SensorSift: Balancing Sensor Data Privacy and Utility in Automated Face Understanding

Miro Enev*, Jaeyeon Jung**, Liefeng Bo*, Xiaofeng Ren*, and Tadayoshi Kohno*

*Department of Computer Science and Engineering, University of Washington
**Microsoft Research, Redmond WA

ABSTRACT

We introduce *SensorSift*, a new theoretical scheme for balancing utility and privacy in smart sensor applications. At the heart of our contribution is an algorithm which transforms raw sensor data into a 'sifted' representation which minimizes exposure of user defined *private* attributes while maximally exposing application-requested *public* attributes. We envision multiple applications using the same platform, and requesting access to public attributes explicitly *not* known at the time of the platform creation. Support for future-defined public attributes, while still preserving the defined privacy of the private attributes, is a central challenge that we tackle.

To evaluate our approach, we apply SensorSift to the PubFig dataset of celebrity face images, and study how well we can simultaneously hide and reveal various policy combinations of face attributes using machine classifiers.

We find that as long as the public and private attributes are not significantly correlated, it is possible to generate a sifting transformation which reduces private attribute inferences to random guessing while maximally retaining classifier accuracy of public attributes relative to raw data (average *PubLoss* = .053 and *PrivLoss* = .075, see Figure 4). In addition, our sifting transformations led to consistent classification performance when evaluated using a set of five modern machine learning methods (linear SVM, kNearest Neighbors, Random Forests, kernel SVM, and Neural Nets).

Categories and Subject Descriptors

K.4 [**Computers and Society**]: Public Policy Issues—*Privacy*; I.2 [**Artificial Intelligence**]: Vision and Scene Understanding—*Modeling and recovery of physical attributes*; I.5 [**Pattern Recognition**]: Models—*Statistical, Neural Nets*; G.1 [**Numerical Analysis**]: Optimization—*Least squares methods*

1. INTRODUCTION

The minimal costs of digital sensors, global connectivity, computer cycles, in addition to advances in machine learning algorithms, have made our world increasingly visible to intelligent computers. The synergy of sensing and AI has unlocked exciting new research horizons and led to qualitative improvements in human-computer interaction. However, alongside these positive developments, novel privacy threats are emerging as digital traces of our lives are harvested by 3rd parties with significant analytical resources. As a result, there is a growing tension between utility and privacy in emerging smart sensor ecosystems.

Table 1: Data sharing models in sensor applications (all terms defined in Section 3).

	Platform	Application	Tradeoffs
M1	sensor data	get features, classify, app. logic	Innovation++ Privacy-
M2	sensor data, get features	classify, app. logic	Innovation- Privacy++
S.Sift	sensor data, sift gen., verify	classify, app. logic	Innovation+ Privacy+

In the present paper we seek to provide a new direction for balancing privacy and utility in smart sensor applications. We are motivated towards this goal by the limitations in the current models of data access in smart sensing applications.

At present there are two conventional modes of data sharing in smart sensing applications and they are either risk carrying or arbitrarily stifling:

In the first mode, an application is given access to all of the raw data produced by a sensor (i.e., the Kinect); the application is then free to do feature extraction, classification, and run the logic that powers its functionality. Although this model of data sharing is great for innovation it leads to a sacrifice in privacy (Table 1, first row M1).

In the second mode, an application is given access to some restricted set of API calls defined by the platform (i.e., Apple's iOS) which restrict access to the raw data produced by a sensor; the application can still perform classification and run its logic, however it no longer has direct access to the data. The benefit of this approach is that privacy can be significantly increased, however innovation is significantly diminished (Table 1, second row M2). Given the limitations of these interaction modes, we seek to find a new model of sensor data access which balances application innovation and user privacy. To this end we develop an information processing scheme called SensorSift which allows users to specify their privacy goals by labeling attributes as private and subsequently enabling applications to use privacy-preserving data access functions called 'sifts' which reveal some non-sensitive (public) attributes from the data (Table 1, third row S.Sift).

Our tool is designed to be understandable and customizable by consumers while defending them from emerging privacy threats based on automated machine inferences. At the same time this tool enables applications access to non-private data in a flexible fash-

ACSAC '12 Dec. 3-7, 2012, Orlando, Florida USA

ion which supports developer innovation. Importantly, while the private attributes must be chosen from a supported list (to enable data protection assurances) the public attributes requested by applications do not need to be known in advance by the SensorSift platform and can be created to meet changing developer demands.

Rather than developing a specific system instance, in this paper we tackle the challenge of protecting sensitive data aspects while exposing non-sensitive aspects. We overcome this challenge by introducing a novel algorithm to balance utility and privacy in sensor data and propose how to embed it in an information processing scheme which could be applied as part of a multi-application trusted platform.

Towards Privacy and Flexibility in Sensor Systems. Suppose that an application running on a camera-enabled entertainment system (like the Kinect) wishes to determine Alice's gender to personalize her avatar's virtual appearance. Suppose also that Alice (the user) has specified that race information should not be available to applications. At present, Alice can either avoid using the application (and thus sacrifice utility) or choose to use the application and forfeit her ability to ensure privacy.

A natural solution to this tension would be to allow data access which is based on pre-defined public and private attributes. While workable for well-known attributes like race and gender, this approach limits innovation as developers are restricted to the predefined public attributes. Under the SensorSift scheme, applications can opt to use standard public and private attributes or can propose novel public attributes not known by the platform in advance (private attributes are still defined by the system in advance and exposed to users as options).

Returning to our example, on a SensorSift supporting platform Alice can specify race as a private attribute. The system would then transform the raw camera data samples to adhere to this policy by maximally removing race information while exposing application-desired attributes. These public attributes could be anything defined by the developers — including attributes not known to the platform designers; for simplicity of exposition, however, we'll use gender as the public attribute.

The transformed sensor data would only be made available to the application if the system successfully verifies (using an ensemble of state-of-the-art classifiers) that the sifted data cannot be used to recognize the private attribute significantly beyond random guessing. If the sift is verified, the target application would receive the transformed data which could then be post-processed to infer the gender value.

Concept Overview. Given a particular sensor context (e.g., optical/image data) and fixed set of data features (e.g., RGB pixel values) the information flow through our scheme is as follows: users define private attributes and applications define (request) public attributes; developers use provided tools to generate a candidate transformation (sift) which attempts to expose the [arbitrarily chosen] public attribute(s) but not the specified private aspects of the data; the user's system checks the quality of the proposed sift using an ensemble of classifiers; and lastly, if the verification is successful the application is allowed access to the transformed data.

Typically we expect that the SensorSift platform will ship with many valid sifts that cater to standard application demands. More importantly, however, we offer support for application-supplied sift transformations which would be verified by the platform either at installation time or when the user changes his or her privacy preferences. Once a particular sift has been invoked and successfully verified it will be applied to each sensor data release. In the case where an application is using a known policy (standard public and private attributes) the platform can automate classification and simply release an attribute label. Alternatively, if the application needs access to a novel public attribute it will need to independently classify the sifted data it receives.

