Optimization in Machine Learning

Recent Developments and Current Challenges

Stephen Wright

University of Wisconsin-Madison

NIPS Workshop, Whistler, 12 December 2008

Summary

1. Sparse / Regularized Optimization
2. SVM Formulations
3. SVM Algorithms (recently proposed)
4. New optimization tools of possible interest.

Themes

- Optimization problems from machine learning are difficult (size, kernel density, ill conditioning)
- Machine learning community has made excellent use of optimization technology. Many interesting adaptations of fundamental algorithms that exploit the structure and fit the requirements of the application.
- Several current topics in optimization may be of interest in solving machine learning problems.

Sparse Optimization

Traditionally, research on algorithmic optimization assumes exact data available and precise solutions needed.

However, in many optimization applications we prefer simple, approximate solutions to more complicated exact solutions.

- simple solutions easier to actuate;
- uncertain data does not justify precise solutions; regularized solutions less sensitive to inaccuracies;
- simple solution more "generalizable."

These new "ground rules" may change the algorithmic approach altogther.

For example, an approximate first-order method applied to a nonsmooth formulation may be preferred to a second-order method applied to a smooth formulation.

Regularized Formulations

Vapnik: "...tradeoff between the quality of the approximation of the given data and the complexity of the approximating function."

Simplicity sometimes manifested as sparsity in the solution vector (or some simple transformation of it).

$$\min \mathcal{F}(x) + \lambda \mathcal{R}(x),$$

- $\mathcal{F}$ is the model, data-fitting, or loss term (the function that would appear in a standard optimization formulation);
- $\mathcal{R}$ is a regularization function;
- $\lambda \geq 0$ is a regularization parameter.

$\mathcal{R}$ can be nonsmooth, to promote sparsity in x (e.g. $\|\cdot\|_1$).

Smooth choices of $\mathcal{R}$ such as $\|\cdot\|_2^2$ (Tikhonov regularization, ridge regression) suppress the size of x and improve conditioning.

Example: Compressed Sensing

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1,$$

where A often combines a "sensing matrix" with a basis, problem is formulated so that there is a sparse x (few nonzeros) satisfying $Ax \approx b$.

Typically A has (many) more columns than rows, and has special properties to ensure that different sparse signals give different "signatures" Ax.

Under these assumptions the "ℓ_2-ℓ_1" formulation above can recover the exact solution of $Ax = b$, in the noise-free case, if A has enough rows.

Control sparsity of recovered solution via λ.

LASSO for variable selection in least squares is similar.

Example: TV-regularized image denoising

Given an image $f : \Omega \to \mathbb{R}$ over a spatial domain Ω, find a nearby u that preserves edges while removing noise. (Recovered u has large constant regions.)

$$\min_u \int_\Omega (u - f)^2 \, dx + \lambda \int_\Omega |\nabla u| \, dx.$$

Here $\nabla u : \Omega \to \mathbb{R}^2$ is the spatial gradient of u.

λ controls fidelity to image data.

Recent work shows that gradient-projection methods on dual or primal-dual are much faster at recovering approximate solutions than methods with fast asymptotic convergence.

Example: Cancer Radiotherapy

In radiation treatment planning, there are an astronomical variety of possibilies for delivering radiation from a device to a treatment area. Can vary beam shape, exposure time (weight), angle.

Aim to deliver a prescribed radiation dose to the tumor while avoiding surrounding critical organs and normal tissue. Also wish to use just a few beams. This makes delivery more practical and is observed to be more robust to data uncertainty.

Example: Matrix Completion

Seek an $m \times n$ matrix X of low rank that (approximately) matches certain linear observations about its contents.

$$\min_X \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \lambda\|X\|_*,$$

where $\mathcal{A}$ is a linear map from $\mathbb{R}^{m\times n}$ to R^p, and $\|\cdot\|_*$ is the *nuclear norm* — the sum of singular values.

Nuclear norm serves as a surrogate for rank of X, in a similar way to $\|x\|_1$ serving as a surrogate for cardinality of x in compressed sensing.

