Advanced Topics in Machine Learning

1. Learning SVMs / Primal Methods

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany

Outline

Outline

9. Subgradient Descent in the Primal

10. Linearization of Nonlinear Kernels

Subgradient Descent

$$\text{minimize } f(\beta, \beta_0; D) := \frac{1}{|D|} \sum_{(x,y)\in D} [1 - y(\beta^T x + \beta_0)]_+ + \frac{1}{2}\lambda||\beta||^2$$

$$\text{subgradient } g(\beta, \beta_0; D) := \begin{pmatrix} -\frac{1}{|D|} \sum_{\substack{(x,y)\in D \\ y(\beta^T x+\beta_0)<1}} yx + \lambda\beta \\ -\frac{1}{|D|} \sum_{\substack{(x,y)\in D \\ y(\beta^T x+\beta_0)<1}} y \end{pmatrix}$$

Subgradient Descent

learn-linear-svm-subgradient-descent-primal(training predictors x, training targets y,
regularization λ, accuracy ϵ,
step lengths η_t) :

$n := |x|$

$\hat{\beta} := 0$

$\hat{\beta}_0 := 0$

$t := 0$

do

$$\Delta\hat{\beta} := -\frac{1}{n} \sum_{\substack{i=1 \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i x_i$$

$$\Delta\hat{\beta}_0 := -\frac{1}{n} \sum_{\substack{i=1 \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i$$

$\quad \hat{\beta} := (1 - \eta_t \lambda)\hat{\beta} - \eta_t \Delta\hat{\beta}$

$\quad \hat{\beta}_0 := \hat{\beta}_0 - \eta_t \Delta\hat{\beta}_0$

$\quad t := t + 1$

while $\eta_t ||\Delta\hat{\beta}|| \geq \epsilon$

return $(\hat{\beta}, \hat{\beta}_0)$

Subgradient Descent (subsample approximation)

Idea:
Do not use all training examples to estimate the error and the gradient, but just a subsample

$$D^{(t)} \subseteq D$$

The subsample may vary over steps t.

Then approximate $f(\cdot; D)$ by $f(\cdot; D^{(t)})$ in step t.

Extremes:

- all samples.
  (subgradient descent)
- just a single (random) sample.
  (stochastic subgradient descent)

Stochastic Subgradient Descent

learn-linear-svm-stochastic-subgradient-descent-primal(training predictors x, training targets y,
regularization λ, accuracy ϵ,
step lengths η_t, stop count t_0) :

$n := |x|$

$\hat{\beta} := 0$

$\hat{\beta}_0 := 0$

$t := 0$

$l^{t'} := 0, \quad t' = 0, \ldots t_0 - 1$

do

    draw i randomly from $\{1, \ldots, n\}$

    $\Delta\hat{\beta} := -\delta_{y_i(\beta^T x_i + \beta_0) < 1} y_i x_i$

    $\Delta\hat{\beta}_0 := -\delta_{y_i(\beta^T x_i + \beta_0) < 1} y_i$

    $\hat{\beta} := (1 - \eta_t \lambda)\hat{\beta} - \eta_t \Delta\hat{\beta}$

    $\hat{\beta}_0 := \hat{\beta}_0 - \eta_t \Delta\hat{\beta}_0$

    $l^{t \bmod t_0} := \eta_t ||\Delta\hat{\beta}||$

    $t := t + 1$

while $\sum_{t'=0}^{t_0-1} l^{t'} \geq \epsilon$

return $(\hat{\beta}, \hat{\beta}_0)$

Subgradient Descent with Subsample Approximation

learn-linear-svm-approx-subgradient-descent-primal(training predictors x, training targets y,
regularization λ, accuracy ϵ,
step lengths η_t, stop count t_0,
subsample size k) :

$n := |x|$

$\hat\beta := 0$

$\hat\beta_0 := 0$

$t := 0$

$l^{t'} := 0, \quad t' = 0, \dots t_0 - 1$

do

  draw subset I randomly from $\{1, \dots, n\}$ with $|I| = k$

$$\Delta\hat\beta := -\frac{1}{k} \sum_{\substack{i \in I \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i x_i$$

$$\Delta\hat\beta_0 := -\frac{1}{k} \sum_{\substack{i \in I \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i$$

  $\hat\beta := (1 - \eta_t \lambda)\hat\beta - \eta_t \Delta\hat\beta$

  $\hat\beta_0 := \hat\beta_0 - \eta_t \Delta\hat\beta_0$

  $l^{t \bmod t_0} := \eta_t ||\Delta\hat\beta||$

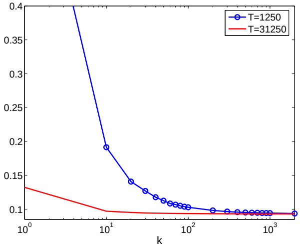
  $t := t + 1$

while $\sum_{t'=0}^{t_0-1} l^{t'} \geq \epsilon$

Subgradient Descent (subsample approximation)

