Learning with Marginalized Corrupted Features

Laurens van der Maaten LVDMAATEN@GMAIL.COM
Delft University of Technology, Mekelweg 4, 2628 CD Delft, THE NETHERLANDS

Minmin Chen MC15@CEC.WUSTL.EDU
Stephen Tyree SWTYREE@WUSTL.EDU
Kilian Q. Weinberger KILIAN@WUSTL.EDU
Washington University, St. Louis, MO 63130, USA

Abstract

The goal of machine learning is to develop predictors that generalize well to test data. Ideally, this is achieved by training on very large (infinite) training data sets that capture all variations in the data distribution. In the case of finite training data, an effective solution is to extend the training set with artificially created examples—which, however, is also computationally costly. We propose to corrupt training examples with noise from known distributions within the exponential family and present a novel learning algorithm, called *marginalized corrupted features* (MCF), that trains robust predictors by minimizing the expected value of the loss function under the corrupting distribution—essentially learning with *infinitely* many (corrupted) training examples. We show empirically on a variety of data sets that MCF classifiers can be trained efficiently, may generalize substantially better to test data, and are more robust to feature deletion at test time.

1. Introduction

In the hypothetical scenario with infinite data drawn from the data distribution $\mathcal{P}$, even a simple classifier such as nearest neighbor (Cover & Hart, 1967) becomes close to optimal (*viz.* its error is twice the Bayes error). In the more realistic scenario with a finite training set, some variations in the data distribution will not be captured and the learned classifier performs worse at test time than during training. In this paper, we propose an algorithm to train a classifier from *infinite* training data, by corrupting the existing finite training examples with a fixed noise distribution. So instead of approximating the exact statistics of $\mathcal{P}$ with *finite* data, we approximate a slightly modified data distribution $\mathcal{P}'$ with *infinite* data.

Our augmented data distribution $\mathcal{P}'$ follows a simple stochastic rule: pick one of the finite training examples uniformly at random and transform it with some predefined corrupting distribution. Many corrupting distributions are possible, but we focus on: i) Poisson corruption and ii) blankout / dropout corruption (random deletion of features). The Poisson corruption model is of interest when the data comprises count vectors, *e.g.*, in document classification. It is particularly appealing as it introduces *no additional hyper-parameters* and, in our results, improves the test accuracy on almost all data sets and loss functions. Robustness to blankout corruption is of interest in settings with heavy-tailed feature distributions, and in the "nightmare at test time" scenario (Globerson & Roweis, 2006) in which some of the features are blanked out during testing (*e.g.*, due to sensors failing or because the feature computation exceeds a time budget).

Previous work (Burges & Schölkopf, 1997) explicitly augments the training set with additional examples that are corrupted through similar transformations. Although the simplicity of such an approach is appealing, it lacks elegance and the computational cost of processing the additional corrupted training examples is prohibitive for most real-world problems. We show that it is efficient to train predictors on an *infinite* amount of corrupted copies of the training data by marginalizing out the corrupting distribution. In particular, we focus on empirical risk minimization and derive analytical solutions for a large family of corrupting distributions and a variety of loss functions.

Appearing in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013.

In summary, we make the following contributions: i) we introduce learning with *marginalized corrupted features* (MCF), a framework that regularizes classifiers by marginalizing out feature corruptions; ii) we derive analytical solutions for quadratic, exponential, and logistic loss functions for a range of corrupting distributions; and iii) on several real-world data sets, we show that training with MCF may lead to better classifiers than training with common l_1 or l_2-norm regularizers.

2. Related work

Next to work that *explicitly* corrupts training data (Burges & Schölkopf, 1997), several prior studies consider *implicit* approaches to classifying objects that are subject to corruptions. Most of these studies minimize the loss under an adversarial worst-case scenario. In particular, Globerson & Roweis (2006) and Dekel & Shamir (2008) propose minimax formulations in which the *maximum* loss of an example over all $\binom{D}{K}$ possible corrupted examples that blank out K features is *minimized*. Others (Bhattacharyya et al., 2004; Shivaswamy et al., 2006; Trafalis & Gilbert, 2007; Xu et al., 2009) also use minimax approaches that minimize the loss under a worst-case scenario, but corrupt the data by adding a constant that is uniformly drawn from $[-u, u]$ to each of the features.

Chechik et al. (2008) propose an algorithm that maximizes the margin in the subspace of the observed features for each training instance to deal with randomly deleted features. Teo et al. (2008) generalize the worst-case scenario to obtain invariances to transformations such as image rotations or translations. Their framework incorporates several prior formulations on learning with invariants as special cases; *e.g.*, Herbrich & Graepel (2004), who generalized SVMs to be invariant under polynomial input transformations. Brückner et al. (2012) study a worst-case scenario in which an adversary changes the data distribution to minimize a function that may be antagonistic to the loss.

Prior work differs from MCF in that the corruption is not computed analytically in expectation. In particular, existing approaches have two disadvantages: i) they are complex and computationally expensive and ii) they minimize the loss of a worst-case scenario that is unlikely to be encountered in practice. By contrast, MCF scales linearly in the number of training examples and considers an average-case instead of a worst-case scenario. Moreover, MCF can readily be used with a variety of loss functions and corruption models.

