**RC24834 (W0908-018) August 3, 2009**
**Computer Science**

# IBM Research Report

## Closed-form Supervised Dimensionality Reduction with Generalized Linear Models

**Irina Rish, Genady Grabarnik, Guillermo Cecchi**
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

**Francisco Pereira**
CSBMB
Green Hall
Princeton University
Princeton, NJ 08540
USA

**Geoffrey J. Gordon**
Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15213
USA

**Research Division**
**Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich**

# Closed-Form Supervised Dimensionality Reduction with Generalized Linear Models

**Irina Rish, Genady Grabarnik, Guillermo Cecchi** {RISH,GENADY,GCECCHI}@US.IBM.COM
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA

**Francisco Pereira** FPEREIRA@PRINCETON.EDU
CSBMB, Green Hall, Princeton University, Princeton, NJ 08540 USA

**Geoffrey J. Gordon** GGORDON@CS.CMU.EDU
Carnegie Mellon University, School of Computer Science, Machine Learning Dept., Pittsburgh, PA 15213 USA

## Abstract

We propose a family of supervised dimensionality reduction (SDR) algorithms that combine feature extraction (dimensionality reduction) with learning a predictive model in a unified optimization framework, using data- and class-appropriate generalized linear models (GLMs), and handling both classification and regression problems. Our approach uses simple closed-form update rules and is provably convergent. Promising empirical results are demonstrated on a variety of high-dimensional datasets.

## 1. Introduction

Dimensionality reduction (DR) is a popular data-processing technique that serves the following two main purposes: it helps to provide a meaningful interpretation and visualization of the data, and it also helps to prevent overfitting when the number of dimensions greatly exceeds the number of samples, thus working as a form of regularization.

When our goal is prediction rather than an (unsupervised) exploratory data analysis, *supervised* dimensionality reduction (SDR) that combines DR with *simultaneously* learning a predictor can significantly outperform a simple combination of unsupervised DR with a subsequent learning of a predictor on the resulting low-dimensional representation (Pereira & Gordon, 2006; Sajama and Alon Orlitsky, 2005). The problem of supervised dimensionality reduction can be viewed as finding a *predictive structure*, such as a low-dimensional representation, which captures the information about the class label contained in the high-dimensional feature vector while ignoring the "noise".

However, existing SDR approaches are often restricted to specific settings, and can be viewed as jointly learning a *particular mapping* (most commonly, a linear one) from the feature space to a low-dimensional hidden-variable space, together with a *particular predictor* that maps the hidden variables to the class label. For example, SVDM (Pereira & Gordon, 2006) learns a linear mapping from observed to hidden variables, effectively assuming Gaussian features when minimizing sum-squared reconstruction loss; on the prediction side, it focuses on SVM-like binary classification using hinge loss. SDR-MM method of (Sajama and Alon Orlitsky, 2005) treats various types of features (e.g., binary and real-valued) but is limited to discrete classification problems, i.e. is not suitable for regression. Recent work on distance metric learning (Weinberger et al., 2005; Weinberger & Tesauro, 2007) treats both classification and regression settings, but is limited, like SVDM, to Gaussian features and linear mappings when learning Mahalanobis distances.

This paper approaches SDR in a more general framework that views both features and class labels as exponential-family random variables, and allows to mix-and-match data- and label-appropriate *generalized linear models*, thus handling both classification and regression, with both discrete and real-valued data[1]. It can be viewed as a discriminative learn-

Appearing in *Proceedings of the 25$^{th}$ International Conference on Machine Learning*, Helsinki, Finland, 2008. 

[1] In other words, the proposed framework provides nonlinear dimensionality reduction methods based on minimizing Bregman divergences that correspond to particular

ing based on minimization of conditional probability of class given the hidden variables, while using as a regularizer the conditional probability of the features given the low-dimensional hidden-variable "predictive" representation.

The main advantage of our approach, besides being more general, is using simple *closed-form* update rules when performing its alternate minimization procedure. This method yields a short Matlab code, fast performance, and is guaranteed to converge. The convergence property, as well as closed form update rules, result from using appropriate auxiliary functions bounding each part of the objective function (i.e., reconstruction and prediction losses). We exploit the additive property of auxiliary functions in order to combine bounds on multiple loss functions.

We perform a variety of experiments, both on simulated and real-life problems. Results on simulated datasets convincingly demonstrate that our SDR approach can discover underlying low-dimensional structure in high-dimensional noisy data, while outperforming SVM and SVDM, often by far, and practically always beating the unsupervised DR followed by learning a predictor. On real-life datasets, SDR approaches continue to beat the unsupervised DR by far, while often matching or somewhat outperforming SVM and SVDM.

## 2. SDR-GLM: Hidden-Variable Model

Let $X$ be an $N \times D$ data matrix with entries denoted $X_{nd}$ where $N$ is the number of i.i.d. samples, and $n$-th sample is a $D$-dimensional row vector denoted $\mathbf{x}_n$. Let $Y$ be an $N \times K$ matrix of class labels for $K$ separate prediction problems (generally, we will consider $K > 1$), where $j$-th column, $1 \leq j \leq K$, provides a set of class labels for the $j$-th prediction problem. Our supervised dimensionality approach relies on the assumption that each data point $\mathbf{x}_n$, $n = 1, ..., N$, is a noisy version of some "true" data point $\theta_\mathbf{n}$ which lives in a low-dimensional space, and that this hidden representation of the noisy data is actually predictive about the class label.

