A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings

Miguel Á. Carreira-Perpiñán **Max Vladymyrov**
EECS, University of California, Merced
http://eecs.ucmerced.edu

Introduction. Given a high-dimensional dataset $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ of $D \times N$, nonlinear embedding (NLE) algorithms seek to find low-dimensional projections $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ of $L \times N$ with $L < D$ by optimizing an objective function $E(\mathbf{X})$ constructed using an $N \times N$ matrix of pairwise similarities $\mathbf{W} = (w_{nm})$ between input data patterns. The algorithms of this type include the elastic embedding (EE) [1], stochastic neighbor embedding (SNE) [4], t-SNE [6], and others.

While there has been a substantial progress in designing faster optimization methods [7] and reducing the computational cost of the iterations [8] for NLE, there is a still the problem that the nonlinear embeddings do not define an "out-of-sample" mapping $\mathbf{F}$: $\mathbb{R}^D \to \mathbb{R}^L$ that can be used to project patterns not in the training set. One common approach is, instead of learning the patterns' low-dimensional projections, to learn directly a parametric mapping $\mathbf{F}$ (e.g. linear or neural net). This involves replacing $\mathbf{x}_n$ with $\mathbf{F}(\mathbf{y}_n)$ in the NLE objective function and optimizing it over the parameters of $\mathbf{F}$. For example, for the elastic embedding the objective function becomes:

$$P(\mathbf{F}) = \sum_{n,m=1}^{N} w_{nm} \left\|\mathbf{F}(\mathbf{y}_n) - \mathbf{F}(\mathbf{y}_m)\right\|^2 + \lambda \sum_{n,m=1}^{N} e^{-\left\|\mathbf{F}(\mathbf{y}_n) - \mathbf{F}(\mathbf{y}_m)\right\|^2} \qquad \lambda > 0. \tag{1}$$

The straightforward approach to train these type of embeddings is to apply the chain rule to compute gradients over the parameters of $\mathbf{F}$ and feed them to a nonlinear optimizer (usually gradient descent or conjugate gradients). This has the problem that a new gradient and optimization algorithm must be developed for each choice of E and $\mathbf{F}$, and that computing the gradient involves $\mathcal{O}(N^2)$ terms each providing a gradient over the entire mapping's parameters, which is very slow.

We propose a very different approach to optimizing parametric embeddings, based on the recently introduced *method of auxiliary coordinates (MAC)* [2, 3]. The idea is to solve an equivalent, constrained problem by introducing new variables (the auxiliary coordinates) and applying a penalty method. Alternating optimization of this over the coordinates and the mapping's parameters results in a step that trains an auxiliary embedding with a "regularization" term, and a step that trains the mapping by solving a regression, both of which can be solved by existing algorithms.

Optimizing a parametric embedding (PE) using auxiliary coordinates. The PE objective function, e.g. (1), can be written as $P(\mathbf{F}) = E(\mathbf{F}(\mathbf{Y}))$ which is a nested function where we first apply $\mathbf{F}$ and then E. The *method of auxiliary coordinates (MAC)* [2, 3], can be used to derive optimization algorithms for such nested systems. We write the nested problem $\min P(\mathbf{F}) = E(\mathbf{F}(\mathbf{Y}))$ as the following, equivalent constrained optimization problem:

$$\min \bar{P}(\mathbf{F}, \mathbf{Z}) = E(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{z}_n = \mathbf{F}(\mathbf{y}_n),\ n = 1, \dots, N \tag{2}$$

where we set an auxiliary coordinate $\mathbf{z}_n$ for each input pattern and a corresponding equality constraint. $\mathbf{z}_n$ can be obviously seen as the output of $\mathbf{F}$ (i.e., the low-dimensional projection) for $\mathbf{x}_n$. The optimization is now on an augmented space with NL extra parameters. We solve the constrained problem using a quadratic-penalty method (it is also possible to use the augmented Lagrangian method), by optimizing the following unconstrained problem and driving $\mu \to \infty$:

$$\min P_Q(\mathbf{F}, \mathbf{Z}; \mu) = E(\mathbf{Z}) + \frac{\mu}{2} \sum_{n=1}^{N} \left\|\mathbf{z}_n - \mathbf{F}(\mathbf{y}_n)\right\|^2 = E(\mathbf{Z}) + \frac{\mu}{2} \left\|\mathbf{Z} - \mathbf{F}(\mathbf{Y})\right\|^2. \tag{3}$$

We optimize P_Q using alternating optimization over the coordinates and the mapping. The step over $\mathbf{F}$ given $\mathbf{Z}$ is a *standard least-squares regression* for a dataset $(\mathbf{Y}, \mathbf{Z})$ using $\mathbf{F}$, and can be solved

