A Computationally Efficient Method for Nonparametric Modeling of Neural Spiking Activity with Point Processes

Todd P. Coleman
colemant@illinois.edu
Department of Electrical and Computer Engineering and the Neuroscience Program, University of Illinois, Urbana, IL 61801, U.S.A.

Sridevi S. Sarma
sree@jhu.edu
Department of Biomedical Engineering and Institute for Computational Medicine, Johns Hopkins University, Baltimore, MD 21218, U.S.A.

Point-process models have been shown to be useful in characterizing neural spiking activity as a function of extrinsic and intrinsic factors. Most point-process models of neural activity are parametric, as they are often efficiently computable. However, if the actual point process does not lie in the assumed parametric class of functions, misleading inferences can arise. Nonparametric methods are attractive due to fewer assumptions, but computation in general grows with the size of the data. We propose a computationally efficient method for nonparametric maximum likelihood estimation when the conditional intensity function, which characterizes the point process in its entirety, is assumed to be a Lipschitz continuous function but otherwise arbitrary. We show that by exploiting much structure, the problem becomes efficiently solvable. We next demonstrate a model selection procedure to estimate the Lipschitz parameter from data, akin to the minimum description length principle and demonstrate consistency of our estimator under appropriate assumptions. Finally, we illustrate the effectiveness of our method with simulated neural spiking data, goldfish retinal ganglion neural data, and activity recorded in CA1 hippocampal neurons from an awake behaving rat. For the simulated data set, our method uncovers a more compact representation of the conditional intensity function when it exists. For the goldfish and rat neural data sets, we show that our nonparametric method gives a superior absolute goodness-of-fit measure used for point processes than the most common parametric and splines-based approaches.

1 Introduction

Characterizing neural spiking activity as a function of environmental stimuli and intrinsic effects such as a neuron’s own spiking history and
concurrent ensemble activity is important in neuroscience. Such a characterization is complex, and there is increasing need for a broad class of models to capture such details. Point-process models have been shown to be useful in characterizing neural spiking activity. For example, they have successfully characterized stimulus-response relationships computed from data recorded in hippocampal cells of a rat (Barbieri, Quirk, Frank, Wilson, & Brown, 2001); retinal ganglion cells in salamander, rabbit, and cat; and lateral geniculate nucleus neurons in cat (Keat, Reinagel, Reid, & Meister, 2001). These models also captured the dynamics of supplementary eye field cells of a macaque monkey (Kass & Ventura, 2001), the influence of one neuron’s spiking activity on another’s in sea hare (*Aplysia californica*) (Brillinger, 1988), and ensemble activity in hippocampal cells of rats (Harris, Csicsvari, Hirase, Dragoi, & Buzsaki, 2003).

Most point-process models based on likelihood analysis are parametric (Brillinger, 1988; Chornoboy, Schramm, & Karr, 1988; Barbieri et al., 2001; Brown, Barbieri, Eden, & Frank, 2003; Paninski, 2004; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005). That is, the conditional intensity function is assumed to belong to a class of parametric functions. Parametric models have several advantages. First, they are often efficiently computable. Second, the parameters may be related back to physiological and environmental factors. Finally, they have good asymptotic properties when the actual distribution lies in the assumed parametric class. However, if the true conditional intensity function does not lie in the assumed class, large errors may occur, resulting in misleading inferences.

Previous models that relax the a priori assumption of the conditional intensity function (CIF) belong to functional classes of either polynomials or piece-wise polynomials (splines), including cardinal cubic splines (Frank, Eden, Solo, Wilson, & Brown, 2002), Zernicke polynomials (Barbieri, Frank, Quirk, Wilson, & Brown, 2002), smoothing splines (Kass & Ventura, 2001), and Bayesian free-knot spline models (Dimatteo, Genovese, & Kass, 2001). These models may provide better alternatives to purely parametric models in that they are broader classes of functions, but they are often computationally expensive or may have low goodness-of-fit performance.

In contrast, there are very few computable nonparametric methods for estimating the conditional intensity function of a point-process model. Furthermore, even if a method is computable, it is difficult to show good asymptotic properties of the estimate. Yet nonparametric estimation methods have the advantage of not assuming that the CIF lies in some known class of functions. Recently Truccolo and Donoghue (2007) introduced a computationally efficient nonparametric approach to model neural point processes using stochastic gradient boosting regression. This approach estimates a discretized CIF to maximize the likelihood function in a computationally efficient manner based on a statistical learning approach. Given finite data samples, they exploit an iterative estimation procedure that introduces a parametric model in each iteration.
We propose an alternative computationally efficient method to build nonparametric point-process models that involves only computing a convex optimization problem subject to linear constraints. We show that point-process estimation is a simple application of a nonparametric nonlinear regression problem (Bertsimas & Tsitsiklis, 1997). The only restriction on the CIF is that its logarithm must be Lipschitz continuous.

We demonstrate a model selection procedure to estimate the Lipschitz parameter from data, akin to the minimum description length principle, by performing a penalized likelihood minimization procedure using the $\epsilon$-entropy of Lipschitz functions of parameter $K$ (Kolmogorov & Tikhomirov, 1959) and the index of resolvability (Barron & Cover, 1991).

We illustrate the effectiveness of our method with simulated neural spiking data, goldfish retinal ganglion neural data, and activity recorded in CA1 hippocampal neurons from an awake behaving rat. For the simulated data set, we showed that our method can uncover a more compact representation of the CIF when it exists. For the goldfish and rat neural data sets, we showed that our nonparametric method gives a superior absolute goodness-of-fit measure than all parametric approaches analyzed.

2 Point-Process Models

Consider the time interval $[0, T]$ as the time window for which our neural spike train is observed. Define $\mathcal{Y}_T$ to be the set of functions $y: (0, T] \rightarrow \mathbb{Z}_+$ that are nondecreasing, right-continuous, and $y_0 = 0$. In other words, $\mathcal{Y}_T$ is the set of point processes on $(0, T]$. Denote the $i$th epoch of point-process sample path $y \in \mathcal{Y}_T$ as

$$\bar{y}_i \triangleq \min \{ t \in [0, T] : y_t \geq i \} \quad i \in \{1, \ldots, y_T\}. \quad (2.1)$$

Succinctly, we can represent a point process as a sample path $y \in \mathcal{Y}_T$ where the jumps in $y$, given by $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{y_T}$, correspond to the times at which spikes occur. Throughout this letter, uppercase letters correspond to random variables or processes, whereas lowercase variables correspond to constants or sample paths realization.