Evaluation and Results. To evaluate our approach, we test how well we can control the exposure of facial attributes in the PubFig database of online celebrity photographs [11]. We leverage the face image attribute scheme of Kumar et al. to provide a quantitative vocabulary through which privacy policies can be defined over facial features [10]. We then choose a set of 90 policies composed using 10 facial attributes (e.g., male [gender], attractive woman, white [race], youth [age], smiling, frowning, no eyewear, obstructed forehead, no beard, outdoors) and show that it is possible to successfully create data 'sifts' which remove selective facial characteristics in a discriminating manner to produce high classification accuracy for public attributes and low accuracy for private attributes (average *PubLoss* = .053 and *PrivLoss* = .075, see Figure 4).

In addition, our sifting transformations lead to consistent classification performance when evaluated using a set of five modern machine learning methods (linear SVM, kNearest Neighbors, Random Forests, kernel SVM, and Feed Forward Neural Networks). As an extension we also show that our approach maintains privacy when applied to complex policies (multiple public and/or private attributes) as well as dynamic video (i.e., sequences of data releases).

To our knowledge our approach is the first solution to verifiably decompose face images into sensitive and non-sensitive features when evaluated against state of the art machine classifiers. In addition, our framework enables on-demand computation and evaluation of sifting functions so that privacy and utility balance can be created for currently unknown but desired attributes (thus supporting application innovation).

2. THREAT AND USAGE MODELS

Smart sensing applications have already been adopted in numerous life-improving sectors such as health, entertainment, and social engagement [2, 9]. Driven by the diminishing costs of digital sensors, growth of computational power, and algorithmic advances even richer sensing applications are on the horizon [1]. In most instances, smart sensor applications create rewarding experiences and assistive services, however, the gathered raw data also presents significant privacy risks given the amount of personal information which modern algorithms can infer about an individual. The potential consequences of these risks are not fully understood given the novelty of the enabling technologies. Nonetheless, we feel that it is critical to develop ways of managing the information exposure in smart sensor applications preemptively rather than reactively.

To mitigate potential privacy threats posed by automated reasoning applications we propose to employ automated defenses. At a high level our scheme is intended to enable a quantitatively verifiable trade off between privacy and utility in participatory smart application contexts. A full description of SensorSift is provided in Section 3, yet intuitively, our goal is to create a trusted clearinghouse for data which transforms raw sensor captures into a sifted (or sanitized) form to maximally fulfill the privacy and utility demands in policies composed of user selected private attributes and application requested public attributes.

We envision a model in which applications are untrustworthy but, in general, not colluding (we discuss collusion in Section 9). Applications might be malicious and explicitly targeting the exposure of private attributes; more likely, however, they are well-intentioned applications that fail to adequately protect the data that

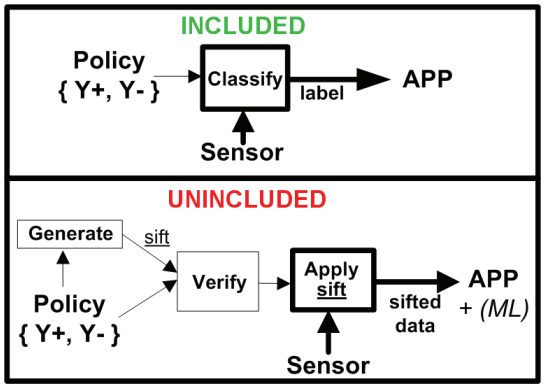

Figure 1: The two modes of operation in the SensorSift framework. Thick lines represent the processing elements that are occurring with each sensor release, while the thin lines indicate operations that are only necessary at application installation or policy revisions.

they harvest. We do not wish to expose private attributes to well-intentioned but possibly weak/insecure applications since those applications might accidentally expose the private information to other third parties. We also define as out of scope the protection of private attributes from adversaries with auxiliary information that might also compromise those private attributes. For example, we cannot protect the privacy of the gender attribute if an application asks for user gender during setup, and if the user supplies that information. We return to a discussion of these limitations in Section 9.

3. SYSTEM DESCRIPTION

While there are numerous potential deployment environments we primarily envision SensorSift running as a service on a trusted multi-application platform/system (like the Kinect) on which applications run locally.

Recall that the goal of the framework is to allow users to specify private attributes and allow applications to request non-private attributes of their choosing. At application install time (or when privacy settings are changed), the user and application declare their respective privacy and utility goals by creating a *policy* which contains the user desired private attribute(s) Y^- and application requested public attribute(s) Y^+. The user selected private attributes must always be known by the system to ensure that they can be verifiably protected; thus, the only viable *private attributes* are those for which the system's verification database has labels. Conversely, applications can request access to non-private (public) attributes which are unknown to the platform (i.e., developer invented). This makes it possible for the system to be one of two operating modes – included or unincluded policy mode.

In included mode (the simpler case), the user chosen private attribute(s) Y^- and the application requested public attribute(s) Y^+ compose a verified policy for which a data processing method is included in the platform. This means that the policy has been previously checked to ensure that the public attribute(s) do not leak information about the private attribute(s), and in addition, the platform has (shipped, or has been updated to include) a trained classification model which can recognize the public attribute(s) from the raw sensor data. As a result, it is possible to simply output the trained classifier's judgment on the public attribute to the application (as a text label) for each sensor sample request (Figure 1 top panel). From the application's perspective this is a straightforward way to get access to the public attribute(s) in the sensor data as all of the inference (pre-processing and classification) traditionally done by application logic is handled by the platform. We expect that many applications will opt to operate in this mode, especially if the list of platform included policies is large and frequently updated.

In some cases, the included list of attributes may not be sufficient to enable the application developers' functionality and utility goals. Whenever this is the case, the application interacts with the platform in unincluded policy mode (Figure 1 bottom panel). In this mode the user has selected some private attribute(s) Y^- (e.g., age) and the application is requesting access to some *novel* public attribute(s) Y^+ (e.g., imagine that eye color is a novel public attribute). Since support for this new policy is not included by the platform it is up to the application to provide a candidate sift (or data access function F) which can be applied to sensor data to balance the removal of private attribute information with the retention of application desired non-sensitive (public) data features. The proposed sift will only be allowed to operate on the sensor data if it can be successfully verified to not expose information about the private attribute(s) specified in the policy. While this scenario is more challenging from an application perspective, it is also more flexible and offers a way to meet the rapidly evolving demands of software developers.

Below we focus our discussion on the usage model for unincluded policies as it is unique to our approach and highlights all of the SensorSift framework's subcomponents.

Sift Generation. To create a candidate sifting function for an unincluded policy, applications can use our PPLS algorithm (defined in Section 4), develop their own method, or potentially use preverified sifts (e.g., crowd sourcing repositories). Code and documentation for the PPLS sifting generating function are freely available at `http://homes.cs.washington.edu/~miro/sensorSift` .

To use the PPLS algorithm, developers need to provide a dataset of sensor data (e.g., face images in our experiments) with binary labels for the public and private attributes. To facilitate the generation of this prerequisite labeled dataset, we imagine that developers will leverage freely available data repositories or use services such as Mechanical Turk.

