Internship - Bibliography

Pierre-Antoine

March 22, 2021

1 Readings

1.1 Implicit differentiation for fast hyperparameter selection in non-smooth convex learning, Bertrand and al.

Goal: select the best set of hyperparameters to optimize a loss function.
Initial solution: Grid search, Random search, Bayesian optimization.
Problem: they scale poorly with the number of hyperparameters to tune.
Solution: use first-order optimization methods to optimize a problem.

The bilevel optimization problem is the problem consisting in optimizing the loss function w.r.t. the hyperparameters (outer loss) with respect to a constraint: minimizing the criterion w.r.t. the parameters of an estimator (inner loss).

One *strong* assumption: the regularization path is well-defined and almost everywhere differentiable.
Main challenge in first-order optimization for the outer loss: evaluating the hypergradient (i.e. gradient of the loss w.r.t. the hyperparameters).

3 algorithms to compute hypergradients:

- Implicit differentiation

- Forward auto differentiation

- Backward auto differentiation (backprop)

Contributions:

- There exist methods to efficiently compute hypergradients for non-smooth functions.

- Leveraging the sparsity of the Jacobian matrix, we propose an efficient implicit differentiation algorithm to compute the hypergradients.

- Implicit differentiation significantly outperforms forward and backward auto differentiation.

Side notes: In practice, (proximal) coordinate descent is much faster for solving LASSO-like problems than accelerated (proximal) gradient descent à la Nesterov. For more details, see Bertrand and Massias, Anderson acceleration of coordinate descent.

1.2 Enhancing Sparsity with Reweighted l1 minimization, Candès and al.

Goal: reconstruct sparse signals (optimization problem).
Initial solution: use a l_1 norm.
Problem: larger coefficients are penalized more heavily in the l_1-norm than smaller coefficients, unlike the more democractic l_0 norm.
Better solution: solve a sequence of weighted l_1-minimization problems where the weights used for the next iteration are computed from the value of the current solution.

For large-dimensional problems ($p >> n$, p is the number of predictors and n the number of samples), there exists an infinite set of solutions. Imposing a structure constraint on the solution form restrains the solution set and tends to yield the right solution.

Constraining the reconstruction problem (compressive sensing) using a l_0-norm yields a NP-hard combinatorial problem. Hence, we rely on the l_1-norm that yields a convex yet sparse surrogate optimization problem.

Contributions:

- An iterative procedure that solve convex subproblems to solve a concave global problem that emulates even better the initial l_0 norm problem.

1st question: how to select the weights to build a weighted convex problem?
As a rule of thumb, the weights should relate inversely to the true signal magnitudes.
Then, how to choose a valid set of weights without knowing a priori the true signal magnitudes?
There must exist weighting matrices based solely on an approximation x to x_0.

The weights actually come from the log-sum penalty function. The log-sum penalty function has the potential to be much more sparsity-encouraging than the l_1-norm.

Question: What is the connection between the chosen weights and the log-sum penalty?
DO THE DEMO

The concave penalty function $f_{\log,\epsilon}$ has slope at the origin that grows roughly has $\frac{1}{\epsilon}$ when $\epsilon \to 0$. Like the l_0-norm, this allows a relatively large penalty to be placed on small nonzero coefficients and more strongly encourages them to be set to zero.
This is a key difference with l_1 norm. As l_1 norm tends to put smaller penalty on small coefficients than large coefficients, small coefficients are not necessarily forced to be zero, thus there remains a residual error when applying l_1-norm.

Question: How to choose ϵ?
As $\epsilon \to 0$, $f_{\log,\epsilon}(t) \to f_0(t)$, one could be tempted to set ϵ arbitrarily small. However, as $\epsilon \to 0$, it becomes more likely for the iterative reweighted l_1 algorithm to be stuck in an undesirable local minimum. In practice, ϵ must be set slightly smaller than the expected nonzero magnitudes of x.