Discriminately Decreasing Discriminability with Learned Image Filters

Jacob Whitehill and Javier Movellan *
Machine Perception Laboratory
University of California, San Diego
{ jake, movellan }@mplab.ucsd.edu

Abstract

In machine learning and computer vision, input signals are often filtered to increase data discriminability. For example, preprocessing face images with Gabor band-pass filters is known to improve performance in expression recognition tasks [1]. Sometimes, however, one may wish to purposely decrease *discriminability of one classification task (a "distractor" task), while simultaneously* preserving *information relevant to another task (the target task): For example, due to privacy concerns, it may be important to mask the identity of persons contained in face images before submitting them to a crowdsourcing site (e.g., Mechanical Turk) when labeling them for certain facial attributes. Suppressing discriminability in distractor tasks may also be needed to improve inter-dataset generalization: training datasets may sometimes contain spurious correlations between a target attribute (e.g., facial expression) and a distractor attribute (e.g., gender). We might improve generalization to new datasets by suppressing the signal related to the distractor task in the training dataset. This can be seen as a special form of supervised regularization. In this paper we present an approach to automatically learning preprocessing filters that suppress discriminability in distractor tasks while preserving it in target tasks. We present promising results in simulated image classification problems and in a realistic expression recognition problem.*

1. Introduction

In machine learning problems, signals are commonly pre-filtered prior to classification to enhance class discriminability. For example, pre-filtering face images with Gabor band-pass filters is known to increase performance in expression recognition tasks [1]. Such filters may be manually constructed or may be learned directly from the data (e.g., using Deep Belief Networks [8] or Independent Components Analysis [2]). However, there also exist scenarios in which it may be useful to intentionally *decrease* discriminability for *one* classification task (a "distractor" task), while enhancing or at least preserving discriminability for *another* task (the target task). Two such scenarios include (1) preservation of privacy during data labeling, and (2) generalization to datasets with different correlation structure.

(1) *Preservation of privacy*: Computer vision is increasingly making use of crowdsourcing services such as the Amazon Mechanical Turk, with anonymous labelers. Sometimes, the data to be labeled may contain sensitive information that should not be released to the public, e.g., the identity of people's faces or the geographical locations of satellite images. It may be useful to first filter the images before uploading them to the Mechanical Turk so that identity/location is removed, but so that the target task remains highly discriminable. For the case of facial identity removal, this process is known as face de-identification [11].

(2) *Generalization to new datasets*: It is not unusual for datasets used to train classifiers to have spurious correlations that impair generalization performance to other datasets. Consider, for example, a classifier to discriminate smiles from neutral facial expressions. Suppose in the training dataset 90% of the male faces have smiles while only 20% of the females smile. A classifier may learn to discriminate smile from neutral expressions by capitalizing on features that discriminate male from female faces. Such a classifier will perform well in cross-validation tests within the dataset, but it will perform badly when tested on a new dataset in which females smile more than males. To improve generalization, standard regularization methods such as L_2 weight penalization can be used; many such methods are equivalent to adding uncorrelated noise to the data in the training set, with the hope that adding such noise will suppress small spurious correlations while still preserving some of the information about the task of interest. In some cases, however, a more targeted regularization approach may be desirable in which we suppress information about a specific attribute in the training set while preserving information about the attribute of interest.

In this paper, we present a novel algorithm for auto-

*The authors gratefully acknowledge NSF grant SBE-0542013.

matically learning data filters that simultaneously *preserve* discriminability of a target task while *suppressing* discriminability of distractor task. In this sense, the filter "discriminately decreases discriminability" of the training data. We focus on *image* filters, but in fact the data can be of arbitrary modalities. In Section 2 we provide a simple example of "discriminately decreasing discriminability" (DDD) . In Section 3 we present the proposed algorithm. We conclude with empirical evaluations on synthetic data and on a realistic facial expression recognition task.

2. Simple example in R^2

Consider the set of 28 data points $\{x_i\}$ (in R^2) shown in Figure 1 (**left**): Each point x_i is given binary labels for two labeling tasks. Points labeled 0 for Task A are shown in magenta, while points labeled 1 for Task A are black. On the other hand, points labeled 0 for Task B are marked as crosses, while points labeled 1 are shown as circles. In their unfiltered original form, both tasks are easily discriminated.

Suppose now that we filter the data using θ_1 (in this case, a general linear transformation), as shown in the **center** part of the figure: Task A (color) is highly discriminable, while Task B (marker) is not – the two marker styles (circles and crosses) appear to overlap. Similarly, we can use θ_2 to suppress discriminability of Task A and preserve discriminability of Task B, in which case we arrive at the filtered points shown in Figure 1 (**right**). The goal of the algorithm in this paper is to learn such linear filters automatically.