It is well known that the CIF completely characterizes the statistical structure of most point processes used in statistical inference of neural data (Brown et al., 2003). The CIF is defined as follows (Daley & Vere-Jones, 2003),

$$\lambda(t \mid \mathcal{H}_t) \triangleq \lim_{\Delta \to 0} \frac{P(Y_{t+\Delta} - Y_t = 1 \mid \mathcal{H}_t)}{\Delta}, \quad (2.2)$$

where $\mathcal{H}_t$ is the $\sigma$-algebra generated by $(Y_\tau : \tau \in [0, t])$, as well as other possible exogenous processes that impact the propensity for a neuron to spike. Succinctly, the conditional intensity specifies the instantaneous probability of spiking per unit time, given previous neural spiking and exogenous inputs.
The history-dependent interspike interval (ISI) density at time $t$, $f_{\text{ISI}}(t \mid \mathcal{H}_t)$, and CIF is in one-to-one correspondence:

\begin{equation}
\begin{aligned}
f_{\text{ISI}}(t \mid \mathcal{H}_t) &= \lambda(t \mid \mathcal{H}_t) \exp \left\{ - \int_{\hat{y}_y}^{t} \lambda(u \mid \mathcal{H}_u) \, du \right\} \\
\lambda(t \mid \mathcal{H}_t) &= \frac{f_{\text{ISI}}(t \mid \mathcal{H}_t)}{1 - \int_{\hat{y}_y}^{t} f_{\text{ISI}}(u \mid \mathcal{H}_u) \, du},
\end{aligned}
\end{equation}

where, from equation 2.1, note that $\hat{y}_y$ is time backward from $t$ of the most recent spike.

Throughout this letter, we assume that $\mathcal{H}_t$ is generated by a random variable $X_t$ on $\mathbb{R}^m$, and we use the notation $\lambda(X_t)$ to explicitly represent equation 2.2 in this manner. Many neuroscience point-process models (e.g., renewal process models and generalized linear models (Brown, Kass, & Frank, 2002) implicitly use this assumption, where $X_t$ can pertain to recent spiking history as well as exogenous processes. Examples of how $X_t$ is interpreted will appear in the experimental results section.

For a sample path $y \in \mathcal{Y}_T$ of a point-process $Y$ with CIF $\lambda(t \mid \mathcal{H}_t) = \lambda(X_t)$ and spike times $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_y\}$, the likelihood or density of $y$ given $x$ is given by (Brown et al., 2003)

\begin{equation}
\begin{aligned}
- \log f_{Y \mid X}(y \mid x; \lambda) &= \exp \left\{ \int_{0}^{T} \log \lambda(x_t) \, dy_t - \lambda(x_t) \, dt \right\}.
\end{aligned}
\end{equation}

By discretizing $[0, T]$ into $n = T/\Delta$ intervals of length $\Delta \ll 1$ so that $dy = (dy_1, \ldots, dy_n)$ with $dy_i \triangleq y_{i+1} - y_i \in \{0, 1\}$, we can approximate equation 2.5 by

\begin{equation}
- \log f_{Y \mid X}(y \mid x; \lambda) \simeq \sum_{i=1}^{n} - \log \lambda(x_i) \, dy_i + \lambda(x_i) \Delta,
\end{equation}

where the discrete time index $i$ corresponds to the continuous interval $[0, T]$ at time $i \Delta$.

Common renewal process (point processes in which ISIs are independent and identically distributed) models in neuroscience are given by the exponential, gamma, and inverse gaussian (Brown, 2005). The generalized linear model (GLM) has recently been used to model CIFs of point processes. In its simplest form, the GLM represents the CIF parametrically as

\begin{equation}
\log \lambda(t \mid \mathcal{H}_t) = \log \lambda(X_t) = \sum_{i} \alpha_i g^{int}_i(X_t) + \sum_{j} \beta_j g^{ext}_j(X_t).
\end{equation}
where \( g_{j}^{ext}(x) \) and \( g_{i}^{int}(x) \) are known basis functions that capture the extrinsic and intrinsic effects (carried in \( X_t \in \mathbb{R}^m \)) on the probability of spiking at time \( t \), respectively. The extrinsic and intrinsic associated parameters to be estimated are \( \alpha_i, \beta_j \in \mathbb{R} \), respectively.

The time-rescaling theorem (Lenglart, 1977; Daley & Vere-Jones, 2003) provides a change of measure for arbitrary point processes via their sample paths and CIF, resulting in a unit-rate Poisson process:

**Theorem 1.** Given a point-process \( Y \in \mathcal{Y}_T \), with associated CIF \( \lambda(t \mid \mathcal{H}_t) \), the random variables

\[
Z_k = \int_{\tilde{Y}_{k-1}}^{\tilde{Y}_k} \lambda(t \mid \mathcal{H}_t) \, dt \quad k = 1, \ldots, Y_T, \tag{2.7}
\]

are independent and identically distributed (i.i.d.) exponential random variables of unit rate.

Goodness-of-fit tests with point processes are widely performed (Brown et al., 2002) by estimating the conditional intensity, \( \hat{\lambda}(t \mid \mathcal{H}_t) \), performing time rescaling (see equation 2.7) on the interspike intervals to construct the sequence \( \{Z_k\} \), constructing an ideally uniform \([0, 1]\) sequence \( \{U_k = 1 - \exp(Z_k)\} \), and performing a Kolmogorov-Smirnov (KS) test. The KS statistic is given by

\[
\max_{t \in [0, 1]} \left| F_{\text{empirical}}(t) - F_{\text{uniform}}(t) \right|, \tag{2.8}
\]

where \( (F_{\text{empirical}}(u)) \) is given by the time-rescaling theorem applied to \( \hat{\lambda}(t \mid \mathcal{H}_t) \), and \( F_{\text{uniform}}(t) \) is the cumulative distribution function (CDF) for a uniform-[0, 1] random variable. We use the KS statistic to test absolute goodness-of-fit for all point-process models computed in our examples. It is important to note that the KS plots were generated on test data so that the models were all cross-validated.

### 3 Dynamic Maximum Likelihood Estimation Problem for Point Processes

We now consider a nonparametric estimation procedure and make the following assumption:

\[
\log \lambda(t \mid \mathcal{H}_t) = z(X_t), \tag{3.1}
\]

where \( X_t \in \mathbb{R}^m \) encodes all the relevant information about \( t \), and \( z : \mathbb{R}^m \to \mathbb{R} \) is an unknown function assumed to be Lipschitz continuous with parameter \( K \). Given this assumption, we discuss a convex optimization procedure to
estimate the structure of \( z \). Recall that a function \( z : \mathbb{R}^m \to \mathbb{R} \) is Lipschitz continuous with parameter \( K \geq 0 \) if

\[
|z(x_1) - z(x_2)| \leq K \|x_1 - x_2\| \tag{3.2}
\]

holds for all \( x_1, x_2 \in \mathbb{R}^m \) (we note that our procedure allows for any norm, and all discussions of complexity and interpolation still hold. Unless mentioned otherwise, we will assume \( \|a\| = \|a\|_\infty = \sup_{i=1,..,m} |a_i| \)).