Sift Verification. Once a candidate sift function has been provided to the platform, SensorSift must ensure that the proposed transformation function does not violate the user's privacy preferences. Indeed, there is no guarantee that a malicious application developer did not construct a sifting transformation function explicitly designed to violate a user's privacy. To verify that the transformation is privacy-preserving, SensorSift will invoke an ensemble of classifiers *ML* on the sifted outputs of an internal database *DB* to ensure that private attributes cannot be reliably inferred by the candidate sift. We discuss these components in more detail below.

Verification: Internal Dataset. The basis upon which we verify privacy assurances is a DB_{verify} dataset of sensor samples (i.e. face images) which would be distributed with each SensorSift install. For our purposes, we assume that the dataset is in matrix format X with n rows and d columns, where n is the number of unique samples (i.e., face images), and d is the dimensionality of each sample (i.e., face features). Large datasets with higher feature dimensionality offer attractive targets since they are more likely to capture real world diversity and produce stronger privacy assurances.

Verification: Classifier Ensemble. The second part of the verification process applies the candidate sift transformation to each sample in the internal database. Next an ensemble of machine classifiers are trained (using a training subset of the internal database) to recognize the private attributes with the sifted data. We leverage

state of the art methods which represent the most popular flavors of mathematical machinery available for classifcation including: a clustering classifier (**k-nearest neighbor** — parameters: q = 9, using euclidean distance metric with majority rule tie break; classifier source: MATLAB knnclassify), linear and non-linear hyperplane boundary classifiers (**linear-SVM** — soft margin penalty C = 10; classifier source: liblinear 1.8; **kernel-SVM** — soft margin penalty C = 10, radial basis kernel function, no shrinking heuristics; classifier source libsvm 3.1), a biologically inspired connection based non-linear classifier (**feedforward neural network** — 100 hidden layer neurons using a hyperbolic tangent sigmoid transfer function trained using gradient-descent backpropagation evaluated using mean squared normalized error, classifier source: MATLAB nnet package), and a recursive partitioning classifier (**random forest** — number of random trees per model = 500; classifier source: `http://code.google.com/p/randomforest-matlab/`).

For each *ML* model, independent training rounds are performed to obtain classifiers optimized for sifts of specific dimensions. A testing subset of the database is then used to evaluate how well the private attribute can be classified after it has been transformed by the proposed sift.

If any of the classifiers can detect the presence and absence of the private attribute(s) with rates significantly above the platform's safety threshold (e.g., 10% better than chance) the sift is rejected because it exposes private information. Alternatively if the private attribute accuracies on the sifted data (from the internal databse) are below the safety threshold the sift is deemed to be safe.

We again stress that while it is important for developers (or their applications) to evaluate the resulting accuracies on both public and private attributes, the system deploying SensorSift would in fact only verify that the private attribute classification accuracy is small.

Sift Application. If a sift has been proposed and successfully verified, it needs to be continuously applied with each data request made by the application. The application itself cannot apply the sifting transformation directly; this is requisite since, if the application had access to the raw sensor data it could be exfiltrated in violation of the privacy goals. Instead, the SensorSift applies the verified sifting transformations and outputs only the transformed data to the application.

Sift Post-Processing. In contrast to included mode where attribute labels are directly provided to the application, the application must post-process the sifted outputs (numerical vectors) that it receives in unincluded mode in order to determine the public attribute. This will likely involve running a classifier on the sifted sensor samples – the classifier can be trained using the database used to generate the sifts; once trained the classifier overhead should be minimal.

4. SIFT ALGORITHM - PPLS

In this work we create sifts using a novel extension of Partial Least Squares (PLS) that we call **Privacy Partial Least Squares**, or **PPLS**. At the heart of our technique is the long standing approach of using correlation as a surrogate for information. Given this perspective we design an objective function which simultaneously aims to maximize the correlations with public attributes and minimizes those with private attributes (while performing the structural projection of PLS). As we later show, this correlation-based PPLS algorithm is easy to use and also very effective within the context of automated face understanding; since our algorithm is domain independent we believe that PPLS is well suited to various datasets but this has not yet been verified.

Intuitively, our approach uses correlation between data features and attribute labels to find 'safe regions' in feature space which

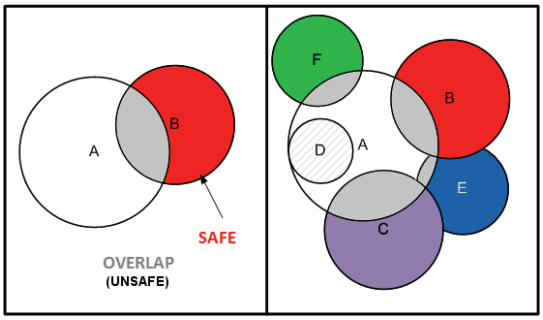

Figure 2: The left panel shows a simplified configuration of feature sets for two distinct attributes A (private) and B (public). The goal of SensorSift is to find the region(s) in feature space which are in the public feature set but not in the private one (i.e. indicated with the color red in the left panel). Raw data can be then re-represented in terms of how strongly it maps to this privacy aware region of the feature space. The right panel depicts how additional public attributes (C-F) which are invented by application developers map onto the feature space of our example. Note that in many cases it is possible to find privacy respecting regions of the region space through which to re-interpret (sift) raw data, however in some instances (attribute D) it may not be possible to separate attributes which have strong causal/correlation relationships (i.e. left eye color from right eye color).

are strongly representative of public but not private attributes (Figure 2). We then project raw sensor data onto these safe regions and call the result of the projection a loading or 'sift' vector.

Privacy Partial Least Squares Now that we have explored the intuition behind our approach we turn our attention to the details. To reiterate, Partial Least Squares (PLS) is a supervised technique for feature transform or dimension reduction [17]: given a set of observable variables (raw features) and predictor variables (attributes), PLS searches for a set of components (called latent vectors) that perform a *simultaneous* decomposition of the observable and predictor variables intended to maximize their mutual covariance. PLS is particularly suited to problem instances where the dimensionality of the observed variables is large compared to the number of predictor variables (this is generally true for rich sensor streams).