We will also assume, following ePCA (Collins et al., 2001), $(GL)^2M$(Gordon, 2002), logistic PCA (Schein et al., 2003) and related extensions of PCA, that noise in the features follows exponential-family distributions with natural parameters $\theta_\mathbf{n}$, with different members of the exponential family used for different dimensions[2], and that the noise is applied independently to each coordinate of $\mathbf{x}_n$. Namely, it is assumed that $N \times D$ parameter matrix $\Theta$ can be represented by a linear model in an $L$-dimensional ($L < D$) space:

$$\Theta_{nd} = \sum_{l=1}^{L} U_{nl} V_{ld} + \Delta_{X_d},$$

where the rows of the $L \times D$ matrix $V$ correspond to the basis vectors, the columns of the $N \times L$ matrix $U$ correspond to the coordinates of the "true points" $\theta_n$, $n = 1, ...N$ in the $L$-dimensional space spanned by those basis vectors, and $\Delta_X$ is the bias vector (corresponding to the empirical mean in PCA). However, to simplify the notation, we will include $\Delta_X$ as the $(L+1)$'s row of the matrix $V$ (i.e., $V_{L+1} = \Delta_X$), and add the $(L+1)$'s column of 1's to $U$, so that we can write $\Theta_X$ as a product of two matrices, $\Theta_{X_{nd}} = (UV)_{nd} = \sum_{l=1}^{L+1} U_{nl} V_{ld}$.

Given the natural parameter $\Theta_{nd}$, an exponential-family noise distribution is defined for each $X_{nd}$ by

$$\log P(X_{nd}|\Theta_{X_{nd}}) = X_{nd}\Theta_{X_{nd}} - G(\Theta_{X_{nd}}) + \log P_0(X_{nd}),$$

where $G_x(\Theta_{nd})$ is the *cumulant* or *log-partition* function that ensures that $P(X_{nd}|\Theta_{nd})$ sums (or integrates) to 1 over the domain of $X_{nd}$. This function uniquely defines a particular member of the exponential family, e.g., Gaussian, multinomial, Poisson, etc.

We can now view each row $U_n$ as a "compressed" representation of the corresponding data sample $\mathbf{x}_n$ that will be used to predict the class labels. We will again assume a noisy linear model for each class label $Y_k$ (column-vector in $Y$) where the natural parameter is represented by linear combination

$$\Theta_{Y_{nk}} = \sum_{l=1}^{L} U_{nl} W_{lk} + \Delta_{Y_k}$$

with linear coefficients $\mathbf{w} = (w_1, ..., w_L)$ and K-dimensional bias vector $\Delta_Y$. As for $\Theta_{X_{nd}}$, we will simplify the notation by including $\Delta_Y$ as the $(L+1)$'s row of the matrix $W$, and write $\Theta_{Y_{nk}} = (UW)_{nk} = \sum_{l=1}^{L+1} U_{nl} W_{lk}$. Using an appropriate type of exponential-family noise $P(Y_{nk}|\Theta_{Y_{nk}})$, we can handle both classification and regression problems. For

members of the exponential family of distribution, including linear (PCA-like) dimensionality reduction as a particular case for Gaussian variables.

[2] It is important to note that in case of standard (linear) PCA corresponding to Gaussian noise the parameters $\theta_n$ live in a linear subspace in the original data space, but for other types of exponential-family noise the low-dimensional space of parameters is typically a nonlinear surface in the original data space.

example, in case of binary classification, we can model $Y_{nk}$ as a Bernoulli variable with parameter $p_{nk}$ and the corresponding natural parameter $\Theta_{nk} = \log(\frac{p_{nk}}{1-p_{nk}})$, and use logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ to write the Bernoulli distribution for $Y_{nk}$ as

$$P(Y_{nk}|\Theta_{Y_{nk}}) = \sigma(\Theta_{Y_{nk}})^{Y_{nk}}\sigma(-\Theta_{Y_{nk}})^{1-Y_{nk}}.$$

In case of regression, $Y_{nk}$ will be a Gaussian variable with the expectation parameter coinciding with the natural parameter $\Theta_{Y_{nk}}$.

In other words, we will use a *generalized linear model (GLM)* $E(X_d) = f_d^{-1}(UV_d)$ for each feature $X_d$ ($d$-th column in $X$, $1 \le d \le D$), and yet another set of GLMs $E(Y_k) = f_k^{-1}(UW)$ for each class label $Y_k$ ($k$-th column in $Y$, $1 \le k \le K$), with possibly different *link functions* $f_d$ and $f_k$ (e.g., the logit link function $f(\mu) = \ln\frac{\mu}{1-\mu} = \sigma^{-1}(\theta)$ is used for binary classification, and identity link function $f(\mu) = \mu$ is used for real-valued regression with Gaussian output).