3. DDD Algorithm: Learning a filter to discriminately decrease discriminability

We pose the task of learning a filter to discriminately decrease discriminability as an optimization problem. The formulation is flexible in that two of the inputs, f and D, can take any form as long as they are differentiable. **Inputs**:

1. A dataset consisting of ordered triples $\{(x_i, l_i^A, l_i^B)\}$, where each $x_i \in \mathcal{R}^d$, $l_i^A \in \{0,1\}$ is the label of x_i for Task A, and $l_i^B \in \{0,1\}$ is the label of x_i for Task B. For example, each x_i might be a face image with d pixels represented as a column vector, Task A might indicate whether x is smiling or not, and Task B might indicate whether x is male or female.

   From the $\{(x_i, l_i^A, l_i^B)\}$, we can define the following matrices (each with d rows), each containing some of the data points as column vectors: X_{0a} contains all the x_i s.t. $l_i^A = 0$, while X_{1a} contains all the x_i s.t. $l_i^A = 1$. Similarly, X_{0b} contains all the x_i s.t. $l_i^B = 0$, while X_{1b} contains all the x_i s.t. $l_i^B = 1$.

2. A filter function $f(\theta, x)$ that filters input vector x using parameters specified by θ. We define the output of the filter on input vector x as $y \doteq f(\theta, x)$.

   For an input matrix $X = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}$ of N column vectors, we also define

   $$\begin{aligned} F(\theta, X) &= F(\theta, \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}) \\ &\doteq \begin{bmatrix} f(\theta, x_1) & \cdots & f(\theta, x_N) \end{bmatrix} \\ &= \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix} \end{aligned}$$

   and then define $Y \doteq \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix}$.

3. A "discriminability metric" $D(X_0, X_1)$ which measures the real-valued "discriminability" of data in matrix X_1 from data in matrix X_0. In our implementation, we use the ratio of between-class variance to within-class variance as the discriminability metric.

Objective function: Given the inputs above, and assuming that Task A is the "target" task while Task B is the "distractor" task, we must select an objective function $R(\theta)$ to minimize w.r.t. the filter parameters θ. R should be small when $D(F(\theta, X_{0a}), F(\theta, X_{1a}))$ (i.e., the discriminability of the *filtered* data for Task A) is large and when $D(F(\theta, X_{0b}), F(\theta, X_{1b}))$ is small. Several choices for R are possible; we use the following *ratio of discriminabilities* formulation:

$$R(\theta) = \log \frac{D(F(\theta, X_{0b}), F(\theta, X_{1b}))}{D(F(\theta, X_{0a}), F(\theta, X_{1a}))} + \frac{1}{2}\beta \operatorname{tr}(\theta^\top \theta)$$

where $\beta \geq 0$ is a scalar regularization parameter on θ. We wish to find θ^* such that

$$\theta^* = \arg\min_\theta R(\theta)$$

While the global minimum may be difficult to find, we can use gradient descent to find a local minimum as long as both f and D are differentiable; hence, the **output** of the DDD algorithm is a θ that locally minimizes R.

Ideally, one chooses f and D with derivatives that are available analytically so that gradients can be computed exactly. Below we suggest some functions with this property:

3.1. Filter function f

For a variety of filter functions f, the derivative $df/d\theta$ can be found analytically. Useful examples include:

- Convolution: $f(\theta, x) = \theta * x$, where θ represents the convolution kernel.
- General linear transformations: $f(\theta, x) = \theta x$, where θ is any matrix that can be right-multiplied by x.
- Pixel-wise "masking": $f(\theta, x) = \theta x$, where θ is a $d \times d$ *diagonal* matrix. In this case, element θ_{ii} represents the strength with which pixel i of the image is allowed to pass through.

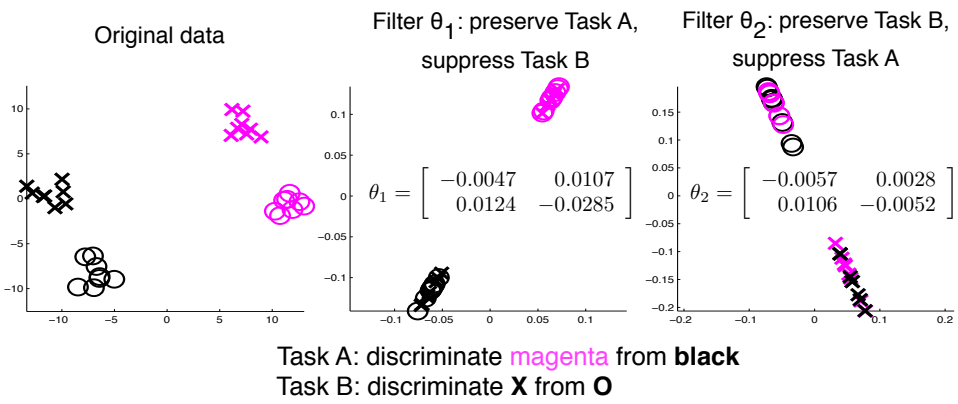

Figure 1. A minimal example in R^2 showing (**left**) unfiltered data, data filtered to preserve Task A's and suppress Task B's discriminability (**center**), and data filtered to suppress Task A's and preserve Task B's discriminability (**right**).

