

Lecture 15 - scribbles - cutting plane

Tuesday, March 13, 2018 14:32

today: • more SVM struct properties
• cutting plane alg.
• FW

more properties of SVM struct dual:

$$w(\alpha) = \frac{1}{n} \sum_i \sum_{\tilde{y} \in \mathcal{Y}_i} \alpha_i(\tilde{y}) \psi_i(\tilde{y})$$

$$w^* = \frac{1}{\lambda} \left(\frac{1}{n} \sum_i \sum_{\tilde{y} \in \mathcal{Y}_i} \alpha_i^*(\tilde{y}) \psi_i(\tilde{y}) \right)$$

$$\text{let } R_i \triangleq \max_{\tilde{y}} \|\psi_i(\tilde{y})\|_2$$

$$\bar{R} \triangleq \frac{1}{n} \sum_i R_i$$

$$\begin{aligned} \text{then } \|w^*\| &\leq \frac{1}{\lambda} \frac{1}{n} \sum_i \left(\sum_{\tilde{y} \in \mathcal{Y}_i} \alpha_i^*(\tilde{y}) \overbrace{\|\psi_i(\tilde{y})\|}^{R_i} \right) \\ &\leq \frac{1}{\lambda} \left(\frac{1}{n} \sum_i R_i \right) = \bar{R} \end{aligned}$$

$$\begin{aligned} \text{kernel trick: } \langle w, \phi(x, y) \rangle &= \frac{1}{n} \sum_i \sum_{\tilde{y}} \alpha_i(\tilde{y}) \underbrace{\langle \psi_i(\tilde{y}), \phi(x, y) \rangle}_{\substack{K(x^{(i)}, y^{(i)}; x, y) \\ = K(x^{(i)}, \tilde{y}; x, y)}} \\ &\leq \frac{1}{n} \sum_i \sum_{\tilde{y}} \alpha_i(\tilde{y}) \underbrace{\langle \psi_i(\tilde{y}), \psi_i(\tilde{y}) \rangle}_{= R_i^2} \\ &\leq \frac{1}{n} \sum_i \sum_{\tilde{y}} \alpha_i(\tilde{y}) R_i^2 \end{aligned}$$

2) suppose rescale features $\tilde{\psi} = b\psi$

$$H_c(y; w) = \ell_i(y) - \langle \tilde{w}, \tilde{\psi}_i(y) \rangle$$

$$\tilde{w} = \frac{1}{\tilde{\lambda}} \frac{1}{n} \sum_i \sum_{\tilde{y}} \alpha_i^*(\tilde{y}) \tilde{\psi}_i(\tilde{y})$$

$$\text{let } \tilde{\lambda} = b^2 \lambda$$

w^*, α^* optimal for (P)
 $\Leftrightarrow \tilde{w} = \frac{w^*}{b}$ and α^* optimal for (\tilde{P}) where $\tilde{\psi} = b\psi$

$$\tilde{x} \text{ is } \tilde{y}$$

$$\Rightarrow \tilde{w}(\alpha^*) = \frac{w^*}{b}$$

and thus \tilde{w} is optimal
rescaled
for new problem

$w = \frac{w^*}{b}$ and α^* optimal for
(P) where $\tilde{w} = b w^*$
 $\tilde{\lambda} = b \lambda$

$$3) \text{ scale loss: } \tilde{l} = b \cdot l \Rightarrow \tilde{\lambda} = \frac{\lambda}{b}$$

4) etc...

