Optimization Algorithms in Support Vector Machines

Stephen Wright

University of Wisconsin-Madison

Computational Learning Workshop, Chicago, June 2009

Summary

Joint investigations with Sangkyun Lee (UW-Madison).

1. Sparse and Regularized Optimization: context
2. SVM Formulations and Algorithms
   - Oldies but goodies
   - Recently proposed methods
   - Possibly useful recent contributions in optimization, including applications in learning.
   - Extensions and future lines of investigation.

Focus on fundamental formulations. These have been studied hard over the past 12-15 years, but it's worth checking for "unturned stones."

Themes

- Optimization problems from machine learning are difficult!
  - number of variables, size/density of kernel matrix, ill conditioning, expense of function evaluation.
- Machine learning community has made excellent use of optimization technology.
  - Many interesting adaptations of fundamental optimization algorithms that exploit the structure and fit the requirements of the application.
- New formulations present new challenges.
  - example: semi-supervised learning requires combinatorial / nonconvex / global optimization techniques.
- Several current topics in optimization may be applicable to machine learning problems.

Sparse / Regularized Optimization

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

However, in many optimization applications we prefer **less complex, approximate** solutions.

- simple solutions easier to actuate;
- uncertain data does not justify precise solutions; regularized solutions less sensitive to inaccuracies;
- a simple solution is more "generalizable" — avoids overfitting of empirical data;
- Occam's Razor.

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 norm 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, chosen so that there is a sparse x (few nonzeros) satisfying $Ax \approx b$.

Typically A has more columns than rows, has special properties (e.g. restricted isometry) 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 $Ax = b$.

Use λ to control sparsity of the recovered solution.

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.)

$$\min_u \frac{1}{2} \int_\Omega (u - f)^2 \, dx + \lambda \int_\Omega |\nabla u| \, dx.$$

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

λ controls fidelity to image data.

First-order methods on dual or primal-dual are much faster at recovering approximate solutions than methods with fast asymptotic convergence. (More later.)

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 believed 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.
- However, some approaches can be "abstracted" across applications, and their properties can be analyzed at a higher level.
- Duality if 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, quasi-Newton) with expensive steps and

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

The latter may be 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):

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

Unconstrained piecewise quadratic. Also can be written as a convex QP.

Dual

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, 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 typically lead to K dense and 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 the kernel form).

Many are based on optimization methods, or can be interpreted using tools from the analysis of optimization algorithms.

Methods compared via a variety of metrics:

- CPU time to find solution of given quality (e.g. error rate).
- Theoretical efficiency.
- Data storage requirements.
- (Simplicity.) (Parallelizability.)

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!

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.

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:

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

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

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

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.

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 calculated at the latest iterate (w^k, b^k). This approach used in many other contexts, e.g. stochastic linear programming with recourse.

In SVM, the subgradients are particularly easy to calculate.

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

Modifications tried (Lee and Wright) by modifying OCAS code:

- partition the sum $R(w, b)$ into p bundles, with cuts generated separately for each bundle. Gives a richer approximation, at the cost of a harder subproblem.
- different heuristics for adding cuts after an unsuccessful step.

Many more ideas could be tried. In the basic methods, each iteration requires computation of the full set of inner products $w^T x_i$, $i = 1, 2, \ldots, N$. Could use strategies like partial pricing in linear programming to economize.

Primal SVM: Stochastic Subgradient

(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

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

Step length η_k usually decreasing with k according to a fixed schedule. Can use rules $\eta_k \sim k^{-1}$ or $\eta_k \sim k^{-1/2}$.