Algorithms can be similar to compressed sensing, but with more complicated linear algebra. (Like the relationship of interior-point SDP solvers to interior-point LP solvers.)

Solving Regularized Formulations

- Different applications have very different properties and requirements, that require different algorithmic approaches.
- Some approaches transfer between applications and can be analyzed at a more abstract level.
- Duality often key to getting a practical formulation.
- Often want to solve for a range of λ values (i.e. different tradeoffs between optimality and regularity).

Often, there is a choice between

(i) methods with fast asymptotic convergence (e.g. interior-point, SQP) with expensive steps and

(ii) methods with slow asymptotic convergence and cheap steps, requiring only (approximate) gradient information.

The latter are more appealling when we need only an approximate solution. The best algorithms may combine both approaches!

SVM Classification: Primal

Feature vectors $x_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$, binary labels $y_i \in \{-1, 1\}$.

Linear classifier: Defined by $w \in \mathbb{R}^n$, $b \in \mathbb{R}$: $f(x) = w_i^T x + b$.

Perfect separation if $y_i f(x_i) \geq 1$ for all i. Otherwise try to find (w, b) that keeps the classification errors ξ_i small (usually a separable, increasing function of ξ_i).

Usually include in the objective a norm of w or (w, b). The particular choice $\|w\|_2^2$ yields a maximum-margin separating hyperplane.

A popular formulation: SVC-C aka L1-SVM (hinge loss):

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi_i,$$

$$\text{subject to } \xi_i \geq 0, \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N.$$

Dual

The SVC-C formulation is a convex QP. Dual is also a convex QP, in variable $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$:

$$\min_{\alpha} \frac{1}{2}\alpha^T K \alpha - \mathbf{1}^T \alpha \qquad \text{s.t. } 0 \leq \alpha \leq C\mathbf{1}, \;\; y^T \alpha = 0,$$

where

$$K_{ij} = (y_i y_j) x_i^T x_j, \;\; y = (y_1, y_2, \ldots, y_N)^T, \;\; \mathbf{1} = (1, 1, \ldots, 1)^T.$$

KKT conditions relate primal and dual solutions:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i,$$

while b is Lagrange multiplier for $y^T \alpha = 0$. Leads to classifier:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b.$$

Kernel Trick, RKHS

For a more powerful classifier, can project feature vector x_i into a higher-dimensional space via a function $\phi : \mathbb{R}^n \to \mathbb{R}^t$ and classify in that space. Dual formulation is the same, except for redefined K:

$$K_{ij} = (y_i y_j)\phi(x_i)^T \phi(x_j).$$

Leads to classifier:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)^T \phi(x) + b.$$

Don't actually need to use ϕ at all, just inner products $\phi(x)^T \phi(\bar{x})$. Instead of ϕ, work with a kernel function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

If k is continuous, symmetric in arguments, and positive definite (Mercer kernel), there exists a Hilbert space and a function ϕ in this space such that $k(x, \bar{x}) = \phi(x)^T \phi(\bar{x})$.

Thus, a typical strategy is to choose a kernel k, form $K_{ij} = y_i y_j k(x_i, x_j)$, solve the dual to obtain α and b, and use the classifier

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b.$$

Most popular kernels:

- Linear: $k(x, \bar{x}) = x^T \bar{x}$
- Gaussian: $k(x, \bar{x}) = exp(-\gamma \|x - \bar{x}\|^2)$
- Polynomial: $k(x, \bar{x}) = (x^T \bar{x} + 1)^d$

These (and other kernels) lead to dense K, often ill conditioned.

Solving the Primal and (Kernelized) Dual

Many methods have been proposed for solving either the primal formulation of linear classification, or the dual (usually with kernel form). Research continues apace.

Most are based on optimization methods, or can be interpreted using the optimization framework.

Methods compared via a variety of metrics:

- CPU time to find solution of given quality (error rate, or target objective value).
- Theoretical efficiency.
- Data storage requirements.
- (Simplicity.) (Parallelizability.)

We'll review several approaches, emphasizing recent developments and large-scale problems.