M3-net example (dual):

$$\text{suppose } \psi(y) = \sum_c \psi_c(y_c)$$

$$w(\alpha) = A\alpha = \sum_i A_i \alpha_i$$

$$\lambda n A_i \alpha_i = \sum_y \alpha_i(y) \psi_i(y) = \sum_y \alpha_i(y) \sum_c \psi_{i,c}(y_c) \triangleq \mu_{i,c}(y_c)$$

marginal variable

$$= \sum_c \sum_{y_c} \psi_{i,c}(y_c) \left[\sum_{\substack{y: y_i = \alpha_i \\ y_c = y_c}} \alpha_i(y) \right]$$

similarly, suppose $\ell_i(y) = \sum_c \ell_{i,c}(y_c)$
define $\tilde{\ell}_{i,c}(y_c) \triangleq \frac{\ell_{i,c}(y_c)}{n}$

$$\langle b, \alpha \rangle = \langle \tilde{b}, \mu(\alpha) \rangle$$

$$A_i \alpha_i = \tilde{A}_i \mu_i \text{ where } (\tilde{A}_i)_{j,c,y_c} = \frac{\psi_{i,j}(y_c)}{\lambda n}$$

#columns $\sum_c |\mathcal{Y}_c|$

$$\alpha_i \in \Delta(\mathcal{Y}_i) \Rightarrow \mu_i \in M_i$$

& marginal polytope

thus we can replace $\max_{\alpha_i \in \Delta(\mathcal{Y}_i)} \rightarrow \frac{\|A\alpha\|^2}{2} + b^T \alpha$ with $\max_{\mu_i \in M_i} -\lambda \frac{\|\tilde{A}\mu\|^2}{2} + \tilde{b}^T \mu$

tractable s.t.
CSP?