- Sum of filters: $f(\theta, x) = \sum_{k=1}^{K} \theta_k f_k(x)$, where θ is a K-vector specifying the weight of a bank of K fixed filter functions $\{f_k\}$. For example, the $\{f_k\}$ might constitute a Gabor filter bank [10] of different spatial frequencies and orientations.

Although the above examples are linear, non-linear filters are also admissible as long as they are differentiable in θ.

3.2. Discriminability metric D

For the discriminability metric, we use D_{LDA} – the maximum ratio, over all unit vectors $p \in \mathcal{R}^d$, of the between-class variance to within-class variance of the data points after projecting them onto p. This discriminability metric was first proposed by Fisher [5] and is used in Fisher's Linear Discriminant Analysis (LDA).

Notation: Suppose that X_1 and X_0 contain the data points labeled 1 and 0, respectively, for some labeling task (Task A or Task B), and suppose that $Y_1 = F(\theta, X_1)$ and $Y_0 = F(\theta, X_0)$. Then we can define the mean vector for class 1 as $\overline{x}_1 \doteq \frac{1}{N_1} X_1 \mathbf{1}$ where N_1 is the number of columns in X_1 and $\mathbf{1}$ is a column vector of N_1 ones. We can also define the "mean matrix" $\overline{X}_1$, consisting of N_1 copies of $\overline{x}_1$, as $\overline{X}_1 \doteq \overline{x}_1 \mathbf{1}^\top$. We define $\overline{x}_0$ and $\overline{X}_0$ analogously. Finally, we define $\overline{y}_1 \doteq \frac{1}{N_1} Y_1 \mathbf{1}$, and $\overline{Y}_1 \doteq \overline{y}_1 \mathbf{1}^\top$ as the mean *filtered* vector and matrix for class 1, and do the same analogously for $\overline{y}_0, \overline{Y}_0$ for class 0. Given this notation, we define

$$D_{\text{LDA}}(X_0, X_1) \doteq \max_p \frac{p^\top B(X_0, X_1) p}{p^\top W(X_0, X_1) p}$$

where the between-class variance is given by $B(X_0, X_1) = (\overline{x}_1 - \overline{x}_0)(\overline{x}_1 - \overline{x}_0)^\top$ and the within-class variance is given by $W(X_0, X_1) = (X_1 - \overline{X}_1)(X_1 - \overline{X}_1)^\top + (X_0 - \overline{X}_0)(X_0 - \overline{X}_0)^\top$.

Computing D_{LDA} requires solving an optimization problem over p. The key feature that makes D_{LDA} useful for the DDD algorithm is that the optimal p can be found analytically [4]. (Note: this contrasts with certain other discriminability measures such as the margin of an SVM, which can only be found numerically.) The optimal p is

$$\begin{aligned} p^*(X_0, X_1) &\doteq \arg\max_p \frac{p^\top B(X_0, X_1) p}{p^\top W(X_0, X_1) p} \\ &= W(X_0, X_1)^{-1} (\overline{x}_1 - \overline{x}_0) \end{aligned}$$

and hence

$$D_{\text{LDA}}(X_0, X_1) = \frac{p^*(X_0, X_1)^\top B(X_0, X_1) p^*(X_0, X_1)}{p^*(X_0, X_1)^\top W(X_0, X_1) p^*(X_0, X_1)}$$

To avoid clutter, we abbreviate D_{LDA} as

$$D_{\text{LDA}}(X_0, X_1) = \frac{p^{*\top} B p^*}{p^{*\top} W p^*}$$

With DDD, the inputs to D_{LDA} are *filtered* data matrices $Y_1 = F(\theta, X_1)$ and $Y_0 = F(\theta, X_0)$ that depend on θ. Hence, B, W, and p^* will implicitly also depend on θ.

3.3. Gradient descent on $R(\theta)$

Putting all the parts together, we can now perform gradient descent on R w.r.t. θ. Below we derive the gradient expressions for the most important terms:

$$\begin{aligned} \frac{\partial R}{\partial \theta_{ij}}(\theta) &= \frac{\frac{\partial}{\partial \theta_{ij}} \left(D_{\text{LDA}}(F(\theta, X_{0b}), F(\theta, X_{1b})) \right)}{D_{\text{LDA}}(F(\theta, X_{0b}), F(\theta, X_{1b}))} - \\ &\quad \frac{\frac{\partial}{\partial \theta_{ij}} \left(D_{\text{LDA}}(F(\theta, X_{0a}), F(\theta, X_{1a})) \right)}{D_{\text{LDA}}(F(\theta, X_{0a}), F(\theta, X_{1a}))} + \beta \theta_{ij} \\ \frac{\partial D_{\text{LDA}}}{\partial \theta_{ij}}(Y_0, Y_1) &= \frac{\frac{\partial}{\partial \theta_{ij}} (p^{*\top} B p^*)}{p^{*\top} W p^*} - \\ &\quad \frac{p^{*\top} B p^*}{(p^{*\top} W p^*)^2} \frac{\partial}{\partial \theta_{ij}} \left(p^{*\top} W p^* \right) \end{aligned}$$