Cheap if $|I_k|$ is small. Extreme case: I_k is a single index, selected randomly. Typical step: Select $j(k) \in \{1, 2, \ldots . N\}$ and set

$$(w^{k+1}, b^{k+1}) \leftarrow (w^k, b^k) - \eta_k g_k,$$

where

$$g_k = \begin{cases} (w, 0) & \text{if } 1 - y_{j(k)}(w^T x_{j(k)} + b) \leq 0, \\ (w, 0) - CNy_{j(k)}(x_{j(k)}, 1) & \text{otherwise.} \end{cases}$$

Stochastic Subgradient

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

Convergence: Roughly, for steplenths $\eta_k = CN/k$, have for fixed total iteration count T and k randomly selected from $\{1, 2, \ldots, T\}$, the expected value of the objective f is within $O(T^{-1} \log T)$ of optimal.

Similar algorithms proposed in (Zhang 2004), (Kivinen, Smola, Williamson 2002) - the latter with a steplength rule of $\eta_k \sim k^{-1/2}$ that yields an expected objective error of $O(T^{-1/2})$ after T iterations.

There's a whole vein of optimization literature that's relevant — Russian in origin, but undergoing a strong revival. One important and immediately relevant contribution is (Nemirovski et al. 2009).

Stochastic Approximation Viewpoint

(Nemirovski et al, SIAM J Optimization 2009) consider the setup

$$\min_{x \in X} f(x) := E_\zeta[F(x, \zeta)],$$

where subgradient estimates $G(x, \zeta)$ are available such that $g(x) := E_\zeta[G(x, \zeta)]$ is a subgradient of f at x. Steps:

$$x^{k+1} \leftarrow P_X(x^k - \eta_k G(x^k, \zeta^k))$$

where ζ^k selected randomly. Some conclusions:

- If f is convex with modulus γ, steplengths $\eta_k = (\gamma k)^{-1}$ yield $E(f(x^k) - f(x^*)] = O(1/k)$.
- Slight differences to the stepsize (e.g. a different constant multiple) can greatly degrade performance.
- If f is convex (maybe weakly), the use of stepsizes $\eta_k \sim k^{-1/2}$ yields convergence at rate $k^{-1/2}$ of a weighted average of iterates in expected function value.
- This is a slower rate, but much less sensitive to the "incorrect" choices of steplength scaling. See this in practice.

Primal-Dual Approaches

Method that solve primal and dual simultaneously by alternating between first-order steps in primal and dual space are proving useful in some apps.

Example: TV Denoising. Given a domain $\Omega \subset \mathbb{R}^2$ and an observed image $f : \Omega \to \mathbb{R}$, seek a restored image $u : \Omega \to \mathbb{R}$ that preserves edges while removing noise.

$$\text{Primal:} \quad \min_u P(u) := \int_\Omega |\nabla u| \, dx + \frac{\lambda}{2} \|u - f\|_2^2.$$

$$\text{Dual:} \quad \max_{w \in C_0^1(\Omega), |w| \leq 1} D(w) := \frac{\lambda}{2} \left[\|f\|_2^2 - \left\| \frac{1}{\lambda} \nabla \cdot w + f \right\|_2^2 \right].$$

Discretized TV Denoising

After regular discretization, obtain a primal-dual pair:

$$\min_v \sum_{l=1}^{N} \|A_l^T v\|_2 + \frac{\lambda}{2}\|v - g\|_2^2,$$

where A_l is an $N \times 2$ matrix with at most 4 nonzero entries ($+1$ or -1).

$$\min_{x \in X} \frac{1}{2}\|Ax - \lambda g\|_2^2$$

$$\text{where} \quad X := \{(x_1; x_2; \ldots; x_N) \in \mathbb{R}^{2N} : x_l \in \mathbb{R}^2, \\ \|x_l\|_2 \leq 1 \text{ for all } l = 1, 2, \ldots, N\},$$

where $A = [A_1, A_2, \ldots, A_N] \in \mathbb{R}^{N \times 2N}$.

First-order method on the dual is quite effective for low-moderate accuracy solutions (Zhu, Wright, Chan 2008). Many other methods proposed: second-order, PDE-based, second-order cone.

