Linear-Time Estimators for Propensity Scores

Deepak Agarwal
Yahoo! Research
Santa Clara, CA, USA
dagarwal@yahoo-inc.com

Lihong Li
Yahoo! Research
Santa Clara, CA, USA
lihong@yahoo-inc.com

Alexander J. Smola
Yahoo! Research
Santa Clara, CA, USA
smola@yahoo-inc.com

Abstract

We present linear-time estimators for three popular covariate shift correction and propensity scoring algorithms: logistic regression(LR), kernel mean matching(KMM) [19], and maximum entropy mean matching(MEMM)[20]. This allows applications in situations where *both* treatment and control groups are large. We also show that the last two algorithms differ only in their choice of regularizer (ℓ_2 of the Radon Nikodym derivative vs. maximum entropy). Experiments show that all methods scale well.

1 Introduction

Propensity scoring [17] has become a staple in the statistical analysis of data obtained through a non-randomized procedure. It aims to answer "what if" questions of the following nature: assume that we would like to test the efficiency of a novel drug. For a number of reasons it may be impossible to select an entirely random set of patients for the treatment — for instance the treatment may come with certain side effects which make its use unethical in relatively healthy patients, thus biasing the treatment towards rather sick patients. Thus quite often the treatment group is anything but random.

Nonetheless we would like to assess the drug. A naive comparison between the treatment and the control set (the patients who did not receive the drug) may lead to quite wrong results: if a cancer drug were only administered to the sickest patients it is likely that the mortality rate in the treatment group is higher than in the control group. However, this does not allow us to conclude that the drug is ineffective. Quite the opposite, the drug might be saving at least a fraction of terminally ill patients. Hence, to answer the question "what if" we had administered the drug to everyone we need to reweight treatment and control groups such as to match their distributions. This is achieved by estimating the Radon-Nikodym derivative (RND) between the treatment and control distributions and by taking a suitable linear combination of scores [17].

This problem is commonly known in machine learning as that of covariate shift correction [14], where training and test set (corresponding to treatment and control populations) are drawn from different distributions. Specifically, we assume access to samples $X_p := \{x_1, \ldots, x_m\}$ and $X_q := \{x'_1, \ldots, x'_{m'}\}$ drawn iid from two unknown distributions p and q, respectively. The quantity needed for propensity scoring is the RND $\beta(x) := \frac{dq(x)}{dp(x)}$, namely, the ratio between the control group's distribution and the treatment group's. The algorithm by [21] has a complexity of $O(mm' + m^\alpha)$,[1] and so does not scale well to problems with a large treatment group.

In this paper, we focus on efficient propensity scoring algorithms that are applicable to problems with large control *and* treatment gropus. The main contributions include:

1. We obtain scalable estimators for β based on X_p and X_q. By scalable we mean estimators whose runtime is $O(m + m')$. We achieve this by presenting online algorithms for three approaches: logistic regression (LR), kernel mean matching (KMM) [8], and maximum entropy mean matching (MEMM)[21]. They are based on online learning convergence results [4, 23].
2. We show the latter two algorithms share a similar optimization setting, differing only in their choice of smoothers (L_2 of the RND vs. relative entropy).
3. We give an experimental evaluation of LR, KMM, and MEMM, using both a UCI benchmark data [3] and a large-scale real data set from a major Web portal.

By representing the RND as a non-negative linear combination of positively valued basis functions, a recent al-

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

[1] Here, α is unspecified but typically $\alpha \in [2, 3]$ for the algorithm proposed [21].

gorithm [9] can be made to have linear complexity. Our scalable algorithms also rely on explicit feature expansion. However, instead of requiring a set of good basis functions, our algorithms aim at solving for the same solution as their corresponding kernel methods, which can be flexibly combined with other approximation techniques; see Section 5.

2 Propensity Scoring and Covariate Shift Correction

In the following, we denote by $\mathcal{X}$ the space of covariates with samples X_p, X_q drawn from p and q respectively. We will refer to p as the treatment (or training set) distribution and to q as the control (or test set) distribution. For many applications it is important to obtain estimates of $\beta(x)$ which closely match the true underlying distribution. We make the following key assumption:

Assumption 1 (Relative Density) *For p and q we assume that the RND of q with respect to p is bounded by some constant $C \geq \beta(x)$ for all $x \in \mathcal{X}$. This ensures that there cannot exist sets of nonzero measure with respect to p that have zero measure with respect to q.*

In the example of patient selection, it means that the control group must not contain any significant component of patients which have a very different distribution of covariates from those in the treatment group.

