Optimal Computational Trade-Off of Inexact Proximal Methods

Multi-Trade-offs in Machine Learning

NIPS workshop 2012

Pierre Machart
Joint work with Sandrine Anthoine and Luca Baldassarre

LIF, Aix-Marseille Université
LSIS, Université du Sud-Toulon-Var

http://www.lif.univ-mrs.fr/~pmachart/
pierre.machart@lif.univ-mrs.fr

December, 7th 2012
Outline of the talk

The Trade-Offs of Learning

Inexact Proximal Methods

Main Contribution

Numerical Simulations

Conclusion
Outline

The Trade-Offs of Learning
  The Big Picture
  Excess Error Decomposition
  Motivation

Inexact Proximal Methods

Main Contribution

Numerical Simulations

Conclusion
Minimizing the Risk

Supervised Statistical Learning:

- Data: $n$ realizations of $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with distribution $D$. 
Minimizing the Risk

Supervised Statistical Learning:

- Data: \( n \) realizations of \((x, y) \in \mathcal{X} \times \mathcal{Y}\) with distribution \( D \).
- Goal: learning a “good” predictor \( h : \mathcal{X} \to \mathcal{Y}\).
Minimizing the Risk

Supervised Statistical Learning:

- Data: \( n \) realizations of \( (x, y) \in \mathcal{X} \times \mathcal{Y} \) with distribution \( D \).
- Goal: learning a “good” predictor \( h : \mathcal{X} \rightarrow \mathcal{Y} \).

- “Goodness” of a prediction measured through a loss function:

\[
\ell : \mathcal{Y}^\mathcal{X} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+
\]
Minimizing the Risk

Supervised Statistical Learning:

- Data: \( n \) realizations of \((x, y) \in \mathcal{X} \times \mathcal{Y}\) with distribution \(D\).
- Goal: learning a “good” predictor \( h : \mathcal{X} \rightarrow \mathcal{Y} \).

- “Goodness” of a \textit{prediction} measured through a \textit{loss function}:

\[
\ell : \mathcal{Y}^\mathcal{X} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+
\]

- “Goodness” of a \textit{predictor} measured through a \textit{risk function}:

\[
R(h) = \mathbb{E}_D \ell(h, x, y)
\]
Minimizing the Risk

Supervised Statistical Learning:

- Data: \( n \) realizations of \((x, y) \in \mathcal{X} \times \mathcal{Y}\) with distribution \(D\).
- Goal: learning a “good” predictor \( h : \mathcal{X} \rightarrow \mathcal{Y} \).

“Goodness” of a prediction measured through a loss function:
\[
\ell : \mathcal{Y}^\mathcal{X} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+
\]

“Goodness” of a predictor measured through a risk function:
\[
R(h) = \mathbb{E}_D \ell(h, x, y)
\]

Absolute best predictor:
\[
h^* := \arg\min_{h \in \mathcal{Y}^\mathcal{X}} R(h)
\]
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$\mathcal{E} := \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.$$
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$\mathcal{E} := \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.$$ 

Small-scale problems:
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$\mathcal{E} := \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.$$ 

Small-scale problems:

Vladimir Vapnik
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$\mathcal{E} := \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.$$ 

Small-scale problems:

Vladimir Vapnik  

Yurii Nesterov
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$\mathcal{E} := \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.$$  

Large-scale problems:
Excess Error Decomposition

Learning algorithms give you $\tilde{h}_n$ with an excess error:

$$E := E_{\text{app}} + E_{\text{est}} + E_{\text{opt}}.$$

Large-scale problems:

Léon Bottou
Consequences of this Trade-Off

- Computational efficiency matters.
  ⇒ How to assess it?

- Optimizing with limited precision.
  ⇒ Are rates of convergence still relevant?