Solving the Dual

$$\min_{\alpha} \frac{1}{2}\alpha^T K\alpha - \mathbf{1}^T\alpha \qquad \text{s.t. } 0 \le \alpha \le C\mathbf{1}, \;\; y^T\alpha = 0.$$

Convex QP with mostly bound constraints, but

a. Dense, ill conditioned Hessian makes it tricky

b. The linear constraint $y^T\alpha = 0$ is a nuisance!

Many methods proposed to work with this formulation.

Dual SVM: Coordinate Descent

(Hsieh et al 2008) Deal with the constraint $y^T\alpha = 0$ by getting rid of it! Corresponds to removing the "intercept" term b from the classifier:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i,$$
$$\text{subject to } \xi_i \geq 0, \quad y_i w^T x_i \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N,$$

Get a convex, bound-constrained QP:

$$\min_{\alpha} \frac{1}{2}\alpha^T K\alpha - \mathbf{1}^T\alpha \qquad \text{s.t. } 0 \leq \alpha \leq C\mathbf{1}.$$

Basic step: for some $i = 1, 2, \ldots, N$, solve this problem in closed form for α_i, holding all components α_j, $j \neq i$ fixed.

- Can cycle through $i = 1, 2, \ldots, N$, or pick i at random.
- Update $K\alpha$ by evaluating one column of the kernel.
- Gets near-optimal solution quickly.

Dual SVM: Gradient Projection

(Dai&Fletcher 2006) Define $\Omega = \{0 \leq \alpha \leq C\mathbf{1},\ y^T\alpha = 0\}$ and solve

$$\min_{\alpha \in \Omega} q(\alpha) := \frac{1}{2}\alpha^T K\alpha - \mathbf{1}^T\alpha$$

by means of gradient projection steps:

$$\alpha_{l+1} = P_\Omega\left(\alpha_l - \gamma_l \nabla q(\alpha_l)\right),$$

where P_Ω denotes projection onto Ω and γ_l is a steplength.

P_Ω not trivial, but not too hard to compute

Can choose γ_l using a Barzilai-Borwein formula together with a nonmonotone (but safeguarded) procedure. Basic form of BB chooses γ_l so that $\gamma_l^{-1} I$ mimics behavior of true Hessian ∇q over the latest step; leads to

$$\gamma_l = \frac{s_l^T s_l}{s_l^T y_l}, \quad \text{where } \ s_l := \alpha_l - \alpha_{l-1}, \ \ y_l := \nabla q(\alpha_l) - \nabla q(\alpha_{l-1}).$$

Dual SVM: Decomposition

Many algorithms for dual formulation make use of *decomposition*: Choose a subset of components of α and (approximately) solve a subproblem in just these components, fixing the other components at one of their bounds. Usually maintain feasible α throughout.

Many variants, distinguished by strategy for selecting subsets, size of subsets, inner-loop strategy for solving the reduced problem.

SMO: (Platt 1998). Subproblem has two components.

SMV$^{\text{light}}$: (Joachims 1998). Use chooses subproblem size (usually small); components selected with a first-order heuristic. (Could use an ℓ_1 penalty as surrogate for cardinality constraint?)

PGPDT: (Zanni, Serafini, Zanghirati 2006) Decomposition, with gradient projection on the subproblems. Parallel implementation.

LIBSVM: (Fan, Chen, Lin, Chang 2005). SMO framework, with first- and second-order heuristics for selecting the two subproblem components. Solves a 2-D QP to get the step.

Heuristics are vital to efficiency, to save expense of calculating components of kernel K and multiplying with them:

- Shrinking: exclude from consideration the components α_i that clearly belong at a bound (except for a final optimality check);
- Caching: Save some evaluated elements K_{ij} in available memory.

Performance of Decomposition:

- Used widely and well for $>$ 10 years.
- Solutions α are often not particularly sparse (many support vectors), so many outer (subset selection) iterations are required.
- Can be problematic for large data sets.

