

Apprentissage statistique à grande échelle

Francis Bach (francis.bach@ens.fr)

De nombreux problèmes d'apprentissage statistique à grande échelle sont formulés comme l'optimisation d'une fonction convexe dont on n'observe que des gradients bruités: cette fonction est typiquement l'erreur de généralisation, et seulement l'erreur sur une observation est disponible à chaque itération. Les algorithmes utilisés en pratique donnent lieu à des garanties de convergence dont l'étude [1] est le but du mémoire.

References

- [1] F. Bach, E. Moulines. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Programming and simulation of hybrid systems

Timothy Bourke (Timothy.Bourke@inria.fr)

Marc Pouzet (Mark.Pouzet@ens.fr)

Embedded systems are usually envisioned as discrete components that interact with a physical environment. This explains much of the success of hybrid modeling tools like Simulink, which have evolved from simulation platforms to development platforms where a single source code is used for simulation, testing, formal verification and the generation of embedded code. Such tools should be built on a firm semantical basis, so that the results of simulation, compilation and verification are provably mutually consistent: simulation results must be reproducible; generated code must be faithful to the original model and efficient; verification results on the model should also be valid for generated code.

The synchronous languages have successfully addressed these issues for purely discrete systems, and they are now routinely used in the development of safety critical systems in domains such as avionics, nuclear power safety, and railways. Recent work has shown how to extend synchronous languages with first Order Differential Equations (ODEs), which, 1) yields an expressive and precise programming language, 2) allows for efficient simulation using off-the-shelf numerical solvers and synchronous compilers, and, 3) provides a sound semantical basis for describing mixed discrete/continuous behaviors and hybrid automata.

For this subject, you must read background papers and book chapters on numerical simulation [1, 2] and synchronous languages [3, 4], and some recent papers describing their combination in a hybrid programming language [5, 6].

References

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, Sept. 2007.
Chapter 17: Integration of Ordinary Differential Equations
- [2] G. Dahlquist and Å. Björck. *Numerical Methods in Scientific Computing: Volume 1*. Society for Industrial and Applied Mathematics, 2008.
Chapter 6: Solving Scalar Nonlinear Equations
- [3] N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous dataflow programming language LUSTRE. *Proc. IEEE*, 79(9):1305–1320, Sept. 1991.
- [4] J.-L. Colaço, B. Pagano, and M. Pouzet. A conservative extension of synchronous data-flow with state machines. In W. Wolf, editor, *Proc. 5th ACM Int. Conf. on Embedded Software (EMSOFT'05)*, pages 173–182, Jersey City, USA, Sept. 2005. ACM Press.

- [5] Benveniste, A., Bourke, T., Caillaud, B., Pouzet, M.: Divide and recycle: types and compilation for a hybrid synchronous language. In: ACM SIGPLAN/SIGBED Conference on Languages, Compilers, Tools and Theory for Embedded Systems (LCTES'11), Chicago, USA (April 2011)
- [6] Benveniste, A., Bourke, T., Caillaud, B., Pouzet, M.: A Hybrid Synchronous Language with Hierarchical Automata: Static Typing and Translation to Synchronous Code. In: ACM SIGPLAN/SIGBED Conference on Embedded Software (EMSOFT'11), Taipei, Taiwan (October 2011)

Fastest-mixing Markov chains

Ana Busic (ana.busic@ens.fr)

Boyd, Diaconis and Xiao consider in [1] the following "fastest-mixing Markov chain" problem. Let $G = (V, E)$ be a finite graph and π a probability distribution on V such that $\pi(i) > 0$ for every i . The goal is to find the fastest-mixing reversible Markov chain with stationary distribution π and transitions allowed only along the edges in E . This is a very important problem because of the use of Markov chains in Markov chain Monte Carlo simulations where the goal is to generate a random object distributed according to π (often a uniform distribution over some set we would like to enumerate) and the Markov chain is constructed only to facilitate the random generation. The efficiency of the simulation depends on the mixing time of the Markov chain. In [1], the authors minimize SLEM (second-largest eigenvalue in modulus). Fill and Kahn propose in [2] to use comparison inequalities for Markov chains to study the mixing time in a much stronger sense - in terms of discrepancy from the stationary distribution at any time step.

References

- [1] Stephen Boyd, Persi Diaconis and Lin Xiao. Fastest Mixing Markov Chain on a Graph. *SIAM Review*, 46(4):667-689, 2004.
- [2] James A. Fill and Jonas Kahn. Comparison Inequalities and Fastest-Mixing Markov Chains. To appear in *Annals of Applied Probability*. Preprint arXiv:1109.6075.

Exploration de terrain

Christoph Dürr et Claire Mathieu (cmathieu@di.ens.fr)

Comment explorer un terrain deux dimensionnel pour trouver une cible dont l'existence est connue mais la localisation inconnue?