This approach is inspired by the methodology of Bertsimas, Gamarnik, and Tsitsiklis (1999). In this point-process setting, we exploit the fact that the point-process likelihood is a convex functional of the conditional intensity, which leads to a computationally efficient dynamic maximum likelihood estimation (DMLE) problem (Bertsimas et al., 1999). Specifically, we introduce decision variables \( \lambda = (\lambda_1, \ldots, \lambda_n) \), express the negative log likelihood of the data by way of equation 2.6, and minimize it subject to the Lipschitz continuity constraint, equation 3.2, at each pair of data points:

\[
\min_{\lambda} \sum_{i=1}^n -\log \lambda_i d y_i + \lambda_i \Delta \tag{3.3a}
\]

\[
s.t. \quad |\log \lambda_{j_1} - \log \lambda_{j_2}| \leq K \|X_{j_1} - X_{j_2}\|, \quad j_1, j_2 = 1, \ldots, n. \tag{3.3b}
\]

By letting \( z_i = \log \lambda_i \), and \( C_{j_1, j_2} = K \|X_{j_1} - X_{j_2}\| \), it follows that equation 3.3 becomes

\[
\min_{\hat{z}} \sum_{i=1}^n -z_i d y_i + \Delta \exp(z_i) \tag{3.4a}
\]

\[
s.t. \quad z_{j_1} - z_{j_2} \leq C_{j_1, j_2} \quad j_1, j_2 = 1, \ldots, n. \tag{3.4b}
\]

Note that equation 3.4 corresponds to minimizing a convex function over a convex set, and thus the problem is, in principle, efficiently solvable.

This procedure for other likelihoods that are convex functionals of the dynamic function was shown in Bertsimas et al. (1999) to have good properties, including consistency. Another useful property of such a procedure is that the optimal solution, \( (\hat{z}_1, \ldots, \hat{z}_n) \), can be used to interpolate a Lipschitz-continuous function at any point \( x \in \mathbb{R}^m \):

\[
\hat{z}(x) \triangleq \max_{1 \leq i \leq n} \{ \hat{z}_i - K \|x - X_i\| \}, \tag{3.5}
\]

where the following properties hold:

- \( \hat{z}(x_i) = \hat{z}_i \).
- \( \hat{z}(x) \) is a Lipschitz continuous function with parameter \( K \).

The proof details follow directly from Bertsimas et al. (1999, proposition 1).
4 Computational Reduction of the DMLE

Although our DMLE, equation 3.3, is a convex optimization problem, it corresponds to minimizing a cost function with \(n\) unknowns and \(O(n^2)\) constraints. Typically neural data involve sample sizes on the order of \(10^6\), rendering a need for a computationally efficient way to compute equation 3.4. In this section, we make a series of observations that lead to a computationally efficient algorithm for solving equation 3.4. The observations are outlined as follows. First, we uncover the separable (Bertsekas, 1999) structure of equation 3.4 and look at the corresponding dual problem, which corresponds to minimizing an unconstrained function of \(O(n^2)\) variables. Then we reduce the complexity of the dual problem by deriving an equivalent nondifferentiable convex optimization problem of \(n\) variables and \(O(n)\) constraints. This can be solved by employing well-known subgradient methods (Bertsekas, 1999), and the calculation of each subgradient corresponds to solving a (particularly efficiently solvable) network flow problem (Bertsimas & Tsitsiklis, 1997).

These two approaches significantly reduce complexity and always converge to the optimal solution.

4.1 Dual Problem Formulation. We now introduce penalties \(\mu_{j_1, j_2}\) for each constraint in equation 3.4b and formulate the Lagrangian,

\[
L(z, \mu) = \sum_{i=1}^{n} f_i(z_i) + \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \mu_{j_1, j_2} (z_{j_1} - z_{j_2} - C_{j_1, j_2}),
\]

where \(f_i(z_i) = -z_i dy_i + \Delta \exp(z_i)\). With some rearranging, it follows that

\[
L(z, \mu) = \sum_{i=1}^{n} L_i(z_i, \mu)
\]

(4.1a)

\[
L_i(z_i, \mu) \triangleq -z_i dy_i + \Delta \exp(z_i) + \left[ \sum_{j_1=1}^{n} \mu_{j_1, i} \left( -z_i - \frac{1}{2} C_{j_1, i} \right) + \sum_{j_2=1}^{n} \mu_{i, j_2} \left( z_i - \frac{1}{2} C_{i, j_2} \right) \right].
\]

(4.1b)

From this, we define the dual function \(q(\mu)\) as

\[
q(\mu) = \min_{\hat{z}} L(\hat{z}, \mu).
\]

(4.2)
For any $\theta \geq 0$, it is straightforward to see that $q(\theta)$ serves as a lower bound to the optimal cost in equation 3.4. To make this lower bound as tight as possible, maximizing over all $\theta \geq 0$ gives a dual problem:

$$\max \quad q(\mu)$$

subject to

$$\mu \geq 0.$$  

Since the original problem is a convex optimization problem, the dual problem, equation 4.3, is also a convex optimization problem, and the optimal cost of that problem is equal to the optimal cost to the primal problem, equation 3.4. This follows from standard convex optimization theory (Bertsekas, 1999).

We now note that by separability in equation 4.1,

$$q(\mu) = \sum_{i=1}^{n} q_i(\mu)$$

$$q_i(\mu) \triangleq \min_{z_i} L_i(z_i, \mu).$$

By defining

$$W_i(\mu) \triangleq dy_i + \sum_{j_1} \mu_{j_1, i} - \sum_{j_2} \mu_{i, j_2},$$

it follows that we can characterize $q_i(\mu)$,

$$0 = \frac{\partial L_i(z_i, \mu)}{\partial z_i} = -W_i(\mu) + \Delta \exp(z_i),$$

and so we have $z_i^* = -\log \Delta + \log W_i(\mu)$. Note that $W_i(\mu) \geq 0$ for all $i, \mu$.

We now simplify the expression for $q(\mu) = \sum_{i=1}^{n} q_i(\mu)$:

By equation 4.6, it follows that

$$\sum_{i=1}^{n} W_i(\mu) = \sum_{i=1}^{n} dy_i.$$

The appearance of the term $-\frac{1}{2}C_{j_1, j_2} \mu_{j_1, j_2}$ in $q(\mu)$ appears twice—once for $q_{j_1}(\mu)$ and once for $q_{j_2}(\mu)$. So it follows that

$$q(\mu) = \left[ \sum_{i=1}^{n} (1 + \log \Delta) dy_i - W_i(\mu) \log W_i(\mu) \right] - \left[ \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} C_{j_1, j_2} \mu_{j_1, j_2} \right],$$
and thus performing our nonparametric regression is equivalent to the following dual problem (D1):

\[
\begin{align*}
\max \ q(\mu) &= \left[ \sum_{i=1}^{n} (1 + \log \Delta) d y_i - W_i(\mu) \log W_i(\mu) \right] \\
&- \left[ \sum_{j=1}^{n} \sum_{j_2=1}^{n} C_{j_1,j_2} \mu_{j_1,j_2} \right] \\
\text{s.t. } \mu &\geq 0, \\
W_i(\mu) &\triangleq d y_i + \sum_{j_1=1}^{n} \mu_{j_1,i} - \sum_{j_2=1}^{n} \mu_{i,j_2}, \\
C_{j_1,j_2} &\triangleq K \| X_{j_1} - X_{j_2} \|.
\end{align*}
\]