$$
\begin{aligned}
\frac{\partial p^*}{\partial \theta_{ij}}(Y_0, Y_1) &= \frac{\partial}{\partial \theta_{ij}} \left(W^{-1}(\overline{y}_1 - \overline{y}_0) \right) \\
&= -W^{-1} \left(\frac{\partial}{\partial \theta_{ij}} W \right) W^{-1}(\overline{y}_1 - \overline{y}_0) + \\
&\quad W^{-1} \left(\frac{\partial}{\partial \theta_{ij}} (\overline{y}_1 - \overline{y}_0) \right) \\
\frac{\partial W}{\partial \theta_{ij}} &= \frac{\partial}{\partial \theta_{ij}} \left((Y_1 - \overline{Y}_1)(Y_1 - \overline{Y}_1)^\top \right) + \\
&\quad \frac{\partial}{\partial \theta_{ij}} \left((Y_0 - \overline{Y}_0)(Y_0 - \overline{Y}_0)^\top \right) \\
\frac{\partial B}{\partial \theta_{ij}} &= \frac{\partial}{\partial \theta_{ij}} \left((\overline{y}_1 - \overline{y}_0)(\overline{y}_1 - \overline{y}_0)^\top \right) \\
\frac{\partial y}{\partial \theta_{ij}} &= \frac{\partial}{\partial \theta_{ij}} f(\theta, x)
\end{aligned}
$$

For general linear transformation filters $f(\theta, x) = \theta x$, $\frac{\partial}{\partial \theta_{ij}} f(\theta, x) = E_j x$, where E_j is a $d \times d$ matrix consisting of all 0's except the (j, j)th entry, which is 1. The gradients of pixel-wise mask filters and discrete convolution filters are given in the Supplementary Materials.

3.4. Reconstruction from filtered images

Gradient descent will find a θ that locally minimizes $R(\theta)$, but there is no guarantee that the filtered images Y will visually resemble the original images X or that humans can interpret them. For machine classification (e.g., when learning a filter to improve inter-dataset performance), this may not matter, but for human labeling applications, it may be necessary to "restore" the filtered images to a more intuitive form. Hence, as an optional step, linear ridge regression can be used to convert the filtered images Y to a form more closely resembling the original images X, while still preserving the property that they are highly discriminability for Task A and not highly discriminable for Task B. In particular, we can compute the $d \times (d+1)$ (the extra +1 is for the bias term) linear transformation P that minimizes

$$
\left\| X - P \begin{bmatrix} Y \\ \mathbf{1} \end{bmatrix} \right\|_{\text{Fr}}^2 + \delta \left\| P \tilde{I} \right\|_{\text{Fr}}^2
$$

where $\delta > 0$ is a scalar ridge strength parameter, $\tilde{I}$ is the identity matrix except that the last diagonal entry is 0 instead of 1 (so that there is no regularization on the bias weight), and Fr means Frobenius norm.

The ridge term in the linear reconstruction is critical: because many of the filters that the gradient descent procedure learns correspond to invertible linear transformations, linear regression without regularization would transform each y_i back to x_i with no loss of information, which would defeat the purpose of filtering at all. With ridge regression, on the other hand, only the "more discernible" aspects of the

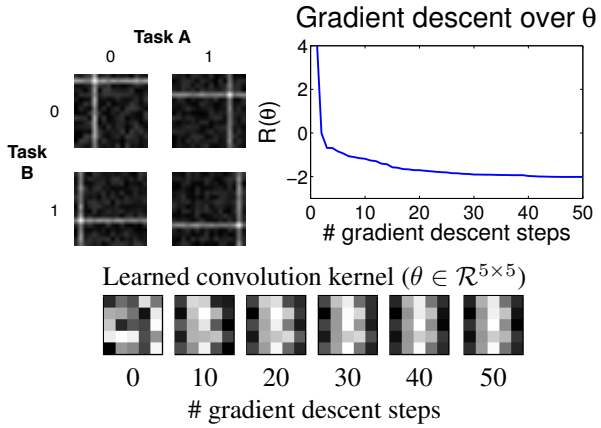

Figure 2. **Upper-left**: Synthetic images consisting of vertical and horizontal lines at different positions. **Upper-right**: gradient descent curve over $R(\theta)$ to a learn a filter to preserve Task A and suppress Task B. **Bottom**: The filters learned at corresponding gradient descent steps.

