ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Optimization by Variational Bounding

Joe Staines and David Barber

University College London - Computer Science
Gower Street, London, WC1E 6BT - United Kingdom

Abstract. We discuss a general technique that forms a differentiable bound on non-differentiable objective functions by bounding the function optimum by its expectation with respect to a parametric variational distribution. We describe sufficient conditions for the bound to be convex with respect to the variational parameters. As example applications we consider variants of sparse linear regression and SVM training.

1 Variational Optimization

We consider the general problem of function minimization, $\min_x f(x)$ for vector x. When f is differentiable and x continuous, optimization methods that use gradient information are typically preferred over non-gradient based approaches since they may take advantage of a locally optimal direction in which to search. However, in the case that f is not differentiable or x is discrete, gradient based approaches are not directly applicable. In that case, alternatives such as relaxation, coordinate-wise optimization and stochastic approaches are popular. Our interest is to discuss another general class of methods that yield differentiable surrogate objectives for discrete x or non-differentiable f. The Variational Optimization (VO) approach is based on the bound

$$f^* = \min_{x\in\mathcal{C}} f(x) \le \left\langle f(x)\right\rangle_{p(x|\theta)} \equiv E(\theta)$$

where $\langle\cdot\rangle_p$ denotes the expectation with respect to the probability density function p defined over the solution space $\mathcal{C}$. The parameters θ of the distribution $p(x|\theta)$ can then be adjusted to minimize the upper bound $E(\theta)$. This bound can be trivially made tight provided the distribution $p(x|\theta)$ is flexible enough to allow all its mass to be placed in the optimal state $x^* = \arg\min_{x\in\mathcal{C}} f(x)$. The variational bound is equivalent to an objective smoothed by convolution with the variational distribution, with the degree of smoothing increasing as the dispersion of the variational distribution increases. The gradient of $E(\theta)$ is given by

$$\frac{\partial E}{\partial\theta} = \frac{\partial}{\partial\theta}\int_{\mathcal{C}} f(x)p(x|\theta)dx.$$

The existence of this derivative depends on interchanging differentiation and integration. We can bring the differential under the integral sign provided:

(i) $f(x)p(x|\theta)$ is Lebesgue integrable and differentiable with respect to θ

(ii) there exists an integrable function $F : \mathcal{C} \to \mathbb{R}$ such that:

$$\left|\frac{\partial}{\partial\theta} f(x)p(x|\theta)\right| < F(x)$$

for all θ. These weak conditions mean that a large class of problems, in which f is non-differentiable or x discrete, have differentiable bounds. For example, consider the non-differentiable objective $f(x) = x$ for $x \geq 0$ and $f(x) = 0$ for $x < 0$. For x normally distributed with mean θ and unit variance, $p(x|\theta) = \mathcal{N}(x|\theta, 1)$, E is smooth, with $\frac{\partial E}{\partial \theta} = \mathcal{N}(\theta|0, 1)$.

We now describe sufficient conditions for $E(\theta)$ to be convex in θ. We first introduce the general concept of an expectation affine distribution.

Definition *(Expectation affine)* A distribution $p(x|\theta)$ is expectation affine if, for linear functions α, β, distribution $q(z)$ and function f,

$$\langle f(x) \rangle_{p(x|\theta)} = \Big\langle f\big(\alpha(\theta)z + \beta(\theta)\big) \Big\rangle_{q(z)}.$$

Theorem 1 *Let $f(x)$ be a convex function and $p(x|\theta)$ an expectation affine distribution. Then $E(\theta) \equiv \langle f(x) \rangle_{p(x|\theta)}$ is convex in θ.*

Proof Defining $\tilde{\lambda} \equiv 1 - \lambda \in [0, 1]$ and using the fact that p is expectation affine, for any two values of θ

$$E(\lambda\theta_1 + \tilde{\lambda}\theta_2) = \left\langle f\Big(\lambda\big(\alpha\left(\theta_1\right) z + \beta\theta_1\big) + \tilde{\lambda}\big(\alpha\left(\theta_2\right) z + \beta(\theta_2)\big)\Big) \right\rangle_{q(z)}.$$