Propensity Scoring In its simplest form [17] propensity scoring attempts to address the issue how much the response $y|x$ changes between treatment and control scenarios. That is, we are interested in the following:

$$\begin{aligned}\Delta :=& \mathbf{E}_{x\sim q}\left[\mathbf{E}_{y|x,\text{treatment}}\left[y\right] - \mathbf{E}_{y|x,\text{control}}\left[y\right]\right] \qquad (1)\\ =& \mathbf{E}_{x\sim p}\left[\beta(x)\mathbf{E}_{y|x,\text{treatment}}\left[y\right]\right] - \mathbf{E}_{x\sim q}\left[\mathbf{E}_{y|x,\text{control}}\left[y\right]\right].\end{aligned}$$

and the corresponding empirical estimate

$$\Delta_{\text{emp}} = \frac{1}{m}\sum_{i=1}^{m}\beta(x_i)y_i - \frac{1}{m'}\sum_{i=1}^{m'}y_i, \qquad (2)$$

with propensity scores $\beta(\cdot)$. The transformation of (2) is possible due to the importance sampling relation

$$\begin{aligned}\mathbf{E}_{x\sim q}[f(x)] &= \int dq(x)f(x)\\ &= \int \frac{dq(x)}{dp(x)}dp(x)f(x) = \mathbf{E}_{x\sim p}[\beta(x)f(x)]. \qquad (3)\end{aligned}$$

Considerably more sophisticated schemes for estimating Δ exist, involving variance reducing schemes for precomputing a smoothed estimate of $y|x$ beforehand. See [10] for a survey of some recent variance reduction techniques for propensity score estimates, such as doubly robust estimators. That said, they all rely on β, hence good estimators for β are desirable.

Covariate Shift Correction In covariate shift correction we assume that $p(y|x) = q(y|x)$. The densities $p(.|x)$ and $q(.|x)$ are conditional densities of response in the training(treatment) and test(control) groups respectively. In several applications, the distribution of covariates in test group is different from training, it is our goal to find a risk minimizer which minimizes the expected risk with respect to $p(y|x)q(x)$ while we only have labels drawn from p. This is possible due to (3) which yields

$$\begin{aligned}R[f] &= \mathbf{E}_{x\sim q}\mathbf{E}_{y|x}[\text{Loss}(x,y,f(x))]\\ &= \mathbf{E}_{x\sim p}\mathbf{E}_{y|x}[\beta(x)\text{Loss}(x,y,f(x))]. \qquad (4)\end{aligned}$$

Given β, empirical estimates for $R_{\text{test}}[f]$ can be obtained directly via

$$R_{\text{emp}}[f] = \frac{1}{m}\sum_{i=1}^{m}\beta(x_i)\text{Loss}(x_i,y_i,f(x_i)). \qquad (5)$$

3 Logistic Regression for Propensity Score Estimation

Conversion to Binary A large number of off-the-shelf estimators exist for the estimation of probabilities, most importantly logistic regression. It is possible to recast the problem of estimating β as a classification problem. For this purpose consider the following distribution

$$ds(x,y) = \frac{1}{2}\delta_{y,1}dp(x) + \frac{1}{2}\delta_{y,-1}dq(x). \qquad (6)$$

By computing estimates for $s(y|x)$ efficiently we immediately have propensity scores:

$$s(y=1|x) = \frac{dp(x)}{d[p(x)+q(x)]}, \text{ so } \beta(x) = \frac{s(y=\text{-}1|x)}{s(y=1|x)}. \qquad (7)$$

Hence, if we use the dataset

$$Z := \{(x_1,1),\ldots,(x_m,1),(x'_1,\text{-}1),\ldots,(x'_{m'},\text{-}1)\}$$

to estimate conditional class probabilities[2] we have a propensity score estimator.

Logistic Regression A particularly popular estimator for conditional probabilities is logistic regression. In it one makes the assumption that, for some $h(\cdot)$

$$p(y|x) = \left[1+e^{-yh(x)}\right]^{-1}, \text{ hence } \beta(x) = e^{-h(x)}. \qquad (8)$$

Quite often we assume h is contained in a linear function space with evaluation functional ϕ: $h(x) = \langle h, \phi(x)\rangle$.

[2] We assume that $m = m'$. Otherwise we would simply need to reweight the instances from X_p and X_q such that the relative weight of both datasets balances out.

This holds, for instance, whenever h is contained in a Reproducing Kernel Hilbert Space with kernel $k(x, x') = \langle \phi(x), \phi(x') \rangle$. h can be obtained efficiently, e.g. by stochastic gradient descent on the penalized log-likelihood

$$\underset{h}{\text{minimize}} \frac{1}{m+m'} \sum_{(x,y)\in Z} \log\left[1 + e^{-yh(x)}\right] + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}}. \tag{9}$$

This yields the following algorithm (we assume that $h(x) = \langle \phi(x), \theta \rangle$ and that initially $\theta = 0$). A host of results showing convergence exist for it [4].

repeat
  Observe (x_t, y_t) from Z
  Compute learning rate $\eta_t \leftarrow ct^{-1}$ and gradient $G_t = \left[1 + e^{y\langle \phi(x_t), \theta \rangle}\right]^{-1}$
  Update $\theta \leftarrow (1 - \eta_t \lambda)\theta - \eta_t G_t y_t \phi(x_t)$
until converged

Theorem 2 *The algorithm converges to the minimum of (9) at rate $O(\frac{1}{T} \log T)$ [4] where T is the number of online examples.*