**SDR-GLM optimization problem**. We formulate SDR as joint optimization of two objective functions corresponding to the reconstruction accuracy (data likelihood) and the prediction accuracy (class likelihood):

$$\mathcal{L}_X(\Theta_X) = \sum_{nd} \log P(X_{nd}|\Theta_{X_{nd}}), \quad (1)$$

$$\mathcal{L}_Y(\Theta_Y) = \sum_{nk} \log P(Y_{nk}|\Theta_{Y_{nk}}), \quad (2)$$

where $\Theta_X = UV$, $\Theta_Y = UW$, and the likelihoods above correspond to particular members of exponential family. Then the SDR optimization problem is

$$\max_{U,V,W} \alpha\mathcal{L}_X(UV) + \mathcal{L}_Y(UW) \quad (3)$$

where the data likelihood can be viewed as a regularization added on top of the class likelihood maximization objective, with the regularization constant $\alpha$ controlling the trade-off between the two likelihoods [3].

**Comparison with SVDM**. Note that the idea of combining loss functions for SDR was also proposed before in SVDM (Pereira & Gordon, 2006), where, similarly to SVD, quadratic loss $||X - UV||_2^2$ was used for data reconstruction part of the objective, while the hinge loss was used for the prediction part (using $U$ as the new data matrix). Herein, we generalize SVDM's sum-squared reconstruction loss to log-likelihoods of exponential family, similarly to ePCA(Collins et al., 2001) and $G^2L^2M$(Gordon, 2002), replace hinge loss by the loglikelihoods corresponding to exponential-family class variables, and provide closed-form update rules rather than perform optimization at each iteration, which results into a significant speed up when compared with SVDM.

[3]In our implementation, an $L_2$-norm regularization was also added on all matrices $U, V, W$ with small regularization constants (0.01), effectively corresponding to assuming Gaussian priors on those matrices.

## 3. Combining Auxiliary Functions

Since the above problem (eq. 3) is not jointly convex in all parameters, finding a globally optimal solution is nontrivial. Instead, we can employ the auxiliary function approach commonly used in EM-style algorithms, and using auxiliary function of a particular form, derive *closed-form* iterative update rules that are guaranteed to converge to a local minimum. We show that an auxiliary function for the objective in eq. 3 can be easily derived for an arbitrary pair of $\mathcal{L}_X$ and $\mathcal{L}_Y$ provided that we know their corresponding auxiliary functions.

**Auxiliary functions.** Given a function $\mathcal{L}(\theta)$ to be maximized, a function $Q(\hat{\theta}, \theta)$ is called an *auxiliary function* for $\mathcal{L}(\theta)$ if $\mathcal{L}(\theta) = Q(\theta, \theta)$ and $\mathcal{L}(\hat{\theta}) \ge Q(\hat{\theta}, \theta)$ for all $\hat{\theta}$. It is easy to see that $\mathcal{L}(\theta)$ is non-decreasing under the update

$$\theta^{t+1} = \arg\max_{\hat{\theta}} Q(\hat{\theta}, \theta^t),$$

i.e., $\mathcal{L}(\theta^{t+1}) \ge \mathcal{L}(\theta^t)$, and thus an iterative application of such updates is guaranteed to converge to a local maximum of $\mathcal{L}$ under mild conditions on $\mathcal{L}$ and $Q$.

We will make use of the following properties of auxiliary functions:

**Lemma 1** *Let $Q_1(\hat{\theta}, \theta)$ and $Q_2(\hat{\theta}, \theta)$ be auxiliary functions for $\mathcal{L}_1(\theta)$ and $\mathcal{L}_2(\theta)$, respectively. Then*

$$Q(\hat{\theta}, \theta) = \alpha_1 Q_1(\hat{\theta}, \theta) + \alpha_2 Q_2(\hat{\theta}, \theta) \quad (4)$$

*is an auxiliary function for $\mathcal{L}(\theta) = \alpha_1\mathcal{L}_1(\theta) + \alpha_2\mathcal{L}_2(\theta)$, where $\alpha_i > 0$ for $i = 1, 2$.*

**Proof.** $Q(\hat{\theta}, \theta) = \alpha_1 Q_1(\hat{\theta}, \theta) + \alpha_2 Q_2(\hat{\theta}, \theta) \le \alpha_1\mathcal{L}_1(\hat{\theta}) + \alpha_2\mathcal{L}_2(\hat{\theta}) = \mathcal{L}(\hat{\theta})$, and $Q(\theta, \theta) = \alpha_1 Q_1(\theta, \theta) + \alpha_2 Q_2(\theta, \theta) = \alpha_1\mathcal{L}_1(\theta) + \alpha_2\mathcal{L}_2(\theta) = \mathcal{L}(\theta)$. ■

Also, it is obvious that a function is an auxiliary for itself, i.e. $Q(\hat{\theta}, \theta) = \mathcal{L}(\hat{\theta})$ is an auxiliary function for $\mathcal{L}(\theta)$. This observations allows us to combine various dimensionality reduction approaches with appropriate predictive loss functions, given appropriate auxiliary functions for both (next section discusses two such combinations). When optimization of an auxiliary functions yields an analytical solution (e.g., for

quadratic functions), it is easy to obtain closed-form update rules for alternating minimization.