The optimal solution is given in closed form simply as

\[
\begin{align*}
z^*_i &= -\log \Delta + \log W_i(\mu^*) \\
\Rightarrow \hat{\lambda}^*_i &= \frac{1}{\Delta} W_i(\mu^*).
\end{align*}
\]

4.2 Complexity Reduction: An Equivalent Dual Problem. Note that the dual problem above has \(O(n^2)\) unknowns. As such, this is also computationally infeasible for moderate data sizes. We now show that the dual can be significantly simplified. Consider, for a fixed \(W \geq 0\), the set

\[
S(W) = \left\{ \mu : \mu \geq 0, \ W_i - d y_i = \sum_{j_1=1}^{n} \mu_{j_1,i} - \sum_{j_2=1}^{n} \mu_{i,j_2}, \ i = 1, \ldots, n \right\},
\]

where \(\sum_i W_i = \sum_i d y_i\). This can be described succinctly by saying that

\[
A \mu = b(W, d y), \\
\mu \geq 0,
\]

where \(A\) is a matrix consisting of 0, 1, and \(-1\) and entries, with \(n\) rows and \(n(n-1)\) columns, and \(b = b(W, d y) = W - d y\) is the column constraint matrix. Note that

\[
\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} (W_i - d y_i) = \left( \sum_{i=1}^{n} W_i \right) - \left( \sum_{i=1}^{n} d y_i \right) = 0. \tag{4.7}
\]
We are fixing the values of $W$ and would like to vary over the remaining free parameters $\mu$, consistent with this fixed $W$, to optimize $q(\mu)$. Note that maximizing $q(\mu)$ subject to this fixed value of $W$ is equivalent to performing the following linear program (LP):

$$\min \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} C_{j_1,j_2} \mu_{j_1,j_2}$$  \hspace{1cm} (4.8a)

subject to

$$A\mu = b(W, dy)$$  \hspace{1cm} (4.8b)

$$\mu \geq 0.$$  \hspace{1cm} (4.8c)

This LP is in standard form, and thus there must exist a basic feasible solution that is optimal (Bertsimas & Tsitsiklis, 1997). Also note that any column of $A$ has exactly two nonzero values, with one being 1 and the other being $-1$. Combining this observation with equation 4.7, it follows that this is a network flow problem (Bertsimas & Tsitsiklis, 1997). We have a graph $G = (V, E)$ with a set $V$ of vertices ranging from 1 to $n$, where each vertex corresponds to a millisecond observation. The edge set $E$ corresponds to directed edges $(j_1, j_2)$ in the graph, and each edge $(j_1, j_2)$ has an associated cost $C_{j_1,j_2}$. This is exactly the $C_{j_1,j_2}$ corresponding to the Lipschitz constraint. Note that the graph is fully connected because we want the Lipschitz constraints to be satisfied for all possible pairs. To enforce absolute value constraints, this means that the directed edges $(j_1, j_2)$ and $(j_2, j_1)$ are both in edge set $E$ and both have the same cost $C_{j_1,j_2}$. There is a flow $b_i = W_i - dy_i$ that is supplied to the network at node $i$.

As an example, suppose we were looking at problem where $n = 3$, and thus

$$\mu = [\mu_{12}, \mu_{13}, \mu_{21}, \mu_{23}, \mu_{31}, \mu_{32}]^T$$

has $n(n-1) = 6$ components. Then $A$ will take the form

$$A = \begin{bmatrix} -1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 & -1 & -1 \end{bmatrix}.$$  \hspace{1cm}

By virtue of the dual, this can also be expressed as

$$\max \sum_{i}^n p_i(W_i - dy_i)$$  \hspace{1cm} (4.9a)

subject to $p_{j_2} - p_{j_1} \leq C_{j_1,j_2}, \ j_1 = 1, \ldots, n, \ j_2 = 1, \ldots, n, \ j_1 \neq j_2.$  \hspace{1cm} (4.9b)
In general, network flow problems are efficiently solvable, and the network simplex method, dual ascent methods, and others allow for extremely efficient solutions (Papadimitriou & Steiglitz, 1998). They are often orders of magnitude faster than implementations of the standard LP simplex method. Thus, we now have the equivalent problem to our dual—a computationally efficient dual problem (*)

\[
\begin{align*}
\min \quad & \bar{q}(W) = \sum_{i=1}^{n} W_i \log W_i + R(W) \\
\text{s.t.} \quad & W \geq 0, \\
& \sum_{i=1}^{n} W_i = \sum_{i=1}^{n} d y_i \\
& R(W) \triangleq \max_{p} \sum_{i} p_i (W_i - d y_i) \\
& \text{s.t.} \quad p_{j_2} - p_{j_1} \leq C_{j_1,j_2}, \quad j_1, j_2 = 1, \ldots, n,
\end{align*}
\]

The optimal solution is given in closed form simply as

\[
\begin{align*}
z^*_i &= -\log \Delta + \log W^*_i \\
\Rightarrow \lambda^*_i &= \frac{1}{\Delta} W^*_i.
\end{align*}
\]  

(4.10)

Note that by exploiting this structure, we have eliminated the \(O(n^2)\) variables, and thus the complexity of the problem has been vastly diminished. Namely, the original problem involved \(O(n^2)\) constraints (which for \(30s = 30,000\) ms of retinal data cannot even be fit into a Matlab matrix without memory overflow errors), and the dual problem involved on the order of \(O(n^2)\) variables that suffer from the same Matlab fate. By reducing our problem as described, there are only \(n\) variables. Every call to the dual function involves the computation of \(R(W)\), which can be done with any special-purpose network flow algorithm. In the appendix, we describe how to efficiently search to optimize across \(W\).

5  An Information-Theoretic Procedure to Select the Lipschitz Constant from Data

For our nonparametric model specified by equation 3.1, it is unclear a priori what the dimension \(m\) is, and likewise for the Lipschitz constant \(K\), and thus they must be learned in a principled way from the data. Selecting the optimal integer \(m\) is a well-established problem and can be performed using the standard minimum description length by penalizing the negative
log likelihood $-\frac{1}{n} \log f_{Y|X}(y | x; m)$ by an amount $\frac{m \log n}{2n}$ for a consistent estimate (Barron, Rissanen, & Yu, 1998). Throughout this discussion, we assume $m$ is fixed. If one needs to optimally select $m$ as well as $K$, the procedure mentioned below can easily be augmented by searching over $m$ and adding the penalty $\frac{m \log n}{2n}$ to the normalized penalized negative log likelihood, given in equation 5.8. As for optimally selecting a Lipschitz constant, we know that for a sequence of increasing constants $\{K_j\}$, by virtue of equation 3.4, the feasible sets are nested increasingly in $j$. Our approach is to embed the problem of selecting $\hat{K} = K_j$ directly into the optimization problem in the spirit of minimum description length (Barron et al., 1998) by selecting the candidate CIF that minimizes the total length in bits required to describe the data (Barron & Cover, 1991; Barron et al., 1998):