Shalev-Shwartz, Singer, and Srebro 2007 experimented with approximations by samples of fixed size k, i.e.,

$$|D^{(t)}| = k, \quad \forall t$$

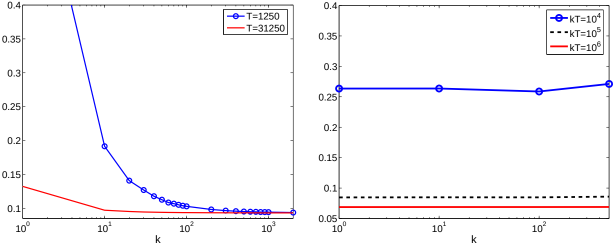


[Shalev-Shwartz, Singer, and Srebro 2007]

Subgradient Descent (subsample approximation)

Shalev-Shwartz, Singer, and Srebro 2007 experimented with approximations by samples of fixed size k, i.e.,

$$|D^{(t)}| = k, \quad \forall t$$



[Shalev-Shwartz, Singer, and Srebro 2007]

Maintaining Small Parameters

Lemma (Shalev-Shwartz, Singer, and Srebro 2007)

The optimal β^* *satisfies*

$$||\beta^*|| \leq \frac{1}{\sqrt{\lambda}}$$

Proof.

Due to strong duality for the optimal β^*, β_0^*:

$$\begin{aligned} f(\beta^*) =& \frac{1}{|D|} \sum_{(x,y)\in D} [1 - y(\beta^{*T} x + \beta_0^*)]_+ + \frac{1}{2}\lambda ||\beta^*||^2 \\ \overset{!}{=} \bar{f}(\alpha^*) =& -\frac{1}{2\lambda}\alpha^{*T}(XX^T \odot yy^T)\alpha^* + \frac{1}{|D|}||\alpha^*||_1 \end{aligned}$$

and with $\beta^* = \frac{1}{\lambda} X^T(y \odot \alpha^*)$

Maintaining Small Parameters

Proof (ctd.).

$$\frac{1}{2}\lambda||\beta^*||^2+\frac{1}{|D|}\sum_{(x,y)\in D}[1-y(\beta^{*T}x+\beta_0^*)]_+ = -\frac{1}{2}\lambda||\beta^*||^2+\frac{1}{|D|}||\alpha^*||_1$$

$$\lambda||\beta^*||^2 = \frac{1}{|D|}||\alpha^*||_1 - \frac{1}{|D|}\sum_{(x,y)\in D}[1-y(\beta^{*T}x+\beta_0^*)]_+$$

$$\leq \frac{1}{|D|}||\alpha^*||_1 \quad \text{and with } 0 \leq \alpha^* \leq 1:$$

$$\leq 1$$

$$\rightsquigarrow ||\beta^*|| \leq \frac{1}{\sqrt{\lambda}}$$

Primal Estimated subgradient solver for SVM (PEGASOS)

Basic ideas:

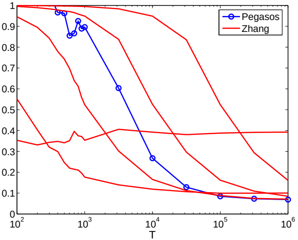
- use subsample approximation with fixed k
  (but $k = 1$, stochastic gradient descent, turns out to be optimal)
- retain $\beta \leq 1/\sqrt{\lambda}$ by rescaling in each step:

$$\beta := \frac{\beta}{\max(1, \sqrt{\lambda}||\beta||)}$$

- Decrease step size over time:

$$\eta_t := \frac{1}{\lambda t}$$

Decrease Step Size Over Time



[Shalev-Shwartz, Singer, and Srebro 2007]

Pegasos

learn-linear-svm-pegasos(training predictors x, training targets y,
    regularization λ, accuracy ϵ,
    stop count t_0, subsample size k) :

$n := |x|$

$\hat{\beta} := 0$

$\hat{\beta}_0 := 0$

$t := 0$

$l^{t'} := 0, \quad t' = 0, \ldots t_0 - 1$

do

  draw subset I randomly from $\{1, \ldots, n\}$ with $|I| = k$

$$\Delta\hat{\beta} := -\frac{1}{k} \sum_{\substack{i \in I \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i x_i$$

$$\Delta\hat{\beta}_0 := -\frac{1}{k} \sum_{\substack{i \in I \\ y_i(\beta^T x_i + \beta_0) < 1}}^{n} y_i$$

  $\eta_t := 1/(\lambda t)$

  $\hat{\beta} := (1 - \eta_t \lambda)\hat{\beta} - \eta_t \Delta\hat{\beta}$

  $\hat{\beta}_0 := \hat{\beta}_0 - \eta_t \Delta\hat{\beta}_0$

  $\hat{\beta} := \hat{\beta} / \max(1, \sqrt{\lambda}||\beta||)$

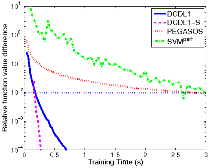
  $l^{t \bmod t_0} := \eta_t ||\Delta\hat{\beta}||$

  $t := t + 1$

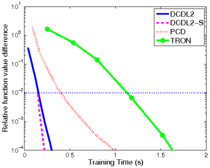
while $\sum_{t'=0}^{t_0-1} l^{t'} \geq \epsilon$

return $(\hat{\beta}, \hat{\beta}_0)$

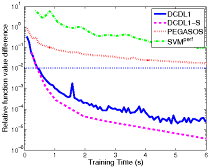
Comparison Dual Coordinate Descent vs. Pegasos