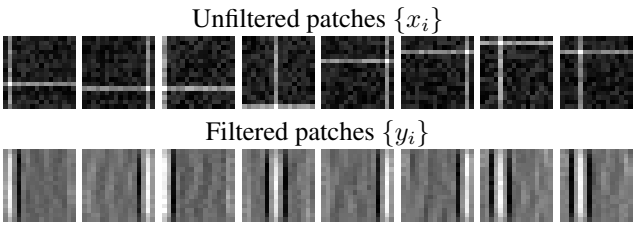

Figure 3. **Top**: unfiltered image patches consisting of superimposed vertical and horizontal lines plus uniform noise. **Bottom**: the same images filtered with a convolution kernel designed to suppress discriminability of Task B (horz. lines) while preserving discriminability of Task A (vert. lines).

image (i.e., the target task) are restored clearly, while the "less discernible" aspects (pertaining to the distractor task) are not. By varying δ, one can cause each "reconstructed" image r_i (where $r_i \doteq P \begin{bmatrix} y_i \\ 1 \end{bmatrix}$) to strongly resemble the mean image $\overline{x}$ (for large δ) or to strongly resemble its unfiltered counterpart x_i (for small δ). In practice, δ is chosen based on visual inspection of the reconstructed training images so that, to the human observer, the target task is clearly discriminable while the distractor task is not.

4. Experiment I: synthetic data

In our first experiment we studied whether the DDD algorithm could operate on images (16×16 pixels) consisting of simple line patterns in order to suppress lines in one direction while preserving them in another. For the filtering operation, we chose "clipped" convolution – $y = f(\theta, x) = \theta * x$, whereby the output image y is the same size as the input image x – using a convolution ker-

Figure 4. **Not DDD**: here, the filter θ optimized $R_{\text{RidgeOnly}}$ to preserve discriminability of Task A *without* specifically suppressing discriminability of Task B. The filter preserves the vertical image components, but does little to suppress the horizontal components.

nel of 5×5 pixels. In this study, all images contained one horizontal line and one vertical line at random locations: In Task A, an image was labeled 0 if its vertical line was in the left half of the image and 1 if the vertical line was in the right half. In Task B, an image was labeled 0 if its horizontal line was in the top half and 1 if it was in the bottom half. Each image $x_i \in R^{16\times16}$ was generated by adding one vertical and one horizontal line (of pixel intensity 1) at random image positions, and then adding uniform noise in $U[0, 0.5)$ to all pixels in the image. Examples are shown in Figure 2 (**upper-left**).

After generating 1000 images according to the procedure above, we initialized the convolution kernel $\theta \in R^{5\times5}$ to random values from $U[0, 1)$ (shown in Figure 2 as the filter kernel at gradient descent step 0). We then applied DDD to learn a filter θ to preserve Task A while suppressing Task B. We set $\beta = 1$. The descent curve is shown in Figure 2 (**upper-right**), and the learned filter kernel at every 10 steps is shown in the **bottom** of the figure. After filtering the images using the convolution kernel learned after 50 descent steps, we arrived at the images shown in Figure 3. Notice how the horizontal lines have been almost completely eradicated, thus decreasing class discriminability for Task B.

4.1. Effect of the ridge term by itself

One might reasonably posit that the eradication of the horizontal line components has more to do with the regularization term $\frac{1}{2}\beta\,\mathrm{tr}(\theta^\top\theta)$ than with the "ratio of discriminabilities" used in the objective function. To test this hypothesis, we created a second objective function $R_{\text{RidgeOnly}}(\theta) = -\log D(F(\theta, X_{0a}), F(\theta, X_{1a})) + \frac{1}{2}\beta\,\mathrm{tr}(\theta^\top\theta)$ that does not explicitly penalize discriminability of Task B (the distractor task). We then optimized the ridge parameter β (by visual inspection, where $\beta \in \{10^0, 10^1, \ldots, 10^7\}$) so that the learned filter θ maximally reduced the visibility of the horizontal components. The result for $\beta = 10^3$ is shown in Figure 4 (though the filter outputs were similar across different β): although the learned filter does preserve the vertical components, its effect on the horizontal line is to "smear" it across the image, leaving it highly discriminable. In contrast, DDD offers a more "surgical" form of regularization that removes specific, undesired components of the data set.

4.2. Comparison to "nullspace filter"

Another plausible method of achieving the "discriminately discriminable" property is to use LDA to find the most discriminable directions of the *distractor* task, and then to reconstruct the images from the nullspace of those directions. More precisely, we can compute the $d \times k$ matrix A whose columns contain the top k directions that maximize D_{LDA} of the distractor task (for some chosen k). Then, each image x can be filtered as $y = \theta_{\text{ns}}x$ where $\theta_{\text{ns}} \doteq BB^\top$ where B is a matrix whose columns span the nullspace of A. This filter will tend to suppress aspects of the image correlated with the distractor task. In practice, however, we found that this method could not suppress the horizontal lines (distractor task) from human perception while still preserving the vertical lines (target task), even for a variety of choices for k. (See Supp. Materials for details.) One reason may be that DDD can be constrained to optimize particular families of filters (e.g., convolution, pixel-wise masking) that may be harder for humans to "invert" than other linear transformations.

5. Experiment II: natural face images

5.1. Preserve expression, suppress gender

We applied the DDD algorithm to natural face images from the GENKI dataset [14], which consists of 60,000 face images, collected from thousands of different persons and geographical locations, and which was used to train a commercial smile detector. Each GENKI image has been manually labeled for 2 binary attributes – smile/non-smile and male/female – as well as the 2D positions of the eyes, nose, and mouth, and the 3D head pose (yaw, pitch, and roll). In this experiment we assessed whether a filter could be learned to *preserve discriminability of expression* (smile/non-smile), while *suppressing discriminability of gender*. For f we used a pixel-wise "mask" filter (see Section 3.1) of the same size as the images (16×16 pixels).