Dual SVM: Active-Set

(Scheinberg 2006)

- Apply a standard QP active-set approach to Dual, usually changing set of "free" components $\alpha_i \in (0, C)$ by one index at each iteration.
- Update Cholesky factorization of "free" part of Hessian K after each change.
- Uses shrinking strategy to (temporarily) ignore components of α that clearly belong at a bound.

(Shilton et al 2005) Apply active set to a min-max formulation (a way to get rid of $y^T \alpha = 0$:

$$\max_b \min_{0 \le \alpha \le C\mathbf{1}} \frac{1}{2} \begin{bmatrix} b \\ \alpha \end{bmatrix}^T \begin{bmatrix} 0 & y^T \\ y & K \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} b \\ \alpha \end{bmatrix}$$

Cholesky-like factorization maintained.

Active set methods good for

- warm starting, when we explore the solution path defined by C.
- incremental, where we introduce data points (x_i, y_i) one by one (or in batches) by augmenting α appropriately, and carrying on.

Dual SVM: Interior-Point

(Fine&Scheinberg 2001). Primal-dual interior-point method. Main operation at each iteration is solution of a system of the form

$$(K + D)u = w,$$

where K is kernel and D is a diagonal. Can do this efficiently if we have a low-rank approximation to K, say $K \approx VV^T$, where $V \in \mathbb{R}^{N \times p}$ with $p \ll N$.

F&S use an incomplete Cholesky factorization to find V. There are other possibilities:

- Arnoldi methods: `eigs` command in Matlab. Finds dominant eigenvectors / eigenvalues.
- Sampling: Nyström method (Drineas&Mahoney 2005). Nonuniform sample of the columns of K, reweight, find SVD.

Low-rank Approx + Active Set

If we simply use the low-rank approximation $K \leftarrow VV^T$, the dual formulation becomes:

$$\min_{\alpha} \frac{1}{2}\alpha^T VV^T\alpha - \mathbf{1}^T\alpha \qquad \text{s.t. } 0 \leq \alpha \leq C\mathbf{1}, \;\; y^T\alpha = 0,$$

which if we introduce $\gamma = V^T\alpha \in \mathbb{R}^p$, becomes

$$\min_{\alpha,\gamma} \frac{1}{2}\gamma^T\gamma - \mathbf{1}^T\alpha \qquad \text{s.t. } 0 \leq \alpha \leq C\mathbf{1}, \;\; \gamma = V^T\alpha, \;\; y^T\alpha = 0,$$

For small p, can solve this efficiently with an active-set QP code (e.g. CPLEX).

Solution is unique in γ, possibly nonunique in α, but can show that the classifier is invariant regardless of which particular α is used.

However limited testing shows that the quality of classifiers is not very good, for small p.

Dual SVM: Other Issues

The expense of calculating K and multiplying by it is a recurring theme, though savings are possible via caching and shrinking.

Another possibility: "Improved Fast Gauss Transformation" (Yang et al 2004) - best suited to short feature vectors x_i.

(Cao et al 2006) Parallel implementation of SMO is fairly straightforward, by distributing rows of K around the available processors. Each processor is responsible for maintaining a subvector of $K\alpha$.

(Catnzaro, Sundaram, Keutzer 2008) GPU implementation of SMO.

Solving the Primal

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i,$$

$$\text{subject to } \ \xi_i \geq 0, \ \ y_i(w^T x_i + b) \geq 1 - \xi_i, \ \ i = 1, 2, \ldots, N.$$

Motivation: Dual solution often not particularly sparse (many support vectors - particularly with a nonlinear kernel). Dual approaches can be slow when data set is very large.

Methods for primal formulations have been considered anew recently.

Limitation: Lose the kernel. Need to define the feature space "manually" and solve a linear SVM.

But see (Chapelle 2006) who essentially replaces feature vector x_i by $[k(x_j, x_i)]_{j=1,2,\ldots,N}$, and replaces $w^T w$ by $w^T K w$. (The techniques below could be applied to this formulation.)