Since f is convex, $f(\lambda x_1 + \tilde{\lambda} x_2) \leq \lambda f(x_1) + \tilde{\lambda} f(x_2)$ and hence

$$\begin{aligned} E(\lambda\theta_1 + \tilde{\lambda}\theta_2) &\leq \lambda \Big\langle f\big(\alpha(\theta_1)z + \beta(\theta_1)\big) \Big\rangle_{q(z)} + \tilde{\lambda} \Big\langle f\big(\alpha(\theta_2)z + \beta(\theta_2)\big) \Big\rangle_{q(z)} \\ &\leq \lambda E(\theta_1) + \tilde{\lambda} E(\theta_2). \end{aligned}$$

2 Lasso Sparse Least Squares Regression

For D-dimensional inputs x^n, outputs y^n, $n = 1, \ldots, N$, and positive regularizing constant λ, lasso sparse regression minimizes [1]

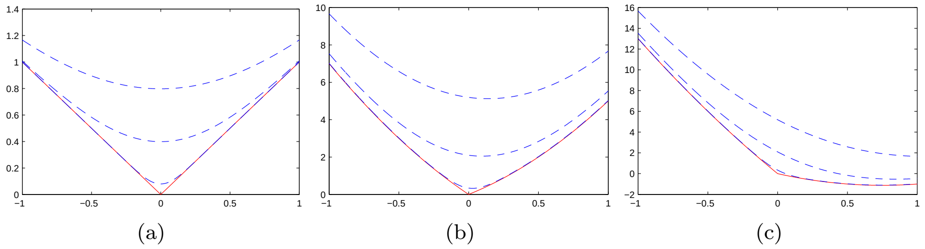
$$f(w) \equiv \sum_{n=1}^{N} \left(y^n - w^\mathsf{T} x^n\right)^2 + \lambda \sum_i |w_i| = c + w^\mathsf{T} b + w^\mathsf{T} A w + \lambda \sum_i |w_i|$$

with $c \equiv \sum_n \left(y^n\right)^2$, $b \equiv -2 \sum_n y^n x^n$, $A \equiv \sum_n x^n x^{n\mathsf{T}}$. The term $\sum_i |w_i|$ is non-differentiable at the origin and hence standard gradient based optimization algorithms cannot be directly applied. We consider a Gaussian variational distribution $p(w|\theta) = \mathcal{N}(w|\mu, \Sigma)$. The Gaussian distribution can be shown to be expectation affine, so from Theorem 1, the bound $E(\mu, C)$, with

$$f^* \leq E(\mu, C) \equiv \sum_n \left\langle \left(y^n - w^\mathsf{T} x^n\right)^2 \right\rangle_{\mathcal{N}(w|\mu,\Sigma)} + \lambda \sum_i \langle |w_i| \rangle_{\mathcal{N}(w_i|\mu_i,\Sigma_{ii})}$$

is jointly convex in (μ, C) for Cholesky decomposition $\Sigma = CC^\mathsf{T}$. This bound can be expressed in closed-form:

$$E(\mu, C) = c + \mu^\mathsf{T} b + \mu^\mathsf{T} A \mu + \operatorname{trace}(A\Sigma) + \lambda \sum_i \mu_i \left(1 - 2\phi\Big(-\frac{\mu_i}{\Sigma_{ii}}\Big)\right) + 2\frac{\Sigma_{ii}}{\sqrt{2\pi}} e^{-\frac{\mu_i^2}{2\Sigma_{ii}^2}}$$



(a) (b) (c)