suppose
 μ_i is tractable

M3-net paper used
"structured SVM" algorithms

1.0 1.1 ... 1.4 ... 1.5 ... 1.6 ... 1.7 ... 1.8 ... 1.9 ... 2.0 ... 2.1 ... 2.2 ... 2.3 ... 2.4 ... 2.5 ... 2.6 ... 2.7 ... 2.8 ... 2.9 ... 3.0 ... 3.1 ... 3.2 ... 3.3 ... 3.4 ... 3.5 ... 3.6 ... 3.7 ... 3.8 ... 3.9 ... 4.0 ... 4.1 ... 4.2 ... 4.3 ... 4.4 ... 4.5 ... 4.6 ... 4.7 ... 4.8 ... 4.9 ... 5.0 ... 5.1 ... 5.2 ... 5.3 ... 5.4 ... 5.5 ... 5.6 ... 5.7 ... 5.8 ... 5.9 ... 6.0 ... 6.1 ... 6.2 ... 6.3 ... 6.4 ... 6.5 ... 6.6 ... 6.7 ... 6.8 ... 6.9 ... 7.0 ... 7.1 ... 7.2 ... 7.3 ... 7.4 ... 7.5 ... 7.6 ... 7.7 ... 7.8 ... 7.9 ... 8.0 ... 8.1 ... 8.2 ... 8.3 ... 8.4 ... 8.5 ... 8.6 ... 8.7 ... 8.8 ... 8.9 ... 9.0 ... 9.1 ... 9.2 ... 9.3 ... 9.4 ... 9.5 ... 9.6 ... 9.7 ... 9.8 ... 9.9 ... 10.0 ... 10.1 ... 10.2 ... 10.3 ... 10.4 ... 10.5 ... 10.6 ... 10.7 ... 10.8 ... 10.9 ... 11.0 ... 11.1 ... 11.2 ... 11.3 ... 11.4 ... 11.5 ... 11.6 ... 11.7 ... 11.8 ... 11.9 ... 12.0 ... 12.1 ... 12.2 ... 12.3 ... 12.4 ... 12.5 ... 12.6 ... 12.7 ... 12.8 ... 12.9 ... 13.0 ... 13.1 ... 13.2 ... 13.3 ... 13.4 ... 13.5 ... 13.6 ... 13.7 ... 13.8 ... 13.9 ... 14.0 ... 14.1 ... 14.2 ... 14.3 ... 14.4 ... 14.5 ... 14.6 ... 14.7 ... 14.8 ... 14.9 ... 15.0 ... 15.1 ... 15.2 ... 15.3 ... 15.4 ... 15.5 ... 15.6 ... 15.7 ... 15.8 ... 15.9 ... 16.0 ... 16.1 ... 16.2 ... 16.3 ... 16.4 ... 16.5 ... 16.6 ... 16.7 ... 16.8 ... 16.9 ... 17.0 ... 17.1 ... 17.2 ... 17.3 ... 17.4 ... 17.5 ... 17.6 ... 17.7 ... 17.8 ... 17.9 ... 18.0 ... 18.1 ... 18.2 ... 18.3 ... 18.4 ... 18.5 ... 18.6 ... 18.7 ... 18.8 ... 18.9 ... 19.0 ... 19.1 ... 19.2 ... 19.3 ... 19.4 ... 19.5 ... 19.6 ... 19.7 ... 19.8 ... 19.9 ... 20.0 ... 20.1 ... 20.2 ... 20.3 ... 20.4 ... 20.5 ... 20.6 ... 20.7 ... 20.8 ... 20.9 ... 21.0 ... 21.1 ... 21.2 ... 21.3 ... 21.4 ... 21.5 ... 21.6 ... 21.7 ... 21.8 ... 21.9 ... 22.0 ... 22.1 ... 22.2 ... 22.3 ... 22.4 ... 22.5 ... 22.6 ... 22.7 ... 22.8 ... 22.9 ... 23.0 ... 23.1 ... 23.2 ... 23.3 ... 23.4 ... 23.5 ... 23.6 ... 23.7 ... 23.8 ... 23.9 ... 24.0 ... 24.1 ... 24.2 ... 24.3 ... 24.4 ... 24.5 ... 24.6 ... 24.7 ... 24.8 ... 24.9 ... 25.0 ... 25.1 ... 25.2 ... 25.3 ... 25.4 ... 25.5 ... 25.6 ... 25.7 ... 25.8 ... 25.9 ... 26.0 ... 