Primal SVM: Cutting Plane

Formulate the primal as

$$\min_{w,b} P(w,b) := \frac{1}{2}\|w\|_2^2 + R(w,b),$$

where R is a piecewise linear function of (w,b):

$$R(w,b) = C\sum_{i=1}^{N} \max(1 - y_i(w^T x_i + b), 0).$$

Cutting-plane methods build up a piecewise-linear lower-bounding approximation to $R(w,b)$ based on a subgradient (possibly more than one) calculated at the latest iterate (w^k, b^k). This approach used in many other contexts.

In SVM, the subgradients are particularly easy to calculate.

(Joachims 2006) implemented as SVM$^{\mathrm{perf}}$. (Franc&Sonnenburg 2008) add line search and monotonicity. Convergence / complexity proved.

Primal SVM: Stochastic Gradient Descent

(Bottou) Take steps in the subgradient direction of a few-term approximation to $P(w, b)$, e.g. at iteration k, for some subset $I_k \subset \{1, 2, \ldots, N\}$, use subgradient of

$$P_k(w, b) := \frac{1}{2}\|w\|_2^2 + C\frac{N}{|I_k|}\sum_{i \in I_k} \max(1 - y_i(w^T x_i + b), 0),$$

evaluated at current iterate (w^k, b^k).

Step length η_k usually decreasing with k according to a fixed schedule.

Cheap if $|I_k|$ is small. Extreme case: I_k is a single index, selected randomly.

(Shalev-Shwartz, Singer, Srebro 2007). Pegasos: After subgradient step, project w onto a ball $\{w \mid \|w\|_2 \leq B\}$. Performance is insensitive to $|I_k|$.

Primal SVM: Subgradient+quasi-Newton

(Yu et al 2008)

- Maintain a quasi-Newton approximation B_k to the inverse "Hessian" of $P(w, b)$
- First guess of step at iteration k is $B_k g_k$, where g_k is from the subgradient of $P(w^k, b^k)$
- Adjust the step to ensure that it gives descent (requires an approximate bundle method - complicated!)
- Do a line search.

Performance similar to cutting-plane methods.

Alternative Formulations: L2-SVM

When ξ_i measures classifier error for point i, use ξ_i^2 rather than ξ_i in the objective:

$$\min_{w,\xi} \frac{1}{2}\|w\|_2^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2, \quad \text{s.t. } \xi_i \geq 0, \quad y_i w^T x_i \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N.$$

(No intercept term.) Eliminate ξ_i to get an unconstrained problem, with discontinuous second derivative:

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \frac{C}{2}\sum_{i=1}^{N}\max(0, 1 - y_i w^T x_i)^2.$$

(Mangasarian 2002; Keerthi&DeCoste 2005): Apply Newton's method with a "Hessian" drawn from the generalized Hessian. Do a line search.

Generalized Newton direction obtained from a regularized linear least-squares problem of the form

$$\min_{\beta} \frac{1}{2}\|Xw - y\|_2^2 + \frac{1}{2C}\|w\|_2^2,$$

where (X, y) contain an "active subset" of the data (x_i, y_i), $i = 1, 2, \ldots, N$. Can use iterative methods (LSQR) to get inexact solution.

Alternative Formulations: $\|w\|_1$.

Replacing $\|w\|_2^2$ by $\|w\|_1$ in the primal formulation gives a linear program (e.g. Mangasarian 2006; Fung&Mangasarian 2004, others):

$$\min_{w,b,\xi} \|w\|_1 + C \sum_{i=1}^{N} \xi_i,$$

$$\text{subject to } \xi_i \geq 0, \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N.$$

Sometimes called "1-norm linear SVM."

Tends to produce sparse vectors w; thus classifiers that depend on a small set of features.

($\|\cdot\|_1$ regularizer also used in other applications, e.g. compressed sensing).

Production LP solvers may not be useful for large data sets; the literature above describes specialized solvers.

Elastic Net

Idea from (Zou&Hastie 2005). Include both $\|w\|_1$ and $\|w\|_2$ terms in the objective:

$$\min_{w,\xi} \frac{\lambda_2}{2}\|w\|_2^2+\lambda_1\|w\|_1+\mathbf{1}^T\xi \text{ s.t. } \xi_i \geq 0, \quad y_i w^T x_i \geq 1-\xi_i, \quad i = 1, 2, \ldots, N.$$

In variable selection, combines ridge regression with LASSO. Good at "group selecting" (or not selecting) correlated w_i's jointly.

