J1.7

MACHINE LEARNING EVOLUTION FOR THE SOURCE DETECTION OF ATMOSPHERIC RELEASES

G. Cervone*[1] and P. Franzese[2]
[1]Dept. of Geography and Geoinformation Science, George Mason University
[2]Center for Earth Observing and Space Research, George Mason University

1. Introduction

Iterative source detection algorithms proved to be an effective tool for the identification of the unknown sources of atmospheric emissions (e.g., Delle Monache et al., 2008). The methodology is based on sensor measurements, numerical models and unsupervised search methodologies. In order to identify a source, multiple forward dispersion simulations from different candidate sources are performed using a numerical transport and dispersion model, and the resulting concentrations are compared to ground observations. The goal of the algorithms is to find the tentative source that minimizes the error between the simulated and the measured concentrations. These methods can be used with any type of dispersion model, can be implemented independently of the amount and type of available data, and can be applied to non-linear processes as well.

Haupt (2005), Haupt et al. (2007) and Allen et al. (2007) developed source detection methodologies based on genetic algorithms (GA) (e.g., Holland, 1975; De Jong, 2008), proving that evolutionary algorithms are well suited for the source detection problem.

We propose an iterative approach based on machine learning non-Darwinian evolutionary processes, which provide an improved search strategy. New candidate solutions are generated by a reasoning process rather than by pseudo-random operators.

In contrast to Darwinian operators of mutation and/or recombination, machine learning classifiers conduct a reasoning process in the creation of new individuals. Specifically, at each step (or selected steps) of evolution, the machine learning method generates hypotheses characterizing differences between high-performing and low-performing individuals. These hypotheses are then instantiated in various ways to generate new individuals.

To understand the advantage of using machine learning to generate new individuals, compared to a traditional Darwinian operation, it is necessary to take into account both the evolution length, defined as the number of function evaluations needed to determine the target solution, and the evolution time, defined as the execution time required to achieve this solution. Choosing between non-Darwinian and Darwinian algorithms involves assessing trade-offs between the complexity of the population generating operators and the evolution length. The machine learning operations of hypothesis generation and instantiation used are more computationally costly than operators of mutation and/or crossover, but the evolution length is typically much shorter than that of Darwinian evolutionary algorithms.

Therefore the use of machine learning as engine of evolution is especially advantageous for problems with high objective function evaluation complexity. In this respect, the problem of source detection of atmospheric pollutants is ideal due to the complexity of the function evaluation, which requires complex numerical simulations.

2. Methodology

Evolutionary algorithms are powerful search techniques to find a solution by simulating an evolutionary process. Most of them share one fundamental characteristic: the use of non-deterministic operators such as mutation and recombination to improve the fitness. These operators are semi-blind and the evolution is not guided by knowledge learned in the past generations.

Because evolutionary computation algorithms evolve a number of individuals in parallel, it is possible to learn from the 'experience' of entire populations. A similar type of biological evolution does not exist because no mechanisms exist in nature to evolve entire species. Estimation-of-Distribution Algorithms (EDA) are a form of evolutionary algorithms where an entire population may be approximated with a probability distribution (Lozano, 2006). New candidate solutions are chosen using statistical information from the sampling distribution, rather than randomly. The aim is to avoid premature convergence and to provide a more compact representation.

Discriminating between best and worst performing individuals could provide additional information on how to guide the evolutionary process. Cervone et al. (2000)a and Cervone et al. (2000)b proposed the Learnable Evolution Model (LEM) methodology in which a machine learning rule induction algorithm was used to learn attributional rules which discriminate between best and worst performing candidate solutions. New individuals are then generated according to inductive hypotheses discovered by the machine learning program. The individuals are thus genetically engineered, in the sense that the values of the variables are not randomly or semi-randomly assigned, but set according to the rules discovered by the machine learning program.

Evolution guided by machine learning represents a fundamentally different approach to evolutionary computation than Darwinian-type evolutionary algorithms. In Darwinian-type evolutionary algorithms, new individuals are generated through various mutation and/or recombination operators. Such operators are domain-independent and easy to execute, which makes them easy to apply to a wide range of problems. They are, however, semi-random, and take into consideration neither the experience of individuals in a given population (as

* *Corresponding author address*: Dept. of Geography and Geoinformation Science, George Mason University, Fairfax, VA 22030; gcervone@gmu.edu

in Lamarckian-type evolution), nor the past history of evolution. As a consequence, such algorithms proceed through a stochastic trial and error search, which may be slow. In all applications where the evaluation of a new individual is computationally very taxing, speeding up the convergence rate is paramount and requires new 'intelligent' operators.