26.1 ... 26.2 ... 26.3 ... 26.4 ... 26.5 ... 26.6 ... 26.7 ... 26.8 ... 26.9 ... 27.0 ... 27.1 ... 27.2 ... 27.3 ... 27.4 ... 27.5 ... 27.6 ... 27.7 ... 27.8 ... 27.9 ... 28.0 ... 28.1 ... 28.2 ... 28.3 ... 28.4 ... 28.5 ... 28.6 ... 28.7 ... 28.8 ... 28.9 ... 29.0 ... 29.1 ... 29.2 ... 29.3 ... 29.4 ... 29.5 ... 29.6 ... 29.7 ... 29.8 ... 29.9 ... 30.0 ... 30.1 ... 30.2 ... 30.3 ... 30.4 ... 30.5 ... 30.6 ... 30.7 ... 30.8 ... 30.9 ... 31.0 ... 31.1 ... 31.2 ... 31.3 ... 31.4 ... 31.5 ... 31.6 ... 31.7 ... 31.8 ... 31.9 ... 32.0 ... 32.1 ... 32.2 ... 32.3 ... 32.4 ... 32.5 ... 32.6 ... 32.7 ... 32.8 ... 32.9 ... 33.0 ... 33.1 ... 33.2 ... 33.3 ... 33.4 ... 33.5 ... 33.6 ... 33.7 ... 33.8 ... 33.9 ... 34.0 ... 34.1 ... 34.2 ... 34.3 ... 34.4 ... 34.5 ... 34.6 ... 34.7 ... 34.8 ... 34.9 ... 35.0 ... 35.1 ... 35.2 ... 35.3 ... 35.4 ... 35.5 ... 35.6 ... 35.7 ... 35.8 ... 35.9 ... 36.0 ... 36.1 ... 36.2 ... 36.3 ... 36.4 ... 36.5 ... 36.6 ... 36.7 ... 36.8 ... 36.9 ... 37.0 ... 37.1 ... 37.2 ... 37.3 ... 37.4 ... 37.5 ... 37.6 ... 37.7 ... 37.8 ... 37.9 ... 38.0 ... 38.1 ... 38.2 ... 38.3 ... 38.4 ... 38.5 ... 38.6 ... 38.7 ... 38.8 ... 38.9 ... 39.0 ... 39.1 ... 39.2 ... 39.3 ... 39.4 ... 39.5 ... 39.6 ... 39.7 ... 39.8 ... 39.9 ... 40.0 ... 40.1 ... 40.2 ... 40.3 ... 40.4 ... 40.5 ... 40.6 ... 40.7 ... 40.8 ... 40.9 ... 41.0 ... 41.1 ... 41.2 ... 41.3 ... 41.4 ... 41.5 ... 41.6 ... 41.7 ... 41.8 ... 41.9 ... 42.0 ... 42.1 ... 42.2 ... 42.3 ... 42.4 ... 42.5 ... 42.6 ... 42.7 ... 42.8 ... 42.9 ... 43.0 ... 43.1 ... 43.2 ... 43.3 ... 43.4 ... 43.5 ... 43.6 ... 43.7 ... 43.8 ... 43.9 ... 44.0 ... 44.1 ... 44.2 ... 44.3 ... 44.4 ... 44.5 ... 44.6 ... 44.7 ... 44.8 ... 44.9 ... 45.0 ... 45.1 ... 45.2 ... 45.3 ... 45.4 ... 45.5 ... 45.6 ... 45.7 ... 45.8 ... 45.9 ... 46.0 ... 46.1 ... 46.2 ... 46.3 ... 46.4 ... 46.5 ... 46.6 ... 46.7 ... 46.8 ... 46.9 ... 47.0 ... 47.1 ... 47.2 ... 47.3 ... 47.4 ... 47.5 ... 47.6 ... 47.7 ... 47.8 ... 47.9 ... 48.0 ... 48.1 ... 48.2 ... 48.3 ... 48.4 ... 48.5 ... 48.6 ... 48.7 ... 48.8 ... 48.9 ... 49.0 ... 49.1 ... 49.2 ... 49.3 ... 49.4 ... 49.5 ... 49.6 ... 49.7 ... 49.8 ... 49.9 ... 50.0 ... 50.1 ... 50.2 ... 50.3 ... 50.4 ... 50.5 ... 50.6 ... 