Bishop (1995) and Webb (1994) proposed approaches that can be viewed as approximations to MCF for additive noise with small variance. By contrast, MCF is exact and can be used with noise distributions with potentially very high variance as well. MCF was inspired by recent successes of denoising autoencoders (Glorot et al., 2011; Vincent et al., 2008). Since autoencoders are non-linear, the marginalization over the corrupting distribution cannot be performed analytically in such models. Linear denoising autoencoders (Chen et al., 2012) can be viewed as a special case of MCF that aim to minimize the expected value of the reconstruction error under blankout corruption.

3. Learning with Marginalized Corrupted Features (MCF)

To derive the *marginalized corrupted features* (MCF) framework, we start by defining a corrupting distribution that specifies how training observations $\mathbf{x}$ are transformed into corrupted versions $\tilde{\mathbf{x}}$. We assume that the corrupting distribution factorizes over dimensions[1] and that each individual distribution is a member of the natural exponential family:

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \prod_{d=1}^{D} P_E(\tilde{x}_d|x_d;\eta_d), \qquad (1)$$

where η_d indicates (user-defined) parameters of the corrupting distribution on dimension d. We also assume that the corrupting distribution is *unbiased*, *i.e.* that $\mathbb{E}[\tilde{\mathbf{x}}]_{p(\tilde{\mathbf{x}}|\mathbf{x})} = \mathbf{x}$. This assumption is necessary because biases in the corrupting distribution may lead to undesired scale changes in the parameters η_d. Most corrupting distributions of interest are unbiased or can easily be made so; examples for P_E are the unbiased "blankout" noise (Vincent et al., 2008), Gaussian noise, Laplace noise, and Poisson noise.

Explicit corruption. A simple approach to improving the generalization of a classifier using a corrupting distribution is to follow the spirit of Burges & Schölkopf (1997) by selecting each element of the training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ and corrupting it M times, following (1). For each $\mathbf{x}_n$, this results in corresponding corrupted observations $\tilde{\mathbf{x}}_{nm}$ (with $m = 1, \ldots, M$). This leads to the construction of a new data set $\tilde{\mathcal{D}}$ of size $|\tilde{\mathcal{D}}| = MN$. This extended data set can be used for training by minimizing:

$$\mathcal{L}(\tilde{\mathcal{D}};\Theta) = \sum_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} L(\tilde{\mathbf{x}}_{nm}, y_n;\Theta),$$

with $\tilde{\mathbf{x}}_{nm} \sim p(\tilde{\mathbf{x}}_{nm}|\mathbf{x}_n)$, Θ the set of model parameters, and $L(\mathbf{x}, y;\Theta)$ the loss function of the model.

[1] For Gaussian noise models, this assumption is unnecessary: we can also work with non-isotropic Gaussian noise.

Distribution	**PDF**	$\mathbb{E}[\tilde{\boldsymbol{x}}_{\boldsymbol{nd}}]_{\boldsymbol{p(\tilde{x}_{nd}\vert x_{nd})}}$	$\boldsymbol{V[\tilde{x}_{nd}]_{p(\tilde{x}_{nd}\vert x_{nd})}}$
Blankout noise	$p(\tilde{x}_{nd}=0)=q_d$ $p(\tilde{x}_{nd}=\frac{1}{1-q_d}x_{nd})=1-q_d$	x_{nd}	$\frac{1}{1-q_d}x_{nd}^2$
Gaussian noise	$p(\tilde{x}_{nd}\vert x_{nd})=\mathcal{N}(\tilde{x}_{nd}\vert x_{nd},\sigma^2)$	x_{nd}	σ^2
Laplace noise	$p(\tilde{x}_{nd}\vert x_{nd})=Lap(\tilde{x}_{nd}\vert x_{nd},\lambda)$	x_{nd}	$2\lambda^2$
Poisson noise	$p(\tilde{x}_{nd}\vert x_{nd})=Poisson(\tilde{x}_{nd}\vert x_{nd})$	x_{nd}	x_{nd}

Table 1. The probability density function (PDF), mean, and variance of corrupting distributions of interest. These quantities can be plugged into Eq. (3) to obtain the expected value under the corrupting distribution of the quadratic loss.

Implicit corruption. Although such an approach is effective, it lacks elegance and comes with high computational costs, as the minimization of $\mathcal{L}(\tilde{\mathcal{D}};\Theta)$ scales linearly in the number of corrupted observations. It is, however, of interest to consider the limiting case in which $M\to\infty$. In this case, we can apply the *weak law of large numbers* and rewrite $\frac{1}{M}\sum_{m=1}^{M}L(\tilde{\mathbf{x}}_m,y_m;\Theta)$ as its expectation (Duda et al., 2001, §2.10.2):

$$\mathcal{L}(\mathcal{D};\Theta)=\sum_{n=1}^{N}\mathbb{E}[L(\tilde{\mathbf{x}}_n,y_n;\Theta)]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)}. \qquad (2)$$

Minimizing the expected value of the loss under the corruption model leads to a new approach for training predictors that we refer to as learning with *marginalized corrupted features* (MCF).

3.1. Specific loss functions

The tractability of (2) depends on the choice of loss function and corrupting distribution P_E. In this section, we show that for linear predictors that employ a quadratic or exponential loss function, the required expectations under $p(\tilde{\mathbf{x}}|\mathbf{x})$ in (2) can be computed analytically for all corrupting distributions in the natural exponential family. For linear predictors with logistic loss, we derive a practical upper bound on the expected loss under $p(\tilde{\mathbf{x}}|\mathbf{x})$, which serves as surrogate loss.