Let X be $[x_1,\cdots,x_{d_x}]^T$ a $n \times d_x$ matrix of observable variables (input features), and $Y = [y_1,\cdots,y_{d_y}]$ a $n \times d_y$ a matrix of predictor variables (attributes), where n is the number of training samples, d_x is the dimension of input features, d_y is the dimension/number of attributes. Without loss of generality, X, Y are assumed to be random variables with zero mean and unit variance. Any unit vector w specifies a projection direction and transforms a feature vector x to $w^T x$. In matrix notation, this transforms X to Xw.The sum of the covariances between the transformed features Xw and the attributes Y can be computed as

$$\mathrm{cov}(Xw,Y)^2 = w^\top X^\top Y Y^\top X w \tag{1}$$

The PLS algorithm computes the best projection w that maximizes the covariance:

$$\begin{aligned} find \quad & \max_w \left[\mathrm{cov}(Xw,Y)^2\right] \\ s.t. \quad & w\top w = 1 \end{aligned} \tag{2}$$

We propose a novel variant of PLS, *Privacy Partial Least Squares* (PPLS), that handles both public attributes and private attributes. Let $Y^+ = [y_1^+,\cdots,y_{d^+}^+]$ be a $n \times d^+$ public attribute matrix, and $Y^- = [y_1^-,\cdots,y_{d^-}^-]$ a $n \times d^+$ private attribute matrix, where d^+ is

Algorithm 1 Privacy Partial Least Squares

1. Set $j = 0$ and cross-product $S_j = X^\top Y^+$
2. if $j > 0$, $S_j = S_{j-1} - P(P^\top P)^{-1} P^\top S_{j-1}$
3. Compute the largest eigenvector w_j: $\left[S_j^\top S_j - X^\top Y^- (Y^-)^\top X\right] w_j = \lambda w_j$
4. Compute $p_j = \frac{X^\top X w_j}{w_j^\top X^\top X w_j}$
5. If $j = k$, stop; otherwise let $P = [p_0, \cdots, p_j]$ and $j = j+1$ and go back to step 2

the number of public attributes and d^- is the number of private attributes. We want to find a projection direction w that both maximizes the covariance $cov(Xw, Y^+)$ and minimizes $cov(Xw, Y^-)$.

This is achieved by optimizing the difference of covariances:

$$find \quad \max_w \left[cov(Xw, Y^+)^2 - \lambda * cov(Xw, Y^-)^2\right] \qquad (3)$$
$$s.t. \quad w^\top w = 1$$

The flow of the PPLS algorithm is outlined in the algorithm box (Algorithm 1). To transform X to more than one dimensions, we follow the PLS approach and develop a sequential scheme: we iteratively apply Equation 3, subtracting away covariances that are already captured in the existing dimensions (Step 2 in the Algorithm). Note that we only remove covariances from $cov(Xw, Y^+)$ but not $cov(Xw, Y^-)$, to ensure that every included dimension w is privacy-perserving by itself for all private attributes.

Free Parameters. There are two key free parameters of the PPLS algorithm, a λ term (privacy emphasis) and the number of sift dimensions to release K. In general we only release several dimensions from these sift vectors as a type of dimensionality reduction step which minimizes the risk of reconstruction. Despite the small size of the outputs sifts we find that public attributes can be correctly inferred with minimal accuracy degradation 7.

The λ term in Equation 3, represents the relative importance of privacy with higher λ values indicating an increased emphasis on removing private features (with a possible loss to utility).

5. DATASET

Our evaluation is based on the the Public Figures Face Database (PubFig) [11] which is a set of 58,797 images of 200 people (primarily Hollywood celebrities) made available as a list of URLs (see `http://www.cs.columbia.edu/CAVE/databases/pubfig/download`). The PubFig images are taken in uncontrolled situations with non-cooperative subjects and as a result there is significant variation in pose, lighting, expression, scene, camera, imaging conditions and parameters. Due to the size and real-world variation in the PubFig dataset we felt that it presents an appropriate foundation on which to evaluate SensorSift.

Validation, Alignment, and Rescaling. We began by downloading the PubFig image URLs using an automated script which would keep track of broken links and corrupted images. At the time of our data collection we found 45,135 valid URLS (77% of the advertised 58,797 images). For each image in the database PubFig provides four pixel coordinates which define the face region; we extracted this face region for each image aligned it to front-center (via affine rotation using the roll parameter). Next we rescaled each image to 128x128 pixels using bicubic interpolation and antialiasing.

Feature Extraction and Normalization. In addition to the raw RGB pixels, we extracted image derivatives of each face image to enrich the feature space of the raw data and provide a larger starting dimensionality to our algorithm. The four features we computed are popular in the computer vision literature and include raw RGB, image intensity, edge orientation, and edge magnitude [12]. After computing these transforms, we apply an energy normalization $(x-\mu)/(2\cdot\sigma)$ to the feature values of each face to remove outliers. Lastly, we concatenate all of the normalized image features for into a row vector and create a matrix to hold the entire dataset (45,135 rows/faces and 98304 columns/features per face).

PCA Compression. Next, we compute a PCA compression which is applied to the entire database (10:1 compaction ratio, $> 95\%$ energy maintained) to decrease the feature dimensionality of our face database and enable the PPLS algorithm to operate within reasonable memory constraints (16GB per node).

6. EXPERIMENTS AND METRICS

The privacy sifts that we compute are intended to provide quantitative assurances which adhere to a specified policy. Policies in turn are based on a set of user declared private attributes and developer requested public attributes. In this section we describe how we selected the attributes to include in the polices we evaluate. In addition we describe the metrics used to evaluate the quality of the sift generated for a particular policy.

6.1 Attribute Selection

The authors of the PubFig database were interested in providing a large vocabulary of attributes over each image to power a text-based 'face search engine' [10] Thus in addition to face coordinates and rotation parameters, each image in the PubFig dataset is annotated with classification scores for 74 different attributes. These scores are numerical judgments produced by a set of machine classifiers each trained for a unique attribute.

For analytical tractability we were interested in reducing the set of 74 available attributes to a more manageable number. Since we are using correlation as a proxy for information in our PPLS algorithm we analyzed the correlations between the available attributes to get a sense for the redundancy in the data.

We found two large clusters of attributes which were centered around *'Male'* and *'Attractive Female'*. The 'Male' attribute was very closely correlated with the attributes: *'Sideburns'*, *'5 oClock Shadow'*, *'Bushy Eyebrows'*, *'Goatee'*, *'Mustache'*, *'Square Face'*, *'Receding Hairline'*, and *'Middle Aged'*. Conversely, *'Attractive Female'* was very closely related to: *'Wearing Lipstick'*, *'Heavy Makeup'*, *'Wearing Necklace'*, *'Wearing Earrings'*, *'No Beard'*, and *'Youth'*.

Given their strong connection to a large set of the available attributes the *'Male'* and *'Attractive Female'* attributes were clear choices for our analysis, however we wanted to also get coverage over other characteristics which might be interesting from a privacy perspective. To this end we chose race (*'White'*), age (*'Youth'*), and emotional indicators(*'Smiling'*, *'Frowning'*), as well as other attributes which were descriptive about distinct regions of the face (*'No Eyewear,' 'Obstructed Forehead,' 'No Beard'*). Lastly we chose the 'Outdoors' attribute as it provides environmental context and it brings our total up to 10.

Policies. Having chosen a base set of 10 attributes we set out to evaluate how different choices of public and private attributes

would impact our goal of balancing utility with privacy. To this end we created 90 simple policies composed of all possible combinations of a single public and a single private attribute (e.g., public:*'Male'*, private:*'Smiling'*)[1].

6.2 Defining Mask Performance: PubLoss and PrivLoss

As previously stated, our system aims to produce data transformations which provide a favorable balance between utility and privacy given a policy instance P, dataset X, and attribute labels Y. Building on these concepts, we now introduce the quantitative measurements *PubLoss* and *PrivLoss* which judge the utility and privacy [respectively] achieved for sifts of a specified dimension within a given policy. *PubLoss* is intended to measure how much classification accuracy is sacrificed when public attributes are sifted (relative to their raw, unsifted versions), while *PrivLoss* is the difference between the highest classification rate of sifted private attributes relative to blind guessing.