Min-Max Formulation

The discrete primal-dual solution (v, x) is a saddle point of

$$\min_{v} \max_{x \in X} \ell(v, x) := x^T A^T v + \frac{\lambda}{2} \|v - g\|_2^2.$$

(Zhu, Chan 2008) solve this with a first-order primal-dual approach:

$$x^{k+1} \leftarrow P_X(x^k + \tau_k \nabla_x \ell(x^k, v^k)) \quad (1)$$

$$v^{k+1} \leftarrow v^k - \sigma_k \nabla_v \ell(x^{k+1}, v^k), \quad (2)$$

for some positive steplengths τ_k, σ_k. They found that this (non-intuitive) choice of steplengths worked well:

$$\tau_k = (.2 + .08k)\lambda, \quad \sigma_k = .5/\tau_k.$$

Why??

PD Method for Semiparametric SVM Regression

(Smola et al, 1999) Add:

- regression (rather than classification) with an ϵ-insensitive margin.
- basis functions $\psi_j(x)$, $j = 1, 2, \ldots, K$, making the regression function partly parametric.

$$\min_{w,\zeta,\zeta^*,\beta} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \max\{0, |y_i - h(x_i; w, \beta)| - \epsilon\},$$

where

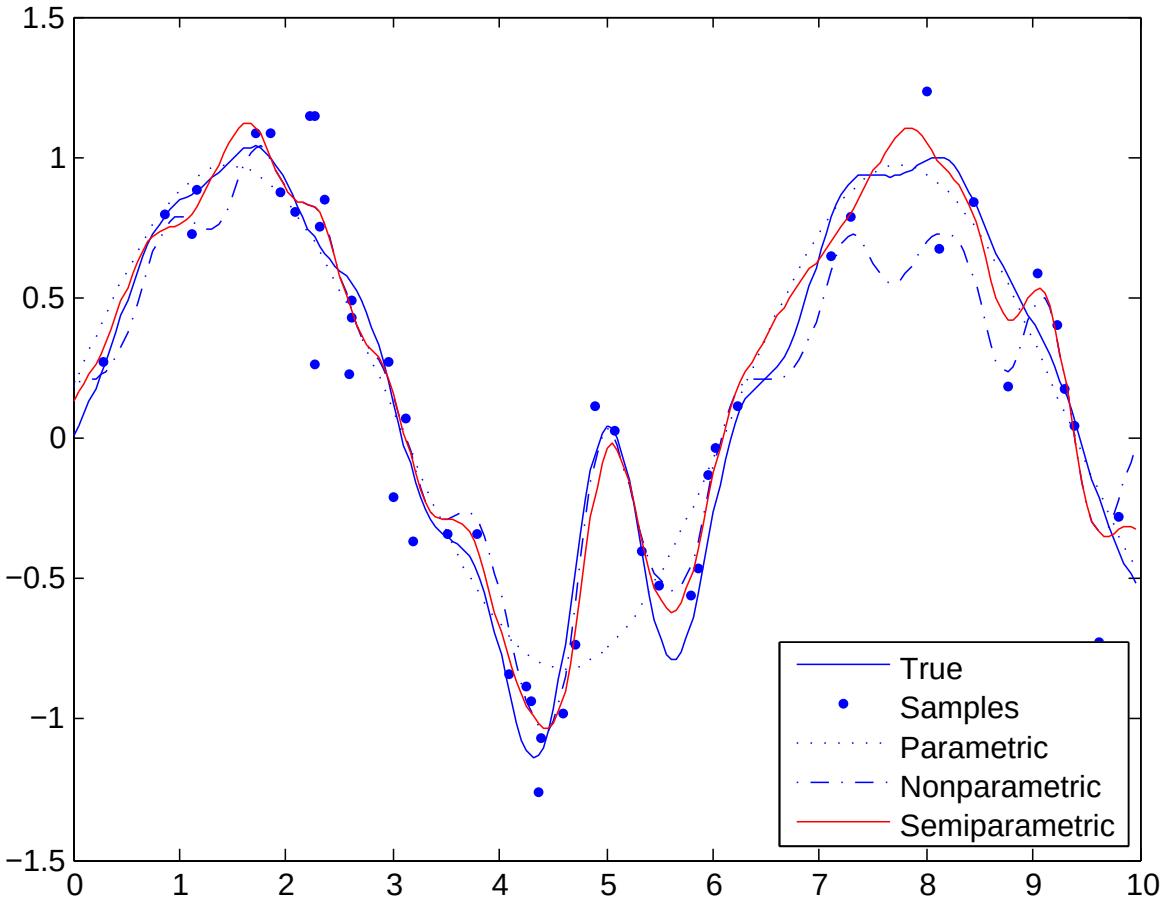
$$h(x; w, \beta) := w^T \phi(x) + \sum_{j=1}^{K} \beta_j \psi_j(x).$$