Quadratic loss. Assuming[2] a label variable $y\in\{-1,+1\}$, the expected value of the quadratic loss under corrupting distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$ is given by:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w})&=\sum_{n=1}^{N}\mathbb{E}\left[\left(\mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n-y_n\right)^2\right]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)}\\&=\mathbf{w}^{\mathrm{T}}\left(\sum_{n=1}^{N}\mathbb{E}[\tilde{\mathbf{x}}_n]\mathbb{E}[\tilde{\mathbf{x}}_n]^{\mathrm{T}}+V[\tilde{\mathbf{x}}_n]\right)\mathbf{w}\\&\qquad-2\left(\sum_{n=1}^{N}y_n\mathbb{E}\left[\tilde{\mathbf{x}}_n\right]\right)^{\mathrm{T}}\mathbf{w}+N, \qquad (3)\end{aligned}$$

where $V[\mathbf{x}]$ is a diagonal $D\times D$ matrix storing the variances of $\mathbf{x}$, and all expectations are under $p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)$. Eq. (3) is convex irrespective of what corruption model is used; the optimal solution $\mathbf{w}^*$ is given by:

$$\mathbf{w}^*=\left(\sum_{n=1}^{N}\mathbb{E}[\tilde{\mathbf{x}}_n]\mathbb{E}[\tilde{\mathbf{x}}_n]^{\mathrm{T}}+V[\tilde{\mathbf{x}}_n]\right)^{-1}\left(\sum_{n=1}^{N}y_n\mathbb{E}\left[\tilde{\mathbf{x}}_n\right]\right).$$

To minimize the expected quadratic loss under the corruption model, we only need to compute the variance of the corrupting distribution, which is practical for all exponential-family distributions. The mean is always x_{nd} because our corrupting distributions are unbiased. Table 1 gives an overview of the variances of corrupting distributions of interest. (In blankout corruption, we scale the value of “preserved” features by $\frac{1}{1-q_d}$ to ensure that the corrupting distribution is unbiased.)

An interesting setting of MCF with quadratic loss occurs when the corrupting distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$ is an isotropic Gaussian distribution with mean $\mathbf{x}$ and variance $\sigma^2\mathbf{I}$. For such a Gaussian corruption model, we obtain as special case (Chapelle et al., 2000):

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w})&=\mathbf{w}^{\mathrm{T}}\left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right)\mathbf{w}\\&\qquad-2\left(\sum_{n=1}^{N}y_n\mathbf{x}_n\right)^{\mathrm{T}}\mathbf{w}+\sigma^2N\mathbf{w}^{\mathrm{T}}\mathbf{w}+N,\end{aligned}$$

which is the standard l_2-regularized quadratic loss with regularization parameter σ^2N. Interestingly, using MCF with Laplace noise also leads to ridge regression (with regularization parameter $2\lambda^2N$).

Exponential loss. The expected value of the exponential loss under corruption model $p(\tilde{\mathbf{x}}|\mathbf{x})$ is:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w})&=\sum_{n=1}^{N}\mathbb{E}\left[e^{-y_n\mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n}\right]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)}\\&=\sum_{n=1}^{N}\prod_{d=1}^{D}\mathbb{E}\left[e^{-y_nw_d\tilde{x}_{nd}}\right]_{p(\tilde{x}_{nd}|x_{nd})}, \qquad (4)\end{aligned}$$

[2] The same derivations are applicable to regression settings in which y is a continuous variable.

Distribution	$\mathbb{E}[\exp(-y_n w_d \tilde{x}_{nd})]_{p(\tilde{x}_{nd}\mid x_{nd})}$
Blankout noise	$q_d + (1-q_d)\exp(-y_n w_d \frac{1}{1-q_d} x_{nd})$
Gaussian noise	$\exp(-y_n w_d x_{nd} + \frac{1}{2}\sigma^2 y_n^2 w_d^2)$
Laplace noise	$(1-\lambda^2 y_n^2 w_d^2)^{-1}\exp(-y_n w_d x_{nd})$
Poisson noise	$\exp(x_{nd}(\exp(-y_n w_d)-1))$

Table 2. Moment-generating functions (MGFs) of various corrupting distributions. These quantities can be plugged into equations (4) and (5) to obtain the expected value of the loss (or surrogate) under the corrupting distribution of the exponential and logistic loss functions, respectively.

where we used the assumption that the corruption is independent across features. The above equation can be recognized as a product of moment-generating functions $\mathbb{E}[\exp(t_{nd}\tilde{x}_{nd})]$ with $t_{nd} = -y_n w_d$. The moment-generating function (MGF) can be computed for all corrupting distributions in the natural exponential family. An overview of these MGFs is given in Table 2. Because the expected exponential loss is a convex combination of convex functions, it is itself convex irrespective of what corruption model is used.

The derivation above can readily be extended to multi-class exponential loss (Zhu et al., 2006) (with K classes) by replacing the weight vector $\mathbf{w}$ by a $D \times K$ weight matrix $\mathbf{W}$, and by replacing the labels y by label vectors $\mathbf{y} = \{1, -\frac{1}{K-1}\}^K$ with $\sum_{k=1}^K y_k = 0$.