Has this been tried for SVM?

Logistic Regression

Seek functions $p_{-1}(x)$, $p_1(x)$ that define the odds of feature vector x having labels -1 and 1, respectively. Parametrize as

$$p_{-1}(x; w) = \frac{1}{1 + \exp w^T x}, \qquad p_1(x; w) = \frac{\exp w^T x}{1 + \exp w^T x}.$$

Given training data (x_i, y_i), $i = 1, 2, \ldots, N$, define log-likelihood:

$$\begin{aligned}\mathcal{L}(w) &= \frac{1}{2} \sum_{i=1}^{N} \left[(1 + y_i) \log p_1(x_i; w) + (1 - y_i) \log p_{-1}(x_i; w) \right] \\ &= \frac{1}{2} \sum_{i=1}^{N} \left[(1 + y_i) \exp w^T x_i - 2 \log(1 + \exp w^T x_i) \right].\end{aligned}$$

Add regularization term $\lambda \|w\|_1$ and solve

$$\min_w \; T_\lambda(w) := -\mathcal{L}(w) + \lambda \|w\|_1.$$

(Shi et al. 2008) Use a *proximal regularized* approach: Given iterate w^k get new iterate z by solving a subproblem with simplified smooth term:

$$\min_z \nabla \mathcal{L}(w^k)^T (z - w^k) + \frac{\alpha_k}{2} \|z - w^k\|_2^2 + \lambda \|z\|_1.$$

Analogous to gradient projection, with $1/\alpha_k$ as line search parameter. Choose α_k large enough to give reduction in T_λ.

For problems with very sparse w (typical), enhance by taking a reduced Newton-like step for $\mathcal{L}$ in the currently-nonzero components only.

Enhancements: Evaluate a random selection of components of $\nabla \mathcal{L}$ (save expense of a full evaluation - like shrinking). Use continuation in λ.

Could also use a nonmonotone algorithm, choosing α_k with a Barzilai-Borwein formula.

(Shi et al. 2008) use long feature vectors x whose components are genetic alleles and their interactions, and label is chance of rheumatoid arthritis.

New Tool: Optimal Gradient Methods

(Nesterov, Nemirovskii, Yudin 1983-2008) Minimize a smooth convex function f, using only information about f and ∇f, to optimize long-term performance.

Make use of parameters L — Lipschitz constant for ∇f — and μ — convexity parameter for f:

$$f(y) \geq f(\bar{x}) + \nabla f(\bar{x})^T(y - \bar{x}) + \frac{\mu}{2}\|y - \bar{x}\|_2^2$$

(possibly zero). (Can use estimates of these instead.)

Look to prove things about convergence of objective values $f(x_k) - f^*$ as well as iterates $\|x_k - x^*\|$.

Basic Gradient Methods

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

gives this estimate for convergence of the objective values:

$$f(x_k) - f^* \leq \frac{2L}{k+4}\|x_0 - x^*\|^2 \sim \frac{1}{k},$$

which is sublinear.

Strongly convex case, a better choice is

$$x_{k+1} = x_k - \frac{2}{L+\mu}\nabla f(x_k)$$

gives this estimate:

$$f(x_k) - f^* \leq \frac{L}{2}\left(\frac{L-\mu}{L+\mu}\right)^{2k}\|x_0 - x^*\|^2$$

which is linear (geometric).

Optimal Rates

- Generate two or three sequences of iterates
- Evaluate gradient once at each iteration
- Possibly save gradient information from previous iterations.