We propose a methodology which uses an evolutionary computation process guided by hypotheses created by a machine learning program that describe areas of the search space most likely to include the global optimum. Such hypotheses are created on the basis of the current and, optionally, also past populations of individuals. Specifically, at each step of evolution, a population is divided into High-performing (H-group) and Low-performing individuals (L-group). These groups are selected from the current population, or a combination of the current and past populations. Then a learning program creates general hypotheses distinguishing between these two groups, which are instantiated in various ways to produce new, candidate individuals. We have developed a specialized machine learning classifier, called AQ4SD, to learn attributional rules that discriminate between the H- and L-group (Cervone et al., 2010). Initial experiments have shown that guiding evolutionary processes by hypotheses generation and instantiation can dramatically speed up convergence.

3. Dispersion experiment and simulations

3.1 *Prairie Grass Experiment*

The search algorithms are applied to identify the characteristics of the source in the Prairie Grass field experiment (Barad and Haugen, 1958). The experiment consisted of 68 consecutive releases of trace gas SO_2 of 10 minutes each from a single source. The mean concentration was measured at sensors positioned along arcs radially located at distances of 50 m, 100 m, 200 m, 400 m and 800 m from the source. For each experiment, data were recorded for the wind direction and speed, temperature and heat fluxes. Using these values it is possible to associate each release to a particular atmospheric class, ranging from A (most stable) to F (most unstable). Neutral atmospheric conditions corresponds to value D.

3.2 *Transport and Dispersion Simulations*

The dispersion simulations are performed using a Gaussian reflected dispersion model, which determines the predicted mean concentration c_s at a location x, y and z generated by a source located at x_s, y_s, and z_s as:

$$c_s = \frac{Qg_1g_2}{2\pi U[(\sigma_s^2+\sigma_y^2)(\sigma_s^2+\sigma_z^2)]^{1/2}} \tag{1}$$

with

$$g_1 = \exp\left[-\frac{(y-y_s)^2}{2(\sigma_s^2+\sigma_y^2)}\right]; \tag{2}$$

$$g_2 = \exp\left[-\frac{(z-z_s)^2}{2(\sigma_s^2+\sigma_z^2)}\right] + \exp\left[-\frac{(z+z_s)^2}{2(\sigma_s^2+\sigma_z^2)}\right] \tag{3}$$

where Q is the source mass emission rate, U is the wind speed, $\sigma_y(x, x_s; \psi)$ and $\sigma_z(x, x_s; \psi)$ are the crosswind and vertical dispersion coefficients (i.e. the plume spreads) where ψ describes the atmospheric stability class (i.e., $\psi = A$ to $\psi = F$), and $\sigma_s^2 = \sigma_y^2(x_s, x_s, \psi) = \sigma_z^2(x_s, x_s, \psi)$ is a measure of the area of the source. The result of the simulation is the concentration field generated by the release along an arbitrary wind direction θ. The dispersion coefficients were computed from the tabulated curves of Briggs (Arya, 1999).

In this study, U, θ and ψ are assumed to be known, with their values set according to the observations reported in the Prairie Grass dataset. Each candidate solution is thus comprised of the 5 variables x_s, y_s, z_s, Q, and σ_s.

3.3 *Error function*

The error function evaluates each candidate solution quantifying the error between the observed concentrations and the corresponding simulated values. This information is used by the search algorithm to drive the stochastic iterative process.

Different measures of accuracy of a dispersion calculation can be adopted (Hanna et al., 1993) and inter-model comparison studies always include several performance metrics (Chang et al., 2003; Chang and Hanna, 2004).

Cervone and Franzese (2010) performed a comparative study of several error functions and show that both the Normalized Root Mean Square Error (NRMSE) and the function defined by Allen et al. (2007) (referred here as AHY2) functions were suitable metrics to quantify the difference between observed and simulated concentrations for source detection algorithms.

$$\text{NRMSE} = \sqrt{\frac{\overline{(c_o - c_s)^2}}{\overline{c_o}\;\overline{c_s}}} \tag{4}$$

$$\text{AHY2} = \sqrt{\frac{\overline{[\log_{10}(c_o+1) - \log_{10}(c_s+1)]^2}}{\overline{[\log_{10}(c_o+1)]^2}}} \tag{5}$$

where c_o is each sensor's observed mean concentration, and the bar indicates an average over all the observations.