## 4. Iterative Update Rules

Recall that natural parameters for $X_{nd}$ and $Y_{nk}$ variables are represented by $\Theta_{X_{nd}} = \sum_{l=1}^{L+1} U_{nl}V_{ld}$ and $\Theta_{Y_{nk}} = \sum_{l=1}^{L+1} U_{nl}W_{lk}$. Let $Q_X(\hat{\Theta}_X, \Theta_X)$ and $Q_Y(\hat{\Theta}_Y, \Theta_Y)$ be the auxiliary functions for the corresponding loglikelihoods in eq. 2, then $Q(\hat{\Theta}_X, \Theta_X, \hat{\Theta}_Y, \Theta_Y) =$

$$= \alpha Q_X(\hat{\Theta}_X, \Theta_X) + Q_Y(\hat{\Theta}_Y, \Theta_Y) \quad (5)$$

is an auxiliary function for the combined log-likelihood in eq. 3 when we fix two out of three parameters and optimize over the remaining one. The update rules for $\hat{U}_{nl}$, $\hat{V}_{ld}$ and $\hat{W}_{nk}$ are obtained by solving

$$\frac{\partial Q}{\partial \hat{U}_{nl}} = 0,\ \frac{\partial Q_X}{\partial \hat{V}_{ld}} = 0,\ \frac{\partial Q_Y}{\partial \hat{W}_{nk}} = 0,\ where$$

$$\frac{\partial Q}{\partial \hat{U}_{nl}} = \alpha \sum_d \frac{\partial Q}{\partial \hat{\Theta}_{X_{nd}}} \frac{\hat{\Theta}_{X_{nd}}}{\partial \hat{U}_{nl}} + \sum_k \frac{\partial Q}{\partial \hat{\Theta}_{Y_{nk}}} \frac{\hat{\Theta}_{Y_{nk}}}{\partial \hat{U}_{nl}}.$$

Note that we get simpler expressions for $V$ and $W$ since they appear only in $Q_X$ or only in $Q_Y$ parts of eq. 5, respectively.

**Auxiliary functions for Bernoulli and Gaussian log-likelihoods**. For a Bernoulli variable $s$ with natural (log odds) parameter $\theta$ we use the variational bound on log-likelihood $\mathcal{L}(\theta) = \log P(s|\theta)$ proposed by (Jaakkola & Jordan, 1997) and used later by (Schein et al., 2003)

$$\mathcal{L}(\hat{\theta}) \geq Q(\hat{\theta}, \theta) = \log 2 - \log\cosh(\theta/2) + + \frac{T\theta^2}{4} + \frac{(2s-1)\hat{\theta}}{2} - \frac{T\hat{\theta}^2}{4}, \quad (6)$$

where $T = \frac{tanh(\theta/2)}{\theta}$. Note that the auxiliary function is quadratic in $\theta$ and taking its derivatives leads to simple closed-form iterative update rules for $U$, $V$ and $W$. For multinomial variables, one can use a recent extension of the above bound to multinomial logistic regression proposed by (Bouchard, 2007).

For a Gaussian variable $s$ with natural parameter $\theta$ that coincides with the mean parameter (identity link function) we do not really have to construct an auxiliary function, since we can simply use the negative squared loss proportional to the Gaussian loglikelihood as an auxiliary function itself, i.e.

$$\mathcal{L}(\hat{\theta}) = Q(\hat{\theta}, \theta) = -c(s-\theta)^2, \quad (7)$$

where $c = (2\sigma)^{-1}$ is a constant, assuming a fixed standard deviation that will not be a part of our estimation procedure here, similarly to the approach of (Collins et al., 2001) and related work; $c$ can be ignored since it will be subsumed by the parameter $\alpha$.

Using the above auxiliary functions, we can combine them into joint auxiliary functions as in eq. 5 for various combinations of Bernoulli and Gaussian variables $X_{nd}$ and $Y_{nk}$. Namely, assuming all $X_{nd}$ are Bernoulli, we get (Schein et al., 2003):

$$Q_X^{Ber}(\hat{\Theta}_{X_{nd}}, \Theta_{X_{nd}}) = \sum_{nd} \log\cosh(\Theta_{X_{nd}}/2) + + \frac{T\Theta_{X_{nd}}^2}{4} + \frac{(2X_{nd}-1)\hat{\Theta}_{X_{nd}}}{2} - \frac{T\hat{\Theta}_{X_{nd}}^2}{4}, \quad (8)$$

while assuming all $X_{nd}$ are Gaussian, we get

$$Q_X^{Gauss}(\hat{\Theta}_{X_{nd}}, \Theta_{X_{nd}}) = -\sum_{nd}(X_{nd} - \hat{\Theta}_{X_{nd}})^2. \quad (9)$$

Note that $Q_Y(\hat{\Theta}_{Y_{nk}}, \Theta_{Y_{nk}})$ for all-Bernoulli or all-Gaussian $Y_{nk}$ is obtained by replacing $X$ with $Y$, $V$ with $W$, and $d$ with $k$ in eq. 8 and 9, respectively.

Due to lack of space, we omit the derivation of the iterative update rules for the four versions of SDR-GLM that we experimented with (for more detail, see (Rish et al., 2008)): *Gaussian-SDR*, that assumes Gaussian $X_{nd}$ and Bernoulli $Y_{nk}$, *Bernoulli-SDR* in case of Bernoulli $X_{nd}$ and Bernoulli $Y_{nk}$, *Gaussian-SDR-Regression* and *Bernoulli-SDR-Regression* in case of Gaussian $Y_{nk}$ (appropriate for real-valued label, i.e. for the regression problem).