Dual can be formulated with $2N$ variables as follows:

$$\min_{\tilde{\alpha}} f(\tilde{\alpha}) := \frac{1}{2} \tilde{\alpha}^T \tilde{K} \tilde{\alpha} + p^T \tilde{\alpha} \text{ s.t. } A\tilde{\alpha} = b, \ 0 \leq \tilde{\alpha} \leq C\mathbf{1}.$$

where $\tilde{K}$ is an extended kernel, still usually positive semidefinite, and A is $K \times 2N$.

Example (Smola et al, 1999): Learn the function $f(x) = \sin x + \mathrm{sinc}(2\pi(x-5))$ from noisy samples $x_i \in [0, 10]$. Use 3 basis functions 1, $\sin x$, $\cos x$ in the parametric part, Gaussian kernel for nonparametric part.



(Kienzle and Schölkopf, 2005) minimal primal-dual (MPD); (Lee and Wright 2009) primal-dual with scaled gradient (PDSG) and decomposition.

Define

$$L(\tilde{\alpha}, \eta) := f(\tilde{\alpha}) + \eta^T (Ax - b),$$

then can formulate as a saddle-point problem:

$$\max_{\eta} \min_{0 \leq \tilde{\alpha} \leq C\mathbf{1}} L(\tilde{\alpha}, \eta).$$

PDSG alternates between steps in

- a subset of $\tilde{\alpha}$ components (decomposition) - using a gradient projection search direction
- η - a Newton-like step in the dual function $g(\eta) := \min_{0 \leq \tilde{\alpha} \leq C\mathbf{1}} L(\tilde{\alpha}, \eta)$.

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):

$$\min_{w,b,\xi} \|w\|_1 + C \sum_{i=1}^{N} \max(1 - y_i(w^T x_i + b), 0).$$

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:

$$\min_{w,\xi} \frac{\lambda_2}{2}\|w\|_2^2 + \lambda_1\|w\|_1 + \sum_{i=1}^{N} \max(1 - y_i(w^T x_i + b), 0).$$

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

Is this useful for SVM?

It would be easy to extend some of the techniques discussed earlier to handle this formulation.

SpaRSA

An extremely simple approach introduced in context of compressed sensing (Wright, Figueiredo, Nowak 2008) can be applied more generally, e.g. to logistic regression. Given formulation

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

and current iterate x^k, find new iterate by choosing scalar α_k and solving

$$\min_z \frac{1}{2\alpha_k}(z - x^k)^T(z - x^k) + \nabla \mathcal{F}(x^k)^T(z - x^k) + \lambda \mathcal{R}(z).$$

Possibly adjust α_k to get descent in the objective, then set $x^{k+1} \leftarrow z$.

- Form a quadratic model of $\mathcal{F}$ around x^k, correct to first order, with simple Hessian approximation $1/\alpha_k$.
- Variants: Barzilai-Borwein, nonmonotonic.
- Useful when the subproblem is cheap to solve.
- **Continuation** strategy useful in solving for a range of λ values (largest to smallest). Use solution for one λ as warm start for the next smaller value.

When $\mathcal{R} = \|\cdot\|_1$ (standard compressed sensing), can solve subproblem in $O(n)$ (closed form).