- Runtime as a limiting resource.
  ⇒ How to take it into account?
Outline

The Trade-Offs of Learning

Inexact Proximal Methods
  Non-Smooth Convex Optimization
  Inexact Proximal Methods
  Rates of Convergence

Main Contribution

Numerical Simulations

Conclusion
Non-smooth convex optimization

General problem:
Minimization of a composite function:

\[ \min_x f(x) := g(x) + h(x), \]

with \( g : \mathbb{R}^n \to \mathbb{R} \) convex, smooth, with \( L \)-LCG and \( h : \mathbb{R}^n \to \mathbb{R} \) lower semi-continuous proper convex.
Non-smooth convex optimization

General problem:
Minimization of a composite function:
\[
\min_x f(x) := g(x) + h(x),
\]
with \( g : \mathbb{R}^n \to \mathbb{R} \) convex, smooth, with \( L \)-LCG and \( h : \mathbb{R}^n \to \mathbb{R} \) lower semi-continuous proper convex.

General framework:
Proximal-Gradient Methods:
\[
x_k = \text{prox}_{h/L} \left[ x_{k-1} - \frac{1}{L} \nabla g(x_{k-1}) \right],
\]
\[
\text{prox}_{h/L}(z) = \arg\min_x \frac{L}{2} \|x - z\|^2 + h(x),
\]
(There exist some accelerated schemes...)
Inexact Proximal Methods

Choices of $h$:

- $L_1$-regularization, indicator of a convex set... ⇒ proximity operator computed in closed-form.
- TV-regularization, norms inducing structured sparsity, and many others... ⇒ no closed-form solution.
Inexact Proximal Methods

Choices of $h$:

- $L_1$-regularization, indicator of a convex set... ⇒ proximity operator computed in closed-form.
- TV-regularization, norms inducing structured sparsity, and many others... ⇒ no closed-form solution.

Numerical solution inducing some approximation:

$$\frac{L}{2} \| x_k - z \|^2 + h(x_k) \leq \epsilon_k + \min_x \left\{ \frac{L}{2} \| x - z \|^2 + h(x) \right\}.$$
Inexact Proximal Methods

Choices of $h$:

- $L_1$-regularization, indicator of a convex set... ⇒ proximity operator computed in closed-form.
- TV-regularization, norms inducing structured sparsity, and many others... ⇒ no closed-form solution.

Numerical solution inducing some approximation:

$$\frac{L}{2} \| x_k - z \|^2 + h(x_k) \leq \epsilon_k + \min_x \left\{ \frac{L}{2} \| x - z \|^2 + h(x) \right\}.$$

where $\{\epsilon_i\}_{i=1}^k$ are optimization hyper-parameters.
Overview of the Algorithm

Algorithm 1 Inexact Proximal Algorithms

Require: initial point $x_0$

for $i = 1$ to $k$ do

$x_{i-\frac{1}{2}} = x_{i-1} - \frac{1}{L} \nabla g(x_{i-1})$ "gradient descent" step

while $\epsilon_i$ is too large do

Increase the precision of $\text{prox}_{h/L}(x_{i-\frac{1}{2}})$

end while

$x_i = \text{prox}_{h/L}(x_{i-\frac{1}{2}})$

end for
Rates of convergence for inexact proximal methods

Convergence rates given by [Schmidt et al., 2011]:

$$f(x_k) - f(x^*) \leq \frac{L}{2k} \left( \|x_0 - x^*\| + 2 \sum_{i=1}^{k} \sqrt{\frac{2\epsilon_i}{L}} + \sqrt{\sum_{i=1}^{k} \frac{2\epsilon_i}{L}} \right)^2.$$  

⇒ Optimal rates when \(\{\epsilon_k\}\) converges faster than \(O\left(\frac{1}{k(2+\delta)}\right)\).
Rates of convergence for inexact proximal methods

Convergence rates given by [Schmidt et al., 2011]:

\[
f(x_k) - f(x^*) \leq \frac{L}{2k} \left( \|x_0 - x^*\| + 2 \sum_{i=1}^{k} \sqrt{\frac{2\epsilon_i}{L}} + \sqrt{\sum_{i=1}^{k} \frac{2\epsilon_i}{L}} \right)^2.
\]

⇒ Optimal rates when \( \{\epsilon_k\} \) converges faster than \( O\left(\frac{1}{k(2+\delta)}\right) \).