4. Results

For each of the 68 Prairie Grass experiments, the algorithm was run 10 times using a different initial random population of candidate solutions. Figure 1 illustrates a typical run of the optimization algorithm for one experiment, showing the convergence for: i) the distance to the real source; ii) Q; iii) σ_s; and iv) NRMSE as functions of algorithm iterations. The results shown in the plot are the averages over ten runs (trendline), and the minimum and maximum values obtained during the runs, identified by the shadowed area. The ideal solution corresponds to: Distance = 0, Q = 45 mg, σ_s = 1 m^2, and NRMSE = 0. In this experiment, the average distance ranges between 10 and 40 m, Q between 60 and 100 mg, σ_s between 1 and 3 m^2, and NRMSE between 0.1 and 0.5. The algorithm converges, on average, within 10 meters of the correct solutions in less than 200 iterations, corresponding to less than 2000 error function (NRMSE or AHY2) evaluations.

Figure 2 shows a summary of the results in terms of distance to the real source for each of the 68 Prairie Grass experiments, sorted from best to worst, and grouped by atmospheric stability class. The graph shows the results obtained

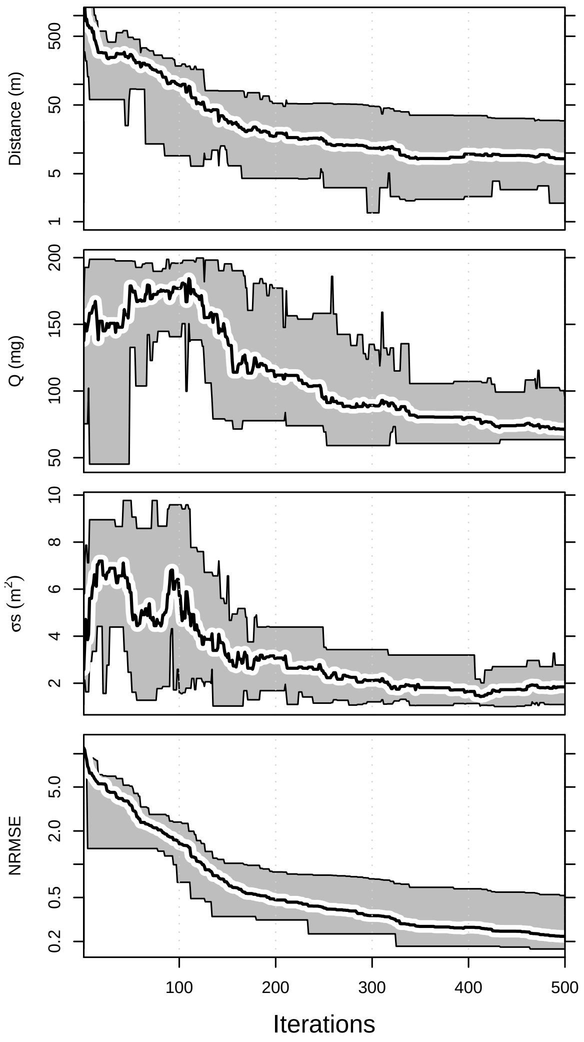

FIG. 1: Evolution of distance to the real source, Q, σ_s, and *NRMSE* as functions of algorithm iterations for one experiment. The plots show the average of 10 runs, and the extent of the values for the runs is shown with the shaded area. The trendline is the average of the 10 runs.

using the NRMSE and AHY2 functions. The two functions determine similar results, but in several cases (i.e., releases 55, 56, 41, 59, 62, 42, 54, 46, 35S, 18, 23, 24, 48, 51, 32, 29, 66 and 36) AHY2 determines a shorter distance from the source. Best results, overall, are obtained for releases in unstable to neutral atmosphere (A -> D). Less accurate predictions are consistently obtained for the releases in stable atmosphere.

Figure 3 illustrates the difference between simulated and observed concentrations for a case with very low error (experiment 55). The graph shows the concentration observed at each sensor plotted along with the concentration profile simulated by the T&D model. The sensors are positioned along five different concentric arcs indicated by alternate white and grey background, located at 50 m, 100 m, 200 m, 400 m and 800 m from the source. Within each arc, the sensors are sorted counter-clockwise.