Define the Lipschitz family of functions on $[0, 1]$ of parameter $K$ as

$$\mathcal{L}(K) \triangleq \{ f : [0, 1] \to \mathbb{R} : | f(a) - f(b) | \leq K | a - b | \}.$$  

The $\epsilon$-entropy $H_\epsilon(\mathcal{S})$ of a set $\mathcal{S}$ of functions is the logarithm of the smallest set $\mathcal{S}'$ of functions such that for any $f \in \mathcal{S}$, there exists an $f' \in \mathcal{S}'$ such that $\| f - f' \|_\infty < \epsilon$. It is known (Kolmogorov & Tikhomirov, 1959) that for sufficiently small $\epsilon$,

$$H_\epsilon(\mathcal{L}(K)) = \frac{K}{\epsilon} + \log \left( \frac{1}{\epsilon} \right). \quad (5.1)$$

For any $\epsilon > 0$, we consider a finite set $\Gamma_K(\epsilon) \subset \mathcal{L}(K)$, where $\log | \Gamma_K(\epsilon) | = H_\epsilon(\mathcal{L}(K))$.

We consider an increasing list $(K_j : j = 1, 2, \ldots)$ of Lipschitz constants, and consider an associated nested list $(\Gamma_{K_j}(\epsilon) : j = 1, 2, \ldots)$ of Lipschitz functions:

$$\Gamma(\epsilon) = \bigcup_{j=1}^\infty \Gamma_{K_j}(\epsilon). \quad (5.2)$$

For any $p \in \cup_{K>0} \mathcal{L}(K)$, and neural spike train $y \in \mathcal{Y}_T$ drawn according to equation 2.5 with log $\lambda = p$, we define the length $R_n(p)$ as the minimum number of bits per input symbol required to represent $y$ with a prefix-free code (Cover & Thomas, 2006).

Define $R_n(p, \epsilon)$ as the minimum number of bits per symbol required to describe $y$ with a prefix-free source code using a description $q \in \Gamma(\epsilon)$. Note that this can be done without loss of generality in three stages: first specifying $j$ with a universal prefix-free code for the set of integers (which requires $O(\log j)$ bits; Rissanen, 1983), then specifying the index of $q$ in $\Gamma_{K_j}$ (which requires $H_\epsilon(\mathcal{L}(K_j))$ bits), and finally encoding the data $y \in \mathcal{Y}_T$ with an optimal prefix-free code, which requires the Shannon code length (Cover
& Thomas, 2006) of $-\log f_{Y|X}(y \mid x; q)$ bits:

$$R_n(p, \epsilon) = \min_{j \in \mathbb{Z}_+, q \in \Gamma_K(e)} \frac{1}{n} \log \frac{f_{Y|X}(y \mid x; q)}{f_{Y|X}(y \mid x; \hat{q}_K)} (5.3)$$

$$\simeq \min_{j \in \mathbb{Z}_+} \frac{1}{n} H_e(\mathcal{L}(K_j)) - \frac{1}{n} \log f_{Y|X}(y \mid x; \hat{q}_K) + D(p \parallel q), (5.4)$$

where equation 5.4 follows for large $n$ and from the Kullback-Leibler divergence interpretation (Cover & Thomas, 2006) of the excess code length for optimally source coding a source with distribution $p$ under the assumption it arose from $q$:

$$D(p \parallel q) \triangleq \mathbb{E}_p \left\{ \log \frac{f_{Y|X}(y \mid x; p)}{f_{Y|X}(y \mid x; q)} \right\}.$$ 

Note that for a fixed $j$ in equation 5.4 and for small $\epsilon$, the inner minimization will result in the maximum likelihood solution $\hat{q}_{K_j}$ from equation 3.3 for a fixed $K_j$. Thus, we have

$$R_n(p, \epsilon) \simeq \min_{j \in \mathbb{Z}_+} \frac{1}{n} H_e(\mathcal{L}(K_j)) - \frac{1}{n} \log f_{Y|X}(y \mid x; \hat{q}_{K_j}) + D(p \parallel \hat{q}_{K_j}). (5.5)$$

It is well known (Barron & Cover, 1991) that for small $\epsilon$, $p \in \mathcal{L}(K)$, and $q \in \Gamma_K(e)$: $D(p \parallel q) \simeq \frac{1}{2} \epsilon^2$. So we can optimize how $\epsilon$ varies with $n$ and $K_j$ in $R_n(p, \epsilon)$ to trade off the increase of $H_e(\mathcal{L}(K_j))$ with the decrease of $D(p \parallel q)$ as $\epsilon \downarrow 0$. It directly follows that to attain the best rate of decay in $n$, we should select $\epsilon^*(n, K_j) \simeq \left(\frac{K_j}{n}\right)^\frac{1}{3}$. As a consequence, we arrive at

$$\hat{j} = \arg\min_{j \in \mathbb{Z}_+} \frac{1}{n} \log f_{Y|X}(y \mid x; \hat{q}_{K_j}) + \left(\frac{K_j}{n}\right)^\frac{2}{3} (5.6)$$

$$\hat{q} = \hat{q}_{K_j}, (5.7)$$

where for future reference, for any fixed $j$ and associated $\hat{q}_{K_j}$, we denote the normalized penalized negative log likelihood as

$$-\frac{1}{n} \log f_{Y|X}(y \mid x; \hat{q}_{K_j}) + \left(\frac{K_j}{n}\right)^\frac{2}{3}. (5.8)$$

Note that in our context, practically speaking, in equations 5.8 and 5.6 $\log f_{Y|X}(y \mid x; \hat{q}_{K_j})$ is given by equation 2.5, approximated practically by equation 2.6, and the conditional intensity in equation 2.6 satisfies
our Lipschitz assumption, using the estimated function from the DMLE:
\[
\log \lambda(t | \mathcal{H}_t) = \hat{q}_{K_j}(X_t).
\]

Under statistical assumptions on the almost-sure convergence of
\[-\frac{1}{n} \log f_{Y|X}(Y | X; p)\] along with \(p \in \Gamma\), it is known that this procedure
is consistent: \(\hat{q} \to p\) a.s. (Barron & Cover, 1991). Moreover, under stronger
statistical assumptions on \(-\frac{1}{n} \log f_{Y|X}(Y | X; p)\), the index of resolvability
(Barron & Cover, 1991) provides a bound on the convergence rate of this
estimation procedure:
\[
D(p \| \hat{q}) \leq O\left(n^{-\frac{2}{3}}\right).
\]

6 Concise Summary of the Proposed Algorithm

In this section, we develop a concise summary of the proposed algorithm:

1. Pick a large \(\bar{j}\) and a sequence of Lipschitz constants, \(K_1 \leq K_2 \leq \cdots \leq K_{\bar{j}}\).
2. For \(j = 1, \ldots, \bar{j}\), execute DMLE\((K_j)\):
   \begin{enumerate}
   \item Initialize \(W_1 = W_\ast(K_{j-1})\)
   \item For \(r = 1, \ldots, \bar{r}\):
     \begin{enumerate}
     \item Calculate the network flow optimal solution \(p_{K_j}(W_r) = R_{K_j}(W_r)\).
     \item Calculate the subgradient \(g_r = -[1 + \log W_r - p_{K_j}(W_r)]\).
     \item Calculate \(W_{r+1} = [W_r + s_r g_r]_+\)
     \end{enumerate}
   \end{enumerate}
3. Construct the optimal \(k\) using the information-theoretic model selection
   procedure:
   \[
   \log \hat{\lambda}(\cdot) = \hat{q}_K(\cdot) \text{ via equations 2.6, 5.6, and 5.7.}
   \]

When initializing \(D2\) for \(K_j\) in step 2.1, an initial “good” feasible solution
is the optimal solution from DMLE\((K_{j-1})\). This follows from the assumption
that \(K_1 \leq K_2 \leq \cdots \leq K_{\bar{j}}\) and the fact that \(\mathcal{L}(K_{j-1}) \subseteq \mathcal{L}(K_j)\). Using
the subgradient method, step 2.2, as a way to solve the DMLE\((K_j)\) is justified
in the appendix. The details of steps 2.2.1 through 2.2.3 can also be found
in the appendix. \(\bar{r}\) can also be adaptively chosen so that step 2.2 terminates
when \(W_{r+1}\) and \(W_r\) deviate by less than an accepted tolerance level.

7 Computing Uncertainty

To compute confidence bounds for our estimate of the CIF, we applied an
algorithm based on the time-rescaling theorem for point processes which is
described in Sarma et al. (2009). This bootstrap method is applicable to both parametric and nonparametric point-process models. Commonly referred to as the Monte Carlo evaluation of the CIF of a point process, one first computes a nonparametric maximum likelihood estimate (MLE) of the CIF, then draws bootstrap samples from the MLE to calculate bootstrap samples of the CIF (Efron, 1982). Asymptotic convergence is shown for the case where the CIF (or any random variable) is infinite-dimensional in Bickel and Freedman (1981), Singh (1981), and Lahiri (2006). Consider a given estimate of a CIF, \( \hat{\lambda}(t \mid H_t) \), for all \( t \in (0, T] \). To compute confidence bounds for our estimate \( \hat{\lambda}(t \mid H_t) \), for all \( t \in (0, T] \), we apply the following bootstrap algorithm, where the \( j \)th bootstrap sample of the statistic for \( j = 1, \ldots, B \) is computed as follows:

1. Draw an i.i.d. sequence \( \{Z^j_k\} \) of unit-rate exponentially distributed random variables for \( j = 1, \ldots, B \) and \( k \in \mathbb{Z}_+ \).

2. Define \( \vec{Y}^j_k \) as the solution to

\[
Z^j_k = \int_{\vec{Y}^j_{k-1}}^{\vec{Y}^j_k} \hat{\lambda}(u \mid H_u) \, du,
\]

with \( \vec{Y}^j_0 = 0 \) for \( j = 1, \ldots, B \). For a fixed \( j \), stop on the \( k \) for which \( \vec{Y}^j_k \leq T \) and \( \vec{Y}^j_k > T \). Thus, \( \{\vec{Y}^j_k\} \) form the jump (i.e., spike times) for the point process \( Y^j \in \mathcal{Y}_T \).

3. For each \( j \), from the bootstrap data \( Y^j \in \mathcal{Y}_T \), estimate \( \lambda(t \mid H_t) \), using the DMLE procedure, equation 5.7, to produce \( \hat{\lambda}^j(t \mid H_t) \). Once all \( B \) bootstrapped spike train samples are generated along with their corresponding CIF estimates, we compute the 95% confidence bounds by ordering the bootstrapped CIFs, \( \hat{\lambda}^j(t \mid H_t) \), for each time \( t \) and then taking the fifth quantile as the lower bound and the ninetieth quantile as the upper bound. An example is provided in Figure 1.

8 Experimental Results

8.1 Simulated Data Set. To test the effectiveness of our approach on a controlled data set we constructed 1.5 s of data. CIF of the spike train was constructed as follows:

\[
X_i \in [-0.8, -0.2]
\]

\[
\log \lambda(i \Delta \mid H_{i\Delta}) \simeq \log \lambda(X_i) = \theta_0 + \frac{K}{\omega} \sin(\omega X_i),
\]

where \( \{X_i : 1 \leq i \leq 1500\} \) are uniformly spaced across \([-0.8, 0.2]\); \( K = 10; \theta_0 = 4.7; \omega = 6\pi \); and \( \Delta = 1 \) ms. Note that in equation 8.2, \( \log \lambda \triangleq p \in \mathcal{L}(10) \). Using a list of increasing candidate \( K \) values, starting 0 with increments of 0.5, the optimum procedure given by equation 5.7 was performed with each
individual $\hat{q}_{K_j}$ solved using the DMLE approach, equation 3.3. Figure 2 (top) illustrates the penalized normalized negative log likelihoods, given by equation 5.7, pertaining to equation 5.6, for various values $K_j$. Note that with this finite $n$, the optimal $K_j$ is 9.5 when the true value is 10. We also compare our estimate $\hat{q}_{9.5}$ of the CIF to the actual $p = \log \lambda$ in Figure 2 (bottom) and note that our nonparametric estimate, which has no prior assumption on the shape of $p$, is able to uncover the sinusoid waveform.

8.2 Goldfish Retinal Ganglion Data. We study a spike train data series from a goldfish retinal ganglion cell neuron recorded in vitro. The retinæ were removed from the goldfish and maintained in a flow of moist oxygen, and recordings were made with an extracellular microelectrode under constant illumination (constant stimulus). The data are 975 spikes recorded over 30 seconds from neuron 78 in Iyengar and Liao (1997). Figure 3 (top) illustrates the histogram of ISIs in these data. The nonparametric estimator treats $x$ as follows: each $x_i \in \{0, 1\}^{10}$ is modeled to capture history with the interpretation that

$$x_{i,j} = d y_{i-j} \quad j \in \{1, \ldots, 10\},$$

with $\Delta = 1$ ms. Using a list of possible $K$ values beginning at 0 with increments of 0.5, equation 5.7 was performed, with each individual $\hat{q}_{K_j}$ solved
using the DMLE approach, equation 3.3. Figure 3 (bottom) shows the penalized normalized negative log-likelihood values, given by equation 5.8, across $K_j$ and the optimal $\hat{K} = 4.5$. The subsequent history-dependent interspike interval density, from $\hat{q}_{\hat{K}}$, is given in Figure 4 (top). With this approach, we can account for bursting phenomena empirically with our
Figure 3: (Top) Histogram of ISIs for the retinal data set. (Bottom) Penalized normalized negative log likelihood, given by equation 5.8, across candidate $K_j$ values for the retinal ganglion data set. $\hat{K} = 4.5$. 
Figure 4: (Top) History-dependent interspike interval density estimated from the optimal nonparametric model, with $\hat{K} = 4.5$. (Bottom) KS plot on cross-validated data using the optimal estimated conditional intensity for the nonparametric model, as well as optimal estimates from parametric models.

approach by estimating the conditional intensity and application of equation 2.3. Note that the ISI density has its maximum delayed, to account for refractoriness. Also, a second local peak at 10 ms appears to be reminiscent of bursting.