Logistic loss. In the case of the logistic loss, the solution to (2) cannot be computed in closed form. Instead, we derive an upper bound, which can be minimized as a surrogate loss:

$$\mathcal{L}(\mathcal{D};\mathbf{w}) = \sum_{n=1}^{N} \mathbb{E}\left[\log\left(1 + e^{-y_n \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n}\right)\right]_{p(\tilde{\mathbf{x}}_n\mid\mathbf{x}_n)}$$

$$\leq \sum_{n=1}^{N} \log\left(1 + \prod_{d=1}^{D} \mathbb{E}\left[e^{-y_n w_d \tilde{x}_{nd}}\right]_{p(\tilde{x}_{nd}\mid x_{nd})}\right). \quad (5)$$

Herein, we have made use of Jensen's inequality to upper-bound $\mathbb{E}[\log z]$. In the upper bound, we again recognize a product of MGFs, which can be computed in closed-form for corrupting distributions in the natural exponential family (see Table 2). The upper bound on the expected logistic loss is convex whenever the moment-generating function is log-linear in w_d, *e.g.*, for blankout corruption.

Again, the above derivation can readily be extended to multi-class logistic loss (by redefining the labels y to be label vectors $\mathbf{y} \in \{0,1\}^K$ with $\sum_{k=1}^K y_k = 1$; and by defining the loss as the logarithm of the softmax-probability of making the correct prediction).

Case study. As an illustrative example, we show the upper bound of the logistic loss (5), where inputs are corrupted with the Poisson distribution:

$$\mathcal{L}(\mathcal{D};\mathbf{w}) \leq \sum_{n=1}^{N} \log\left(1 + \exp\left(\sum_{d=1}^{D} x_{nd}(e^{-y_n w_d} - 1)\right)\right).$$

Remarkably, this regularized loss does not have any additional hyper-parameters. Further, the sum over all features can still be computed efficiently for sparse data by summing only over non-zero entries in $\mathbf{x}_n$.

Computational complexity. The use of MCF does not impact the computational complexity of training: the complexity of the training algorithms remains linear in N. The additional training time for minimizing quadratic loss with MCF is negligible, because the computation time is dominated by the inversion of a $D \times D$-matrix. For exponential and logistic loss, we empirically found that computing the gradient of the MCF loss was 2× and 10× slower than computing the gradient of the "normal" loss, respectively.

4. Experiments

We perform experiments comparing MCF predictors with standard predictors on three tasks: i) document classification based on word-count features using MCF with blankout and Poisson corruption; ii) image classification based on bag-of-visual-word features using MCF with blankout and Poisson corruption; and iii) classification of objects in the "nightmare at test time" scenario using MCF with blankout corruption. The three experiments are described separately below. Code to reproduce the results of our experiments is available from `http://homepage.tudelft.nl/19j49/mcf`.

4.1. Document classification

We first test MCF predictors with blankout and Poisson corruption on document classification tasks. Specifically, we focus on three data sets: the Dmoz data set, the Reuters data set, and the Amazon review benchmark set (Blitzer et al., 2007).

Data sets. The Dmoz open directory (`http://www.dmoz.org`) contains a large collection of webpages arranged into a tree hierarchy. The subset we use contains $N = 8,980$ webpages from the $K = 16$ categories in the top level of the hierarchy. Each webpage is represented by a bag-of-words representation with $D = 16,498$ words. The Reuters data set is a collection of documents that appeared on the Reuters newswire in 1987. Documents with multiple labels were removed from the data, resulting in a set of $N = 8,293$ documents from $K = 65$ categories. The bag-of-words