From the whole GENKI dataset we selected a training set consisting of 1740 images (50% male and 50% female; 50% smile and 50% non-smile) whose yaw, pitch, and roll parameters were all within 5° of frontal. All of the images were registered to a common face cropping using the center of the eyes and mouth as anchor points. They were then downscaled to a resolution of 16×16 pixels. In addition, we similarly extracted a separate testing set consisting of 100 images (50 males, 50 females, and 50 smiling, 50 non-smiling) with the same 3D pose characteristics. The filter θ was initialized component-wise by sampling from $U[0, 1)$.

Using the training set for learning the filter, and setting the regularization parameter $\beta = 1$, we applied conjugate gradient descent for 100 function evaluations. The learned filter was then applied to all of the training images. Finally, we applied the image reconstruction technique from Section

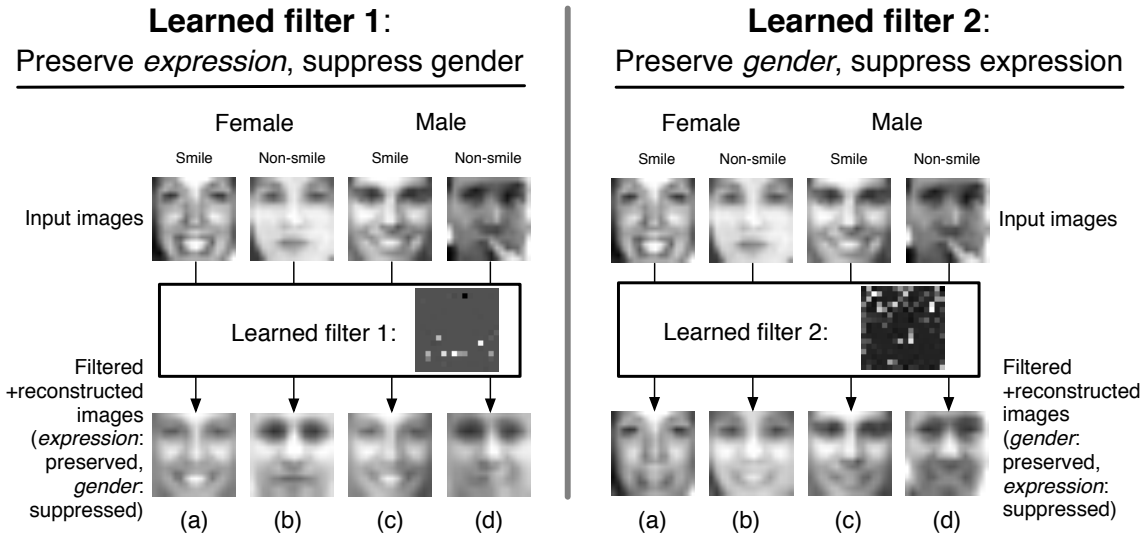

Figure 5. Face images from the GENKI dataset that have been filtered to preserve expression and suppress gender (**left**), or to preserve gender and suppress expression (**right**). Filters were learned using the algorithm presented in Section 3.

3.4 to restore the filtered images to a form more easily analyzable by humans. The reconstruction ridge parameter δ was selected, by looking only at the training images, so that smile appeared well discriminable whereas gender did not (in this case, $\delta = 6 \times 10^{-2}$). Examples of the input images as well as the filtered (+ reconstructed) images are shown in Figure 5 (**left**). The learned filter mask is shown to the right of the text "Learned filter 1". As shown in the figure, the filter allows most of the smile information to pass through, but it removes most of the gender information, as intended.

To assess quantitatively the ability of the learned filter to preserve expression and suppress gender, we posted a labeling task to the Amazon Mechanical Turk (AMT) consisting of 50 randomly selected pairs of *filtered* images selected from the *testing* set using the filter learned according to the above procedure. Each pair contained 1 smiling image and 1 non-smiling image presented in random order (Left or Right), and the labeler was asked to select which image – Left or Right – was "smiling more". The entire set of 50 image pairs was presented to 10 AMT workers, and their opinions on each pair were combined using Majority Vote, with ties resolved by selecting the "Right" image. Accuracy of the AMT labelers compared to the official GENKI labels was measured as the probability of correctness on a 2 alternative forced choice task (2AFC), which is equivalent to the Area under the Receiver Operating Characteristics curve (A' statistic) that is commonly used in the automatic facial expression recognition literature (e.g., [9]). We similarly generated a set of 50 randomly selected pairs of filtered images containing 1 male and 1 female and asked AMT workers to select the image (Left or Right) appeared "more female". As a baseline, we compared gender and smile labeling accuracy of the filtered images to similar tasks for the *unfiltered* images.