- **PubLoss:** Decrease in F sifted public attribute classification accuracy relative to the achievable accuracy using raw (unsifted) data, computed as:

$$PubLoss = ML_m(X, Y^+) - ML_m(F_{Y^+,Y^-}(X,K), Y^+)$$

- **PrivLoss:** F sifted private attribute classification accuracy relative to chance, computed as:

$$PrivLoss = ML_m(F_{Y^+,Y^-}(X,K), Y^-) - .5$$

Where $ML_m(X,Y)$ denotes the Class Avg. Accuracy (Section 6.3) computed via classifier m using a 50%-50% split of training vs testing instances given data samples X with ground truth labels Y; and, $F_{c,d}(X,K)$ indicates the K dimensional privacy sift computed using data samples X and public and private labels Y^+ and Y^-.

A poor quality F would yield transformed samples whose public attributes are unintelligible and whose private attributes are easily identified (high *PubLoss* and *PrivLoss*). Conversely, an ideal sifting transformation would have no impact on the raw classification rates of public attributes while completely obscuring private attributes (no *PubLoss* and *PrivLoss*).

6.3 Classification Measures

The performance criteria we have selected (*PubLoss* and *PrivLoss*) are heavily dependent on measures of classification accuracy. Thus to provide stronger privacy claims, we now describe a robust approach to computing classification accuracy.

Class Average Accuracy. A common method of reporting classification accuracy is based on the notion of *aggregate accuracy* shown in Eq (4). Although this metric is suitable to many problem instances, whenever attributes have unequal distributions of positive vs negative samples (e.g., 78% of faces in our dataset lack eyewear) classifiers can achieve high *aggregate accuracy* scores by exploiting the underlying statistics (and always guessing 'no eyewear') rather than learning a decision boundary from training data. To avoid scores which mask poor classifier performance and warp our *PubLoss* and *PrivLoss* measures we opt to use **Class Avg. Accuracy** which is a more revealing gauge of classification success and is calculated as in Eq (5):

[1]We did not consider policies where the same attribute is both public and private

Table 2: Achievable accuracy for each attribute using raw data features computed using the maximum classification score across our five classifiers. Columns one and two use the *aggregate accuracy* metric and respectively represent our attribute recognition scores and state of the art performance (ICCV09 accuracies are reported from [11]). The remaining column provides the Class Average Accuracy measure.

Attribute	ICCV09	Agg. Accuracy	Class Avg. Accuracy
Male	81.22	94.18	92.86
Attr. Female	81.13	87.33	84.26
White	91.48	88.07	86.97
Youth	85.79	83.27	79.97
Smiling	95.33	92.11	87.69
Frowning	95.47	89.98	85.35
No Eywear	93.55	87.01	82.86
Obst. Forehead	79.11	81.01	77.86
No Beard	89.53	88.60	86.13
Outdoor	–	88.18	84.83

$$AggregateAccuracy = (tP + tN)/tS \quad (4)$$

$$ClassAvgAccuracy = (tP/(tP + fP) + tN/(tN + fN))/2 \quad (5)$$

Where tP is the number of True Positives (correct identifications), fP is the number of False Positives (type 1 errors), tN is the number of True Negatives samples (correct identifications), fN is the number of False Positive samples (type 2 errors), and tS is the number of Total samples ($tP + fP + tN + fN$).

As can be seen from equation (5) above, **Class Avg. Accuracy** places equal weight on correctly identifying attribute presence (positive hit rate) and attribute absence (negative hit rate) which in turn emphasizes classifier precision and offers less sensitivity to attributes with imbalanced ratios of positive to negative data.

6.4 Achievable Accuracies

Achievable Accuracy is a term we use to refer to the correct classification rates that we were able to obtain using the PubFig dataset. As mentioned in Section 6.1, images in the PubFig dataset are annotated with 74 numerical judgments produced by a set of 74 machine classifiers each trained to recognize a unique attribute. These scores are positive whenever the classifier has determined that an attribute is present and negative if the attribute is deemed to be absent (higher absolute values indicate additional confidence)[2]. To produce these numbers each attribute classifier was trained using 2000 hand labeled (ground truth) samples produced using Mechanical Turk [11]. Unfortunately due to the liability policy of Columbia University these ground truth labels cannot be released, instead we treat the classifier outputs as a proxy ground truth.

In the first two columns of Table 2, we use the *aggregate accuracy* metric to compare attribute recognition performance of our classifiers against state of the art methods. The third column provides the more robust **Class Average Accuracy** measure which we'll be using as the basis for result discussions. Note that all of the results in Table 2 are computed raw [unsifted] data features.

In the first column of Table 2 we report the correct classification rates of our 10 attributes from the original PubFig publication. These scores are based on the notion of *aggregate accuracy* shown in in Eq (4). In the second column of Table 2 we also use the *aggregate accuracy* method however we now apply classification

[2]Each scores indicates the distance of a sample from the SVM separation hyperplane

models which we train using the features described in Section 5. This serves as a verification that we are able to match state of the art results (in fact outperform for the first two attributes). In the last column we report the more robust classification measure - class average accuracy - which we use as a reference for the PubLoss computations for the remainder of the paper.

When looking at these accuracy rates it is important to note that the results could be improved with additional data, access to ground truth labels, and novel computer vision features. However we are not seeking maximal identification accuracy; instead the achievable accuracy serves as a reference point, and we are interested in how our sifting methods operate around it.

7. RESULTS

Below we describe the results of our experiments on the PubFig dataset. First we set a conservative privacy threshold and determine the sift output dimensionality that meets this criteria when measured against our ensemble of classifiers. Next we look at the *PubLoss* and *PrivLoss* computed from the 90 policies using one public and one private attribute, and describe the factors influencing the results. We follow this with an extension of our algorithm suited to complex policies (multiple public and/or private attributes). We also discuss how our approach can be applied to sequential sensor samples (i.e. video) and provide a details from a case study. Lastly we compare our approach to the closest method in the literature.

Sift Dimensionality and Multiple Classifiers. Recall that the output of our system is a transformation which can be applied to any input feature vector (i.e., face image) to produce a sifted output intended to uphold a given policy. Our results indicate that the average (across all policies) *PubLoss* monotonically decreases while the average *PrivLoss* monotonically increases as the number of sift dimensions exposed to classifiers grows. This is reasonable since very low dimensional sifts do not carry enough information to classify public attributes while high dimensional sifts provide an increased risk of information leakage.

In our evaluation we adopt a conservative threshold, and set the acceptable *PrivLoss* to inferences that are 10% better than chance (i.e., maximal allowed private classification accuracy is 60%). Given this constrain we find that an output sift dimensionality of $K = 5$, and $\lambda = 1$ yield the best average tradeoffs across policies (with one public and one private attribute accross all tested classifiers). Figure 3 provides examples of our system's output for two policies (which use the same attributes in exchanged public/private order) in which classification accuracy is shown as a function of sift dimensionality.