For comparison, parametric models were developed with $X_t \in \mathbb{R}$ to the time (in ms) since the last spike. The parametric models built in comparison are:
1. A renewal process model with $f_{\text{ISI}}(t | H_t)$ pertaining to an exponential distribution on the interspike intervals
2. A renewal process model with $f_{\text{ISI}}(t | H_t)$ pertaining to a gamma distribution on the interspike intervals
3. The conditional intensity pertains to a generalized linear model:

$$\lambda(i \Delta | H_{i\Delta}) = \exp \left\{ \sum_{j=1}^{10} \alpha_j \sum_{k=l(j,1)}^{l(j,2)} d y_{i-k} \right\}$$

with $l = [1, 3; 4, 6; 7, 8; 9, 10; 11, 20; 21, 30; 31, 40; 41, 50; 51, 60; 61, 70; 71, 80; 81, 90; 91, 100; 101, 110; 111, 120]$.

The optimal order of the GLM model was estimated using the minimum description length procedure for parametric families (Barron et al., 1998), which penalizes the normalized negative log likelihood, $\frac{-1}{n} \log f_{y|x}(y|x;d)$, with a penalty $\frac{d \log(n)}{2n}$ - a model of order $d$.

In order to assess goodness of fit and cross-validation, each of the parametric models, along with $\hat{q}_{K,j}$, was optimally estimated using maximum likelihood on 15 s of data and then cross-validated on an adjacent nonoverlapping 15 s of data. For the GLM model, the minimum description length model selection approach was used to estimate the optimal order of 16. The time-rescaling theorem and Kolmogorov-Smirnov plots with 95% confidence intervals were used, and subsequent KS plots on the time-rescaled interspike intervals on the cross-validated data set are shown in Figure 4 (bottom). Note that the nonparametric model has the best KS plot across all cross-validated parametric models, including the GLM model.

### 8.3 Rat CA1 Hippocampal Data.

We apply our nonparametric method to a spike train recorded from a CA1 hippocampal neuron in a rat foraging in an open circular environment. Both place cell microelectrode array recordings and position data were captured from a Long-Evans rat freely foraging in a circular environment 70 cm in diameter with walls 30 cm high and a fixed visual cue. The data we used are 137 spikes recorded over the first 3 seconds from one CA1 neuron. Details of the experiment can be found in Barbieri et al. (2004). Figure 5 illustrates the position trajectory of the rat and the spiking histogram relative to the position of the rat over the first 20 seconds. Barbieri et al. (2004) built a point-process model on the same data set using spatial Zernike polynomials via maximum likelihood estimation. Specifically, the CIF took the form

$$\log \lambda_z(x_t) = \sum_{l=0}^{L} \sum_{m=-l}^{l} \alpha_{l,m} Z_{l,m}(\rho(t), \phi(t)).$$
Figure 5: (Top) Rat trajectory over 20 seconds. (Bottom) Spiking histogram of single place cell over 20 seconds.
where $\rho(t) = r^{-\frac{1}{2}}[(x(t) - \eta_1)^2 + (y(t) - \eta_2)^2]^{\frac{1}{2}}$, $\phi(t) = \tan^{-1}[(x(t) - \eta_1)(y(t) - \eta_2)^{-1}]$, $(\eta_1, \eta_2) = (35 \text{ cm}, 35 \text{ cm})$ is the center of the circular environment, and $r = 35 \text{ cm}$ is the radius of the circular environment, and $Z_{l,m}$ is the $m$th component of the $l$th order Zernike polynomial,

$$Z_{l,m}(\rho(t), \phi(t)) = \begin{cases} R_{l,m}(\rho(t)) \sin(m \phi(t)) & m > 0 \\ R_{l,m}(\rho(t)) \cos(m \phi(t)) & m < 0 \end{cases}$$

$$R_{l,m}(\rho(t)) = \begin{cases} \frac{(l-m)!}{2} \frac{(-1)^{l-j} l!}{j!(\frac{l+m}{2}-j)!(\frac{l-m}{2}-j)!} & (l-m) \text{ even} \\ 0 & o.w. \end{cases}$$

where $0 \leq |m| \leq l$. More details are found in Barbieri et al. (2004).

We compare our nonparametric method to estimate the CIF as a function of the two-dimensional position vector $x_t$ and compare it to the Zernike polynomial modeling approach described above. An optimal number of Zernike basis functions was selected using the minimum description length procedure (Barron et al., 1998) by penalizing the negative log likelihood, $-\frac{1}{n} \sum_{i=1}^{n} f(y_i | x_i; d)$, with $\frac{d \log(n)}{2n}$ for the Zernike model with $d$ parameters. The optimal Zernike model we use for comparison had four basis functions. We used the nonparametric procedure, equation 5.7, with $x_i \in \mathbb{R}^2$ corresponding to the position within a circle of radius 2 centered at (0, 0). Figure 6 plots the penalized normalized negative log likelihood, equation 5.8, as a function of the Lipschitz candidate constants $K_j$. Note that the optimal nonparametric model is for $K = 5$. Figure 7 (top) shows the contour plots of each CIF estimate and their corresponding KS plots. Note that the nonparametric method appears to provide a model of spiking more visually consistent with the spiking histogram than that of the Zernike approach. This is also manifested quantitatively in a better KS plot for the nonparametric method in Figure 6.

9 Discussion

Nonparametric approaches applied to point processes, as evidenced by the retinal data set and hippocampal place cell data set, do not always fit the data using standard approaches (such as point-process GLM scenarios). In this letter, we are proposing the use of a slightly more general approach, with many fewer assumptions, that still guarantees statistical consistency under appropriate assumptions. We hope that such procedures are used in complicated settings where off-the-shelf parametric approaches are not necessarily successful. In such settings, the structure of the dependence of the conditional intensity on other covariates of interest, as estimated by our algorithm, can provide alternative structured ways to parameterize...
Figure 6: (Top) Penalized normalized negative log likelihood, given by equation 5.8, across candidate $K_j$ values for the hippocampal place cell data set. (Bottom) KS plots of optimal nonparametric model and optimal Zernike model on a cross-validated set of 20 s of position and spiking.

the data. Rather than claiming that we want to replace all other parametric or nonparametric approaches, we are simply providing an alternative approach that balances mathematical rigor under fewer assumptions with algorithmic efficiency.
Figure 7: (Top) CIF contour for the optimal Zernike model learned from four base functions. (Bottom) CIF contour for nonparametric model.
Appendix