Results are shown in Table 1 and indicate that the learned filter substantially reduced discriminability of gender (from 98% to 58%) while maintaining high discriminability of expression (94% to 96%) compared to the unfiltered images.

Comparison to manually constructed filter: In the case of expression and gender, one might reasonably argue that the "optimal filter" for preserving smile/non-smile and suppressing male/female information would be simply to crop and display only the mouth region of each face. Hence, we performed an additional experiment in which we compared Mechanical Turk labeling accuracy on 50 pairs of filtered images, generated similarly as described above, using a manually constructed mask filter consisting of just the mouth region (rows 11 through 15 and columns 4 through 13 of each 16×16 face image). Results are in Table 1: while smile discriminability is equally high as the learned filter 1, gender discriminability using the manually constructed filter was substantially higher (74% compared to 58%), indicating that the hand-crafted filter actually allowed considerable gender information to pass through. This suggests that a learned filter can work better than a manually constructed one even when strong prior domain knowledge exists.

5.2. Preserve gender, suppress expression

Analogously to Section 5.1, we also learned a filter to preserve *gender* and suppress *expression*, using an identical training procedure to that described above. Examples of the filtered (+ reconstructed) images ($\delta = 9 \times 10^{-3}$) are

Table 1. Accuracy (2AFC) of workers on Mechanical Turk when labeling filtered GENKI images

Filter method	**Expression**	**Gender**
Unfiltered (baseline)	94%	98%
Learned filter 1: Preserve expr., suppress gender	96%	58%
Manually constructed filter: show mouth region only	96%	74%
Learned filter 2: Preserve gender, suppress expr.	64%	86%

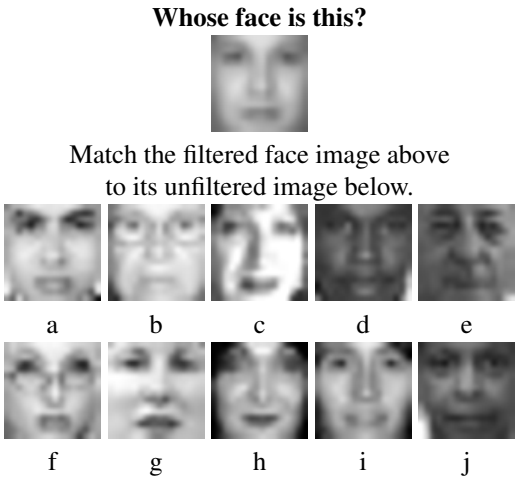

Figure 6. The preserve-smile, suppress-gender filter both allows smile/non-smile information to pass through, and also serves as a "face de-identification" mechanism, as illustrated in the face recognition task above. The correct face match is (f).

shown in Figure 5 (**right**). Notice how, for face image (b), the filter not only "suppressed" the expression of the non-smiling female, but actually seems to "flip" the smile/non-smile label so that the woman appears to be smiling. The accuracy compared to baseline (unfiltered) images is shown in Table 1. While accuracy of gender labeling did drop from 98% to 86%, it dropped much more for the smiling labeling (94% to 64%) compared to unfiltered images.

6. Experiment III: Preserving privacy in face images (face de-identification)

The filter learned in Section 5 to preserve smile while suppressing gender information was not designed specifically to suppress the faces' identity. In practice, however, we found that the identity of the people shown was very difficult to discern in the filtered images. Indeed, it is possible that gender represents one of the first "principal components" of face space, and that, by removing gender, one implicitly removes substantial identity information as well.

To test the hypothesis that identity was effectively masked by suppressing gender, we created a face recognition test consisting of 40 questions similar to Figure 6: a single face must be matched to one of 10 unfiltered candidate face images. In half of the questions, the face to be matched was *filtered* using the preserve-expression, suppress-gender filter (Section 5). In this case, the matching task was very challenging. In the other half of the questions, the face to be matched was *unfiltered*, and hence the matching task was nearly trivial. The order of the questions presented to the labelers was randomized, and we obtained results from 10 workers on the Amazon Mechanical Turk.

Results: For the *unfiltered* images, the rate of successful match was 100% for each of the 10 labelers. For the *filtered* images, the rate of successful match, using Majority Vote, was 15%, indicating that the preserve-smile, suppress-gender filter also removed identity. The highest successful matching rate of the filtered images for any one labeler was 30%. Baseline rate for guessing was 10%.

7. Experiment IV: Filtering to improve generalization across datasets

Here we show a proof-of-concept of learning a filter to improve generalization to novel datasets. Consider a dataset of face images such as GENKI with a positive correlation between gender and smile. If a gender classifier were trained on these data, then it might learn to distinguish gender not just by male/female information alone, but also by the correlated presence of smile. When tested on a dataset with different covariance structure, e.g., with negative correlation between smile and gender, the classifier would likely perform badly. If we first filter the data to suppress smile but preserve gender, then the performance of the trained classifier might not suffer when applied to the new dataset.