50.7 ... 50.8 ... 50.9 ... 51.0 ... 51.1 ... 51.2 ... 51.3 ... 51.4 ... 51.5 ... 51.6 ... 51.7 ... 51.8 ... 51.9 ... 52.0 ... 52.1 ... 52.2 ... 52.3 ... 52.4 ... 52.5 ... 52.6 ... 52.7 ... 52.8 ... 52.9 ... 53.0 ... 53.1 ... 53.2 ... 53.3 ... 53.4 ... 53.5 ... 53.6 ... 53.7 ... 53.8 ... 53.9 ... 54.0 ... 54.1 ... 54.2 ... 54.3 ... 54.4 ... 54.5 ... 54.6 ... 54.7 ... 54.8 ... 54.9 ... 55.0 ... 55.1 ... 55.2 ... 55.3 ... 55.4 ... 55.5 ... 55.6 ... 55.7 ... 55.8 ... 55.9 ... 56.0 ... 56.1 ... 56.2 ... 56.3 ... 56.4 ... 56.5 ... 56.6 ... 56.7 ... 56.8 ... 56.9 ... 57.0 ... 57.1 ... 57.2 ... 57.3 ... 57.4 ... 57.5 ... 57.6 ... 57.7 ... 57.8 ... 57.9 ... 58.0 ... 58.1 ... 58.2 ... 58.3 ... 58.4 ... 58.5 ... 58.6 ... 58.7 ... 58.8 ... 58.9 ... 59.0 ... 59.1 ... 59.2 ... 59.3 ... 59.4 ... 59.5 ... 59.6 ... 59.7 ... 59.8 ... 59.9 ... 60.0 ... 60.1 ... 60.2 ... 60.3 ... 60.4 ... 60.5 ... 60.6 ... 60.7 ... 60.8 ... 60.9 ... 61.0 ... 61.1 ... 61.2 ... 61.3 ... 61.4 ... 61.5 ... 61.6 ... 61.7 ... 61.8 ... 61.9 ... 62.0 ... 62.1 ... 62.2 ... 62.3 ... 62.4 ... 62.5 ... 62.6 ... 62.7 ... 62.8 ... 62.9 ... 63.0 ... 63.1 ... 63.2 ... 63.3 ... 63.4 ... 63.5 ... 63.6 ... 63.7 ... 63.8 ... 63.9 ... 64.0 ... 64.1 ... 64.2 ... 64.3 ... 64.4 ... 64.5 ... 64.6 ... 64.7 ... 64.8 ... 64.9 ... 65.0 ... 65.1 ... 65.2 ... 65.3 ... 65.4 ... 65.5 ... 65.6 ... 65.7 ... 65.8 ... 65.9 ... 66.0 ... 66.1 ... 66.2 ... 66.3 ... 66.4 ... 66.5 ... 66.6 ... 66.7 ... 66.8 ... 66.9 ... 67.0 ... 67.1 ... 67.2 ... 67.3 ... 67.4 ... 67.5 ... 67.6 ... 67.7 ... 67.8 ... 67.9 ... 68.0 ... 68.1 ... 68.2 ... 68.3 ... 68.4 ... 68.5 ... 68.6 ... 68.7 ... 68.8 ... 68.9 ... 69.0 ... 69.1 ... 69.2 ... 69.3 ... 69.4 ... 69.5 ... 69.6 ... 69.7 ... 69.8 ... 69.9 ... 70.0 ... 70.1 ... 70.2 ... 70.3 ... 70.4 ... 70.5 ... 70.6 ... 70.7 ... 70.8 ... 70.9 ... 71.0 ... 71.1 ... 71.2 ... 71.3 ... 71.4 ... 71.5 ... 71.6 ... 71.7 ... 71.8 ... 71.9 ... 72.0 ... 72.1 ... 72.2 ... 72.3 ... 72.4 ... 72.5 ... 72.6 ... 72.7 ... 72.8 ... 72.9 ... 73.0 ... 73.1 ... 73.2 ... 73.3 ... 73.4 ... 73.5 ... 73.6 ... 73.7 ... 73.8 ... 73.9 ... 74.0 ... 74.1 ... 74.2 ... 74.3 ... 74.4 ... 74.5 ... 74.6 ... 74.7 ... 74.8 ... 74.9 ... 75.0 ... 75.1 ... 75.2 ... 75.3 ... 