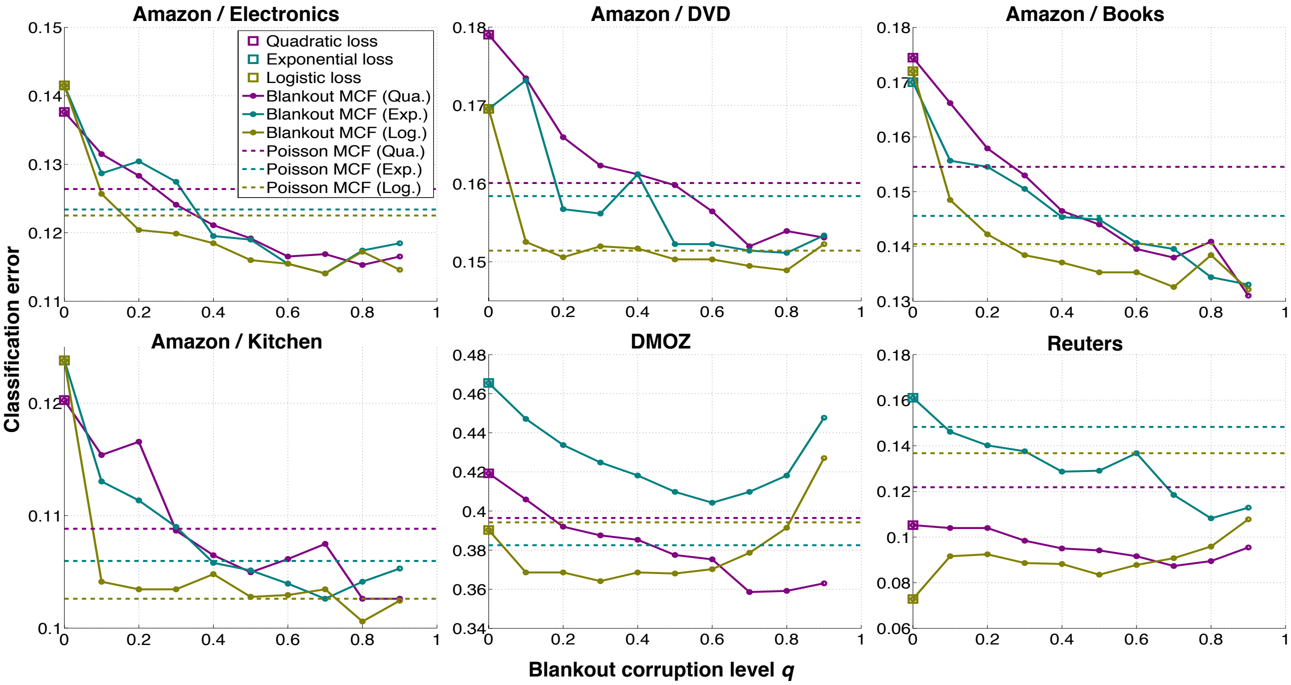

Figure 1. Classification errors of MCF predictors using blankout and Poisson corruption – as a function of the blankout corruption level q – on the Amazon, Dmoz, and Reuters data sets for l_2-regularized quadratic, exponential, and quadratic loss functions. Classification errors are represented on the y-axis, whereas the blankout corruption level q is represented on the x-axis. The case of MCF with blankout corruption and $q = 0$ corresponds to a standard l_2-regularized classifier. Figure best viewed in color.

representation contains $D = 18,933$ words for each document. The four Amazon data sets consist of approximately $N = 6,000$ reviews of four types of products: books, DVDs, electronics, and kitchen appliances. Each review is represented by a bag-of-words representation of the $D = 20,000$ most common words. On the Dmoz and Reuters data sets, the task is to classify the documents into one of the predefined categories. On the Amazon data set, the task is to decide whether a review is positive or negative.

Setup. On the Dmoz and Reuters data sets, we use a fixed training set of 75% of the data, and evaluate the performance of our predictors on a fixed test set of 25% of the data. On the Amazon data set, we follow the experimental setup of Blitzer et al. (2007) by using a fixed division of the data into approximately 2,000 training examples and about 4,000 test examples (the exact numbers vary slightly between tasks). All experiments are performed using linear classifiers that are trained with l_2-regularization; the amount of l_2-regularization is determined via cross-validation. We consider quadratic, exponential, and logistic loss functions (both with and without MCF). The minimization of the (expected) exponential and logistic losses is performed by running Mark Schmidt's `minFunc` implementation of L-BFGS until convergence or until a predefined maximum number of iterations is reached.

All predictors included a bias term that is neither regularized nor corrupted. In our experiments with MCF using blankout corruption, we use the same noise level for each feature, *i.e.* we assume that $\forall d : q_d = q$. On all data sets, we first investigate the performance of MCF as a function of the corruption level q (but we still cross-validate over the l_2-regularizer). In a second set of experiments, we cross-validate over the blankout corruption parameter q and study to what extent the performance (improvements) of MCF depend on the amount of available training data. (MCF with Poisson corruption has no additional hyper-parameters, as a result of which it requires no extra cross-validations.)

Results. Figure 1 shows the test error of our MCF predictors on all data sets as a function of the blankout corruption level q. Herein, corruption level $q = 0$ corresponds to the baseline predictors, *i.e.* to predictors that do not employ MCF. The results show: i) that MCF improves over standard predictors for both blankout corruption (for all corruption levels q) and Poisson corruption on five out of six tasks; ii) that MCF with Poisson corruption leads to significant performance improvements over standard classifiers whilst introducing no additional hyperparameters; and iii) that the best performance tends to be achieved by MCF with blankout corruption with high corruption levels, *i.e.* when q is in the order of 0.8. The best-performing MCF classifiers reduce the test errors by up to 22% on the Amazon data if q is properly set. In many of the experiments with MCF-trained losses (in particular, when blankout corruption is used), we also observe that the optimal level of l_2-regularization is 0. This shows that MCF has a regularizing effect in itself, rendering additional regularization superfluous.

Figure 2 presents the results of a second set of experiments on Dmoz and Reuters in which we study how the performance of MCF depends on the amount of training data. For each training set size, we repeat the experiment five times with randomly sub-sampled training sets; the figure reports the mean test errors and the corresponding standard deviations. The results show that classifiers trained with MCF (solid curves) significantly outperform their counterparts without MCF (dashed curves). The performance improvement is consistent irrespective of the training set size, *viz.* up to 25% on the Dmoz data set.

Explicit vs. implicit feature corruption. Figure 3 shows the classification error on Amazon (books) when a classifier without MCF is trained on the data set with additional *explicitly* corrupted samples, as formulated in (3). Specifically, we use the blankout corruption model with q set by cross-validation for each setting, and we train the classifiers with quadratic loss and l_2-regularization. The graph shows a clear trend that the error *decreases* when the training set contains more corrupted versions of the original training data, *i.e.* with higher M in eq. (3). The graph illustrates that the best performance is obtained as M approaches infinity, which is equivalent to MCF with blankout corruption (marker in the bottom right; $q = 0.9$).

4.2. Image classification

We performed image-classification experiments on the CIFAR-10 data set (Krizhevsky, 2009), which is a subset of the 80 million tiny images (Torralba et al., 2008). The data set contains RGB images with 10 classes of size 32×32, and consists of a fixed training and test set of $50,000$ and $10,000$ images, respectively.