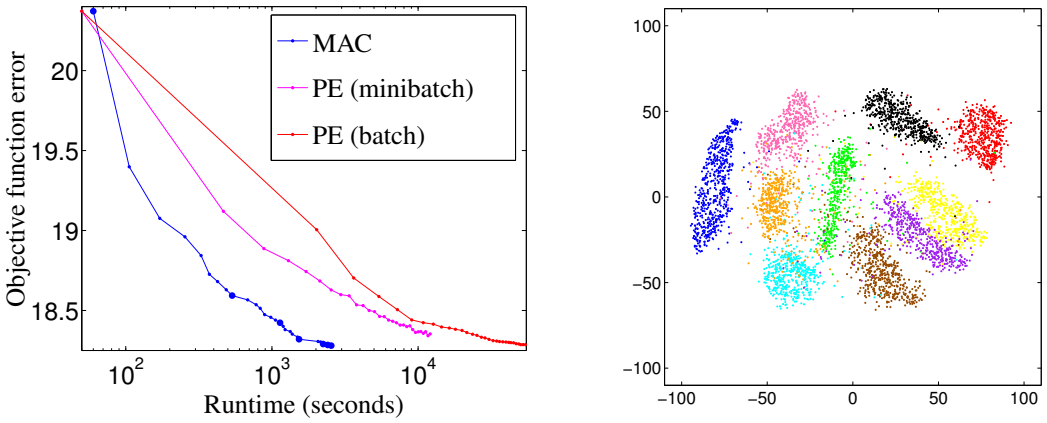

Figure 1: *Right*: learning curves. Each marker indicates one iteration. For MAC, the solid markers indicate iterations where μ increased. *Left:* 2D parametric embedding of 60 000 MNIST images, using t-SNE with a neural net. We show a sample of 5 000 points for each of the embedding plots.

using existing, well-developed code for many classes of mappings. The step over $\mathbf{Z}$ given $\mathbf{F}$ is a *regularized embedding*, since $E(\mathbf{Z})$ is the original embedding objective function and $\|\mathbf{Z} - \mathbf{F}(\mathbf{Y})\|^2$ is a quadratic regularization term on $\mathbf{Z}$ with weight $\frac{\mu}{2}$.

Using the MAC approach has many advantages compared to the direct, chain-rule optimization. First, the intricacies of nonlinear optimization (line search, method parameters, etc.) remain confined within the regression for $\mathbf{F}$ and within the embedding for $\mathbf{Z}$, *separately from each other*. Designing an optimization algorithm for an arbitrary combination of embedding and mapping is simply achieved by alternatingly calling existing algorithms for the embedding and for the mapping. Second, since the chain rule gradients are not used, we can use non-differentiable mappings (such as a decision tree), because the optimization over the non-differentiable part is confined within the regression step over the mapping. Finally, since the step over $\mathbf{Z}$ becomes a regularized embedding, we can benefit from recent advances in NLE optimization, such as the spectral direction [7] and fast multipole method approximation [8], which reduce the number and the cost of the iterations tremendously.

Experiments. We used t-SNE NLE with a neural net mapping (architecture 28×28–500–500–2000–2, initialized with pretraining[5]). We compared MAC with using a direct chain-rule optimization of the direct parametric embedding [5] using either minibatches or with a batch gradient. The chain-rule optimization converged to a different local optimum than MAC and with a larger objective function value. The learning curves in fig. 1 (left) show that MAC is considerably faster than the chain-rule optimization: almost $5\times$ faster than using minibatch (the approximate PE objective) and $20\times$ faster than the exact, batch mode. This is partly thanks to the use of N-body methods in the $\mathbf{Z}$ step. The runtimes were (excluding the time taken by pretraining, 40'): MAC: 42'; PE (minibatch): 3.36 h; PE (batch): 15 h; free embedding: 63". Fig. 1 (right) shows that the parametric t-SNE embedding preserves the overall structure of the MNIST digits very well.

References

[1] M. Á. Carreira-Perpiñán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, 2010.

[2] M. Á. Carreira-Perpiñán and W. Wang. Distributed optimization of deeply nested systems. Unpublished manuscript, arXiv:1212.5921, Dec. 24 2012.

[3] M. Á. Carreira-Perpiñán and W. Wang. Distributed optimization of deeply nested systems. *AISTATS*, 2014.

[4] G. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS*, 2003.

[5] L. J. P. van der Maaten. Learning a parametric embedding by preserving local structure. In *AISTATS*, 2009.

[6] L. J. P. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *JMLR*, 9:2579–2605, 2008.

[7] M. Vladymyrov and M. Á. Carreira-Perpiñán. Partial-Hessian strategies for fast learning of nonlinear embeddings. In *ICML*, 2012.

[8] M. Vladymyrov and M. Á. Carreira-Perpiñán. Linear-time training of nonlinear low-dimensional embeddings. In *AISTATS*, 2014.