**Prediction step**. Once we learn the parameters $U$, $V$ and $W$, and thus $\Theta_X$ and $\Theta_Y$, we can use them on test data in order to predict labels for previously unseen data. Given the test data $X^{test}$, we only need to find the corresponding low-dimensional representation $U^{test}$ by performing the corresponding iterative updates only with respect to $U$, keeping $V$ and $W$ fixed. Once $U^{test}$ is found, we predict the class labels as $Y^{test} = U^{test}W$.

## 5. Empirical Evaluation

We evaluated our SDR-GLM methods on both simulated and real-life datasets, in both classification and regression settings. We varied the low-dimensionality parameter $L$ from 2 to 10, and evaluated a range of regularization parameters $\alpha = 0.0001, 0.001, 0.01, 0.1, 1, 10, 100$, selecting those giving the best average error among several dimensions[4].

[4]Selecting the best $\alpha$ separately for each dimension can only improve the results, but would be more computationally intensive.

### 5.1. Classification Problems

In the classification setting, we compared *Bernoulli-SDR* and *Gaussian-SDR* versus linear SVM[5] and versus unsupervised dimensionality reduction followed by SVM and by logistic regression, which we refer to as SVM-UDR and *Logistic-UDR*, respectively. In both cases, unsupervised dimensionality reduction is performed first using the data-appropriate DR method (i.e., PCA for real-valued data and Logistic-PCA for binary data; this is equivalent to removing the prediction loss in eq. 3); then SVM or logistic regression are performed on the low-dimensional data representation $U$. For datasets with real-valued features, we also compared the above methods to SVDM (Pereira & Gordon, 2006). We performed $k$-fold cross-validation with $k = 10$ on the datasets with less than 1000 dimensions, and with $k = 5$ otherwise.

*Simulated data.* In order to test our methods first on the data with controllable low-dimensional structure, we simulated high-dimensional datasets that indeed were noisy versions of some underlying easily-separable low-dimensional data. Particularly, a set of $N = 100$ samples from two classes was generated randomly in two-dimensional space so that the samples were linearly separable with a large margin.

Next, we simulated two sets of exponential-family random variables $X_{nd}$, a Bernoulli set and a Gaussian set, using the coordinates of the above points for natural parameters $\Theta_{nd}$, where the number of samples was $N = 100$ and the dimensionality of a "noisy" dataset varied from $D = 100$ to $D = 1000$. We then combined the data with the labels generated in the low-dimensional space and provided them as an input to our algorithms.

*Simulated data: Bernoulli noise.* Figures 1a summarize the results for Bernoulli-noise dataset. Supervised DR methods (Bernoulli-SDR and Gaussian-SDR) versus SVM and versus unsupervised DR followed by learning a predictor (Logistic-UDR and SVM-UDR); the reduced dimensionality parameters is set to $L = 2$ and the regularization constant is $\alpha = 0.0001$ (the choice of those parameters is discussed below). We can see that Bernoulli-SDR significantly outperforms all other methods, including SVM, and both Bernoulli-SDR and Gaussian-SDR also outperform the unsupervised DR followed by either logistic regression or SVM. Apparently, Bernoulli-SDR is able to reconstruct correctly the underlying separable two-dimensional dataset and is robust to noise, as its error remains zero for up to 700 dimensions, and only increases slightly up to 0.05 for 1000 dimensions. On the other hand, SVM has zero-error prediction only in the lowest-dimensional case ($D = 100$), and is much more sensitive to noise when the dimensionality of the data increases, making incorrect predictions in up to 14% to 21% cases when the dimensionality increases above $D = 300$. Apparently, SVM was not able to exploit the underlying separable low-dimensional structure disturbed by high-dimensional noise, while supervised dimensionality reduction easily detected this structure.

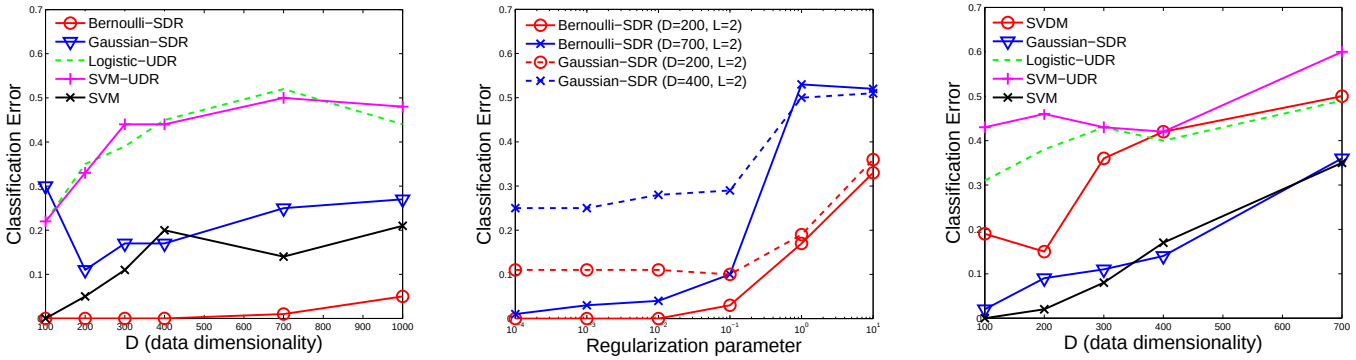
Also, using the Bernoulli model instead of Gaussian when features are binary is clearly beneficial, and thus, as noted previously, proper extensions of PCA to exponential-family data must be used (Collins et al., 2001; Schein et al., 2003). However, previous work on logistic PCA by (Schein et al., 2003) demonstrated advantages of using the Bernoulli vs Gaussian assumption only for *reconstruction* of the original data, while this paper investigates the impact of such assumptions on the *generalization* error in supervised learning case. This is less obvious, since a good fit to the training data does not necessarily imply a good generalization ability, as shown by our experiments with unsupervised dimensionality reduction followed by learning a classifier.

