# Variational Inference for Sparse Linear Models

July 17, 2015

## 1 Introduction

In this note we provide a (simple) probabilistic model for sparse linear models, as applied to Compressive Sensing. It allows the modelling of sparsity patterns from heavy-tailed distributions (for example, Laplace or Student-t).

In particular, we present a method to fully reconstruct the full posterior distribution on the latent variables; as opposed to finding the most probable (MAP) solution.

However, computing the exact posterior under heavy-tailed priors is not tractable. We instead present a variational technique where we approximate the true posterior by a parameterised Gaussian, which allows closed form computations.

## 2 Variational Bayes for Sparse Linear Models

Consider an undersampled linear system:

$$y = Sx + n \tag{2.0.1}$$

where $y, n \in \mathbb{R}^m$, $S \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $n$ is white Gaussian noise with variance 1.

We can model simply as follows:

$$p\left(y|x\right) = \mathcal{N}\left(y; Sx, 1\right) \approx \exp\left(-\frac{1}{2}\left(y - Sx\right)^T\left(y - Sx\right)\right) \tag{2.0.2}$$

and we place a Laplace prior over the signal $x$:

$$p\left(x\right) \approx \exp\left(-\lambda \sum_{i=1}^{n} |x_i|\right) \tag{2.0.3}$$

we choose $\lambda = \sqrt{2 * logn}$ to simply the discussion.

The entire posterior distribution is given by:

$$
\begin{aligned}
p\left(x|y\right) &= p\left(y|x\right)p\left(x\right) \\
&= \exp\left(-\left(\frac{1}{2}\left(y-Sx\right)^T\left(y-Sx\right)+\lambda\sum_{i=1}^{n}|x_i|\right)\right) \qquad (2.0.4)
\end{aligned}
$$

This leads to the familiar MAP estimate:

$$
\hat{x}_{MAP} = \arg\min_x \frac{1}{2}\left(y-Sx\right)^T\left(y-Sx\right)+\lambda\sum_{i=1}^{n}|x_i| \qquad (2.0.5)
$$

However, computing the entire posterior 2.0.4 is difficult. In this case we can approximate it with an appropriately parameterised Gaussian distribution:

$$
\begin{aligned}
q\left(x|y\right) &\approx P(y|x)\exp\left(\beta^T x-\frac{1}{2}x^T\Gamma^{-1}x\right) \qquad (2.0.6) \\
&= \mathcal{N}\left(x;\hat{x}_Q,A^{-1}\right) \qquad (2.0.7)
\end{aligned}
$$

where

$$
\hat{x}_Q = A^{-1}b \qquad (2.0.8)
$$

$$
A = S^TS+\Gamma^{-1} \qquad (2.0.9)
$$

$$
b = S^Ty+\beta \qquad (2.0.10)
$$

and

$$
\Gamma = \operatorname{diag}\left(\gamma\right) \qquad (2.0.11)
$$

and we minimise the KL divergence:

$$
KL\left(p\left(x|y\right)||q\left(x|y\right)\right) \qquad (2.0.12)
$$

This can be expressed as:

$$
\begin{aligned}
\mathcal{L}_{KL}\left(\beta,\Gamma\right) &= q\left(x|y\right)\log p\left(x|y\right)-q\left(x|y\right)\log q\left(x|y\right) \qquad (2.0.13) \\
&= \frac{1}{2}||y-Sx||_2^2+\lambda||x||_1-\frac{1}{2}\left(x-A^{-1}b\right)^TA^{-1}\left(x-A^{-1}b\right) \qquad (2.0.14)
\end{aligned}
$$

# 3 Bregman ADMM

The alternating direction method of multipliers [?], (ADMM), algorithm solves problems of the form

$$\begin{aligned} \arg\min_x f(x) + g(z) \\ \text{s.t } Ux + Vz = c \end{aligned} \tag{3.0.15}$$

where $f$ and $g$ are assumed to be convex function with range in $\mathbb{R}$, $U \in \mathbb{R}^{p\times n}$ and $V \in \mathbb{R}^{p\times m}$ are matrices (not assumed to have full rank), and $c \in \mathbb{R}^p$.

ADMM consists of iteratively minimising the augmented Lagrangian

$$L_p(x, z, \eta) = f(x) + g(z) + \eta^T (Ux + Vz - c) + \frac{\rho}{2}\|Ux + Vz - c\|_2^2$$

($\eta$ is a Lagrange multiplier), and $\rho$ is a parameter we can choose to make $g(z)$ smooth [?], with the following iterations:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{3.0.16}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{3.0.17}$$

$$\eta^{k+1} := \eta^k + \rho\left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{3.0.18}$$

The alternating minimisation works because of the decomposability of the objective function: the $x$ minimisation step is independent of the $z$ minimisation step and vice versa.

Bregman ADMM differes from the standard ADMM by replacing the quadratic penalty with a KL divergence:

$$L_{KL}(x, z, \eta) = f(x) + g(z) + \eta^T (Ux + Vz - c) + KL(Ux||Vz)$$

so that the updates now become

$$x^{k+1} := \arg\min_x f(x) + \eta^T (Ux + Vz - c) + \rho KL\left(c - Ux||Vz^k\right) \tag{3.0.19}$$

$$z^{k+1} := \arg\min_z g(z) + \eta^T (Ux + Vz - c) + \rho KL\left(Vz||c - Ux^{k+1}\right) \tag{3.0.20}$$

$$\eta^{k+1} := \eta^k + \rho\left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{3.0.21}$$