75.4 ... 75.5 ... 75.6 ... 75.7 ... 75.8 ... 75.9 ... 76.0 ... 76.1 ... 76.2 ... 76.3 ... 76.4 ... 76.5 ... 76.6 ... 76.7 ... 76.8 ... 76.9 ... 77.0 ... 77.1 ... 77.2 ... 77.3 ... 77.4 ... 77.5 ... 77.6 ... 77.7 ... 77.8 ... 77.9 ... 78.0 ... 78.1 ... 78.2 ... 78.3 ... 78.4 ... 78.5 ... 78.6 ... 78.7 ... 78.8 ... 78.9 ... 79.0 ... 79.1 ... 79.2 ... 79.3 ... 79.4 ... 79.5 ... 79.6 ... 79.7 ... 79.8 ... 79.9 ... 80.0 ... 80.1 ... 80.2 ... 80.3 ... 80.4 ... 80.5 ... 80.6 ... 80.7 ... 80.8 ... 80.9 ... 81.0 ... 81.1 ... 81.2 ... 81.3 ... 81.4 ... 81.5 ... 81.6 ... 81.7 ... 81.8 ... 81.9 ... 82.0 ... 82.1 ... 82.2 ... 82.3 ... 82.4 ... 82.5 ... 82.6 ... 82.7 ... 82.8 ... 82.9 ... 83.0 ... 83.1 ... 83.2 ... 83.3 ... 83.4 ... 83.5 ... 83.6 ... 83.7 ... 83.8 ... 83.9 ... 84.0 ... 84.1 ... 84.2 ... 84.3 ... 84.4 ... 84.5 ... 84.6 ... 84.7 ... 84.8 ... 84.9 ... 85.0 ... 85.1 ... 85.2 ... 85.3 ... 85.4 ... 85.5 ... 85.6 ... 85.7 ... 85.8 ... 85.9 ... 86.0 ... 86.1 ... 86.2 ... 86.3 ... 86.4 ... 86.5 ... 86.6 ... 86.7 ... 86.8 ... 86.9 ... 87.0 ... 87.1 ... 87.2 ... 87.3 ... 87.4 ... 87.5 ... 87.6 ... 87.7 ... 87.8 ... 87.9 ... 88.0 ... 88.1 ... 88.2 ... 88.3 ... 88.4 ... 88.5 ... 88.6 ... 88.7 ... 88.8 ... 88.9 ... 89.0 ... 89.1 ... 89.2 ... 89.3 ... 89.4 ... 89.5 ... 89.6 ... 89.7 ... 89.8 ... 89.9 ... 90.0 ... 90.1 ... 90.2 ... 90.3 ... 90.4 ... 90.5 ... 90.6 ... 90.7 ... 90.8 ... 90.9 ... 91.0 ... 91.1 ... 91.2 ... 91.3 ... 91.4 ... 91.5 ... 91.6 ... 91.7 ... 91.8 ... 91.9 ... 92.0 ... 92.1 ... 92.2 ... 92.3 ... 92.4 ... 92.5 ... 92.6 ... 92.7 ... 92.8 ... 92.9 ... 93.0 ... 93.1 ... 93.2 ... 93.3 ... 93.4 ... 93.5 ... 93.6 ... 93.7 ... 93.8 ... 93.9 ... 94.0 ... 94.1 ... 94.2 ... 94.3 ... 94.4 ... 94.5 ... 94.6 ... 94.7 ... 94.8 ... 94.9 ... 95.0 ... 95.1 ... 95.2 ... 95.3 ... 95.4 ... 95.5 ... 95.6 ... 95.7 ... 95.8 ... 95.9 ... 96.0 ... 96.1 ... 96.2 ... 96.3 ... 96.4 ... 96.5 ... 96.6 ... 96.7 ... 96.8 ... 96.9 ... 97.0 ... 97.1 ... 97.2 ... 97.3 ... 97.4 ... 97.5 ... 97.6 ... 97.7 ... 97.8 ... 97.9 ... 98.0 ... 98.1 ... 98.2 ... 98.3 ... 98.4 ... 98.5 ... 98.6 ... 98.7 ... 98.8 ... 98.9 ... 99.0 ... 99.1 ... 99.2 ... 99.3 ... 99.4 ... 99.5 ... 99.6 ... 99.7 ... 99.8 ... 99.9 ... 100.0