Regarding the choice of the regularization parameter $\alpha$, we experimented with a range of values from 0.0001 to 10, and concluded that the smallest value was the most beneficial for both SDR algorithms; this is intuitive since it effectively puts most of the weight on the predictive loss. There is a clear trend (Figure 1b) in error decrease with the parameter decrease, where sufficiently low values of $\alpha \leq 0.1$ yield quite similar low errors, but increasing $\alpha$ further, especially above 1, leads to a drastic increase in the classification error, especially in higher-dimensional cases. Note, however, that such tendency is not present in the other datasets we experimented with, where the effect of regularization constant can be non-monotonic, and thus underscores the importance of using cross-validation or other parameter-tuning approaches[6].

Regarding the choice of the reduced-dimensionality parameter $L$, for low values of $\alpha$, we did not observe any significant variation in the results with increasing this parameter up to $L = 10$, e.g. the results for different $L$ were practically identical when $\alpha \leq 0.1$, while for higher values variance was more significant.

[5]We used the SVM code by A. Schwaighofer available at http://ida.first.fraunhofer.de/~anton/software.html.

[6]Bayesian approach to selecting regularization parameter may prove beneficial, as shown, for example, in (Y. Lin and D. Lee, 2006)




(a) (b) (c)

*Figure 1.* (a) Results for Bernoulli noise. (b) Effects of the regularization parameter $\alpha$ (the weight on the data reconstruction loss) - Bernoulli noise. (c) Results for Gaussian noise.

*Simulated data: Gaussian noise.* Figure 1c shows the results for the Gaussian noise: supervised dimensionality reduction provides a clear advantage over unsupervised ones, although the performance of SDR versus SVM is somewhat less impressive, as Gaussian-SDR is comparable to SVM. However, Gaussian-SDR seems to outperform considerably another supervised dimensionality method, SVDM, proposed by (Pereira & Gordon, 2006). SVDM was used with its regularization constant set to 100 since it provided the best SVDM performance among the same values of $\alpha$ as before. Interestingly, the performance of SVDM does not show any monotonic dependence on this parameter.

*Real-life datasets.* First, we considered several datasets with *binary features*, such as a 41-dimensional *Sensor network* dataset where the data represent connectivity (0 or 1) between all pairs of sensor nodes in a network of 41 light sensors (see Table 1). Note that Bernoulli-SDR with $L = 10$ and regularization parameter $\alpha = 0.1$ outperformed SVM, Gaussian-SDR with $L = 8, 10$ and same $\alpha = 0.1$ matched SVM performance, while both the unsupervised dimensionality reduction methods followed by SVM and logistic regression - *SVM-UDR* and *Logistic-UDR*, respectively - performed worse. (Best results for each method are shown in the boldface.) Also, using the Bernoulli model for binary data instead of Gaussian seems to pay off: Bernoulli-SDR performs somewhat better than Gaussian-SDR. It is interesting to note that for really low dimensionality $L = 2$, all of the above methods have same error of 0.2, while increasing the dimensionality allows for much better performance, although this effect is non-monotonic.

Another dataset related to network performance management, of somewhat larger dimensionality ($N = 169$, $D = 168$), contains pairwise end-to-end network latency (ping round-trip times) collected by the PlanetLab measurement project (http://www.pdos.lcs.mit.edu/∼simstrib/pl_app) discretized by applying a threshold as follows: above the average latency is considered "bad" (1) while the below-average latency is considered "good" (0). We selected the first column (latencies of all 169 nodes towards the node 1) as the label, and predicted them given the remaining data. The results are shown in Table 2. The regularization parameters $\alpha$ selected by cross-validation were different here for different SDR methods: for Bernoulli-SDR, $\alpha = 100$ turned out to be the best, while Gaussian-SDR performed better with $\alpha = 1$. Overall, the results for different methods and

*Table 1.* Results for *Sensor network* dataset ($N = 41, D = 41$): classification errors of different methods for different reduced dimension parameter, $L$.

| *method*\ L | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| *Bernoulli-SDR* | 0.20 | 0.24 | 0.27 | 0.15 | **0.12** |
| *Gaussian-SDR* | 0.20 | 0.22 | 0.22 | **0.17** | 0.17 |
| *Logistic-UDR* | **0.20** | 0.22 | 0.20 | 0.20 | 0.24 |
| *SVM-UDR* | **0.20** | 0.27 | 0.22 | 0.27 | 0.24 |
| SVM | 0.17 | | | | |