For strongly convex case, can obtain faster (geometric) rates:

$$f(x_k) - f^* \leq C(x_0, L, \mu) \left(1 - \sqrt{\frac{\mu}{L}}\right)^k.$$

When $\mu = 0$, can obtain:

$$f(x_k) - f^* \leq C(x_0, L) \frac{1}{k^2}.$$

Typical Schemes

Strongly convex ($\mu > 0$): Start with $y_0 = x_0$ and generate:

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k); \tag{1}$$

$$y_{k+1} = x_{k+1} + \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}(x_{k+1} - x_k). \tag{2}$$

Weakly convex: more complicated linear combination. x step is still (1), but y step is

$$y_{k+1} = x_{k+1} + \beta_k(x_{k+1} - x_k),$$

where

$$\beta_k = \alpha_k(1 - \alpha_k)/(\alpha_k^2 + \alpha_{k+1}),$$
$$\alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 + q\alpha_{k+1}$$

with $q = \mu/L$, $y_0 = x_0$, and some $\alpha_0 \in (0, 1)$.

Extensions

There are variants for convex constrained problems

$$\min_{x \in \Omega} f(x)$$

regularized optimization (with simple nonsmooth regularizers):

$$\min_{x} f(x) + \lambda P(x),$$

nonsmooth problems (see next slide) and min-max problems (Nemirovski 2005).

These methods have had some recent computational success in compressed sensing and other applications.

The nonsmooth variant might be useful for primal SVM...

Nesterov: Smoothing Nonsmooth Problems

(Nesterov 2005) Many convex functions f can be expressed as

$$f(x) := \max_{u \in \mathcal{Q}} u^T(Ax - b) - \phi(u)$$

for some convex set $\mathcal{Q}$ and convex function ϕ. Define a smoothed version $f_\mu(x)$ (for $\mu > 0$) by choosing a strongly convex d (with strong convexity parameter 1) and setting

$$f_\mu(x) := \max_{u \in \mathcal{Q}} u^T(Ax-b)-\phi(u)-\mu d(u), \qquad \text{Gradient:} \quad \nabla f_\mu(x) = A^T u_\mu(x),$$

where u_μ is the arg max in the smoothed evaluation.

Now apply optimal gradient methods to f_μ.

Given an iteration budget N, fix $\mu \sim N^{-1}$, run N iterations of an optimal method on the smoothed problem to obtain objective accuracy $O(N^{-1})$.

New Tool: Primal-Dual Gradient Projection

Consider a saddle point (min-max) problem

$$\min_{v \in V} \max_{x \in X} \ell(v, x),$$

with $\ell(\cdot, x)$ convex for all $x \in X$ and $\ell(v, \cdot)$ concave for all $v \in V$, and V and X are convex sets.

(When ℓ is quadratic and V, X are polyhedral, this is Rockafellar's ELQP.)

Many convex optimization problems can be formulated naturally in this framework.

Primal-dual gradient projection procedure is

$$x^{k+1} \leftarrow P_X(x^k + \tau_k \nabla_x \ell(v^k, x^k)),$$
$$v^{k+1} \leftarrow P_V(v^k - \sigma_k \nabla_v \ell(v^k, x^{k+1})),$$

where τ_k and σ_k are positive steplengths.

We'd get standard gradient projection on the "primal" problem of maximizing the function $q(x) := \min_{v \in V} \ell(v, x)$ over $x \in X$ if we were to replace the gradient step in v by

$$v^{k+1} = \arg\min_{v \in V} \ell(v, x^{k+1}).$$

However, it's sometimes faster to use the "inexact" v^{k+1}. Why?

Application to image denoising:

$$\ell(v, x) = v^T A x + \frac{\lambda}{2} \|v - y\|_2^2,$$

while $V = \mathbb{R}^N$ and X is a simple bounded set (Cartesian product of circles). A is sparse (discretized difference operator).

Exhaustive tests show that non-intuitive choice of step length works best:

$$\tau_k = (.2 + .08k)\lambda = O(k), \qquad \sigma_k \approx \frac{1}{2\tau_k} = O(k^{-1}).$$

(Because of the projection onto X, steps in x are short, despite $\tau_k \to \infty$.)

- There are many ways to view this approach, but none has yet yielded an understanding of its excellent practical performance.
- Does its usefulness extend beyond image reconstruction, e.g. to SVM formulations?

THE END