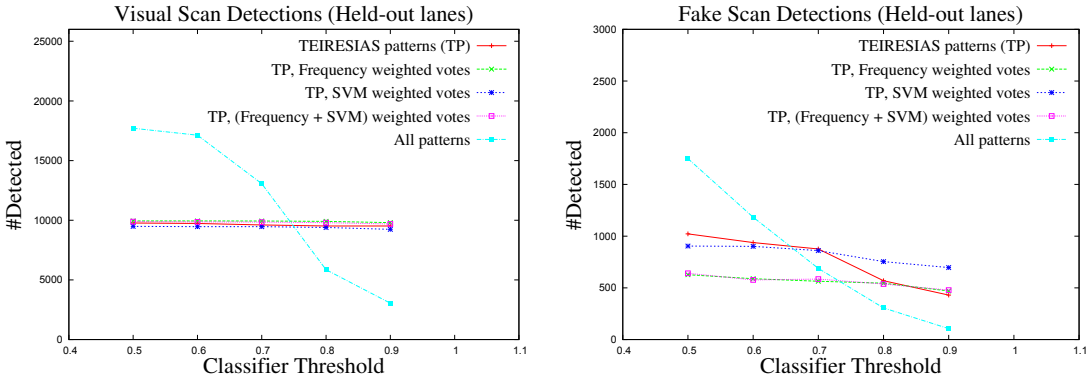
6.4 Results

As described in Section 4.1, each primitive in a transaction stream is assigned a true or fake scan label based on whether its maximum posterior likelihood for the corresponding class exceeds a certain threshold. We vary this threshold value from 0.5 to 0.9 in steps of 0.1 and plot the true and fake scan detection performances. The number of detected true (or visual) and fake scans for data from the held out lanes of one particular division of all the data is plotted in Fig. 4. In addition to the above, the detection of genuine fake scans is an important measure of performance. Fig. 5a shows the average true positive recall rate for the held out lanes across all the 20 divisions of the entire data along with the standard errors. The average false positive rates and their standard errors for the held out lanes across all the training/held-out divisions is plotted in Fig 5b.

7. DISCUSSION

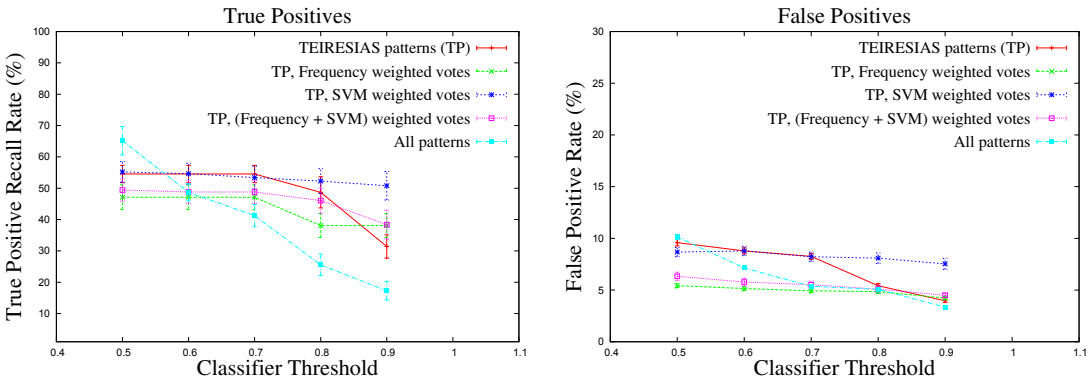
Based on the plot of Fig. 4a, we see that the true (or visual) scan detection using Teiresias patterns is quite stable across the entire range of classifier thresholds. This implies that the posterior probabilities resulting from the pattern matching are high for the true class primitives regardless of the weighting scheme used. Using all patterns leads to increased uncertainty in assigning a primitive to either of the two classes as is evident from the drastic reduction in the number of both true and fake scan detections with increasing classifier threshold. This suggests that using more patterns to model cashier checkout activities is not always beneficial with the proposed pattern matching and classification scheme but learning from data is helpful.

From Fig. 4b, the false positive detection rate decreases with increasing classifier threshold in the case of unweighted Teiresias patterns and all patterns but remains relatively constant using any of the weighting schemes. This suggests that using weighted votes, the confidence in classifying a primitive into the fake scan class is usually higher as implied by the posterior likelihoods. Among all the weighting schemes, SVM weights lead to an increased false positive rate (7.5% to 8.8%) for all classifier threshold values. The other two weighting schemes have a false positive rate within about 4.0% to 6.3%. The range is much larger, between 3.3% to 10.1% for the unweighted Teiresias and all patterns. A similar analysis of the results (Table 2) from a state-of-the-art video-based surveillance system [7, 8] on the same dataset resulted in an average false positive rate of 4.4% with a standard error of 0.12%. Note that similar to our case of frequency weighted votes, there is no true notion of held out data in this case as the detections remain the same for all the divisions.



(a) True (Visual) Scan Detections (b) Fake Scan Detections

4: True (visual) and fake scan detection performances on transactions of the held-out lanes of one particular division of the entire data into training/held-out sets. The total #detections are plotted separately for the true (left) and fake(right) scans as a function of the classification threshold. The various curves are for results using the Teiresias patterns without any weighting or the same patterns using different weighting schemes (See Section 5). A curve for the brute force method of considering all possible patterns is also plotted (See Section 6.2). The total number of successfully registered items is 18,357.