Un exemple jouet de problème de ce type: le cas uni dimensionnel [5]. On doit trouver une cible sur la ligne en minimisant le déplacement dans le pire des cas.

Un exemple de problème de ce type: le "Ray searching". On a des rayons, partant tous d'une même origine. Il existe une cible quelque part et il faut la trouver avec le moins de déplacements possibles [1].

Un autre exemple, qui n'a pas encore été étudié en algorithmique: la localisation de victime d'avalanche [6]. La technologie utilisée [7] a des difficultés à gérer la présence de victimes multiples. Comment modéliser ce problème dans un context algorithmique ? Quelle est la bonne fonction objectif à étudier ?

Le but de ce travail est de lire et présenter la littérature sur les "search games" [4], en particulier [2], et d'imaginer un modèle adapté au problème de recherche de victimes d'avalanches.

References

- [1] S. Angelopoulos, A. López-Ortiz and K. Panagiotou, Multi-target Ray Searching Problems. *Algorithms and Data Structures*, 37–58, 2011.
- [2] Jeż, Artur and Łopuszański, Jakub. On the two-dimensional cow search problem. IPL 2009.
http://www.ii.uni.wroc.pl/~mbi/grants/mnisw_2006/papers/2009-ipl-cowsearch-2d.pdf
- [3] M. Chrobak, A princess swimming in the fog looking for a monster cow, *ACM Sigact news*, 35(2), 74-78 (2004).
- [4] Search games. http://en.wikipedia.org/wiki/Search_games
- [5] A. Beck and D.J. Newman. Yet More on the linear search problem. *Israel J. Math.* (1970). MY Kao, JH Reif and SR Tate, Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem, *SODA* 1993.
- [6] http://en.wikipedia.org/wiki/Avalanche_transceiver
- [7] <http://pistehors.com/backcountry/wiki/Avalanches/Rechercher-Victime-Avalanche-ARVA>

Stratégies et universalité des systèmes de réécriture

Maribel Fernández (maribel@di.ens.fr)

Max Dauchet [1] a montré qu'une machine de Turing peut être codée par un système de réécriture avec une seule règle, avec une stratégie de réduction particulière.

Le but de ce projet est de comprendre ce système et spécifier formellement la stratégie de Dauchet dans un langage du style ELAN [2], Stratego [3] ou PORGY [4].

References

- [1] M. Dauchet. Simulation of Turing Machines by a left-linear rewrite rule. Proc. RTA 1989, LNCS 355, Springer.
- [2] P. Borovansky, C. kirchner, H. Kirchner, P.-E. Moreau, C. Ringeissen. An overview of ELAN. ENTCS 15, 1998.
- [3] E. Visser. Stratego: A language for program transformation based on rewriting strategies. System description. Proc. RTA 2001, LNCS 2051, Springer.
- [4] O. Andrei, M. Fernandez, H. Kirchner, G. Melançon, O. Namet, B. Pinaud. PORGY: Strategy-driven interactive transformation of graphs. Proc. TERMGRAPH 2011, EPTCS 2011.

Epidémies et graphes aléatoires

Marc Lelarge (marc.lelarge@ens.fr)

Les réseaux complexes (complex networks) font l'objet de recherches actives à l'intersection des mathématiques, de l'informatique, de la physique, de l'économie et de la sociologie. Le but de ce mémoire est de comprendre de manière rigoureuse les modèles de graphes aléatoires qui ont été développés récemment [2] puis dans un deuxième temps, de comprendre les comportements de processus stochastiques d'épidémies sur ces graphes.

References

- [1] M. Draief, L. Massoulié. Epidemics and Rumours in Complex Networks. Cambridge University Press 2010.
- [2] R. van der Hofstad. Random Graphs and Complex Networks.

Représentation Parcimonieuse de Signaux et d'Images

Stéphane Mallat (stephane.mallat@ens.fr)

Beaucoup de problèmes de traitement du signal ou de l'image nécessitent de construire une représentation parcimonieuse, ce qui veut dire représenter un signal de grande dimension avec un nombre réduit de paramètres. Les applications concernent la compression de signaux ou d'images, le débruitage, la restauration d'images dégradées ainsi que des problèmes de classification. Le but de ce projet est d'étudier les techniques de représentation parcimonieuses et une application particulière qui pourra être choisie par l'élève. Le travail se fera à partir d'un article scientifique qui dépendra de l'application choisie et inclura des simulations numériques pour tester les algorithmes étudiés. Différents stages sont possibles dans des laboratoires travaillant sur ces problèmes, aussi bien en France qu'aux US. Ce projet nécessite de suivre le cours de traitement du signal.