Setup. We follow the experimental setup of Coates et al. (2011): we whiten the training images and extract a set of 7×7 image patches on which we apply k-means clustering (with $k = 2048$) to construct a codebook. Next, we slide a 7×7 pixel window over each image and identify the nearest prototype in the

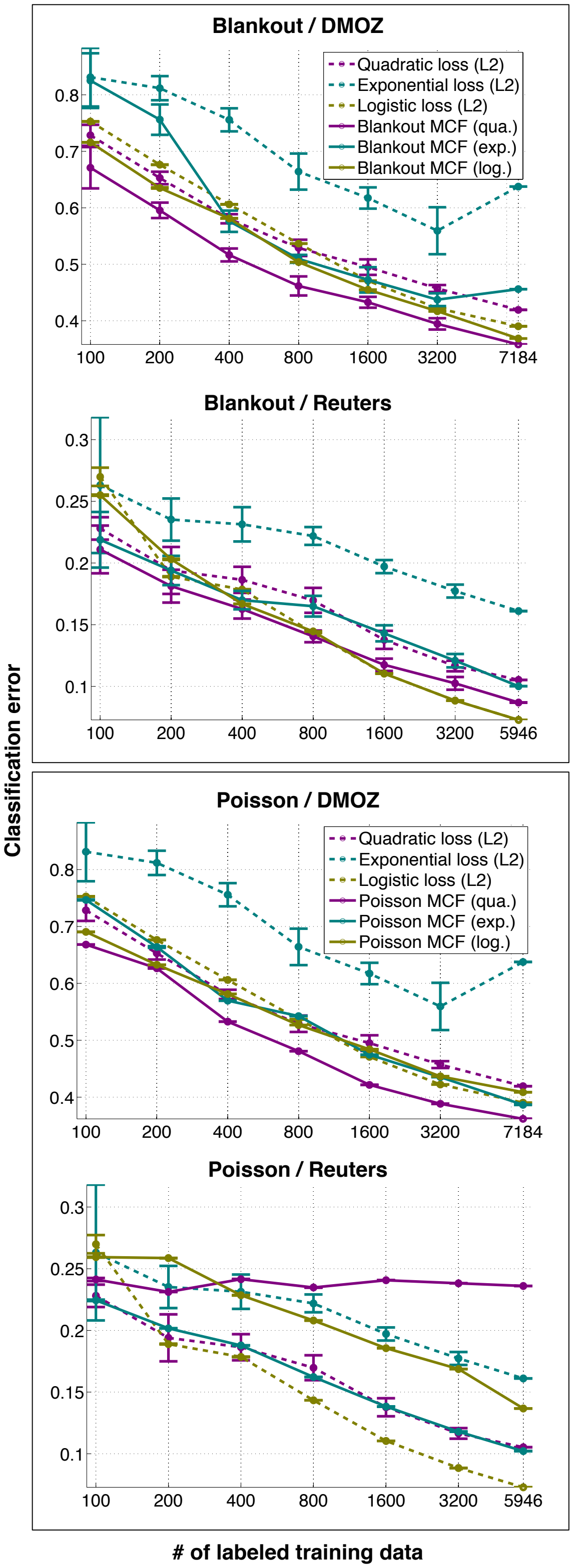

Figure 2. The performance of standard and MCF classifiers with blankout and Poisson corruption models as a function of training set size on the Dmoz and Reuters data sets. Both the standard and MCF predictors employ l_2-regularization. Figure best viewed in color.

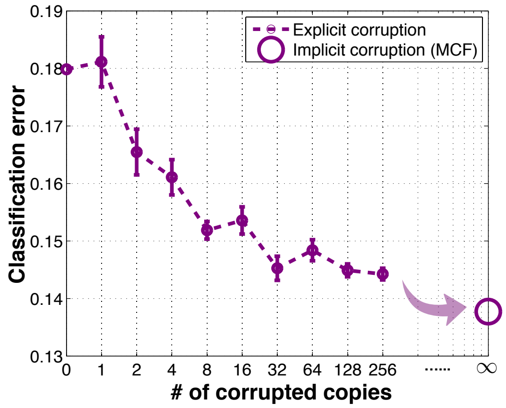

Figure 3. Comparison between MCF and explicitly adding corrupted examples to the training set (for quadratic loss) on the Amazon (books) data using blankout corruption. Training with MCF is equivalent to using infinitely many corrupted copies of the training data.

codebook for each window location. We construct an image descriptor[3] by subdividing the image into four equally sized quadrants and counting the number of times each prototype occurs in each quadrant. This leads to a descriptor of dimensionality $D = 4 \times 2048$. We do not normalize the descriptors, because all images have the same size. We train MCF predictors with blankout and Poisson corruption on the full set of training images, cross-validating over a range of l_2-regularization parameters. The generalization error is evaluated on the test set.

Results. The results are reported in Table 3. The baseline classifiers (without MCF) are comparable to the 68.8% accuracy reported by Coates et al. (2011) with exactly the same experimental setup (except for exponential loss). The results illustrate the potential of MCF classifiers to improve the prediction performance on bag-of-visual-words features, in particular, when using quadratic or logistic loss in combination with a Poisson corruption model. Although our focus in this section is to merely illustrate the potential of MCF on image classification tasks, it is worth noting that the best results in Table 3 match those of a highly non-linear mean-covariance RBMs trained on the same data (Ranzato & Hinton, 2010), despite our use of very simple visual features and of linear classifiers.