To test this hypothesis, we partitioned the GENKI images used in Section 5 into a training set (4062 images) and test set (970 images). As before, all images were 16×16 pixels. In the training set, the correlation between smile and gender was $+0.64$, whereas in the test set, it was -1. We then trained two support vector machine (SVM) classifiers with radial basis function (RBF) kernels to classify gender. One classifier was trained on *filtered* training images, using the gender-preservation, smile-suppression filter learned in Section 5.2, and the other was trained on *unfiltered* images. The RBF width γ was optimized independently ($\gamma \in \{10^{-8}, 10^{-7}, \ldots, 10^{+4}\}$) for each classifier using a holdout set (a randomly selected 20% subset of the training images). The SVM trained on unfiltered images was then applied to the unfiltered test set, and the SVM trained on filtered images was applied to the filtered test set.

Results: Filtering the data using the gender-preservation, smile-suppression filter increased gener-

alization performance substantially: 2AFC accuracy was 0.92 for the SVM trained on filtered images, whereas it was only 0.79 for the SVM trained on unfiltered images.

Comparison to LMNN: We also compared DDD to an existing supervised learning method for learning a data transformation to increase classification accuracy – Large Margin Nearest Neighbors (LMNN [15]). LMNN uses semidefinite programming to find a transformation L that decreases the distance between each data point and its k nearest neighbors of the same class, while maximally increasing the distance to data of a different class. It is conceivable that such an approach would also aid generalization across datasets of different covariate structure.

In our experiment, we used LMNN to learn a filter L to increase gender classification accuracy. The LMNN parameter k was selected to maximize gender classification accuracy on the holdout set after applying the learned data transformation L_k associated with k. Then, after fixing L for the best k, we re-trained an SVM on the transformed training set and then applied it to the transformed test set.

Results: Using LMNN to learn a filter to increase gender discriminability improved classification accuracy to 0.87 (2AFC). While this is better than 0.79 for the unfiltered images, it is still less than the 0.92 achieved by DDD.

8. Related work

We are unaware of any work that specifically learns filters to simultaneously preserve and suppress different image attributes. However, our approach is somewhat reminiscent of work by Birdwell and Horn [3], in which an optimal combination of a fixed set of filters is learned to minimize the conditional entropy of class labels given filtered inputs.

In terms of applications to data privacy, our method is related to "face de-identification" methods such as [11, 7, 6]. Such methods identify faces which are similar either in terms of pixel space ([11, 6]), eigenface space ([11]), or Active Appearance Model parameters ([7]), and then replace clusters of k similar faces with their mean face, thus guaranteeing that no face can be identified more specifically than to a cluster of k candidates. However, in contrast to our proposed algorithm, these methods cannot be "reversed" to maximally preserve identity while minimizing discriminability of a given face attribute.

For inter-dataset generalization, our work is related to covariate shift [13] and transfer learning [12]. The method proposed in our paper is useful when dataset differences are known a priori – the learned filter helps to overcome covariate shift by altering the underlying images themselves. In addition, our work is related to Subclass Discriminant Analysis [16] which partially overcomes covariate shift by learning important subclasses (e.g., smiling males) of a given task (e.g., male versus female).

9. Summary

We have presented a novel method for learning filters that can preserve binary discriminability of a target task while suppressing the discriminability of a distractor task. The effectiveness of the approach was demonstrated on natural face images. Interestingly, the suppression of gender implicitly removed considerable facial identity information, which renders the technique useful for labeling tasks where personal identity should remain private. Finally, we demonstrated that "discriminately decreasing discriminability" may help classifiers to generalize across datasets.

Matlab code for the DDD algorithm is available at `http://mplab.ucsd.edu/~jake`.

References

[1] A. Ashraf and S. Lucey. Re-interpreting the application of gabor filters as a manipulation of the margin in linear svms. *Pattern Analysis and Machine Intell.*, 2010. 1

[2] T. Bell and T. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 1997. 1

[3] J. Birdwell and R. Horn. Optimal filters for attribute generation and machine learning. In *Conference on Decision and Control*, 1990. 8

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

[5] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936. 3

[6] R. Gross, E. Airoldi, and L. Sweeney. Integrating utility into face de-identification. In *Workshop on Privacy-Enhancing Technologies*, 2005. 8

[7] R. Gross, L. Sweeney, F. de la Torre, and S. Baker. Model-based face de-identification. In *CVPR*, 2006. 8

[8] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009. 1

[9] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset: A complete dataset for action unit and emotion-specified expression. In *CVPR Workshop on Human-Comm. Behavior*, 2010. 6

[10] J. R. Movellan. Tutorial on gabor filters. `http://mplab.ucsd.edu/tutorials/gabor.pdf`. 3

[11] E. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *IEEE Transactions on knowledge and data engineering*, 2005. 1, 8

[12] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 8

[13] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 2000. 8

[14] `http://mplab.ucsd.edu`. MPLab GENKI Dataset. 5

[15] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Neural Information Processing Systems*, 2006. 8

[16] M. Zhu and A. Martinez. Subclass discriminant analysis. *Pattern Analysis and Machine Intell.*, 2006. 8