vbsr: Variational Bayes Spike regression

Ben Logsdon

September 27, 2011

1 Example 1

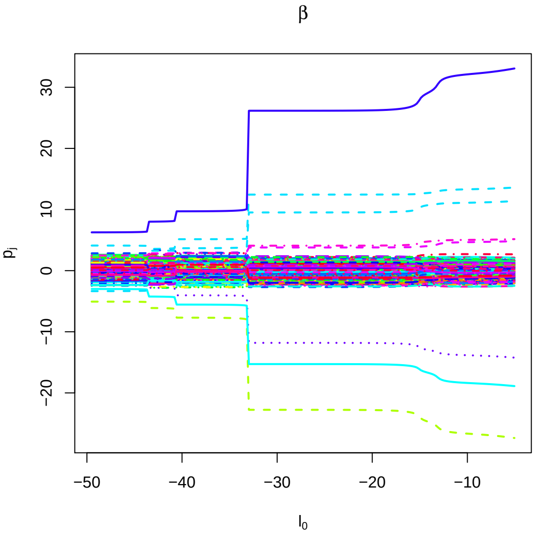
We first consider the case of uncorrelated features, and a linear response, with a sparse true model:

```
> library(vbsr)
> library(MASS)
> set.seed(1)
> n <- 100
> m <- 500
> ntrue <- 10
> e <- rnorm(n)
> X <- matrix(rnorm(n * m), n, m)
> tbeta <- sample(1:m, ntrue)
> beta <- rep(0, m)
> beta[tbeta] <- rnorm(ntrue, 0, 2)
> y <- X %*% beta + e
> res <- vbsr_net(y, X, regress = "LINEAR")
```

```
[1] 100 501
Initializing marginal analysis...
Sum of squares pre-compute...
Initialized marginal model...
Model run...
Initializing model...
Scaling...
Initialized model...
Model run...
Results computed..
```

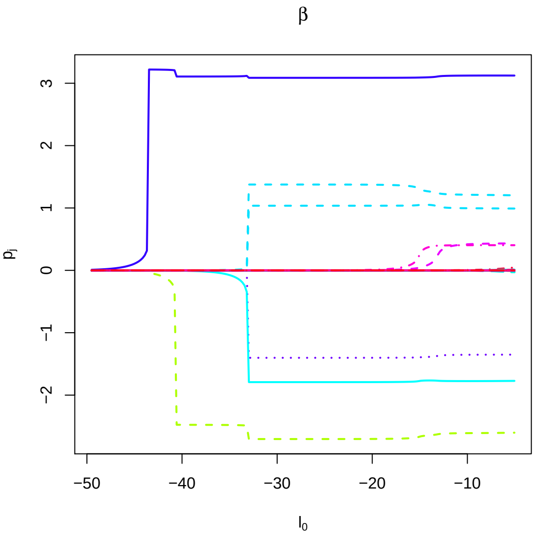
Next we look at the following solutions along the path of the penalty parameter for this, starting with the normally distributed test statistic:

```
> plot_vbsr_beta_chi(res)
```



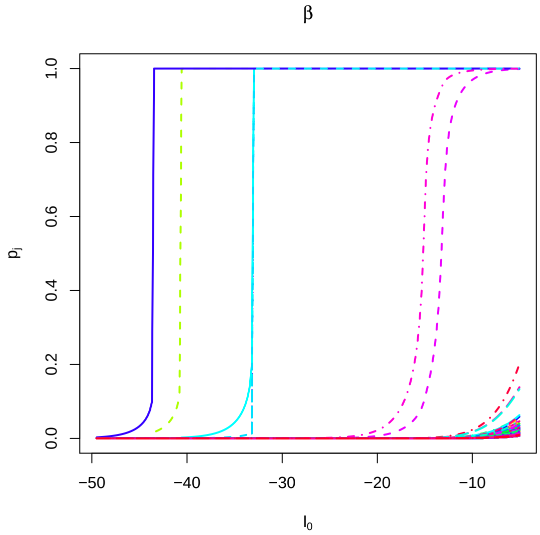
The expectation of the regression coefficients:

```
> plot_vbsr_e_beta(res)
```