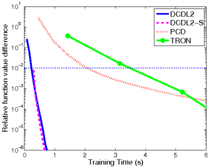
(a) L1-SVM: astro-physic



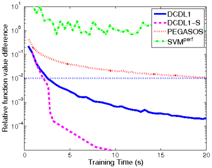
(b) L2-SVM: astro-physic



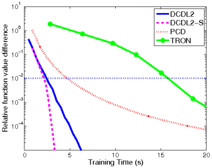
(c) L1-SVM: news20



(d) L2-SVM: news20



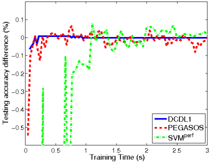
(e) L1-SVM: rcv1



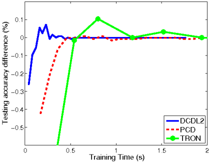
(f) L2-SVM: rcv1

[C. J. Hsieh et al. 2008]

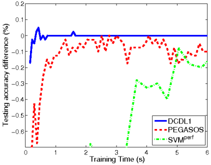
Comparison Dual Coordinate Descent vs. Pegasos



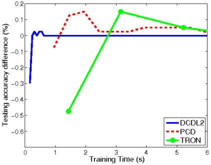
(a) L1-SVM: astro-physic



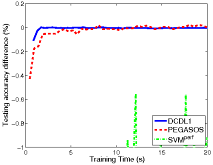
(b) L2-SVM: astro-physic



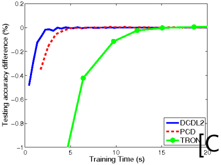
(c) L1-SVM: news20



(d) L2-SVM: news20



(e) L1-SVM: rcv1



(f) L2-SVM: rcv1

[C. J. Hsieh et al. 2008]

Outline

9. Subgradient Descent in the Primal

10. Linearization of Nonlinear Kernels

Basic Idea

Instead of using a nonlinear kernel, e.g., the polynomial kernel of degree d

$$K(x, z) := (\gamma x^T z + r)^d$$

with hyperparameters d, γ and r for data $x, z \in \mathbb{R}^n$,
use the explicit embedding, e.g., for $d = 1$ and $r = 1$:

$$\phi(x) := (1, \sqrt{2\gamma}x_1, \ldots, \sqrt{2\gamma}x_n, \gamma x_1^2, \ldots, \gamma x_n^2, \sqrt{2\gamma}x_1 x_2, \ldots, \sqrt{2\gamma}x_{n-1}x_n)$$

or more simple

$$\phi(x) := (1, x_1, \ldots, x_n, x_1^2, \ldots, x_n^2, x_1 x_2, \ldots, x_{n-1}x_n)$$

of dimension $\frac{(n+d)!}{n!d!}$.

Comparison Linearized Nonlinear vs. Nonlinear Kernel

	Linear (LIBLINEAR)			RBF (LIBSVM)			
Data set	C	Time (s)	Accuracy	C	γ	Time (s)	Accuracy
a9a	32	5.4	84.98	8	0.03125	98.9	85.03
real-sim	1	0.3	97.51	8	0.5	973.7	97.90
ijcnn1	32	1.6	92.21	32	2	26.9	98.69
MNIST38	0.03125	0.1	96.82	2	0.03125	37.6	99.70
covtype	0.0625	1.4	76.35	32	32	54,968.1	96.08
webspam	32	25.5	93.15	8	32	15,571.1	99.20

Table 4: Comparison of linear SVM and nonlinear SVM with RBF kernel. Time is in seconds.

Data set	Degree-2 Polynomial					Accuracy diff.	
	C	γ	Training time (s) LIBLINEAR	Training time (s) LIBSVM	Accuracy	Linear	RBF
a9a	8	0.03125	1.6	89.8	85.06	0.07	0.02
real-sim	0.03125	8	59.8	1,220.5	98.00	0.49	0.10
ijcnn1	0.125	32	10.7	64.2	97.84	5.63	−0.85
MNIST38	2	0.3125	8.6	18.4	99.29	2.47	−0.40
covtype	2	8	5,211.9	NA	80.09	3.74	−15.98
webspam	8	8	3,228.1	NA	98.44	5.29	−0.76

Table 5: Training time (in seconds) and testing accuracy of using the degree-2 polynomial mapping.

[Y. Chang et al. 2010]

References

Chang, Yin-Wen et al. (Aug. 2010): *Training and Testing Low-degree Polynomial Data Mappings via Linear SVM*. In: *J. Mach. Learn. Res.* 11, 1471–1490.

Hsieh, C. J et al. (2008): *A dual coordinate descent method for large-scale linear SVM*. In: *Proceedings of the 25th international conference on Machine learning*, 408–415.

Shalev-Shwartz, S., Y. Singer, and N. Srebro (2007): *Pegasos: Primal estimated sub-gradient solver for svm*. In: *Proceedings of the 24th international conference on Machine learning*, 807–814.