Still cheap when

$$\mathcal{R}(x) = \sum_l \|x_{[l]}\|_2, \quad \mathcal{R}(x) = \sum_l \|x_{[l]}\|_\infty$$

where $x_{[l]}$ are disjoint subvectors. (Group LASSO.)

Not so clear how to solve the subproblems cheaply when

- subvectors $x_{[l]}$ are not disjoint in the group-lasso formulation
- regularized $\mathcal{R}$ chosen to promote a hierarchical relationship between components of x
- $\mathcal{R}(x)$ is a TV-norm.

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.

Enhancements:

- For problems with very sparse w (typical), take a reduced Newton-like step for $\mathcal{L}$ in the currently-nonzero components only.
- Evaluate a random selection of components of $\nabla \mathcal{L}$ (save expense of a full evaluation - like shrinking).
- Use continuation in λ: solution for one value of λ used to warm-start a the next smaller value in the sequence.

References

Compressed Sensing.

1. Candès, E., "Compressive Sampling," Proceedings of the International Congress of Mathematicians, Madrid, 2006.
2. `www.compressedsensing.com` (maintained at Rice University)
3. `http://nuit-blanche.blogspot.com/` (blog with news on compressed sensing).
4. Candès, E., Romberg, J., and Tao. T., "Signal recovery from incomplete and inaccurate information," Communications in Pure and Applied Mathematics 59 (2005), pp. 1207–1233.
5. Lee, S. and Wright, S. J., "Implementing algorithms for signal and image processing on graphical processing units," Optimization Technical Report, Commputer Sciences Department, University of Wisconsin-Madison, October 2008.
6. Wright, S. J., Nowak, R., and Figueiredo, M., "Sparse Reconstruction for Separable Approximation," Technical Report, University of Wisconsin-Madison, 2008.

Image Denoising (See the following and references therein):

1. Zhu, M., Wright, S. J., and Chan, T., "Duality-based algorithms for total varition image restoration," Computational Optimization and Applications, to appear (2009).
2. Zhu, M. and Chan, T., "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," CAM Report 08-34, Mathematics Departmetn, UCLA, 2008.

Matrix Completion:

1. Recht, B., Fazel, M., and Parrilo, P., "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm mininization," Technical Report, California Institute of Technology, 2008.
2. Candès, E. and Recht, B., "Exact matrix completion via convex optimization," Technical Report, California Institute of Technology, 2008.
3. Cai, J. F., Candès, E., and Shen, Z., "A singular value thresholding algorithm for matrix completion," Technical Report, California Institute of Technology, 2008.

Dual SVM (papers cited in talk):

1. Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S., "A Dual coordinate descent method for large-scale linear SVM", Proceedings of teh 25th ICML, Helsinki, 2008.
2. Dai, Y.-H. and Fletcher, R., "New algorithms for singly linearly consrtained quadratic programming subject to lower and upper bounds," Mathematical Programming, Series A, 106 (2006), pp. 403–421.
3. Platt, J., "Fast training of support vector machines using sequential minimal optimization," Chapter 12 in *Advances in Kernel Methods* (B. Schölkopf, C. Burges, and A. Smola, eds), MIT Press, Cambridge, 1998.
4. Joachims, T., "Making large-scale SVM learning practical," Chapter 11 in *Advances in Kernel Methods* (B. Schölkopf, C. Burges, and A. Smola, eds), MIT Press, Cambridge, 1998.
5. Zanni, L., Serafini, T., and Zanghirati, G., "Parallel software for training large-scale support vector machines on multiprocessor systems," JMLR 7 (206), pp. 1467–1492.
6. Fan, R.-E., Chen, P.-H., and Lin, C.-J., "Working set selection using second-order information for training SVM," JMLR 6 (2005), pp. 1889–1918.
7. Chen, P.-H., Fan, R.-E., and Lin, C.-J., "A study on SMO-type decomposition methods for support vector machines," IEEE Transactions on Neural Networks 17 (2006), pp. 892–908.
8. Scheinberg, K. "An efficient implementation of an active-set method for SVMs," JMLR 7 (2006), pp. 2237-2257.
9. Shilton, A., Palaniswami, M., Ralph, D., and Tsoi, A. C., "Incremental training of support vector machines," IEEE Transactions on Neural Networks 16 (2005), pp. 114–131.
10. Fine, S., and Scheinberg, K., "Efficient SVM training using low-rank kernel representations," JMLR 2 (2001), pp. 243–264.
11. Drineas, P. and Mahoney, M., "On the Nystrom method for approximating a Gram matrix for improved kernel-based learning," JMLR 6 (2005), pp. 2153–2175.
12. Yang, C., Duraiswami, R., and Davis, L., "Efficient kernel machines using the improved fast Gauss transform," 2004.
13. Cao, L. J., Keerthi, S. S., Ong, C.-J., Zhang, J. Q., Periyathamby, U., Fu, X. J., and Lee, H. P., "Parallel sequential minimal optmization for the training of support vector machines,"IEEE Transactions on Neural Networks 17 (2006), pp. 1039–1049.
14. Catanzaro, B., Sundaram, N., and Keutzer, K., "Fast support vector machine training and classification on graphics processors," Proceedings of the 25th ICML, Helsinki, Finland, 2008.