From an adversarial standpoint, the output of our system represents an 'un-sifting' challenge which can be tackled with any available tool(s). In general we find that for low dimensional sifts, classifier accuracies are similar despite differences in the algorithmic machinery used for inference; however as the sift dimensionality grows the classifiers increasingly differ in performance — when we look across classifiers using the 90 simple policy combinations possible with one public and one private attribute, we find that 5 dimensional sifts have an avg. public attribute accuracy standard deviation of 3.86% and an avg. private attribute accuracy standard deviation of 3.77%; whereas 15 dimensional sifts have significantly larger deviations as avg. public attribute accuracy standard deviation is 8.25% and avg. private attribute accuracy standard deviation is 14.16%. Another interesting observation is that the linear-SVM and kernel-SVM classifiers consistently produced the lowest *PubLoss* while the linear-SVM and randomForest classifiers produced the highest *PrivLoss*. The high performance of linear-SVM is not surprising given the linear nature of our PPLS algorithm.

Policy Results. We evaluated sifts created for each of our 90 policies (10 attributes paired with all others, excluding self matches) using each of our 5 classification methods. For each policy, we report the lowest *PubLoss* and highest *PrivLoss* obtained across all 5 classifiers in Figure 4. In these matrices, the attribute enumeration used in the rows and columns is: (1) Male - *M*, (2) Attractive Female - *AF*, (3) White - *W*, (4) Youth - *Y*, (5) Smiling - *S*, (6) Frowning - *F*, (7) No Eyewear - *nE*, (8) Obstructed Forehead - *OF*, (9) No Beard - *nB*, and (10) Outdoors - *O*. Recall that the *PubLoss* results are relative to the achievable accuracies reported in the third column of Table 2.

Our results indicate that we can create sifts that provide strong privacy and minimize utility losses at ($K = 5$ dimensions) for the majority of policies we tested (average *PubLoss* = 6.49 and *PrivLoss* = 5.17). This is a significant finding which highlights the potential of policy driven privacy and utility balance in sensor contexts!

Performance Impacting Factors Based on our analysis we find that the PPLS algorithm is able to produce high performing sifts as long as there are not significant statistical interactions between the public and private attributes. This is to be expected given the structure of the problem we are trying to solve. In the extreme case, if we consider a policy which includes the same attribute in its public and private set it seems obvious that any privacy enforcing algorithm will have a hard time balancing between utility and privacy since obscuring the private attribute prevents recognition of the [same] public attribute.

To formalize the intuition above we use two quantitative measures to capture the levels of statistical interactions in policies: correlation and overlap. Correlation is the traditional statistical measure of the probabilistic dependence between two random variables (in our case attributes). Overlap is a metric we introduce to describe the degree to which two attributes occupy the same regions in feature space. Overlap is computed as in equation (6) and normalized to 1 across our 90 policies. The Correlation and Overlap matrices in Figure 4 show the correlation and overlap for each attribute pair in our tested policies.

$$find \sum\sum \max_{w} \left[cov(Xw, Y^{+})^2 * cov(Xw, Y^{-})^2 \right] \quad (6)$$
$$s.t. \quad w^{\top} w = 1$$

To help illustrate correlation and overlap we provide a set of examples from our analysis. Consider the attributes Male and No Beard. These attributes are highly correlated ($r = -.72$). Male and Attractive Female are another highly correlated attribute pair ($r = -.66$). Using our domain knowledge we can reason about these numerical dependencies as follows: if you know about the presence of facial hair (i.e., No Beard is false) then Maleness is easily predicted, similarly if an individual is an ttractive Female it is highly unlikely that they are Male.

Although correlations provide a key insight into the interactions between attributes a deeper level of understanding is obtained by investigating overlap. Returning to our examples, Male and No Beard have an overlap (.29) which is less than half of the overlap of Male and Attractive Female (.72). The reason for this is that No Beard is a relatively localized attribute (i.e., pixels around around the mouth/chin) and does not depend on features in many of the regions used to determine Male-ness. Conversely, Attractive Female and Male have high overlap because they are determined using many of the same feature regions (i.e., eyebrows, nose, bangs) as can be seen in Figure 5.

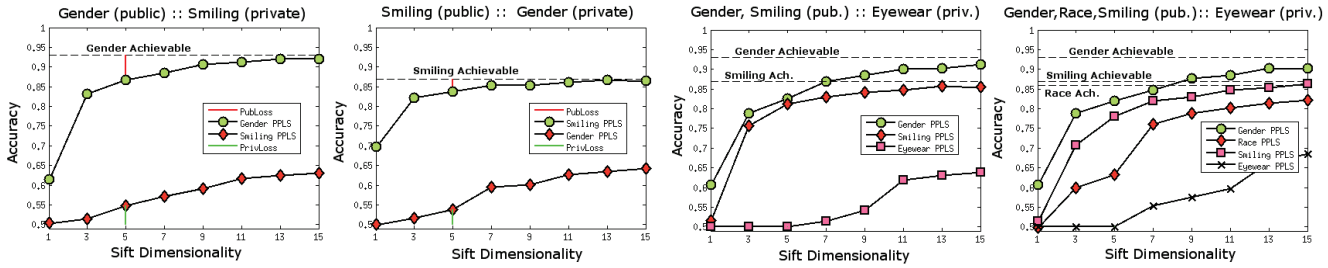

Figure 3: Left: *PubLoss* **and** *PrivLoss* **performance (classification accuracy) as a function of sift dimensionality for two simple policies. Right:** *PubLoss* **and** *PrivLoss* **performance for complex policies. In all figures, the lowest** *PubLoss* **and highest** *PrivLoss* **is reported across all five classifiers. Dashed lines represent the maximum achievable accuracies using raw (unsifted) data which serve as upper bounds for** *PubLoss* **performance.**

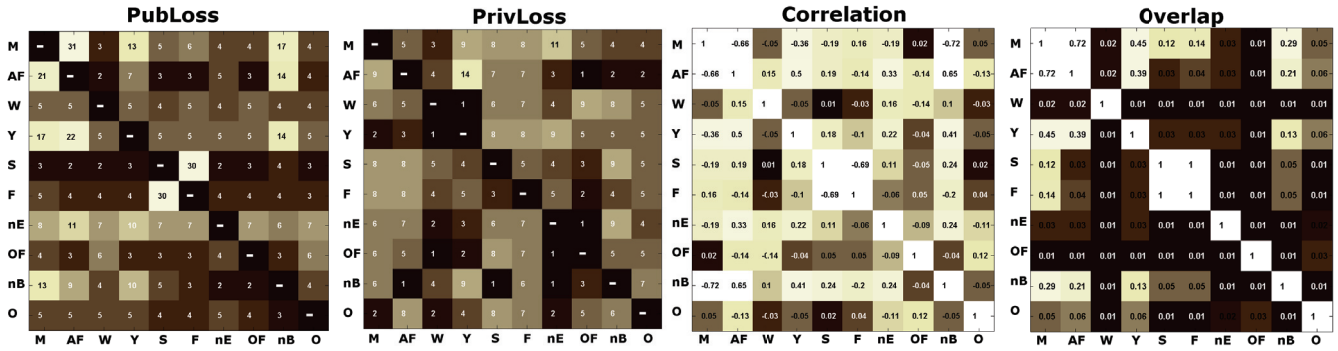

Figure 4: *PubLoss*, *PrivLoss*, **Correlation, and Overlap matrices for our 90 simple policy combinations. Rows denote the public attribute, columns represent the private attribute, while cells represent policies which combine the row and column attributes. In the case of** *PubLoss* **and** *PrivLoss* **lower values are desirable as they indicate minimal utility and privacy sacrifices respectively. Correlation values are shown using absolute values and higher cell values indicate significant information linkages between attributes. Lastly, high Overlap values indicate that attributes occupy the same regions in feature space.**

