FAST ℓ_1 MINIMIZATION FOR GENOMEWIDE ANALYSIS OF mRNA LENGTHS

Iddo Drori and Victoria C. Stodden

Stanford University
Department of Statistics

Evan H. Hurowitz

University of California, San Francisco
Biopharmaceutical Sciences

ABSTRACT

Application of the Virtual Northern method to human mRNA allows us to systematically measure transcript length on a genome-wide scale [1]. Characterization of RNA transcripts by length provides a measurement which complements cDNA sequencing. We have robustly extracted the lengths of the transcripts expressed by each gene for comparison with the Unigene, Refseq, and H-Invitational databases [2, 3].

Obtaining an accurate probability for each peak requires performing multiple bootstrap simulations, each involving a deconvolution operation which is equivalent to finding the sparsest non-negative solution of an underdetermined system of equations. This process is computationally intensive for a large number of simulations and genes. In this contribution we present an efficient approximation method which is faster than general purpose solvers by two orders of magnitude, and in practice reduces our processing time from a week to hours.

1. INTRODUCTION

In previous work we presented a baseline deconvolution model for robustly extracting lengths of RNA transcripts [1]. Our analysis includes recovering a small set of underlying peaks from noisy microarray data. In this setting the peaks form a sparse non-negative representation. We briefly describe the convolutional model used for recovering peaks by ℓ_1 norm minimization - for a more comprehensive and detailed description of our analysis the reader is referred to [1, 4].

1.1. ℓ_1 norm minimization

In an abstract fashion, our problem can be formulated as finding the sparsest non-negative solution to an underdetermined system of equations, which is the solution with the smallest number of nonzero elements. We would therefore like to solve the optimization problem:

$$(P_0) \qquad \min \|x\|_0 \text{ subject to } y = Ax, x \geq 0.$$

However, it is well-known that this non-convex combinatorial optimization problem is NP-hard and therefore we consider the convex optimization problem:

$$(P_1) \qquad \min \|x\|_1 \text{ subject to } y = Ax, x \geq 0,$$

which can be cast as a standard linear program, and solved using interior point methods [5]. When the solution is sufficiently sparse there exists equivalence between (P_0) and (P_1) [6, 7]. In most practical applications, we observe noisy data [8] and would like to solve the problem:

$$(P_{1,\varepsilon}) \qquad \min \|x\|_1 \text{ subject to } \|y - Ax\| \leq \varepsilon$$

1.2. Baseline deconvolution

In our setting A denotes a convolution operator, y the observation, and x the underlying set of non-negative peaks. Solving $(P_{1,\varepsilon})$ in this case amounts to l_1 norm deconvolution in the presence of noise. While such a model accounts for spurious peaks within noisy data, it is not sufficiently suitable for our purposes. Roughly speaking, we would also like to accommodate signals with low frequency content which the deconvolution model represents as multiple peaks. We therefore incorporate a rolling baseline into the model such that the observation is represented as a smooth baseline u with peaks on top v and solve the optimization problem:

$$\min \|x\|_1 + \mu\|\beta\|_1 + \lambda\|r\|_2^2 \text{ subject to } u + v + r = y,$$

such that $Ax = v$ and $\Delta^2 u = \beta$, where Δ^2 denotes the second difference operator. This means that we solve for the underlying peaks which when convolved with the kernel and added to a smooth baseline result in the observed data.

1.3. Parametric bootstrap

In order to ascertain a confidence for each peak, we assign a probability to each range of mRNA lengths for every gene. This value describes the likelihood that what we have uncovered is a true peak. Each value of the signal is sampled with Poisson noise to create multiple parametric bootstrap replicates. Given the baseline deconvolution result according to our model, we simulate the experiment by convolution and a Poisson model. More specifically, for each gene we consider each peak independently and perform multiple simulations of the type $\tilde{y}_i = A * \tilde{x} + z_i$, where z_i is Poisson and $\tilde{x}$ each peak in the baseline deconvolution result. Next, we reconstruct each simulation by deconvolution to obtain multiple bootstrap measurements $\hat{x}_i$. Generating multiple reconstructed replicates $\hat{x}_1, \ldots, \hat{x}_s$ requires solving many instances of the problem $(P_{1,\varepsilon})$ which is computationally intensive, in particular for a large number of simulations ($s = 100$) and genes (22,385). We therefore apply a rapid method for finding a sparse solution of underdetermined linear systems of equations as described next.