Garantir l'exécution à mémoire bornée de réseaux de Kahn

Louis Mandel (louis.mandel@lri.fr)

Les applications vidéo telles qu'on les trouve dans les télévisions manipulent des flots de données infinis auxquels sont appliqués des filtres successifs. Un exemple d'une de ces applications est le Picture in Picture dont l'objet est d'incruster une image dans une autre. Ces applications ont les contraintes suivantes :

- temps-réel : la fréquence d'affichage des pixels d'une image ne peut pas être diminuée ;
- performance : effectuer 1000 opérations élémentaires par pixel d'une image haute définition, nécessite 150 milliards d'opérations par secondes ;
- sûreté : on veut garantir que l'exécution se fera toujours sans blocage et que des données ne seront pas perdues.

Le langage Lucy-n est dédié à la programmation de ce type d'applications. Il repose sur la théorie des réseaux de Kahn [2] qui permet de décrire des systèmes qui sont à la fois parallèles et déterministes. Plus précisément, il étend le modèle de programmation synchrone [1] avec de la communication par files, appelées *buffers*.

Un des rôles du compilateur est de dimensionner automatiquement la taille des buffers en garantissant que le programme pourra s'exécuter en mémoire bornée et sans blocage. Afin de réaliser cette tâche, il utilise des informations sur les rythmes de consommation et de production des flots qui sont données sous forme de mots ultimement périodiques [4].

Pour cet exposé, vous devrez lire la thèse de Florence Plateau [3] qui présente le langage Lucy-n et les algorithmes de dimensionnement des buffers.

References

- [1] Albert Benveniste, Paul Caspi, Stephen A. Edwards, Nicolas Halbwachs, Paul Le Guernic and Robert de Simone, The Synchronous Languages 12 Years Later, Proceedings of the IEEE 91.1, 2003.
- [2] Gilles Kahn, The Semantics of Simple Language for Parallel Programming, Proceedings of IFIP 74 Conference, 1974.
- [3] Florence Plateau, Modèle n-synchrone pour la programmation de réseaux de Kahn à mémoire bornée, Thèse de doctorat de l'Université Paris-Sud 11, janvier 2010.
- [4] Jean Vuillemin, On circuits and numbers, IEEE Transactions on Computers 1994.

Traveling salesman problem

Claire Mathieu (cmathieu@di.ens.fr)

The traveling salesman problem is one of the most famous of combinatorial optimization. NP-complete even in the metric case, it has led to wonderful progress in the design of approximation algorithms. The first interesting heuristic, due to Christofides [3], uses a magical algorithm by Edmonds for minimum weight perfect matching [5] that extends a simple but clever algorithm for unweighted maximum matching [4]. It produces a tour with length at most $3/2$ times optimal. Since then, improving on that ratio has been an outstanding open problem of Algorithms. In the 1990s, a major breakthrough happened for low-dimensional Euclidian metrics [6]. In 2011, another major breakthrough happened for unweighted graph metrics [2], followed by further progress in 2012 for a variant of the problem [1]. These two very recent papers open exciting possibilities for gaining new understanding of the combinatorial structure of TSP.

The main goal of the project is to read, understand and present [2, 1].

References

- [1] Hyung-Chan An, Robert Kleinberg, David B. Shmoys: Improving christofides' algorithm for the s-t path TSP. STOC 2012: 875-886
- [2] Tobias Mömke, Ola Svensson: Approximating Graphic TSP by Matchings. FOCS 2011: 560-569
- [3] The Christofides algorithm for metric TSP. 1976. http://en.wikipedia.org/wiki/Christofides_algorithm
- [4] The blossom algorithm for maximum matching. Jack Edmonds, 1961. http://en.wikipedia.org/wiki/Blossom_algorithm
- [5] Schrijver, Alexander. Combinatorial Optimization: Polyhedra and Efficiency. Algorithms and Combinatorics. Springer.
- [6] Vijay V. Vazirani: Approximation algorithms. Springer 2001, isbn 978-3-540-65367-7, pp. I-IXI, 1-378. Arora's algorithm is presented in chapter 11.

Logique de séparation et vérification de programmes

Xavier Rival (rival@di.ens.fr)

La vérification de programme a pour but de formaliser des preuves de correction de fragments de code, à l'aide d'annotations qui peuvent être soit écrites à la main, soit calculée par un analyseur statique. La logique de Hoare [1] formalise de telles annotations sous la forme preCpost où C est un fragment de programme, pre une pré-condition et post une post-condition. Dans le cas de programmes manipulant des structures de données avec pointeurs telles que des listes ou des graphes, il est particulièrement difficile de traiter le cas des *alias*, au cours des preuves. La logique de séparation [2] propose une description très fine de la structure de la mémoire, et apporte une solution à ce problème. Ce sujet d'exposé propose d'étudier ce problème et le principe de la logique de séparation.

References

- [1] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, October 1969.
- [2] J. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS*, pages 55–74, 2002.