4 Convex Duality and Operator Mean Matching

4.1 Mean Matching

Recently, some rather less traditional algorithms have been proposed to deal with the problem of covariate shift correction. At their core they rely on approximating the expectation operator of a distribution directly [7]. For linear function classes with evaluation operator

$$\phi(x) : f \longrightarrow f(x) = \langle \phi(x), f \rangle \tag{10}$$

we may write expectations via

$$\begin{aligned} &\mathbf{E}_{x\sim p}[f(x)] = \mathbf{E}_{x\sim p}[\langle \phi(x), f \rangle] \\ &= \langle \mathbf{E}_{x\sim p}[\phi(x)], f \rangle =: \langle \mu[p], f \rangle . \end{aligned} \tag{11}$$

Representing p via $\mu[p]$ has a number of advantages. Firstly, one may show that empirical averages $\mu[X] := \frac{1}{m}\sum_{i=1}^m \phi(x_i)$ converge at rate $O(m^{-\frac{1}{2}})$ to $\mu[p]$ under fairly benign conditions (e.g. bounded $\|\phi(x)\|$ in an RKHS, bounded Rademacher averages, etc.) [2]. Secondly, an approximation of $\mu[p]$, e.g. by μ' immediately also implies an approximation in expectation:

$$\begin{aligned} &\sup_{\|f\|\leq 1} [\mathbf{E}_{x\sim q}[f(x)] - \langle f, \mu' \rangle] \\ &= \sup_{\|f\|\leq 1} \langle f, \mu[q] - \mu' \rangle = \|\mu[q] - \mu'\| . \end{aligned}$$

This means that if we succeed in approximating $\mu[q]$ by a weighted combination of samples drawn from p we immediately also enjoy guarantees in terms of weighted averages as they are needed for covariate shift correction.

As discussed in Section 2, we are interested in computing expectations with respect to q, i.e. we want to approximate $\mu[q]$ by a weighted combination of instances drawn from p. This is a convex constrained optimization problem:

$$\underset{\hat{p}\in\mathcal{P}}{\text{minimize}}\, \Omega[\hat{p}, p] \text{ subject to } \frac{1}{2}\left\|\mathbf{E}_{x\sim\hat{p}(x)}\left[\phi(x)\right] - \mu\right\|^2 \leq \epsilon, \tag{12}$$

where $\mu = \mu[q]$ or $\mu = \mu[X_q]$. Here $\Omega[\hat{p}, p]$ quantifies the proximity of $\hat{p}$ to the treatment (training) distribution, $\mathcal{P}$ is the probability simplex on $\mathcal{X}$, and μ denotes the (empirical) expectation operator associated with q.

Whenever $\mu[\hat{p}]$ is close to μ we only incur a small amount of bias. On the other hand, having a large value of $\Omega[\hat{p}, p]$ usually implies that some of the observations drawn from p will have a rather large sample weight, which should be avoided lest we suffer from a high variance estimate. We now show that Kernel Mean Matching (KMM) [8] and Maximum Entropy Mean Matching [21] both arise from the same framework, albeit with different choices in Ω. In the following we show that the following relations hold for Kernel Mean Matching, Bounded Kernel Mean Matching, and Maximum Entropy Mean Matching, respectively:

$$\Omega[\hat{p}, p] = \left\|\frac{d\hat{p}}{dp}\right\|^2_{L_2,p} = \int \gamma^2(x) dp(x) \tag{13}$$

$$\Omega[\hat{p}, p] = \left\|\frac{d\hat{p}}{dp}\right\|_{L_\infty} = \sup_{x\in \mathrm{Supp}[p]} \gamma(x) \tag{14}$$

$$\Omega[\hat{p}, p] = \mathrm{D}(\hat{p}\|p) = \int \gamma(x) \log \gamma(x) dp(x) \tag{15}$$

Here we defined $\gamma(x) = \frac{d\hat{p}(x)}{dp(x)}$ to be the RND of the estimate of the density correction.

4.2 Kernel Mean Matching

Assume that we would like to ensure that the Radon-Nikodym derivative is as uniform as possible. This is desirable since uniform sample weights lead to low-variance estimators, since the smallest value of (13) is obtained for uniform γ. Substituting the empirical average on the treatment group for p yields the optimization problem solved by [8] which can be solved by standard QP codes.

Theorem 3 *Let $K_{ij} := k(x_i, x_j)$ and $u_i := m'^{-1}\sum_{j=1}^{m'} k(x_i, x'_j)$ be a kernel matrix and the pointwise evaluation of $\mu[X_q]$ respectively. In this case solving (12) using (13) is achieved by solving the quadratic program*

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\alpha^\top [K + \lambda \mathbf{1}]\alpha - \alpha^\top u \tag{16}$$
subject to $\alpha^\top 1 = 1$ and $\alpha_i \geq 0$.

for some value $\lambda > 0$ and for $\gamma = m\alpha$. Moreover, (14) subject to (12) is equivalent to

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\alpha^\top K\alpha - \alpha^\top u \quad (17)$$
$$\text{subject to } \alpha^\top 1 = 1 \text{ and } \alpha_i \in [0, \lambda] \text{ for some } \lambda \geq m^{-1}.$$

Proof We begin by rewriting the constraint in terms of $\alpha_i = m^{-1}\gamma(x_i)$. We have

$$\underbrace{\sum_{i,j}^{m} \alpha_i\alpha_j k(x_i, x_j)_{ij}}_{=\gamma^\top K\alpha} - 2\underbrace{\sum_{i=1}^{m} \alpha_i \frac{1}{m'}\sum_{j=1}^{m'} k(x_i, x'_j)}_{=\alpha^\top u} + \|\mu\|^2 \leq \epsilon. \quad (18)$$

Moreover, (13) can be expressed in vector notation as $\|\alpha\|^2_{L_{q,p}} = m\alpha^\top \mathbf{1}\alpha$. Likewise, (14) can be expressed as $\|\alpha\|_\infty = \max_i \alpha_i$ since $\alpha \geq 0$ by default. Finally, the normalization constraint which ensures that $\hat{p}$ is a proper distribution amounts to $\alpha^\top 1 = 1$.