*Table 2.* Results for *PlanetLab* dataset ($N = 169, D = 168$): classification errors of different methods for different reduced dimension parameter, $L$.

| *method*\ L | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| *Bernoulli-SDR* | 0.10 | **0.07** | 0.10 | 0.12 | 0.15 |
| *Gaussian-SDR* | 0.10 | 0.11 | 0.08 | **0.07** | 0.08 |
| *Logistic-UDR* | 0.11 | 0.10 | **0.08** | 0.09 | 0.10 |
| *SVM-UDR* | 0.10 | 0.09 | 0.08 | 0.08 | **0.07** |
| SVM | 0.10 | | | | |

*Table 3.* Results for *Mass-spectrometry* dataset ($N = 50, D = 38573$): classification errors of different methods for different reduced dimension parameter, $L$.

| *method*\ L | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| *Gaussian-SDR* | 0.04 | 0.02 | 0.02 | 0.02 | 0.02 |
| *Logistic-UDR* | 0.5 | 0.18 | 0.08 | 0.04 | 0.02 |
| *SVM-UDR* | 0.54 | 0.2 | 0.02 | 0.06 | 0.02 |
| *SVDM* | 0.42 | 0.04 | 0.02 | 0.06 | 0.04 |
| SVM | 0.02 | | | | |

*Table 4.* Results for *fMRI* dataset ($N = 84, D = 14043$): classification errors of different methods for different reduced dimension parameter, $L$.

| *method*\ L | 5 | 10 | 15 | 20 | 25 |
|---|---|---|---|---|---|
| *Gaussian-SDR* | 0.21 | 0.26 | 0.23 | **0.20** | 0.23 |
| *Logistic-UDR* | 0.44 | 0.42 | **0.29** | 0.30 | 0.26 |
| *SVM-UDR* | **0.49** | 0.52 | 0.56 | 0.57 | 0.55 |
| *SVDM* | 0.32 | 0.25 | **0.21** | 0.23 | 0.23 |
| SVM | 0.21 | | | | |

varying dimensions $L$ were surprisingly similar to each other, with both SDR methods achieving the lowest error of 0.07 for $L = 4$ and $L = 8$, respectively, that was also achieved by SVM-UDR at $L = 10$, slightly outperforming SVM (0.10 error). Very similar results (not shown here due to lack of space) were also obtained on the *Advertisement* dataset from UCI ML repository.

We experimented next with several extremely high-dimensional datasets from biological experiments that had *real-valued features*. The first dataset, called here *Proteomics data*, containing mass-spectrometry data for $D = 38573$ proteins (features), showing their "expression levels" in $N = 50$ female subjects (examples), 25 of which were pregnant (class 1), and the others were not (class 0). The results are shown in Table 3, comparing Gaussian-SDR with $\alpha = 0.001$ (that yields lowest average SVDM error (among all dimensions) on this dataset) versus SVM, Logistic-UDR, SVM-UDR (both using the Gaussian assumption, i.e. PCA, for dimensionality reduction), and SVDM with its best-performing parameter 0.01. Despite its very high dimensionality, this dataset turned out to be easy: both SVM and Gaussian-SDR achieved an error of 0.02, and Gaussian-SDR used only $L = 4$ dimensions to achieve it. On the contrary, unsupervised DR followed by predictor learning (Logistic-UDR and SVM-UDR), suffered from really high errors at low dimensions, and only managed to achieve same low error at $L = 10$. SVDM was able to reach its lowest error (same 0.02) a bit earlier ($L = 6$), although for $L = 2$ it incurred a huge error of 0.42, while Gaussian-SDR had 0.04 error at that same level of reduced dimensionality.

Another truly high-dimensional dataset we used contained the fMRI recordings of subject's brain activity (measured using changes in the brain oxygenation levels) while the subject was observing on a screen words of two types, representing tools or buildings (see (Pereira & Gordon, 2006) for details). The task was to learn a mind-reading classifier that would predict, given the fMRI data, what type of the word the subject was looking at. The features here correspond to fMRI voxels ($D = 14043$ voxels were selected after some preprocessing of the data, as described in (Pereira & Gordon, 2006)), and there are $N = 84$ samples (i.e., word instances presented to a subject). This dataset was clearly more challenging then the previous one (both SVM's and SDR's errors were around 0.2).

The results for all methods are shown in Table 4; for Gaussian-SDR we used $\alpha = 0.0001$, while SVDM was best at 0.001 (as before, we used the average error over all dimensions to select best $\alpha$). For those values of $\alpha$ parameter, Gaussian-SDR matches SVM's errors of 0.21 using just five dimensions ($L = 5$), while SVDM reaches same error at $L = 15$ dimensions. Again, learning a predictor on "compressed" data obtained via *unsupervised* dimensionality reduction was consistently worse than both supervised methods.

### 5.2. Regression Problems

Finally, we did some preliminary experiments with the regression version of our SDR approach that makes Gaussian assumption about the label (response variable) $Y$ and thus uses sum-squared predictive $\mathcal{L}_Y(UW)$ loss in equation 3, comparing it with the state-of-art sparse-regression technique called the *Elastic Net*, which was shown to improve over both *Lasso* and *ridge* regression using a convex combination of the $L1$- and $L2$-norm regularization terms (Zou & Hastie, 2005).