structured SVM algorithm

block-coordinate ascent using pair of variables at a time
[similar to pairwise FW]

$$(D) \max -\frac{\lambda \|A\alpha\|^2}{2} + \langle b, \alpha \rangle \quad \alpha \in \Delta \left(\sum_{i=1}^n \gamma_i \right)$$

$w(\alpha) = \sum_{i=1}^n \alpha_i \tilde{y}_i$

constraint generation algorithm:

[Isochrantzidis & al. JMLR 2005]

1-slab version

$$(P) \min_{w, \xi} \frac{\lambda \|w\|^2}{2} + \sum_i \xi_i$$

s.t. $\xi_i \geq \frac{1}{n} \sum_i H_i(\tilde{y}_i; w) \quad \forall \tilde{y}_i \in \mathcal{Y}_i, i=1, \dots, n$

$\sum_i |\mathcal{Y}_i|$ constraints

want to solve

$$(P) \min_{w, \xi} \frac{\lambda \|w\|^2}{2} + \sum_i \xi_i$$

s.t. $\xi_i \geq H_i(\tilde{y}_i; w) \quad \forall \tilde{y}_i \in \mathcal{Y}_i$
 $\xi_i \geq 0$

N-slab version

$\sum_i |\mathcal{Y}_i|$ constraints

(D) $\max -\frac{\lambda \|A\alpha\|^2}{2} + \langle b, \alpha \rangle$
s.t. $\alpha_i \in \Delta(\mathcal{Y}_i)$

exponential # constraints
variables

[ML 2009] 1-slab SVM struct paper

$O(\frac{1}{\epsilon})$ # iterations
big memory saving as 1 constraint per iteration

SVM struct algorithm:

iterate solving QPs with more and more constraints

1) start with no constraint $\Rightarrow w^{(0)} = 0, \xi^{(0)} = 0$

2) repeat: for each i , find $\hat{y}_i = \arg\max_{\tilde{y} \in \mathcal{Y}_i} H_i(\tilde{y}; w^{(t)})$

• add $\xi_i \geq H_i(\hat{y}_i; w)$ constraint to QP } $O(n)$ step

• resolve QP(w, ξ) with these constraints to get $w^{(t+1)}, \xi^{(t+1)}$ [e.g. using CVXopt]

for LP, this is called "column generation"
"constraint generation alg." aka here as "cutting plane"

loss augmented dualing

stop when primal-dual gap $\leq \epsilon$

[in 2005, showed that algorithm stops after $O(\frac{1}{\epsilon^2})$ iterations]
 refined later to $O(\frac{1}{\epsilon})$

1940

Frank-Wolfe algorithm

↳ ^{for} Smooth constrained optimization

(motivation dual of SVM struct:

$$\min_{a \in \Delta^d} \frac{1}{2} \|Aa\|^2 - b^T a$$
)

1940s: simplex algorithm to solve LPs

1956: Mangenite Frank & Phil Wolfe

→ non-linear optimization by iterating LPs

Setup: $\min_x f(x)$
 s.t. $x \in M$

• f is L -smooth i.e. ∇f is L -Lipschitz and f is convex

• M is convex and bounded set

(implicit assumption is that LMO is efficient)
 "Linear minimization oracle"

(more generally, can get rates for f continuously differentiable)
 e.g. ∇f is Hölder α

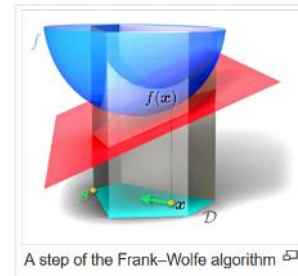
FW algorithm:

start with $x_0 \in M$
 for $t = 0, \dots$

compute $S_t = \arg \min_{S \in M} \langle S, \nabla f(x_t) \rangle$

$f(s) \geq f(x_t) + \langle \nabla f(x_t), s - x_t \rangle \quad \forall s \in M$
 minimize → linear approx of f at x_t

stopping criteria



SEM

stopping criteria

[Let $g_t \triangleq \langle s_t - x_t, \nabla f(x_t) \rangle$ (FW gap) if $g_t \leq \epsilon$; output x_t]

$x_{t+1} = (1-\gamma_t)x_t + \gamma_t s_t$ (convex combo between x_t & s_t)

$$= x_t + \gamma_t \underbrace{(s_t - x_t)}_{d_t} \quad \gamma_t \in [0,1]$$

end
output x_{t+1}

step size choice: $\gamma_t = \begin{cases} \text{universal choice } \frac{2}{t+2} \\ \text{line-search } \gamma_t = \arg \min_{\gamma \in [0,1]} f(x_t + \gamma(s_t - x_t)) \\ \text{adaptive: } \frac{g_t}{L\|s_t - x_t\|^2} \text{ or } \frac{g_t}{c_g} \text{ truncated at 1} \end{cases}$
 c_g affine invariant constant