We can move the constraint into the respective objective functions by multiplying them by a suitable Lagrange multiplier. Moreover, the terms $\|\mu\|^2$ and ϵ are independent of α, hence we can omit them from the resulting partial Lagrange function. For (12) subject to (13) this yields

$$\Lambda\alpha^\top K\alpha - 2\Lambda\alpha^\top u + \frac{m}{2}\alpha^\top \mathbf{1}\alpha \text{ subject to } \alpha^\top 1 = 1 \text{ and } \alpha_i \geq 0. \quad (19)$$

Defining $\lambda = \frac{m}{2\Lambda}$ and dividing (19) by 2Λ proves (16). An analogous transformation of the constraints and a move of the norm bound on γ into the constraints proves (17). ■

The optimization problem bears more than a passing resemblance to the single class SVM optimization problem [18] with the linear soft-margin loss replaced by a quadratic soft-margin loss. It is, in fact, equivalent to such a problem with adaptively chosen threshold.

Theorem 4 *The optimization problems (16) and (17) are up to constants the duals of the following*

$$\underset{\theta,b}{\text{minimize}} \frac{1}{2}\|\theta\|^2 + b + \frac{1}{2\lambda}\sum_{i=1}^{m}(u_i - \langle\phi(x_i), \theta\rangle - b)_+^2 \quad (20)$$

$$\underset{\theta,b}{\text{minimize}} \frac{1}{2}\|\theta\|^2 + b + \lambda\sum_{i=1}^{m}(u_i - \langle\phi(x_i), \theta\rangle - b)_+ \quad (21)$$

Moreover, the weighting coefficients γ_i are given by $\gamma_i = \frac{m}{\lambda}(u_i - \langle\phi(x_i), \theta\rangle - b)_+$ in the first case and $\gamma_i = \lambda m\frac{1}{2}(1 + \operatorname{sgn}[u_i - \langle\phi(x_i), \theta\rangle - b])$ in the second case. Here we used the threshold function $(\xi)_+ := \max(0, \xi)$.

Proof The proof is analogous to that of [6, 18]. We sketch it for the purpose of self-consistency and to explain the connection between θ, γ and α. We begin by rewriting (20) as a constrained convex optimization problem

$$\underset{\theta,b}{\text{minimize}} \frac{1}{2}\|\theta\|^2 + b + \frac{1}{2\lambda}\sum_{i=1}^{m}\xi_i^2 \quad (22)$$
$$\text{subject to } \xi_i \geq u_i - \langle\phi(x_i), \theta\rangle - b$$

Computing the Lagrange function yields

$$L = \frac{1}{2}\|\theta\|^2 + b + \frac{1}{2\lambda}\sum_{i=1}^{m}\xi_i^2 + \sum_{i=1}^{m}\alpha_i[u_i - \langle\phi(x_i), \theta\rangle - b - \xi_i] \quad (23)$$

Plugging the associated first-order conditions back into L yields the dual problem

$$\underset{\alpha}{\text{maximize}} -\frac{1}{2}\alpha^\top[K + \lambda\mathbf{1}]\alpha + \alpha^\top u \quad (24)$$
$$\text{subject to } \alpha^\top 1 = 1 \text{ and } \alpha_i \geq 0.$$

To see the connection between θ and α we use the first order optimality conditions on (20):

$$\theta = \sum_i \alpha_i\phi(x_i) = \frac{1}{\lambda}\sum_{i=1}^{m}(u_i - \langle\phi(x_i), \theta\rangle - b)_+$$

and so $\alpha_i = \frac{1}{\lambda}(u_i - \langle\phi(x_i), \theta\rangle - b)_+$. The proof connecting (17) and (20) is similar and therefore omitted. ■

Note that dropping the constraint $\sum_i \alpha_i = 1$ is equivalent to removing b from (20). The main motivation for Theorem 4 is that it will allow us to recover propensity scores via a stochastic gradient descent procedure for KMM.

4.3 Entropy Regularization

Another penalty is to minimize the Kullback-Leibler divergence between $\hat{p}$ and p. In other words, we aim to maximize the relative entropy of $\hat{p}$ when using p as a base measure (15). The resulting optimization problem belongs to the family of constrained maximum entropy estimation problems and its dual has a functional form similar to penalized maximum likelihood, as described in [2].