Fig. 1: (a) The solid line represents the function $|w|$ and the dashed lines the bound $\langle |w| \rangle_{\mathcal{N}(w|\mu,\sigma^2)}$ plotted as a function of μ for three different values $\sigma \in \{0.1, 0.5, 1\}$ (from bottom to top in the figure). (b) The lasso objective function $f(w)$ plotted for $A = 2$, $\lambda = 4$, $b = -1$, $c = 0$. In this case the optimal w based on the lasso objective is $w = 0$ but the optimal w for VO is slightly larger than zero. (c) $A = 2$, $\lambda = 4$, $b = -7$, $c = 0$. The optimal w for both lasso and VO are non-zero and numerically very close. As we move further away from the non-differentiable point, the bound more closely matches the objective, resulting in a closer match of their optima.

where $\phi(x) = \int_{-\infty}^{x} e^{-y^2/2}/\sqrt{2\pi}dy$.

2.1 Lasso Experiments

We generated a sparse $D = 200$ dimensional parameter vector with components

$$u_i \sim \begin{cases} 0 & \text{with probability } 0.5 \\ \mathcal{N}(u_i|5,1) & \text{with probability } 0.25 \\ \mathcal{N}(u_i|-5,1) & \text{with probability } 0.25. \end{cases}$$

We then created a set of $N = 200$ training inputs x^n in D-dimensional space from the standard multivariate normal distribution $\mathcal{N}(x|0, I)$. The outputs are $y^n = u^{\mathsf{T}} x^n + \epsilon^n$ where ϵ^n is Gaussian random noise with mean zero and standard deviation $\frac{0.1}{N}\sum_{n=1}^{N} |u^{\mathsf{T}} x^n|$, chosen to obscure but not dominate the clean outputs. We set $\lambda = 30D$ to give solutions with sparsity roughly the same as the initial parameter vector u. To minimize $E(\mu, C)$, at each iteration we fixed $\Sigma = CC^{\mathsf{T}}$ and updated μ using a diagonal approximation to the Newton method (to avoid the cost of inverting the full Hessian). For gradient E'_i and Hessian elements E''_{ii} the updates were $\mu_i^{new} = \mu_i^{old} - 0.1\frac{E'_i}{E''_{ii}}$. We used an initial covariance matrix of $\Sigma = 0.1I$ and reduced it by a factor of 0.9 at each iteration. The initial estimate for μ was a vector of zeros. We terminated when the mean absolute difference in the solution elements between iterations was less than 10^{-15} of the mean absolute value of the elements of the current solution and took this terminal μ to be the estimate for $\arg\min_w f(w)$.

We tested our method against a number of standard `minFunc` lasso solvers.[1] Since the true optima are not known, we measured the success of the algorithms

[1] `www.di.ens.fr/∼mschmidt/Software/minFunc.htm`

Algorithm	Solution time (SD)	Relative Error (SD)
Variational optimization	0.1308 (0.0405)	1.923×10^{-15} (6.01×10^{-16})
Shooting	0.0676 (0.0209)	2.656×10^{-16} (3.17×10^{-16})
Iterated ridge regression	0.1645 (0.0844)	4.004×10^{-14} (1.19×10^{-13})
Smoothing (integral sigmoid)	1.2305 (0.3381)	2.638×10^{-15} (9.19×10^{-16})
Smoothing ($\sqrt{x^2+\epsilon}$)	1.1385 (0.3673)	1.471×10^{-15} (1.86×10^{-15})
Projection	0.3121 (0.0726)	5.325×10^{-16} (2.85×10^{-16})
Sub-gradients	1.3478 (0.3273)	6.038×10^{-11} (9.42×10^{-10})

Table 1: Performance and time taken (in seconds) for algorithms solving lasso problems. All algorithms were implemented in MATLAB and run on a 2.27GHz 4GB Windows machine. Results are the average of 500 experiments.