4.3. Nightmare at test time

To test the performance of our MCF predictors with blankout corruption under the “nightmare at test time” scenario, we perform experiments on the MNIST handwritten digits data set. The MNIST data set contains $N = 60,000$ training and $10,000$ test images of size $D = 28 \times 28 = 784$ pixels with $K = 10$ classes.

	Quadr.	**Expon.**	**Logist.**
No MCF	32.6%	39.7%	32.5%
Poisson MCF	29.1%	39.5%	30.0%
Blankout MCF	32.3%	37.9%	29.4%

Table 3. Classification errors obtained on the CIFAR-10 data set with MCF classifiers trained on simple spatial-pyramid bag-of-visual-words features.

Setup. We train our predictors on the full training set, and evaluate their performance on versions of the test set in which a certain percentage of the pixels are randomly blanked out, *i.e.* set to zero. We compare the performance of our MCF-predictors (using blankout corruption) with that of standard predictors that use l_1 or l_2-regularized quadratic, exponential, logistic, and hinge loss. As before, we use cross-validation to determine the optimal value of the regularization parameter. For MCF predictors, we also cross-validate over the blankout corruption level q (again, we use the same noise level for each feature, *i.e.* $\forall d : q_d = q$). In addition to the comparisons with standard predictors, we also compare the performance of MCF with that of FDROP (Globerson & Roweis, 2006), which is a state-of-the-art algorithm for the “nightmare at test time” setting that minimizes the hinge loss under an adversarial worst-case scenario.

The performances are reported as a function of the feature-deletion percentage in the test set, *i.e.* as a function of the probability with which a pixel in the test set is switched off. Following the experimental setting of Globerson & Roweis (2006), we perform the cross-validation for each deletion percentage independently, *i.e.* we create a small validation set with the same feature-deletion level and use it to determine the best regularization parameters and blankout corruption level q for that percentage of feature deletions.

Results. Figure 4 shows the performance of our predictors as a function of the percentage of deletions in the test images. The figure shows the performance for all three loss functions with MCF (solid lines) and without MCF (dashed lines). The performance of a standard predictor using hinge loss is shown as a red dashed line; the performance of FDROP is shown as a black dashed line. The results presented in Figure 4 clearly illustrate the ability of MCF with blankout corruption to produce predictors that are robust to the “nightmare at test time” scenario: MCF improves the performance substantially for all three loss func-

[3] This way of extracting the image features is referred to by Coates et al. (2011) as k-means with hard assignment, average pooling, patch size 7×7, and stride 1.

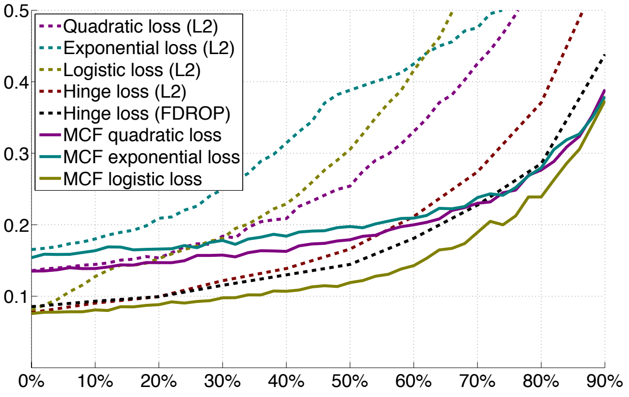

Figure 4. Classification errors of standard and MCF predictors with a blankout corruption model – trained using three different losses – and of FDROP (Globerson & Roweis, 2006) on the MNIST data set using the "nightmare at test time" scenario. Classification errors are represented on the y-axis, whereas the percent of features that are deleted at test time is represented on the x-axis. Figure best viewed in color.

tions considered. For instance, in the case in which 50% of the pixels in the test images is deleted, the performance improvements obtained using MCF for quadratic, exponential, and logistic loss are 40%, 47%, and 60%, respectively. Further, the results also indicate that MCF-losses may outperform FDROP: our MCF logistic loss outperforms FDROP's worst-case hinge loss across the board[4]. This is particularly impressive as the standard hinge loss performs surprisingly better than standard logistic loss on this data set (the improvement of FDROP over the generic hinge-loss is relatively modest). This result suggests that it is better to consider an average-case than a worst-case scenario in the "nightmare at test time" setting.

5. Discussion and Future Work

We presented an approach to learn classifiers by marginalizing corrupted features (MCF). Specifically, MCF trains predictors by introducing corruption on the training examples, which is marginalized out in the expectation of the loss function. We minimize the expected loss with respect to the model parameters. Our experimental results show that MCF predictors with blankout and Poisson corruption perform very well in the context of bag-of-words features. MCF with Poisson corruption is particularly interesting for such count features, as it improves classification performances without introducing any additional hyper-parameters. As a disclaimer, care must be taken when applying MCF with Poisson corruption on data sets with outliers. Poisson corruption may emphasize outliers in the expected loss because the variance of a Poisson distribution is equal to its mean, and because our loss functions are not robust to outliers. A solution to this problem may be to redefine the corruption distribution to $p(\tilde{x}_d|x_d) = Pois(\tilde{x}_{nd}|\min\{x_{nd}, u\})$ for some cutoff parameter $u \geq 0$.