However, this imposes a STRICT control over the approximations.
Rates of convergence for inexact proximal methods

Convergence rates given by [Schmidt et al., 2011]:

\[
f(x_k) - f(x^*) \leq \frac{L}{2k} \left( \|x_0 - x^*\| + 2 \sum_{i=1}^{k} \sqrt{\frac{2\epsilon_i}{L}} + \sqrt{\sum_{i=1}^{k} \frac{2\epsilon_i}{L}} \right)^2.
\]

\[\Rightarrow\text{Optimal rates when } \{\epsilon_k\} \text{ converges faster than } O\left(\frac{1}{k(2+\delta)}\right).\]

However, this imposes a STRICT control over the approximations.

Remember:

- Computational efficiency matters.
  \[\Rightarrow\text{How to assess it?}\]
- Optimizing with limited precision.
  \[\Rightarrow\text{Are rates of convergence still relevant?}\]
- Runtime as a limiting resource.
  \[\Rightarrow\text{How to take it into account?}\]
Outline

The Trade-Offs of Learning

Inexact Proximal Methods

Main Contribution

Computational cost

Main result

Numerical Simulations

Conclusion
Defining and Optimizing the Cost

Global cost of the optimization procedure:

\[ C_{\text{glob}}(k, \{l_i\}_{i=1}^k) = C_{\text{in}} \sum_{i=1}^k l_i + kC_{\text{out}}. \]
Defining and Optimizing the Cost

Global cost of the optimization procedure:

\[ C_{\text{glob}}(k, \{l_i\}_{i=1}^k) = C_{\text{in}} \sum_{i=1}^{k} l_i + kC_{\text{out}}. \]

The “fastest” strategy can be retrieved by solving an optimization problem:

\[ \min_{k, \{l_i\}_{i=1}^k} \quad C_{\text{in}} \sum_{i=1}^{k} l_i + kC_{\text{out}} \quad \text{s.t.} \quad f(x_k) - f(x^*) \leq \rho. \]
Precision and Number of Iterations

\[ \min_{k, \{l_i\}_{i=1}^k} C_{\text{in}} \sum_{i=1}^k l_i + kC_{\text{out}} \quad \text{s.t.} \quad f(x_k) - f(x^*) \leq \rho. \]

The proximal point is approximated via iterative algorithms with sub-linear convergence rate:

\[ \epsilon_i = \frac{A}{\ell^\alpha}. \]
Precision and Number of Iterations

\[
\min_{k, \{l_i\}_{i=1}^k} C_{in} \sum_{i=1}^{k} l_i + kC_{out} \quad \text{s.t. } f(x_k) - f(x^*) \leq \rho.
\]

The proximal point is approximated via iterative algorithms with sub-linear convergence rate:

\[
\epsilon_i = \frac{A}{l_i^{\alpha}}.
\]

Gives rise to parameterized bound on \( f(x_k) - f(x^*) \):

\[
f(x_k) - f(x^*) \leq B(k, \{l_i\}_{i=1}^k),
\]

with

\[
B(k, \{l_i\}_{i=1}^k) = \frac{L}{2k} \left( \|x_0 - x^*\| + 3 \sum_{i=1}^{k} \sqrt{\frac{2A}{Ll_i^{\alpha}}} \right)^2.
\]
Optimal Strategy

Define \( C(k) = \frac{\sqrt{L}}{3\sqrt{2}A} \left( \frac{2k\rho}{L} - \|x_0 - x^*\| \right) \).