relative to the best solution f_{best} found by any of the algorithms on each problem instance. In Table 1 we give the mean and standard deviation of relative errors $(f - f_{\text{best}})/(f_{\text{best}})$. Whilst the shooting method is optimal, VO provides solutions of similar quality to two other `minFunc` smoothed methods (the first approximates the L_1 norm using an integral of two sigmoid functions and the second uses the bound $\sqrt{x^2+\epsilon'}$, for $\epsilon' \geq 0$).[2] The results indicate that VO is capable of approximating the global optimum well in moderate time. For $\Sigma = \sigma^2 I$ we show in [2] that VO finds the optimum of f to within a specified maximal error Δ_f, provided

$$\sigma \leq \frac{1}{\sqrt{\text{trace}\,(A)}} \left(\sqrt{\left(\frac{\lambda^2 D^2}{2\pi} + \Delta_f \text{trace}\,(A) \right)} - \frac{\lambda D}{\sqrt{2\pi}} \right).$$

2.2 Fused Lasso Sparse Regression

The shooting algorithm solves for each component w_i, keeping the others fixed, and cycles through components to convergence. This is very effective in the standard lasso problem since the objective only weakly couples the components of the vector w. In contrast, the fused lasso problem induces additional sparsity between adjacent elements by using the regularization

$$\lambda_1 \sum_{i=1}^{D} |w_i| + \lambda_2 \sum_{i=2}^{D} |w_i - w_{i-1}|.$$

The additional second term introduces strong dependencies between adjacent components and componentwise shooting methods struggle to converge [3]. As for the standard lasso case, we can readily obtain a bound using a Gaussian variational distribution and analytic expressions for the gradient and Hessian. For the experiments, u_1 was generated as in the previous problem and all subsequent

[2] The improvement of VO over other smoothing approaches is most likely accounted for by implementation differences and our reduction of Σ at each iteration. For further discussion of the impact of changing covariance see [2].

elements were sampled from

$$u_{i>1} \sim \begin{cases} u_{i-1} & \text{with probability } 0.5 \\ 0 & \text{with probability } 0.25 \\ \mathcal{N}(u_i|5,1) & \text{with probability } 0.125 \\ \mathcal{N}(u_i|-5,1) & \text{with probability } 0.125. \end{cases}$$

For each experiment we created $N = 5000$ training inputs x^n each of dimension $D = 500$ from the standard multivariate normal distribution. The outputs were given by $y^n = u^\mathsf{T} x^n + \epsilon^n$ where ϵ^n is random noise with mean zero and standard deviation $\frac{0.1}{N}\sum_{n=1}^N |u^\mathsf{T} x^n|$. Regularization parameters $\lambda_1 = 500$ and $\lambda_2 = 200$ were chosen to yield solutions w with sparsity similar to that of u. We used initial standard deviation 0.1, and shrunk it by a factor 0.9 at each iteration. For comparison we used the SLEP package [4] which is based on a version of Nesterov's method [5], a two step gradient method with backtracking line search; this has very competitive performance compared to other state-of-the-art solvers.

Using VO with a shrinking variance and convergence tolerance of 10^{-6}, the relative error compared to SLEP was consistently small, having mean 1.59×10^{-4} over the 1000 experiments and standard deviation 3.79×10^{-5}. The mean of the relative distances $L_2(w_s - w_v)/L_2(w_s)$ between each VO solution w_v and SLEP solution w_s was 0.0135. The mean CPU time for VO was 0.0947s and for the SLEP algorithm 0.0724s.[3] Despite its simplicity, VO therefore does not suffer from the convergence problems of the shooting method and has good performance compared to the state of the art.