2. ITERATIVE SOFT THRESHOLDING

A fast solution to the optimization problem $(P_{1,\varepsilon})$ is obtained by a simple *Iterative Soft Thresholding* algorithm. Let $\delta_t(x)$ denote the soft thresholding operator:

$$(\delta_t(x))_i = \textit{sign}(x_i)(\|x_i\| - t)_+.$$

Consider the iteration

$$x^{\ell+1} = x^\ell + \rho\delta_{t_\ell}(A^T(y - Ax^\ell)), \qquad (1)$$

where x^l is the l-th approximate solution, δ_t is soft thresholding at amplitude t and the threshold t_l decreases with increasing iteration count. Here $0 < \rho \leq 1$, we start this iteration from $x^0 = 0$, and t_l decreases from iteration to iteration by a factor $\mu = 1 - \rho$. Each iteration requires applications of A and A^T. Figure 1 provides a detailed description of the algorithm in pseudocode.

A variation which accelerates the basic iteration in Eq. 1 is a solution in which the matrix A is partitioned into blocks $A = [B_1, B_2 \ldots B_J]$ by taking random disjoint columns. Then the least-squares projection of B_j is applied in computing the correlations:

$$x^{l+1,j} = x^{l,j} + \rho\delta_{t_l}((B_j^T B_j)^{-1} B_j^T (y - Ax^l)).$$

Iterative soft thresholding naturally extends to a parallel solution of multiple problems of the same form by replacing the observation and solution vectors with matrices and solving:

$$X^{l+1} = X^l + \rho\delta_{t_l}(A^T(Y - AX^l)).$$

Using iterative soft thresholding, computation time of the bootstrap is reduced from being quadratic in the input to linear with a small constant, and in practice by two orders of magnitude. For example, for our purposes the bootstrap computation for a single gene using a general purpose solver is performed in 18 seconds on a 1.5GHz machine whereas using iterative soft thresholding the computation is performed in 30 milliseconds. Processing time of the entire set of genes is reduced from a week to hours. For more details, we provide `SparseLab` [9] - a collection of Matlab functions which, among others, includes this application and solvers. In the spirit of *reproducible research*, we are making the software available to the research community at:
http://www-stat.stanford.edu/˜sparselab/.

```
Input: $n \times p$ matrix $A$, $n < p$, and observation vector $y$.

Output: solution of $(P_{1,\varepsilon})$.

Algorithm:

Init:
iteration $l = 0$.
solution $x^l = 0$.
residual $r^l = y$.
correlation $c^l = A'r^l$.
threshold $t_l = \max(\|c^l\|)$.

Step:
while $\|r^l\|_2 \geq \varepsilon$
    iteration $l = l + 1$.
    solution $x^l = x^{l-1} + \rho\delta_{t_{l-1}}(c^{l-1})$.
    residual $r^l = y - Ax^l$.
    correlation $c^l = A'r^l$.
    threshold $t_l = \mu t_{l-1}$.
end while
```

Fig. 1. Iterative Soft Thresholding pseudocode.

3. REFERENCES

[1] Evan H. Hurowitz, Iddo Drori, Victoria C. Stodden, David L. Donoho, and Patrick O. Brown, “Virtual northern analysis of the human genome,” *In progress*.

[2] David L. Wheeler, Deanna M. Church, and Ron Edgar et al., “Database resources of the national center for biotechnology information: update,” *Nucleic Acids Res.*, vol. 32, pp. 35–40, 2004.

[3] Tadashi Imanishi, Takeshi Itoh, and Yutaka Suzuki et al., “Integrative annotation of 21,037 human genes validated by full-length cDNA clones,” *PLOS Biology*, vol. 2, pp. 1–20, 2004.

[4] Evan H. Hurowitz and Patrick O. Brown, “Genome-wide analysis of mRNA lengths in saccharomyces cerevisiae,” *Genome Biology*, vol. 5, no. 1, pp. 1–14, 2003.

[5] Michael A. Saunders and Byunggyoo Kim, “PDCO: Primal-dual interior method for convex objectives,” http://www.stanford.edu/group/SOL/software/pdco.html.

[6] David L. Donoho, “For most underdetermined systems of linear equations, the minimal ℓ^1-norm near-solution approximates the sparsest near-solution,” *Comm. Pure and Appl. Math.*, 2004.

[7] David L. Donoho and Jared Tanner, “Sparse nonnegative solutions of underdetermined linear equations by linear programming,” *Proc. Natl. Acad. Sci. USA*, vol. 102, pp. 9446–9451, 2005.

[8] David L. Donoho, Miki Elad, and Vladimir N. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Trans. on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[9] David L. Donoho, Iddo Drori, Victoria C. Stodden, and Yaakov Tsaig, “Sparselab web site,” http://www-stat.stanford.edu/˜sparselab/.