We used the fMRI data from the 2007 Pittsburgh Brain Activity Interpretation Competition (PBAIC)(Pittsburgh EBC Group, 2007), where the fMRI data were recorded while subjects were playing a videogame, and the task was to predict several real-valued response variables, such as level of anxiety the subject was experiencing etc. We used the data for the two runs (games) by the first subject, and for the *Instructions* response variable, learning from run 1 and predicting on run 2. The dataset contained $N = 704$ samples (measurements over time) and approximately $D = 33,000$ features (voxels). Figure 2 compares the performance of Elastic Net and Gaussian-SDR-Regression, where the $L$ parameter denotes the number of active variables (voxels selected)

by the sparse EN regression. We can see that SDR regression is comparable with the state-of-the art EN (or even slightly better) when the number of hidden dimensions is not too low.

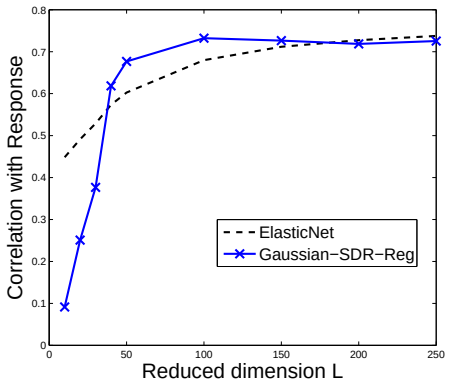



*Figure 2.* Results for the fMRI dataset from PBAIC. Performance is measured by correlation between the actual and predicted response variable (*Instructions*, in this case).

## 6. Conclusions and Future Work

This paper proposes a family of SDR algorithms that use generalized linear models to handle various types of features and labels, thus generalizing previous approaches in a unifying framework. Our SDR-GLM approach is fast and simple: it uses closed-form update rules at each iteration of alternating minimization procedure, and is always guaranteed to converge. Experiments on a variety of datasets show that this approach is clearly promising, although more empirical investigation is needed in case of SDR-regression.

Although we only tested our approach with Gaussian and Bernoulli variables, it can be easily extended to multinomial variables (and thus to multiclass classification) using a recent extension of the variational bound proposed in (Jaakkola & Jordan, 1997) to multinomial logistic regression (soft-max)(Bouchard, 2007). Deriving closed-form update rules for other members of the exponential family (e.g., Poisson) remains a direction of future work. Another possible extension is to obtain closed-form SDR update rules for alternative DR methods, such as non-negative matrix factorization (NMF)(Lee & Seung, 2000), simply plugging in NMF's auxiliary function instead of (unconstrained) PCA-like quadratic-loss.

Other potential applications of our approach include dimensionality reduction on mixed-type data, weighted dimensionality reduction schemes (e.g., assigning different weight to reconstruction error of different coordinates in PCA and similar DR techniques), multitask learning, as well as semi-supervised learning, including only the reconstruction loss part of the objective for unlabeled date, while keeping both reconstruction and prediction losses for the labeled ones. Finally, a more principled selection of the regularization constant $\alpha$ (e.g., using Bayesian approaches) is another open research direction.

## References

Bouchard, G. (2007). Efficient bounds for the softmax and applications to approximate inference in hybrid models. In *Nips workshop on approximate inference in hybrid models*.

Collins, M., Dasgupta, S., & Schapire, R. (2001). A generalization of principal component analysis to the exponential family. *NIPS2001*.

Gordon, G. (2002). Generalized2 Linear2 Models. *NIPS2002* (pp. 577–584).

Jaakkola, T., & Jordan, M. (1997). A variational approach to bayesian logistic regression problems and their extensions. *Sixth International Workshop on Artificial Intelligence and Statistics*.

Lee, D. D., & Seung, S. H. (2000). Algorithms for non-negative matrix factorization. *NIPS* (pp. 556–562).

Pereira, F., & Gordon, G. (2006). The Support Vector Decomposition Machine. *ICML2006* (pp. 689–696).

Pittsburgh EBC Group (2007). PBAIC Homepage: http://www.ebc.pitt.edu/2007/competition.html.

Rish, I., Grabarnik, G., Cecchi, G., Pereira, F., & Gordon, G. (2008). *Closed-Form Supervised Dimensionality Reduction with Generalized Linear Models* (Technical Report). IBM T.J. Watson Research Center.

Sajama and Alon Orlitsky (2005). Supervised dimensionality reduction using mixture models. *ICML '05: Proceedings of the 22nd International Conference on Machine learning* (pp. 768–775). New York, NY, USA: ACM.

Schein, A., Saul, L., & Ungar, L. (2003). A generalized linear model for principal component analysis of binary data. *Ninth International Workshop on Artificial Intelligence and Statistics*.

Weinberger, K., Blitzer, J., & Saul, L. (2005). Distance Metric Learning for Large Margin Nearest Neighbor Classification. *NIPS2005*.

Weinberger, K., & Tesauro, G. (2007). Metric learning for kernel regression. In M. Meila and X. Shen (Eds.), *Eleventh international conference on artificial intelligence and statistics*, 608–615. Puerto Rico: Omnipress.

Y. Lin and D. Lee (2006). Bayesian L1-norm sparse learning. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2006)*.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B*, *67*, 301–320.