Primal SVM papers (cited in talk):

1. Chappelle, O., "Training a support vector machine in the primal," Neural Computation, in press (2008).
2. Shalev-Schwartz, S., Singer, Y., and Srebro, N., "Pegasos: Primal estimated sub-gradient solver for SVM," Proceedings of the 24th ICML, Corvallis, Oregon, 2007.
3. Joachims, T., "Training linear SVMs in linear time," KDD '06, 2006.
4. Franc, V., and Sonnenburg, S., "Optimized cutting plane algorithm for support vector machines," in Proceedings of the 25th ICML, Helsinki, 2008.
5. Bottou, L. and Bousquet, O., "The tradeoffs of large-scale learning," in *Advances in Neural Information Processing Systems* 20 (2008), pp. 161–168.
6. Yu, J., Vishwanathan, S. V. N., Günter, S., and Schraudolph, N. N., "A quasi-Newton approach to nonsmooth convex optimization," in Proceedings of the 25th ICML, Helsinki, 2008.
7. Keerthi, S. S. and DeCoste, D., "A Modified finite Newton method for fast solution of large-scale linear SVMs," JMLR 6 (2005), pp. 341–361.
8. Mangasarian, O. L., "A finite Newton method for classification," Optimization Methods and Software 17 (2002), pp. 913–929.
9. Mangasarian, O. L., "Exact 1-norm support vector machines via unconstrained convex differentiable minimization," JMLR 7 (2006), 1517–1530.
10. Fung, G. and Mangasarian, O. L., "A Feature selection Newton method for support vector machine classification," Computational Optimization and Applications 28 (2004), pp. 185–202.
11. Zou, H. and Hastie, T., "Regularization and variable selection via the elastic net," J. R¿ Statist. Soc. B 67 (2005), pp. 301–320.

Logistic Regression:

1. Shi, W., Wahba, G., Wright, S. J., lee, K., Klein, R., and Klein, B., "LASSO-Patternsearch algorithm with application to ophthalmology data," Statistics and its Interface 1 (2008), pp. 137–153.

Optimal Gradient Methods:

1. Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A., "Robust stochastic approximation approach to stochastic programming," SIAM J. Optimization 19 (2009), pp. 1574–1609.
2. Nesterov, Y., "A method for unconstrained convex problem with the rate of convergence $O(1/k^2)$," Doklady AN SSR 269 (1983), pp. 543–547.
3. Nesterov, Y., *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, 2004.
4. Nesterov. Y., "Smooth minimization of nonsmooth functions," Mathematical Programming A 103 (2005), pp. 127–152.
5. Nemirovski, A., "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," SIAM J. Optimization 15 (2005), pp. 229–251.