Intuitively highly correlated attributes with significant overlap should prevent utility and privacy balance. This is indeed what we see when we match up the results of the *PubLoss* and *PrivLoss* matrices with the correlation and overlap matrices (Figure 4).

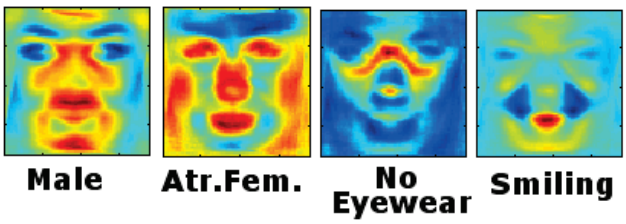

Figure 5: The image features (from the red component of the raw RGB values) which most strongly covary with several attributes (red values indicate strong positive correlations, blue values indicate strong negative correlations).

A regression analysis model attributes 63% of the *PubLoss* estimate to correlation and 37% to overlap. In the case of *PrivLoss* the weights correspond to 67% and 33% for correlation and overlap respectively. Furthermore, using regression we find that correlation alone as a predictor leads to a SSE (sum of squared error) term which is 315% larger than if correlation and overlap are used together. These findings suggest that correlation and overlap should be considered together when analyzing sifting performance.

Extensions: Complex Policies. Although the bulk of our analysis has focused on policies in which there is one public and one private attribute, our algorithm can be augmented to support multiple public and multiple private attributes. To illustrate the potential for more complex policies, we modified the PPLS objective function to produce the largest average gap between multiple public and multiple private attribute covariances relative to data features.

$$\begin{array}{ll} find & \max_w [avg(cov(Xw, Y_1^+), cov(Xw, Y_2^+), ...)^2 \\ . & \quad -\lambda * avg(cov(Xw, Y_1^-), cov(Xw, Y_2^-), ...)^2] \\ s.t. & w^\top w = 1 \end{array}$$

Using this averaging method, we were able to find high performing masks for various policies which include several public and/or several private attributes. An example of two such policies is provided in Figure 3.

Complex policies can include arbitrary ratios of (public:private) 1:2, 1:3, 2:1, 2:2, 3:1 (i.e., public: *'Male'* + *'Smiling'*, private: *'White'* + *'Youth'*). The number of complex policy combinations is very large, however in our tests using (35 complex policies) we found that the same principles from Section 7 apply. Just as in the case of simple policies correlations and overlap have a big impact on *PubLoss* and *PrivLoss*. In general as polices grow to include many attributes the likelihood of significant correlation/overlap grows thus increasing the chance of diminishing utility and privacy balance. A more detail analysis of complex policies is a deep topic which is certainly an attractive target for future work.

Extensions: Streaming Content. So far we have focused our analysis on static sensor samples (i.e., still photos), however dynamic data (i.e. streaming video) is also of importance. To evaluate the SensorSift scheme in a dynamic context we used the Talking Face dataset [3]. The data consists of 5000 frames taken from a 29fps video of a person engaged in a natural conversation lasting roughly 200 seconds. Using the annotations provided from the dataset we first cropped the face region from each frame. Next we

extracted image features as described in Section 5. Subsequently we used the Face.com [16] labeling tool to determine the frames in which the individual was smiling.

As evaluation, we applied the sift for the policy Male (public) Smiling (private) to concatenated sets of 10 sequential frames (identified as smiling) together prior to computing *PubLoss* and *PrivLoss*. As an additional pre-processing step we made sure that the sequences of frames we used as our concatenated samples did not occur at the boundaries of smiling events). We find that the PrivLoss accuracy increases by only 2.3% while PubLoss accuracy decreased by 4.5% (using 5 dimensional sifts and a $\lambda = 5$).

This is an encouraging result and suggests that the SensorSift technique can be applied to dynamic sensor contexts, however, in instances where samples are accumulated over longer time sequences (i.e., days, months) the dynamics of privacy exposure are likely to change and so will the optimal parameter settings for sift output dimensionality and privacy emphasis (λ). This is certainly an important area for further research as dynamic sensing becomes more ubiquitous (i.e. Microsoft Face Tracking Software Development Kit in Kinect for Windows [2]).

Comparison to Related Work. The most similar publication to our present effort is a recent article by Whitehill and Movellan [18] which uses image filters (applied to a face dataset) to intentionally decrease discriminability for one classification task while preserving discriminiability for another (smiling and gender). This work uses a ratio of discriminability metrics (based on Fisher's Linear Discriminant Analysis) to perform a type of linear feature selection. Perhaps the most significant difference between [18] and SensorSift is that the authors evaluate the quality of their privacy filters against human judgments whereas we target automated inferences.

To compare against [18] we used the methods and demo dataset provided on their website. The dataset consists of 870 grayscale images (16x16 pixel 'face patches'). It also provides labels for smiling and gender thus enabling analysis of two policies (1) gender (public) : smiling (private), and (2) smiling (private) : gender (public).

For each policy we evaluated 3 different combinations of training and testing data splits (using different 80% 20% splits of training and testing respectively). For each combination we generated 100 discriminability filters using the provided algorithm (total of 300 filters for each policy) and subsequently used a linear SVM classifier to evaluate their quality. We found that even though these filters were reported to prevent successful human judgement on the private attribtue, even the best filter we found was not able to deter machine inference.

In particular the lowest private attribute accuracy for the gender (public) smiling (private) policy was 81.21% (average 86.32%). Conversely the lowest private attribtue accuracy for the smiling (public) gender (private) policy was 77.65% (average 83.12%). The public attribute accuracy decreased by 4% on average relative to classification performance on unfilted (raw) images.

8. RELATED WORK

Below we touch on the related literature in the broader context of balancing utility and privacy and subsequently describe efforts withing the more focused area of face-privacy on which we base our experimental evaluation.

Utility Privacy Balance. There are several classes of approaches which have been proposed for finding a utility and privacy balance in database and/or information sharing contexts. Among these, the developments in differential privacy and cryptographic techniques are only remotely connected to our present discussion as they focus on statistical databases and very limited homomorphic encryption respectively [7]. More pertinent are the systems based approaches which typically use proxies/brokers for uploading user generated content prior to sharing with third-parties. These approaches use access control lists, privacy rule recommendation, and trace audit functions; while they help frame key design principles, they do not provide quantitative obfuscation algorithms beyond degradation of information resolution (typically for location data) [14].