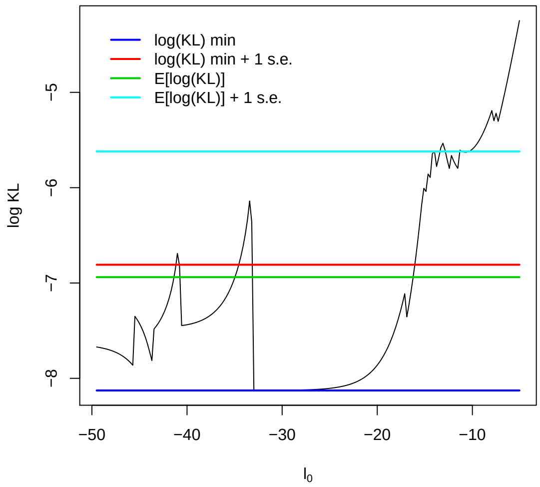
as well as the posterior probability of being non-zero:

```
> plot_vbsr_beta_p(res)
```



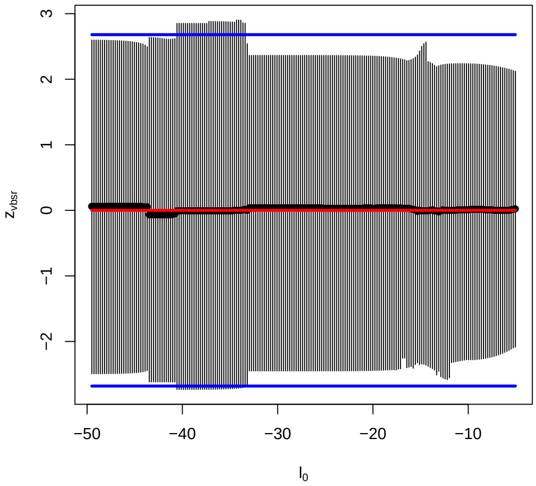
the Kullback-Leibler divergence computed along the path:

```
> plot_vbsr_kl(res)
```



and finally another diagnostic of the goodness of fit of the null features to a normal distribution along the path:

```
> plot_vbsr_boxplot(res)
```



Let's look at what the solution at the KL minimum plus 2 standard errors looks like:

```
> w_sol = which.min(abs(res$kl - res$kl_min - 2 * res$kl_se))
> res$l0_path[w_sol]

[1] -9.296557

> print(sort(tbeta))

 [1]  12  38  77  92 126 183 286 396 427 494

> which(res$beta_p[-1, w_sol] > 0.99)

[1]  12  38  77 126 286 427 494

> which(res$beta_chi[-1, w_sol]^2 > qchisq(1 - 0.05/1000, 1))

[1]  12  38  77  92 126 286 427 494
```

2 Example 2

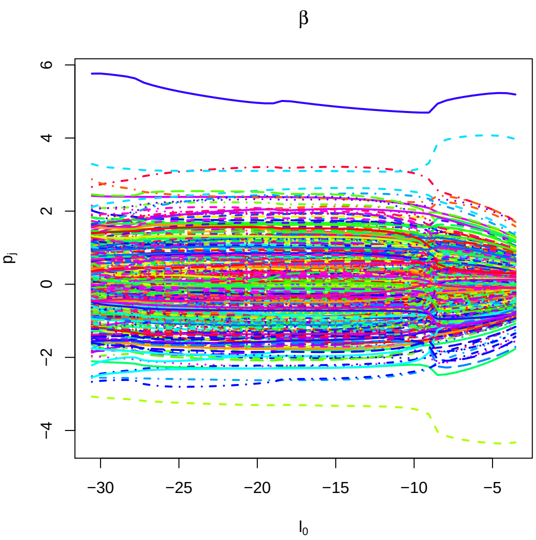
We first consider the case of uncorrelated features, and a logistic response, with a sparse true model:

```
> pred_val <- X %*% beta
> y <- rep(0, n)
> for (i in 1:n) {
+     y[i] <- rbinom(1, 1, 1/(1 + exp(-pred_val[i])))
+ }
> res <- vbsr_net(y, X, regress = "LOGISTIC", n_orderings = 1,
+     path_length = 50)
```

```
[1] 100 501
Initializing marginal analysis...
Sum of squares pre-compute...
Initialized marginal model...
Model run...
Initializing model...
Scaling...
Initialized model...
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Maximum iterations exceeded!
Model run...
Results computed..
```