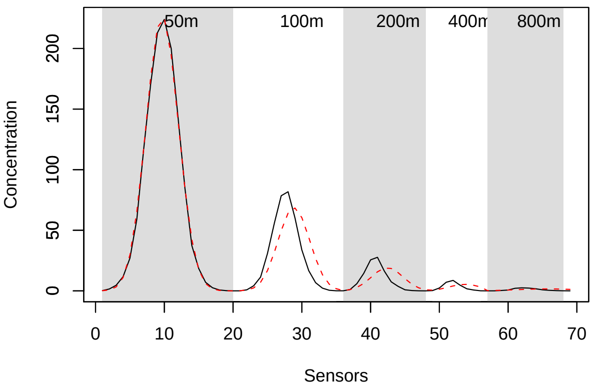

FIG. 3: Difference between simulated and observed concentrations for experiment 55. The graph shows the concentration observed at each sensor (solid line) along with the concentration profile simulated by the T&D model with the parameters determined by the search algorithm (dashed line). The sensors are positioned along five different concentric arcs indicated by alternate white and grey background, located at 50 m, 100 m, 200 m, 400 m and 800 m from the source. Within each arc, the sensors are sorted counter-clockwise.

5. Acknowledgments

Work performed under this project has been partially supported by the NSF through Award 0849191.

REFERENCES

Allen, C. T., G. S. Young, and S. E. Haupt, 2007: Improving pollutant source characterization by better estimating wind direction with a genetic algorithm. *Atmos. Environ.*, **41**(11), 2283–2289.

Arya, P. S., 1999: *Air pollution meteorology and dispersion*. Oxford University Press, 310 pp.

Barad, M., and D. Haugen, 1958: *Project Prairie Grass, a field program in diffusion.* United States Air Force, Air Research and Development Command, Air Force Cambridge Research Center.

Cervone, G., and P. Franzese, 2010: Stochastic source detection of atmospheric releases. *Computers & Geosciences*, (in press).

Cervone, G., P. Franzese, and A. P. Keesee, 2010: AQ learning for the source detection of atmospheric releases. *WIREs: Computational Statistics*, **2**(March/April), 18 pages.

Cervone, G., K. Kaufman, and R. Michalski, 2000: Experimental validations of the learnable evolution model. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on Evolutionary Computation*, volume 2.

Cervone, G., R. Michalski, K. Kaufman, and L. Panait, 2000: Combining machine learning with evolutionary computation: Recent results on lem. *Proceedings of the Fifth*