Lastly, there are several papers which have looked at the question of privacy and utility balance from a trust modeling and information theoretic perspectives [5, 6]. While these are very valuable problem characterizations which we use to motivate our formal analysis, we go beyond their framing and develop an algorithmic defense tool which we apply to a real world problem. Furthermore we introduce an information processing scheme for embedding our algorithm into a trusted platform for potential deployment in smart sensor applications.

Previous Approaches to Face Privacy. Prior work on preserving the privacy of face images and videos has been almost exclusively focused on data transformations aimed at identity anonymization. The methods range from selectively masking or blurring regions of the face or the whole face [4], perturbing the face ROI (region of interest) with noise through lossy encoding [13], and face averaging schemes (k-Same, and its variants [8, 15]) aimed at providing k-anonymity guarantees (each de-identified face relates ambiguously to at least k other faces). Whereas these methods emphasize recognition deterrence their methods of limiting information exposure are unconstrained in what face attribute details they perturb. The only notable exception is the multi-factor (ε, k)-map algorithm [8] which demonstrates a selective ability to enhance the representations of facial expressions in k-anonymity de-identified faces, however this approach does not consider privacy granularity below the level of identity protection.

9. DISCUSSION

Our approach aims to mitigate the emerging privacy threats posed by automated reasoning applied to harvested digital traces of personal activity by using algorithmic defenses that enable selective information exposure – private information should remain private, while other non-private information can be harvested and used. We believe that this is a promising approach towards offering quantitative privacy assurances in the rapidly growing market of smart sensing applications.

A critical strength of the SensorSift design is the built in support for innovation by future application developers. We provide an algorithm for generating sifting transformations which can be used by developers to unlock access to novel data features. As long as the sifting transformation functions can be verified to yield minimal sensitive information exposure our system will allow it to operate over the sensor data. This ability to dynamically generate and verify novel privacy respecting data access functions enables flexibility and provides a way to keep up with the rapidly evolving needs of software providers.

Limitations. We stress that, as with many systems, privacy is not binary. Indeed, it may be impossible to achieve absolute privacy in any useful sensor-based system. Our goal, therefore, is to explore new directions for increasing privacy for sensor-based systems while flexibly supporting the desired functionality.

An important point to consider is that multiple applications may request different privacy views (i.e., sift functions) of the image data. In the present work, we do not consider collusion between applications – two applications may be able to combine their func-

tions to reconstruct information greater than that granted to each application alone. We do note, however, that some simple measures can be used to protect against collusion (e.g., apply SensorSift to all the applications running on a system in unity rather than to each application by itself, or only allow one application access to facial attributes over some period of time).

A second potential weakness of our approach is that adversaries may have additional knowledge sources at their disposal which can reveal private information that SensorSift is unable to counteract. Our goal is to explore how to protect against unauthorized privacy disclosures from the sensed data itself, not to defend against auxiliary information sources. Indeed, auxiliary information can almost always break any privacy or anonymity-preserving system. As an extreme example, suppose the private attribute is race but that the application asks the user to complete a biographical form – which includes race – during the application installation process.

Third, our approach leverages classification metrics to verify that the data exposed to applications does not reveal significant information about private attributes; it is possible that future machine learning tools can significantly outperform our benchmarks. To mitigate this evolving algorithmic threat, our scheme uses an ensemble of multiple machine classification tools which span the space of state of the art linear and non-linear methods. Further, the design is meant to support plug in modules so that new classifiers can be added on demand to enrich the privacy metrics.

10. CONCLUSION

Given the growing demand for interactive systems, the low cost of computational resources, and the proliferation of sophisticated sensors (in public/private locations and mobile devices) digital traces of our identities and activity patterns are becoming increasingly accessible to third parties with analytics capabilities. Thus, although sensor systems enhance the quality and availability of digital information which can aid technical innovation, they also give rise to security risks and cause privacy tensions. To address these concerns we proposed a theoretical framework for quantitative balance between utility and privacy though policy based control of sensor data exposure.

In our analysis we found promising results when we evaluated the PPLS algorithm within the context of optical sensing and automated face understanding. However, the algorithm we introduce is general, as it exploits the statistical properties of the data; and in the future it would be exciting to evaluate SensorSift in other sensor contexts.

Acknowledgements

We would like to thank the members of the UW SecurityLab and Dr. Daniel Halperin for their insightful feedback during the writing process. This work was supported in part by the Intel Science and Technology Center for Pervasive Computing and NSF Grant CNS-0846065.

References

[1] Inside google project glass part 1, 2012. http://www.fastcompany.com/1838801/exclusive-inside-google-x-project-glass-steve-lee.

[2] Kinect for windows sdk, 2012. http://www.microsoft.com/en-us/kinectforwindows/develop/new.aspx.

[3] Talking face video, 2012. http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html.

[4] Mukhtaj S. Barhm, Nidal Qwasmi, Faisal Z. Qureshi, and Khalil El-Khatib. Negotiating privacy preferences in video surveillance systems. In *IEA/AIE*, volume 6704 of *Lecture Notes in Computer Science*, pages 511âĂŞ–521, 2011.

[5] Supriyo Chakraborty, Haksoo Choi, and Mani B. Srivastava. Demystifying privacy in sensory data: A qoi based approach. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 38 –43, 2011.

[6] Supriyo Chakraborty, Haksoo Choi Zainul Charbiwala, Kasturi Rangan Raghavan, and Mani B. Srivastava. Balancing behavioral privacy and information utility in sensory data flows. In *Preprint*, 2012.

[7] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, 2009.

[8] R. Gross, L. Sweeney, F. de la Torre, and S. Baker. Semi-supervised learning of multi-factor models for face de-identification. In *CVPR*, pages 1–8, 2008.

[9] David Kotz, Sasikanth Avancha, and Amit Baxi. A privacy framework for mobile health and home-care systems.

[10] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, pages 340–353, 2008.

[11] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[12] Shan Li and David Sarno. Advertisers start using facial recognition to tailor pitches, 2011. http://www.latimes.com/business/la-fi-facial-recognition-20110821,0,7327487.story.

[13] Isabel Martnez-ponte, Xavier Desurmont, Jerome Meessen, and Jean franois Delaigle. Robust human face hiding ensuring privacy. In *WIAMIS*, 2005.

[14] Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal data vaults: a locus of control for personal data streams. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 17:1–17:12, 2010.

[15] Elaine M. Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE Trans. Knowl. Data Eng*, 17(2):232–243, 2005.

[16] Yaniv Taigman and Lior Wolf. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition, August 2011.

[17] Cajo J. F. ter Braak and Sijmen de Jong. The objective function of partial least squares regression. *Journal of Chemometrics*, 12(1):41–54, 1998.

[18] Jacob Whitehill and Javier Movellan. Discriminately decreasing discriminability with learned image filters. In *CVPR*, 2012.