Theorem 5 *Solving the optimization problem (12) using (15) is equivalent to solving*

$$\underset{\theta}{\text{minimize}}\, g(\theta) - \langle\theta, \mu\rangle + \frac{1}{2\lambda}\|\theta\|^2 \quad (25)$$
$$\text{with } g(\theta) = \log\sum_{i=1}^{m} e^{\langle\phi(x_i),\theta\rangle}.$$

In this case, $\gamma(x)$ are given by $\gamma(x) = e^{\langle\phi(x),\theta\rangle - g(\theta)}$.

This shows that the Kullback-Leibler smoothed optimization problem with a rather ad-hoc RKHS regularization as proposed by [21] does have a more profound interpretation as a relative maximum entropy problem subject to a moment matching condition, sharing much of its motivation with the kernel mean matching algorithm. Instead of the Shannon Entropy we could obviously use any other choice of f-divergence between distributions, thus yielding a convex optimization problem [1].

Proof To keep this presentation self-contained we give a direct proof. A much shorter version could be obtained by using the results of [2] and by appealing to the Fenchel duality theorem. As before we generate a Lagrange function

$$L = \int dp(x)\gamma(x)\log\gamma(x) + \lambda\left[\frac{1}{2}\left\|\mathbf{E}_{x\sim p}\left[\gamma(x)\phi(x)\right] - \mu\right\|^2 - \epsilon\right] + \Lambda\left[\mathbf{E}_{x\sim p}[\gamma(x)] - 1\right]$$

Note that we omitted the constraint $\gamma(x) \geq 0$. However, as we shall see, this constraint is always satisfied by the solution we will be obtaining, hence it is unnecessary to add this constraint explicitly. Taking a variational derivative of L with respect to γ yields the first order conditions

$$0 = \log\gamma(x) + 1 + \underbrace{\lambda\left[\mathbf{E}_{x\sim p}\left[\gamma(x)\phi(x)\right] - \mu\right]^\top}_{:=-\theta}\phi(x) + \Lambda \tag{26}$$

and consequently $\log\gamma(x) = \langle\theta, \phi(x)\rangle - g(\theta)$ where $g(\theta) = \log\mathbf{E}_{x\sim p}[\exp\langle\phi(x), \theta\rangle]$. Plugging this back into L we notice that the third term vanishes. Furthermore, the second term is given by $\frac{1}{2\lambda}\left\|\theta\right\|^2 - \lambda\epsilon$. Finally, the first term amounts to

$$\int dp(x)\gamma(x)\left[\langle\theta, \phi(x)\rangle - g(\theta)\right] = \left\langle\theta, \mu - \lambda^{-1}\theta\right\rangle - g(\theta) \tag{27}$$

which leads to $L = \langle\theta, \mu\rangle - g(\theta) - \frac{1}{2\lambda}\left\|\theta\right\|^2 - \lambda\epsilon$. Hence, for a suitable choice of λ the dual problem is maximized by solving (25). ■

5 Online Algorithms for Distribution Matching

In the previous section we showed that many covariate shift correction algorithms can be derived in a common framework. By design they scale extremely well in m' since $|X_q|$ only matters in calculating μ and $\langle\mu, \phi(x_i)\rangle$ respectively. The remainder is a constrained quadratic program (for KMM) or a general convex problem (for MEMM). Unfortunately, this scaling behavior which was also observed for MEMM by [21] is insufficient for large scale industrial problems. We address this problem by establishing online algorithms with $O(m + m')$ runtime behavior.

Assumption 6 *We assume that $\phi(x)$ can be obtained explicitly as feature vector.*

This assumption, while seemingly contradictory to the idea of having a nonparametric estimator in Hilbert Space, is quite valid. Recent techniques for feature space decomposition, such as random kitchen sinks [15] or the quadratic feature expansion for sparse data, as is common in the VowpalWabbit online solver [11], provide ample empirical and theoretical evidence. This allows us to compute a finite-dimensional approximation of μ in linear time.

5.1 Kernel Mean Matching

The key idea in obtaining an online algorithm for kernel mean matching is to solve problem described in Theorem 4 instead of the original convex program. Multiplying (20) by $\lambda' := \frac{\lambda}{m}$ and (21) by $\lambda' := \frac{1}{\lambda m}$ respectively we obtain data dependent terms for L_2 and L_∞ RND penalty, respectively

$$l_t(\theta) := \frac{\lambda'}{2}\left\|\theta\right\|^2 + \lambda' b + \frac{1}{2}(u_t - \langle\phi(x_t), \theta\rangle - b)_+^2 \tag{28}$$

$$l_t(\theta) := \frac{\lambda'}{2}\left\|\theta\right\|^2 + \lambda' b + (u_t - \langle\phi(x_t), \theta\rangle - b)_+ \tag{29}$$

We obtain a simple stochastic gradient descent procedure for (28) (with initialization $\theta = 0$):

Precompute $\mu = \frac{1}{m'}\sum_{i=1}^{m'}\phi(x_i')$
repeat
  Observe x_t from X_p
  Compute learning rate $\eta_t \leftarrow ct^{-\frac{1}{2}}$
  Compute gradient $G_t = (\langle\phi(x_t), \mu\rangle - \langle\phi(x_t), \theta\rangle - b)_+$
  Update $\theta \leftarrow (1 - \eta_t\lambda')\theta + \eta_t G_t\phi(x_t)$
  Update $b \leftarrow b - \eta_t[\lambda' - G_t]$
until converged

The modification when using (29) is $G_t = \frac{1}{2}[1 + \mathrm{sgn}(\langle\phi(x_t), \mu\rangle - \langle\phi(x_t), \theta\rangle - b)_+]$. While the problem remains strongly convex in θ it ceases to be so in b. Hence, the slower $O(t^{-\frac{1}{2}})$ for η_t.