Proposition

If \( \rho < 6\sqrt{2LA}\|x_0 - x^*\| \), the solution of our optimization problem:

\[
\min_{k,\{l_i\}_{i=1}^k} \sum_{i=1}^k l_i + kC_{\text{out}} \quad \text{s.t.} \quad B(k, \{l_i\}_{i=1}^k) \leq \rho,
\]

is:

\[
\forall i, l_i^* = \left( \frac{C(k^*)}{k^*} \right)^\frac{-2}{\alpha}, \text{ with } k^* = \arg\min_{k \in \mathbb{N}^*} kC_{\text{in}} \left( \frac{C(k)}{k} \right)^{-\frac{2}{\alpha}} + kC_{\text{out}}.
\]
Optimal Strategy

Define $C(k) = \frac{\sqrt{L}}{3\sqrt{2A}} \left( \sqrt{\frac{2k\rho}{L}} - \|x_0 - x^*\| \right)$.

Proposition

If $\rho < 6\sqrt{2LA}\|x_0 - x^*\|$, the solution of our optimization problem:

$$\min_{k,\{l_i\}_{i=1}^k} C_{in} \sum_{i=1}^k l_i + kC_{out} \quad \text{s.t.} \quad B(k, \{l_i\}_{i=1}^k) \leq \rho,$$

is:

$$\forall \ i, l_i^* = \left( \frac{C(k^*)}{k^*} \right)^{-\frac{2}{\alpha}}, \ \text{with} \ k^* = \arg\min_{k \in \mathbb{N}^*} kC_{in}(\frac{C(k)}{k})^{-\frac{2}{\alpha}} + kC_{out}.$$

Remarks:

- Constant number of inner iterations (hence $\epsilon_i$).
Optimal Strategy

Define $C(k) = \frac{\sqrt{L}}{3\sqrt{2A}} \left( \sqrt{\frac{2k \rho}{L}} - \|x_0 - x^*\| \right)$.

Proposition

If $\rho < 6\sqrt{2LA}\|x_0 - x^*\|$, the solution of our optimization problem:

$$\min_{k, \{l_i\}_{i=1}^k} C_{in} \sum_{i=1}^k l_i + kC_{out} \quad \text{s.t.} \quad B(k, \{l_i\}_{i=1}^k) \leq \rho,$$

is:

$$\forall i, l_i^* = \left( \frac{C(k^*)}{k^*} \right)^{-\frac{2}{\alpha}}, \text{ with } k^* = \arg\min_{k \in \mathbb{N}^*} kC_{in}\left( \frac{C(k)}{k} \right)^{-\frac{2}{\alpha}} + kC_{out}.$$

Remarks:

- Constant number of inner iterations (hence $\epsilon_i$).
- $l_i^*$ such that the bound $B$ exactly equals $\rho$ for $k^*$ outer iterations.
Outline

The Trade-Offs of Learning

Inexact Proximal Methods

Main Contribution

Numerical Simulations

Conclusion
Some simulations on a TV-reg deblurring problem

Classical setting: deblurring Lena.

![Graph showing computational cost vs. $F_k - F^*$ for different $\varepsilon_k = 1/K^{2+\delta}$ and SIP (tol = 1e-8).]
Outline

The Trade-Offs of Learning

Inexact Proximal Methods

Main Contribution

Numerical Simulations

Conclusion
Conclusions and Future work

- A new finite-time analysis (as opposed to asymptotical ones).
- Computationally optimal strategies to provably get $\rho$-accurate solutions.
- A new practical strategy SIP that seems to perform extremely well.
Conclusions and Future work

- A new finite-time analysis (as opposed to asymptotical ones).
- Computationally optimal strategies to provably get $\rho$-accurate solutions.
- A new practical strategy SIP that seems to perform extremely well.

Main open question:

- Same methodology to optimize the computational efficiency in other settings?
Conclusions and Future work

- A new finite-time analysis (as opposed to asymptotical ones).
- Computationally optimal strategies to provably get $\rho$-accurate solutions.
- A new practical strategy SIP that seems to perform extremely well.

Main open question:

- Same methodology to optimize the computational efficiency in other settings?