3 Discussion

Due to space restrictions, we provided details for only a single application. However, the method is easily applicable to other problems. As a second example application, we briefly discuss SVM training. For a dataset with inputs x^n and class labels $y^n = \pm 1$, $n = 1, \ldots, N$, the hinge-loss form of the SVM minimizes (see *e.g.* [6])

$$f(\beta, b) = \beta^\mathsf{T} K \beta + C \sum_{n=1}^N \max\{1 - \sum_{m=1}^N K_{nm}\beta^m - b y^n, 0\}$$

for kernel K_{nm}. This objective is convex but non-differentiable. For VO we use Gaussian distributions over the parameters: β distributed with mean μ_β and covariance $\sigma^2 I$ and b independently with mean μ_b and variance σ^2, giving the upper bound $f^* \leq E$, with

$$E \equiv \mu_\beta^\mathsf{T} K \mu_\beta + \mathrm{trace}(\sigma^2 K) + C \sum_{n=1}^N \nu^n \phi\left(\nu^n/\varsigma^n\right) + \varsigma^n e^{-\frac{1}{2}\left(\frac{\nu^n}{\varsigma^n}\right)^2}/\sqrt{2\pi}$$

where $\nu^n = 1 - \sum_{m=1}^N K_{nm}\mu_\beta^m + \mu_b y^n$ and $(\varsigma^n)^2 = \sigma^2 + \sum_{m=1}^N K_{nm}\sigma^2$. Due to the convexity of f, E is jointly convex in μ, σ. In [2] we compare VO against

[3] SLEP is coded in C, so comparing speed to our MATLAB implementation is inconclusive.

Chapelle's primal approach [6] and a range of classical SVM solvers, showing that VO again has excellent empirical performance. See [2] for full details.

To our knowledge, little attention has been given to VO or its relation to other approaches. In [7], minimizing $f(x)$ is considered by first defining the distribution $\tilde{p}(x) = \frac{1}{Z}e^{\beta f(x)}$, $\beta \geq 0$, where Z normalizes $\tilde{p}$. We can find an approximation to $\tilde{p}(x)$ by minimizing

$$\mathrm{KL}(p|\tilde{p}) \equiv \underbrace{\left\langle \log p(x|\theta) \right\rangle_{p(x|\theta)} - \beta \left\langle f(x) \right\rangle_{p(x|\theta)}}_{\tilde{E}(\theta)} + \text{constant}.$$

Here $p(x|\theta)$ is chosen to ensure that $\tilde{E}(\theta)$ is tractably computable. Whilst this method does not in general provide a bound on f^*, if the entropic term $\left\langle \log p(x|\theta) \right\rangle_{p(x|\theta)}$ is constant with respect to θ, then minimizing $\tilde{E}(\theta)$ is equivalent to VO, namely maximizing $\left\langle f(x) \right\rangle_{p(x|\theta)}$. This occurs for Gaussian $p(x|\theta) = \mathcal{N}(x|\mu, \Sigma)$ and fixed Σ – in general, however, the two approaches are different.

Estimation of distribution algorithms (EDAs) are a broad set of optimization algorithms for the problem $\min_w f(w)$. An EDA starts with a prior distribution $p_0(w)$ over the solution space. At each iteration this is then used to generate a new set of candidates $\{w^n\}$. The distribution of next generation candidates is then $p_{t+1}(w) = F\big(p_t(w), \{f(w^n)\}, \{w^n\}\big)$ where F characterizes the particular form of the EDA. Berny [8] considers a similar approach for binary optimization problems, using sampling to approximate the expectations; this can be viewed therefore as an approximate version of VO. More generally, when VO is used with expectations approximated by sampling, it can be classified as an EDA.

References

[1] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[2] J. Staines and D. Barber. Variational Optimization. Technical Report arXiv:1212.4507, Department of Computer Science, University College London, 2012.

[3] J. Friedman, H. Hastie, H. Höfling, and R. Tibshirani. Pathwise Coordinate Optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[4] J. Liu, S. Ji, and J. Ye. SLEP: Sparse Learning with Efficient Projections. Technical report, Computer Science and Engineering Arizona State University, 2011.

[5] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[6] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[7] M. Gallagher and M. Frean. Population-based continuous optimization, probabilistic modelling and mean shift. *Evolutionary Computation*, 13(1):29–42, 2005.

[8] A. Berny. Selection and Reinforcement Learning for Combinatorial Optimization. In *Parallel Problem Solving from Nature VI*, pages 601–610. Springer, 2000.