Theorem 7 *The algorithm converges to the minimum of (20) at rate $O(T^{-\frac{1}{2}})$.*

This establishes a linear-time algorithm for convergence of the kernel mean matching algorithm: first solve the stochastic gradient descent problem for θ and b and subsequently convert θ into γ.

Proof In order to apply the guarantees from [23] we need to show boundedness in b and θ. To see that θ is bounded, evaluate (20) for $b = 0$ and $\theta = 0$. That value is an upper bound for $\frac{\lambda'}{2}\left\|\theta\right\|^2$. Moreover, it is also an upper bound on b^2, by virtue of the bias-variance decomposition. ■

5.2 Markov Chain Monte Carlo Estimation for Maximum Entropy Mean Matching

Consider the optimization problem (25). The computation of $\partial_\theta g(\theta)$ costs $O(m)$ time. Since the problem is strongly convex in θ we may apply a gradient descent with line search procedure to obtain fast $O(m \log \frac{1}{\epsilon})$ convergence for a batch solver. This is a direct application of the line search algorithm described in [5] (as before we initialize $\theta = 0$):

repeat
  Initialize $g = 0$ and $G = 0$
  for $i = 1$ **to** m **do**
    $g = g + e^{\langle \phi(x), \theta \rangle}$ and $G = G + \phi(x) e^{\langle \phi(x), \theta \rangle}$
  end for
  Compute gradient $G = \frac{G}{g} - \mu + \lambda^{-1}\theta$
  Perform linesearch for $\theta - \eta G$ with respect to (25) and update $\theta = \theta - \eta^* G$.
until converged

The disadvantage of the above algorithm is that we require a significant number of passes through the data to compute the gradients and for the line search. We address this by performing MCMC estimation of $\partial_\theta g(\theta)$ instead.

The key insight is that $\partial_\theta g(\theta)$ induces a distribution over $\phi(x_i)$ with unnormalized weights $e^{\langle \phi(x_i), \theta \rangle}$. More specifically, $\partial_\theta g(\theta) = E_{x \sim w} \phi(x)$, where w is a discrete distribution on X_p with mass $w(x_i) \propto e^{\langle \phi(x_i), \theta \rangle}$ at x_i. Drawing from this distribution can be achieved by a simple Metropolis-Hastings sampler with uniform proposal distribution with mass m^{-1} at each x_i. The Metropolis-Hastings step accepts the proposal x' given x with probability $\alpha = \min\left(1, e^{\langle \phi(x') - \phi(x), \theta \rangle}\right)$. Hence, we may interleave a stochastic gradient step with respect to θ with several MCMC steps to obtain a gradient estimate for $g(\theta, Z)$. A similar procedure was used for learning intractable CRFs by [22]. We have the following algorithm:

Initialize x with random sample from X_q and let $\theta = 0$.
repeat
  Draw random sample x' from X_q and draw k random samples $x_1, \ldots, x_k$ from X_p
  for $i = 1$ **to** k **do**
    With probability $\min\left(1, e^{\langle \phi(x_i) - \phi(x), \theta \rangle}\right)$ replace $x \leftarrow x_i$
  end for
  Update $\theta \leftarrow (1 - \frac{\eta_t}{\lambda})\theta + \eta_t \left[\phi(x') - \phi(x)\right]$
until converged

The advantage of the above algorithm is that each update now only requires us to see $O(k)$ instances. The independence Metropolis sampler mixes exponentially fast. Let $\psi(x, A)$ denote the one-step transition probability of the Metropolis sampler from x to set A. Since we use an independence sampler, the transition probability does not depend on the current state x. Also, let $\psi(x, A)^T$ denote the transition probability starting from x to set A after T draws from the Metropolis sampler. To show that the sampler mixes exponentially, it is sufficient to show that the distance of transition probability measure after T steps from the stationary distribution w w.r.t. the total variation norm decays exponentially fast in T. From [16], since $w(x_i) \leq m \cdot \frac{1}{m} = 1$, it follows that $\|\psi(x, \cdot)^T - w(\cdot)\|_{\text{TV}} \leq 2(1 - 1/m)^T$.

5.3 Moving Average Estimation for Log Partition Function

An alternative to drawing from $\partial_\theta g(\theta)$ via a Metropolis-Hastings sampler is to compute a running average R_t for $g(\theta)$ and to use this approximation in place of the true denominator. That is, we use the stochastic approximations $\phi(x) \frac{e^{\langle \phi(x), \theta \rangle}}{R_t}$. We begin by stating the gradient descent procedure which is quite similar to the one proposed in Section 5.2.