A.1 Executing the Efficient Dual Problem D2 in Detail. We now discuss some properties of problem D2 that allow us only to compute a sequence of network flow problems to attain a solution. Consider problem D2:

$$\begin{align*}
\text{max } & q_K(W) = \sum_{i=1}^{n} (1 + \log \Delta) dy_i - W_i \log W_i - R_K(W) \\
\text{s.t. } & W \in S,
\end{align*}$$

(A.1a)

where $S$ is the simplex:

$$S = \left\{ W : W \geq 0, \sum_{i=1}^{n} W_i = \sum_{j=1}^{n} dy_j \right\}$$

and $R_K(W)$ is the optimal solution to the network flow problem (in dual form):

$$\begin{align*}
\text{max } & p'(W - dy) \\
\text{s.t. } & p \in P_K \tag{A.2a} \\
\mathcal{P}_K & \triangleq \left\{ p : p_i - p_j \leq K \mid X_i - X_j \mid, \forall i, j \right\}. \tag{A.2c}
\end{align*}$$

We now provide the following lemma, which describes an equivalent dual problem D2 that can be implemented with lower complexity:

**Lemma 1.** For any $K \geq 0$ and any $W \in S$, by virtue of equation A.2c, $R_K(W) = K R_1(W)$ and if $p^*_1$ is an optimal dual solution associated with $R_1(W)$, then $K p^*_1$ is an optimal dual solution associated with $R_K(W)$.

Note that the dual problem, specified as equation A.1, is convex but is nondifferentiable because of the piecewise linearity of $R_K(W)$ in $W$. We now consider using a subgradient method (Bertsekas, 1999) to solve this.

**Definition 1 (Subgradient).** Given a convex function $f$, we say that $\underline{d}$ is a subgradient of $f$ at $\underline{x}$ if

$$f(\underline{z}) \geq f(\underline{x}) + (\underline{z} - \underline{x})^T \underline{d}$$

for all $\underline{z}$. If $f$ is concave, we say that $\overline{d}$ is a subgradient of $f$ at $\underline{x}$ if $-\overline{d}$ is a subgradient of the convex function $-f$ at $\underline{x}$. 
Note that any dual optimal solution $p_k(W)$ to $R_k(W)$ is a subgradient to $R_k(W)$, and thus $g = -[1 + \log W - p_k(W)]$ is a subgradient to $q_k(W)$. The subgradient method generates a sequence of dual feasible points according to the iteration

$$W_{r+1} = [W_r + s_r g_r]_+,$$

where $g_r$ is a subgradient to $q_k(W_r)$ and $[.]_+$ denotes projection onto the simplex $S$. Selection of the positive scalar step sizes $\{s_r\}$ can be done using a number of procedures, outlined in Bertsekas (1999). The selection procedure for the step size in this letter is given by method a in Bertsekas (1999, p. 613). Finally, solving D2 can also be hastened by using a standard cutting plane algorithm (Bertsekas, 1999) to adaptively add constraints in the network flow problem $R(W)$, as required. A certificate of feasibility routine can be implemented in C++ to interface with Matlab to identify a small set of violated constraints. Adaptively adding more constraints, and stopping once a certificate of feasibility is generated, usually requires significantly fewer constraints and is guaranteed to return the optimal solution. $R(W)$ can be executed using a standard network flow solver. Our implementation used MCFZIB (Frangioni & Manca, 2006, http://www.di.unipi.it/optimize/software/MCF.html).

### A.2 Projections onto Convex Sets.

**Lemma 2.** For any $W_0 \in \mathbb{R}^n$, the projection of $W_0$ onto $S$, given by

$$[W_0]_+ = \arg \min_{W \in S} \|W - W_0\|^2,$$

is given by

$$W^*_i = (W_{0,i} - \nu)^+,$$

where $\nu$ is chosen to satisfy (Bertsekas, 1999) $\sum_i W^*_i = \sum_j dy_j$.

**Proof.** The KKT optimality conditions for convex optimization over a simplex (see Bertsekas, 1999, example 2.1.2) are that

$$W^*_i > 0 \Rightarrow \frac{\partial f(W^*)}{\partial W_i} \leq \frac{f(W^*)}{\partial W_j} \quad \forall j,$$

which equivalently means that all coordinates that are positive at the optimum must have minimal (and equal) partial cost derivatives. In our
case,
\[ f(W) = \sum_{i=1}^{n} (W_i - W_{0,i})^2 \Rightarrow \frac{\partial f(W)}{\partial W_i} = 2(W_i - W_{0,i}). \]

By defining \( S = \sum_{j} d y_j \), it follows that \( W_i^* \) must satisfy
\[ W_i^* = (W_{0,i} - \nu)^+ \]  
(A.3a)
\[ \frac{\partial f(W^*)}{\partial W_i} = 2(W_i^* - W_{i,0}) = -2\nu, \ \forall \ i \ s.t. \ W_i^* > 0 \]  
(A.3b)
\[ \nu \geq \max_{i: W_i^* = 0} W_{0,i} \]  
(A.3c)
\[ \nu \text{ satisfies } \sum_{i=1}^{n} W_i^* = S, \]  
(A.3d)

where the maximum of the empty set is \(-\infty\). We can set up an algorithm to do this as follows:

1. If \( \sum_{i} W_{0,i}^+ = S \), then set \( \nu = 0 \) and go to 4.
2. Define \( W_s \) to be \( W_0 \) sorted in descending order so that \( W_{s,1} \geq W_{s,2} \geq \cdots \geq W_{s,n} \geq W_{s,n+1} \triangleq -\infty \). Set \( k = n \).
3. Define \( \nu_k \) as
\[ \nu_k = \frac{1}{k} \left\{ \left( \sum_{i=1}^{k} W_{s,i} \right) - S \right\}. \]  
(A.4)

If \( W_{s,k+1} \leq \nu_k \leq W_{s,k} \), \( (A.5) \)
then \( \nu = \nu_k \); else set \( k = k - 1 \) and go to 3.
4. Set \( W_i^* = (W_{0,i} - \nu)^+ \).

**Proof of optimality of the algorithm.** Equation A.3a is satisfied because of step 4 in the algorithm. Equation A.3b is satisfied because of the second inequality in equation A.5, and the definition of \( W_i^* \) in step 4. Equation A.3c is satisfied because of the first inequality in equation A.5 and the definition of \( W_i^* \) in step 4. Equation A.3d is satisfied because of equations A.4 and A.5, and the definition of \( W_i^* \) in step 4.

**Acknowledgments**

We thank Angelia Nedich and Stephen Boyd for thoughtful recommendations on implementation of the subgradient method and cutting plane algorithms, Maxim Raginsky and Munther Dahleh for useful comments.
on nonparametric statistical learning, and Antonio Frangioni for assistance with the MCFZIB library for implementing the network flow solution solver in C++. Support of this research is through the AFOSR Complex Networks Program, award FA9550-08-1-0079 to T.P.C, and Burroughs Wellcome Fund and L’Oreal USA FWIS to S.V.S.

References