In most of our experiments with MCF, the l_2-regularizer parameter (which was set by cross-validation) ended up very close to zero. This implies that MCF in itself has a regularizing effect. At the same time, MCF with blankout corruption also appears to prevent *weight undertraining* (Sutton et al., 2005): it encourages the weight on each feature to be non-zero, in case this particular feature survives the corruption. Our experiments also reveal that MCF with blankout corruption produces predictors that are more robust to the "nightmare at test time" scenario, making it useful in learning settings in which features in the test data may be missing. Learning with MCF is quite different from previous approaches for this setting (Dekel & Shamir, 2008; Globerson & Roweis, 2006): it does not learn under a worst-case scenario, but the (arguably) more common average-case scenario by considering all possible corrupted observations. This has the advantage that it is computationally much cheaper and that it allows for incorporating prior knowledge. For instance, if the data is generated by a collection of unreliable sensors, knowledge on the sensor reliability may be used to set the q_d-parameters.

In future work, we intend to explore extensions of MCF to regression and structured prediction, as well as to investigate if MCF can be employed for kernel machines. We also plan to explore in more detail what corruption models $p(\tilde{\mathbf{x}}|\mathbf{x})$ are useful for what types of data, and how these corruption models regularize classifiers (Ng, 2004). Further, MCF could be used in the training of neural networks with a single layer of hidden units: blankout noise *on the hidden nodes* can improve the performance of the networks (LeCun et al., 1990; Sietsma & Dow, 1991; Hinton et al., 2012) and can be marginalized out analytically. A final interesting direction is to investigate the effect of marginalizing corrupted *labels* (Lawrence & Schölkopf, 2001).

Acknowledgements

LvdM is supported by FP7 Social Signal Processing (SSPNet). KQW, MC, and ST are supported by NIH grant U01 1U01NS073457-01 and NSF grants 1149882 and 1137211. The authors thank Fei Sha for helpful discussions.

[4] Quadratic and exponential losses perform somewhat worse because they are less appropriate for linear classifiers, but even they outperform FDROP for large numbers of feature deletions in the test data.

References

Bhattacharyya, C., Pannagadatta, K.S., and Smola, A.J. A second order cone programming formulation for classifying missing data. In *Advances in Neural Information Processing Systems*, pp. 153–160, 2004.

Bishop, C.M. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

Blitzer, J., Dredze, M., and Pereira, F. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics*, volume 45, pp. 440, 2007.

Brückner, M., Kanzow, C., and Scheffer, T. Static prediction games for adversarial learning problems. *Journal of Machine Learning Research*, 12:2617–2654, 2012.

Burges, C.J.C. and Schölkopf, B. Improving the accuracy and speed of support vector machines. *Advances in Neural Information Processing Systems*, 9:375–381, 1997.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, pp. 416–422, 2000.

Chechik, G., Heitz, G., Elidan, G., Abbeel, P., and Koller, D. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9(Jan): 1–21, 2008.

Chen, M., Xu, Z., Weinberger, K.Q., and Sha, F. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the International Conference on Machine Learning*, pp. 767–774, 2012.

Coates, A., Lee, H., and Ng, A.Y. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence & Statistics, JMLR W&CP 15*, pp. 215–223, 2011.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13 (1):21–27, 1967.

Dekel, O. and Shamir, O. Learning to classify with missing and corrupted features. In *Proceedings of the International Conference on Machine Learning*, pp. 216–223, 2008.

Duda, R.O., Hart, P.E., and Stork, D.G. *Pattern Classification*. Wiley Interscience Inc., 2001.

Globerson, A. and Roweis, S. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the International Conference on Machine Learning*, pp. 353–360, 2006.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning*, pp. 513–520, 2011.

Herbrich, R. and Graepel, T. Invariant pattern recognition by semidefinite programming machines. In *Advances in Neural Information Processing Systems*, volume 16, pp. 33, 2004.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors, 2012.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lawrence, N.D. and Schölkopf, B. Estimating a kernel Fisher discriminant in the presence of label noise. In *Proceedings of the International Conference in Machine Learning*, pp. 306–313, 2001.

LeCun, Y., Denker, J.S., and Solla, S.A. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pp. 598–605, 1990.

Ng, A.Y. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of International Conference on Machine Learning*, pp. 78–85, 2004.

Ranzato, M. and Hinton, G.E. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2551–2558, 2010.

Shivaswamy, P.K., Bhattacharyya, C., and Smola, A.J. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

Sietsma, J. and Dow, R.J.F. Creating artificial neural networks that generalize. *Neural Networks*, 4:67–79, 1991.

Sutton, C., Sindelar, M., and McCallum, A. Feature bagging: Preventing weight undertraining in structured discriminative learning. Technical Report IR-402, University of Massachusetts, 2005.

Teo, C.H., Globerson, A., Roweis, S., and Smola, A. Convex learning with invariances. *Advances in Neural Information Processing Systems*, 20:1489–1496, 2008.

Torralba, A., Fergus, R., and Freeman, W.T. 80 million tiny images: A large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

Trafalis, T. and Gilbert, R. Robust support vector machines for classification and computational issues. *Optimization Methods and Software*, 22(1):187–198, 2007.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning*, pp. 1096–1103, 2008.

Webb, A.R. Functional approximation by feed-forward networks: a least-squares approach to generalization. *IEEE Transactions on Neural Networks*, 5(3):363–371, 1994.

Xu, H., Caramanis, C., and Mannor, S. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, 2009.

Zhu, J., Rosset, S., Zou, H., and Hastie, T. Multi-class AdaBoost. Technical Report 430, Department of Statistics, University of Michigan, 2006.