Initialize $\theta = 0$ and $R = 1$.
repeat
  Set learning rates $\eta_t = ct^{-\frac{1}{2}}$ and $\bar{\eta}_t = \bar{c}t^{-\frac{1}{2}}$
  Draw $x \in X_p$ and $x' \in X_q$
  Draw k random samples $x_1, \ldots, x_k$ from X_p
  Update normalization $R \leftarrow (1 - \bar{\eta}_t)R + \frac{\bar{\eta}_t}{k} \sum_{i=1}^k e^{\langle \phi(x_i), \theta \rangle}$
  Update $\theta \leftarrow (1 - \frac{\eta_t}{\lambda})\theta + \eta_t \left[\phi(x') - \frac{e^{\langle \phi(x), \theta \rangle}}{R} \phi(x)\right]$
until converged

The key difference to the MCMC sampler is that we now use a running average to get a handle on the denominator R. The moving average to compute R can be shown to converge, albeit with variance $O(T^{-\frac{1}{4}})$. The slower rate is due to the fact that the weight of the updates decreases with $O(T^{-\frac{1}{2}})$ and the variance decreases at half of this rate. The uncertainty in R adds a small amount of error in the gradient estimate. This can be addressed using the additive SGD error bounds of [12].

6 Experiments

In this section, we will first apply the online kernel mean matching algorithms to a small-scale UCI dataset to verify they indeed produce the same propensity score estimates (in the limit) as their batch counterparts. Then, we demonstrate the use of these online algorithms in a real-world problem with millions of data and thousands of features.

The first dataset is GISETTE from UCI [13]. The problem is to separate the highly confusable handwritten digits "4" and "9". It has 5000 features, 6000 training data, and 6500 test data. We first obtained the first principal component in the training data, and then computed the projected value of training data onto the principal component. Let m and $\bar{m}$ be the minimum and mean of the projection. We then subsampled a subset from the training data us-

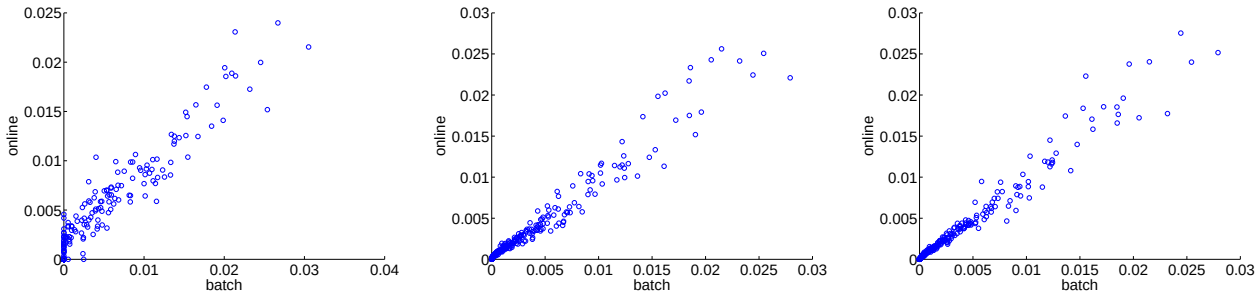

Figure 1: Online vs. batch computed propensity scores in GISETTE. From left to right: ℓ_2-KMM, MEMM w/ M-H sampling, MEMM with adaptive averaging.

ing their projected values according to a normal distribution $\mathcal{N}(m + \frac{\bar{m}-m}{3}, \frac{\bar{m}-m}{4})$. We ended up with 324 data.

We used $\lambda = 0.1$ for ℓ_2-regularized KMM and $\lambda = 1$ for MEMM. Figure 1 shows the propensity scores estimated by the three online algorithms indeed converge to the same solutions of their corresponding batch optimization problems, which is consistent with the analysis in the previous section. A closer look at the results revealed that the estimates are close to the true propensity scores, especially for small scores. For larger scores, the covariate overlap is small and results in high variance. Regularization thus reduces variance but increases bias, hence the discrepancy is larger for higher propensity scores.

The next problem we consider is estimating average number of user visits to Yahoo! front page. It records real user visits to Yahoo! front page for about 4 million *bcookies*[3] randomly selected from all bcookies during March 2010. Each bcookie is associated with a sparse binary feature vector of size around 5000. These features describe browsing behavior as well as other information (such as age, gender, and geographical location) of the bcookie. We chose a time window in March 2010. and calculated the number of visits of each of the selected bcookies during this window. To summarize, this dataset contains 4 million data, $D = \{(b_i, x_i, v_i)\}_{i=1,\dots,N}$ for $N \approx 400000$, where b_i is the i-th (unique) bcookie, x_i is the corresponding feature vector, and v is the number of visits.

If we can sample from D uniformly at random, the sample mean of v_i will be an unbiased estimate of the true average number of user visits. However, in many situations, it may be difficult to ensure a uniform sampling scheme due to practical constraints, thus the sample mean may not reflect the true quantity of interest. We used a similar PCA sampling trick as in the previous problem to compute the probability of each bcookie being subsampled. On average, the subsampling probability is around $1/4$. The set of subsampled bcookies is called the "treatment group". Treatment groups sampled this way are indeed biased: in our dataset, the average number of visits in a treatment group is around 29.5, in contrast to the actual number 23.8, resulting in a huge bias of 24%. Propensity scoring methods are therefore necessary to eliminate this sampling bias.