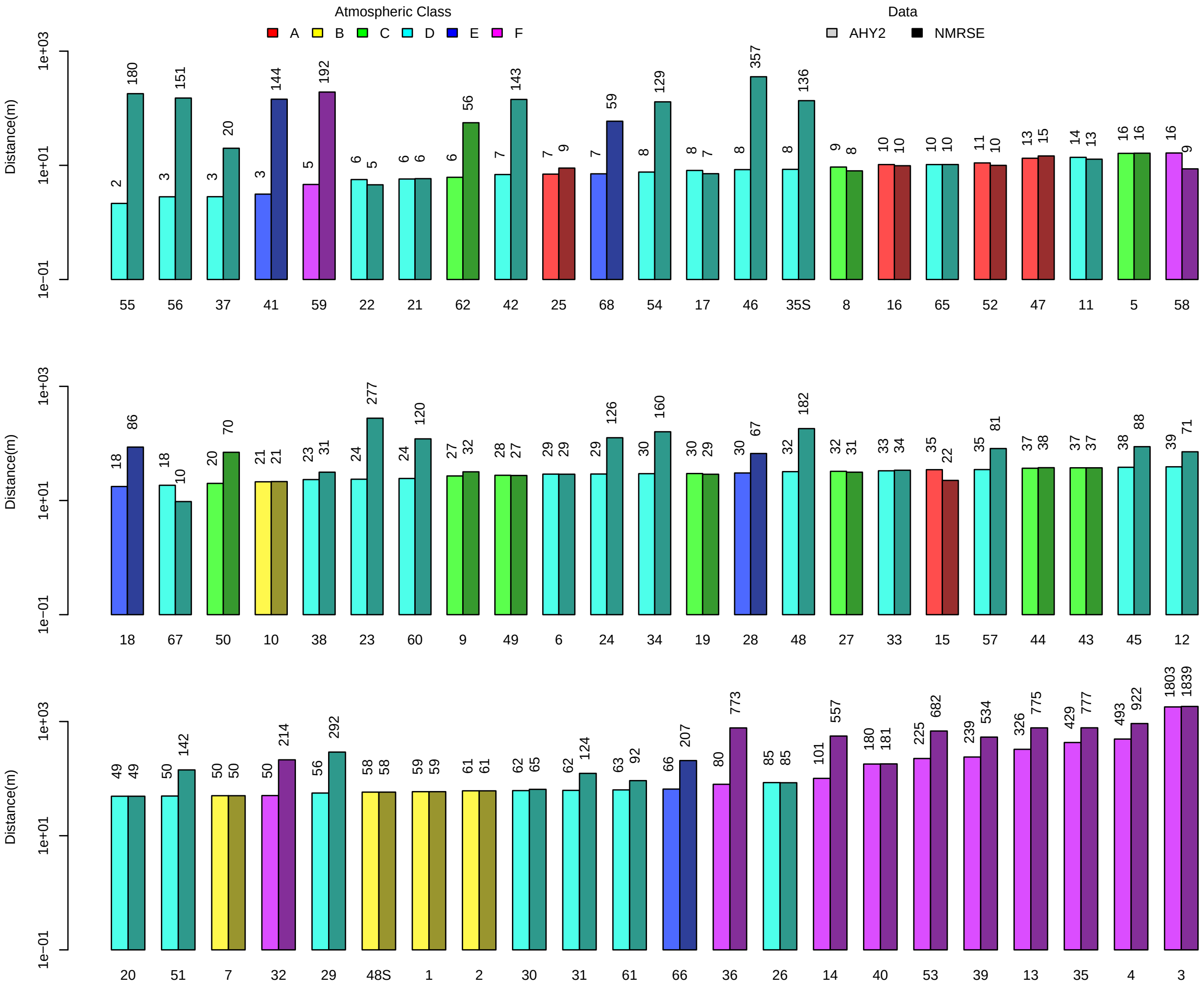

FIG. 2: Distance from the real source estimated by the algorithm for all 68 releases of the Prairie Grass experiments using the NMRSE and AHY2 error functions. Each bar is the average of 10 runs, is color coded according to the atmospheric class of the experiment, and the results are sorted from best to worst.

International Workshop on Multistrategy Learning (MSL-2000), Guimaraes, Portugal, 41–58.

Chang, J. C., P. Franzese, K. Chayantrakom, and S. R. Hanna, 2003: Evaluations of CALPUFF, HPAC, and VLSTRACK with two mesoscale field datasets. *Journal of Applied Meteorology*, **42**(4), 453–466.

Chang, J. C., and S. R. Hanna, 2004: Air quality model performance evaluation. *Meteor. Atmos. Phys*, **87**, 167–196.

De Jong, K., 2008: Evolutionary computation: a unified approach. *Proceedings of the 2008 GECCO Conference on Genetic and Evolutionary Computation*, ACM New York, NY, USA, 2245–2258.

Delle Monache, L., J. Lundquistand, B. Kosović, G. Johannesson, K. Dyer, R. Aines, F. Chow, R. Belles, W. Hanley, S. Larsen, G. Loosmore, J. Nitao, G. Sugiyama, and P. Vogt, 2008: Bayesian inference and Markov Chain Monte Carlo sampling to reconstruct a contaminant source on a continental scale. *J. Appl. Meteor. Climatol.*, **47**, 2600–2613.

Hanna, S. R., J. C. Chang, and G. D. Strimaitis, 1993: Hazardous gas model evaluation with field observations. *Atmos. Environ*, **27A**, 265–2285.

Haupt, S. E., 2005: A demonstration of coupled receptor/dispersion modeling with a genetic algorithm. *Atmos. Environ.*, **39**(37), 7181–7189.

Haupt, S. E., G. S. Young, and C. T. Allen, 2007: A genetic algorithm method to assimilate sensor data for a toxic contaminant release. *Journal of Computers*, **2**(6), 85–93.

Holland, J., 1975: *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, MA.

Lozano, J., 2006: *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. Springer.