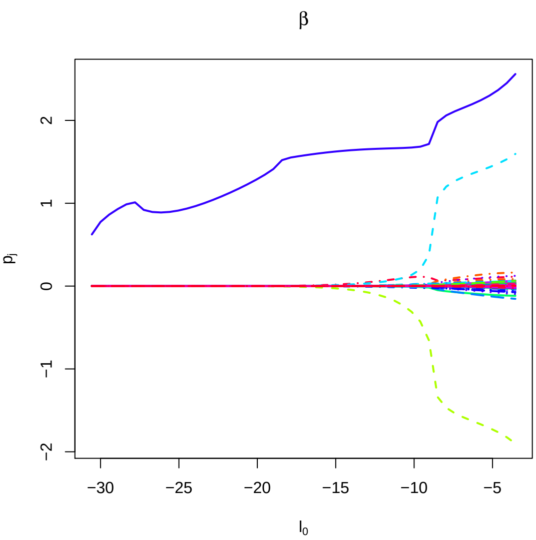
Next we look at the following solutions along the path of the penalty parameter for this, starting with the normally distributed test statistic:

```
> plot_vbsr_beta_chi(res)
```



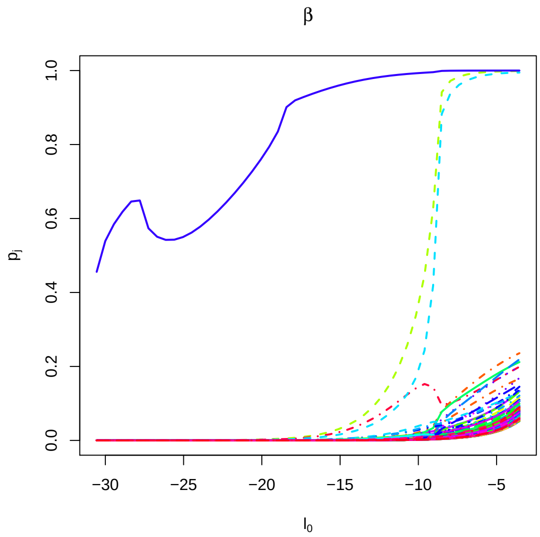
The expectation of the regression coefficients:

```
> plot_vbsr_e_beta(res)
```



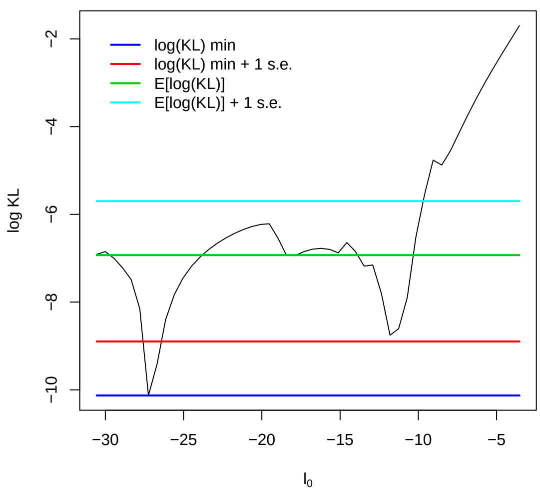
as well as the posterior probability of being non-zero:

```
> plot_vbsr_beta_p(res)
```



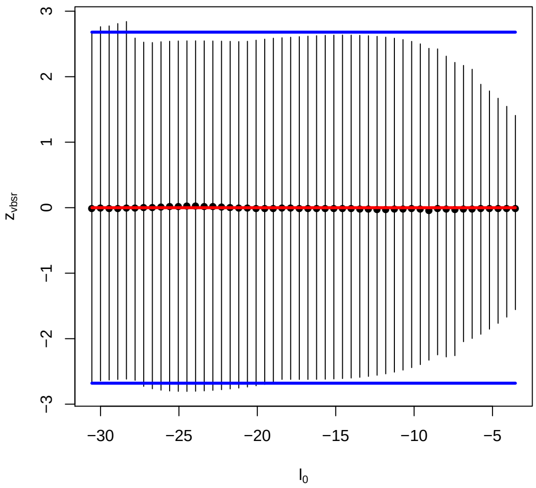
the Kullback-Leibler divergence computed along the path:

```
> plot_vbsr_kl(res)
```



and finally another diagnostic of the goodness of fit of the null features to a normal distribution along the path:

```
> plot_vbsr_boxplot(res)
```



Let's look at what the solution at the KL minimum plus 2 standard errors looks like:

```
> w_sol = which.min(abs(res$kl - res$kl_min - 2 * res$kl_se))
> res$l0_path[w_sol]

[1] -12.36114

> print(sort(tbeta))

 [1]  12  38  77  92 126 183 286 396 427 494

> which(res$beta_p[-1, w_sol] > 0.99)

integer(0)

> which(res$beta_chi[-1, w_sol]^2 > qchisq(1 - 0.05/1000, 1))

[1] 286
```