(a) True Positive Recall Rate (b) False Positive Rate

5: Average true positive recall rate (left) as a function of classifier threshold for the various methods (weighted, unweighted Teiresias patterns and all patterns). True positive rate is defined to be the fraction of all the human annotated fake scans detected by the particular method. The curves show the averaged rates across all the held-out sets of the 20 divisions of the entire data into training and held-out sets. Similarly, the average false positive rate (right) across the held-out lanes for the different pattern sets and weighting schemes is plotted. False positive rate is the number of detected false positives as a fraction of the total number of detected true scans. The vertical bars in the above plots are the standard errors of the corresponding means.

Lane	1	2	3	4	5	6
#Visual scans	2497	2186	4463	3195	3533	4244
#Fake scans (FS)	67	176	209	145	114	165
#True positives	1	1	1	1	2	2
#Ground truth FS	3	2	1	1	4	2

2: Detection statistics of the video-based system across the 6 lanes. The number of manually detected or ground truth fake scans (FS) for each lane are also shown in the bottom row.

From the plot of Fig. 5a, we observe that the SVM weighting scheme helps achieve the best true positive detection performance across the entire range of classifier thresholds (50.8% to 55.2%). Using unweighted Teiresias patterns results in a similar performance (54.5%) but only for lower values of the classifier threshold. On the other hand, the frequency weighting scheme results in a relatively smaller average true positive recall rate (38.1% to 47.1%). Combining the two weighting schemes in an additive fashion results in an average true positive recall rate ranging between 38.3% to 49.4%. The corresponding statistic for the video-based surveillance system is 63.6% with a standard error of 3.01%. We would like to point out that the video-based system uses the time duration of primitives in more ways than is being made use of in our approach.

8. CONCLUSIONS

In this paper we address the problem of detecting fake item checkouts by cashiers in retail stores. A new detection framework is proposed by merging data from video and TLog streams recorded at the POS. Using an efficient pattern discovery technique called Teiresias, we learn variable patterns of cashier checkout activity and use them to build simple complementary models for true and fake item checkouts. These models are employed in a pattern matching and classification scheme to detect fake scans robustly in the presence of noise in either video or TLog data. We experiment with a few different weighted voting schemes for the discovered patterns and compare the true and false positive detection performances. Our results suggest that a pattern frequency based weighting scheme leads to reduced false positive rates while a SVM weighting scheme results in higher true positive detection rates. A hybrid scheme combining the two weights in an additive manner helps achieve a trade-off between the two.

We compared the performance of our system with a state-of-the-art video surveillance system and obtained similar performances in terms of false positive detections. This is achieved without making use of the time information of the primitives in any systematic way as is already being done by the video-based system. This is perhaps also the reason for a slightly inferior true positive detection rate in our case relative to theirs. Our future work will include utilizing this information in various stages of our algorithm. We also intend to deploy the proposed system in a few retail stores for checkout lane monitoring.

9. ACKNOWLEDGMENTS

We would like to thank Sachiko Miyazawa, Jiyan Pan and Juan Moreno for their efforts in annotating the video data.

10. REFERENCES

[1] http://www.stoplift.com.

[2] http://www.agilenceinc.com.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[5] Y. L. Cun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2*, pages 598–605, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[7] Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, and A. Hampapur. Recognition of repetitive sequential human activity. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:943–950, 2009.

[8] Q. Fan, A. Yanagawa, R. Bobbitt, Y. Zhai, R. Kjeldsen, S. Pankanti, and A. Hampapur. Detecting sweethearting in retail surveillance videos. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '09, pages 1449–1452. IEEE Computer Society, 2009.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[10] T. Imam, K. Ting, and J. Kamruzzaman. z-svm: An svm for improved classification of imbalanced data. *AI 2006: Advances in Artificial Intelligence*, 4304:264–273, 2006.

[11] J. R. Kender, M. L. Hill, A. P. Natsev, J. R. Smith, and L. Xie. Video genetics: a case study from youtube. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 1253–1258, New York, NY, USA, 2010. ACM.

[12] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm [published erratum appears in Bioinformatics 1998;14(2):229]. *Bioinformatics*, 14(1):55–67, February 1998.

[13] I. Rigoutsos and T. Huynh. Chung-kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (spam). In *First Conference on Email and Anti-Spam*, 2004.

[14] H. Trinh, Q. Fan, S. Pankanti, P. Gabbur, J. Pan, and S. Miyazawa. Detecting human activities in retail surveillance using hierarchical finite state machine. In *ICASSP*, 2011.

[15] V. Vapnik. *Statistical Learning Theory*. Wiley Interscience, 1998.

[16] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60, 1999.

[17] F. Vilarino, P. Spyridonos, P. Radeva, and J. Vitria. Experiments with svm and stratified sampling with an imbalanced problem: Detection of intestinal contractions. *Lecture Notes in Computer Science*, 3687:783–791, 2005.

[18] A. Wespi, M. Dacier, and H. Debar. Intrusion detection using variable-length audit trail patterns. In *In Proceedings of the 2000 Recent Advances in Intrusion Detection*, pages 110–129. Springer-Verlag, 2000.