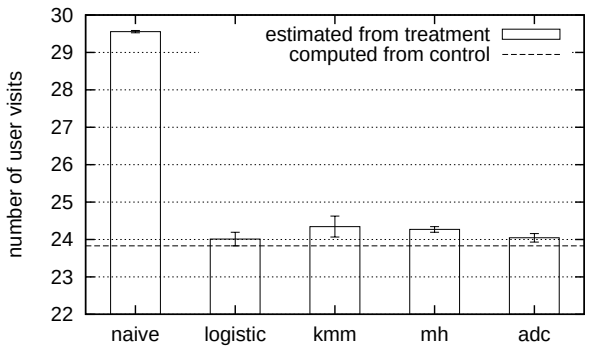

Figure 2: Estimates of average number of user visits to Yahoo! front page. Here, "kmm" uses ℓ_2 regularization, while "mh" and "adc" use entropy regularization with Metropolis-Hastings sampling and approximate denominator control, respectively.

The parameters (λ, η_t, k) were hand tuned using a separate treatment group collected from a *different* time period. We then used these parameters on 10 randomly sampled treatment groups from our dataset D.[4] The regularization parameter was as follows: 0.00001 for logistic regression, 0.001 for ℓ_2-regularized KMM, and 0.5 for MEMM. The sampling size for the MEMM was 30. The learning rate followed the decay scheme described in Section 5 with constant c tuned separately for each algorithm.

Figure 2 plots the average estimate as well as the error bar for each algorithm; "naive" corresponds to the estimate

[3] A bcookie is unique string that identifies a user. Strictly speaking, one user may correspond to multiple bcookies, but it suffices to equate a bcookie with a user for our purpose here.

[4] Since logistic regression requires data from a comparably-sized control group, we also subsampled 10 control groups.

that does not use propensity scores (equivalently, uniform β_i). The horizontal line in the figure is the average number of user visits estimated from uniformly sampled bcookies (and thus called "control"). This line is considered as "ground truth"; the closer an estimate is to this line, the more accurate it is. The results clearly demonstrate effectiveness of all the proposed online algorithm. While a naive estimate can cause a large estimation error, the propensity-score-weighted estimate are all very close to target value.

7 Conclusions

In this paper, we showed that propensity scoring and covariate shift correction coefficients can be estimated efficiently by linear-time online algorithms. This is a significant improvement over previous work which heavily relies on batch convex solvers, thus limiting the size of the treatment (or training) set. Moreover, we demonstrated that recent work on covariate shift correction can be unified in an approximate moment-matching framework, the main difference being the criteria for smoothing over the class of Radon Nikodym derivatives. This explains why these methods perform quite similarly in practice.

Our experiments showed that while all methods perform very well for propensity scoring (when compared to ground truth obtained by uniformly random sampling), the humble logistic regression is just as good as the more modern techniques. Given the widespread access to efficient logistic regression codes it is definitely one of the tools to be evaluated when dealing with propensity scoring. Out of the moment matching methods, the entropic algorithms perform slightly better, albeit at the expense of a somewhat more difficult parameter setting (we need to approximate the log-partition function, too). In this context, probably the Markov Chain Monte Carlo algorithm is the one which should be a first choice in practice.

References

[1] S.M. Ali and S.D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[2] Y. Altun and A.J. Smola. Unifying divergence minimization and statistical inference via convex duality. In H.U. Simon and G. Lugosi, editors, *Proc. Annual Conf. Computational Learning Theory*, LNCS, pages 139–153. Springer, 2006.

[3] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[4] P.L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.

[6] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A.J. Smola. A kernel method for the two sample problem. Technical Report 157, MPI for Biological Cybernetics, 2008.

[8] A. Gretton, A.J. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors, *Covariate Shift and Local Learning by Distribution Matching*, pages 131–160, Cambridge, MA, 2008. MIT Press.

[9] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

[10] D. Lambert and D. Pregibon. More bang for their bucks: assessing new features for online advertisers. *SIGKDD Explorations*, 9(2):100–107, 2007.

[11] J. Langford, L. Li, and A.L. Strehl. Vowpal Wabbit (fast online learning), 2007. http://hunch.net/˜vw/.

[12] J. Langford, A.J. Smola, and M. Zinkevich. Slow learners are fast. arXiv:0911.0491, 2009.

[13] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998.

[14] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2008.

[15] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[16] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2004.

[17] P. Rosenbaum and D. Rubin. The central role of propensity score in observational studies for causal effects. *Biometrica*, 70:41–55, 1983.

[18] B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001.

[19] L. Song, J. Huang, A.J. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions. In *ICML*, 2009.

[20] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 20*, pages 1433–1440, Cambridge, MA, 2008.

[21] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. In *SDM*, pages 443–454, 2008.

[22] S.V.N. Vishwanathan, N.N. Schraudolph, M. Schmidt, and K. Murphy. Accelerated training conditional random fields with stochastic gradient methods. In *ICML*, pages 969–976, New York, NY, USA, 2006. ACM Press.

[23] M. Zinkevich. Online convex programming and generalised infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.