⊛ big motivation for FW is that LMO is often much cheaper than projections and cheap for many sets M appearing in ML

properties: 1) $f(x_t) - \min_{x \in M} f(x) \leq O(\frac{1}{t})$
 $\frac{x \in M}{\triangleq f^*}$

2) FW-gap $g_t \geq f(x_t) - f^* \rightarrow$ certificate of suboptimality

$$\min_{s \in S} g_s \leq O(\frac{1}{t})$$

$$3) x_t = p_0^t x_0 + \sum_{u=1}^t p_u^t s_{u-1}$$

$$\text{where } \sum_{u=0}^t p_u^t = 1 \quad p_u^t \geq 0$$

$\rightarrow x_t$ has "sparse" expansion in terms of the FW-coners $\{s_u\}_{u=1}^{t-1}$

sparse method \rightarrow popular in ML



(see here can run FW on SVM structural dual assuming can compute LMO)

4) FW is affine co-variant (like Newton)

Thm 9.5 2) gap bound: $f(s) \geq f(x_t) + \langle \nabla f(x_t), s - x_t \rangle \quad \forall s$ by convexity

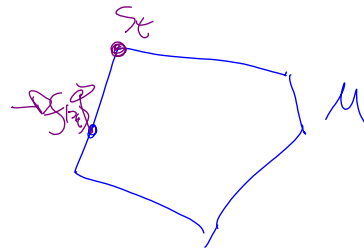
min
with respect
to SEM on
both sides

$$\begin{aligned} & \downarrow \qquad \qquad \qquad \downarrow \\ f^* & \geq f(x_t) + \underbrace{\langle \nabla f(x_t), s_t - x_t \rangle}_{-g_t} \end{aligned}$$

$$\Rightarrow \boxed{g_t \geq f(x_t) - f^* \geq 0}$$

interpretations: s_t is min. linear approx. of f

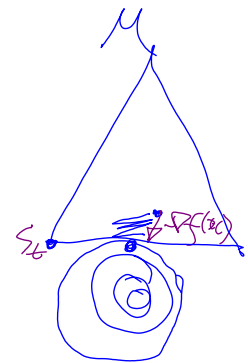
or max inner product with $-\nabla f(x_t)$



comment: FW step moves mass uniformly away from active set to FW-corner s_t

$$x_t = \sum_u \alpha_u s_u \quad x_{t+1} = (1 - \delta_t) x_t + \delta_t s_t$$

$$x_{t+1} = \sum_u \underbrace{\alpha_u (1 - \delta_t)}_{\text{shrinking previous coordinate}} s_u + \delta_t s_t$$



unless step-size $\delta_t = 1$,

FW is never removing a corner from its expansion

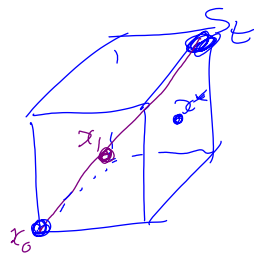
\Rightarrow zig-zag close to boundary on polytopes

(this is why $O(\frac{1}{t})$ rate)

⇒ Zig-zag close to boundary on polytopes

(this is why $O(\frac{1}{\epsilon})$ rate even if f is μ -strongly convex)

but FW has no problem on "strongly convex sets"
 ↳ sublevel set of a strongly convex fct.



when solution is in the interior



($O(\exp(-\mu t))$) convergence rate

(11)

Away-step FW fix: (solve the zig-zagging problem)

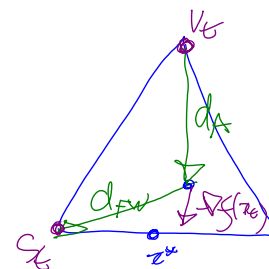
in addition to compute $S_t = \arg \min_{S \in M} \langle \nabla f(x_t), S \rangle$

$x_t = \sum_u \alpha_u s_u$
 active-set(x_t) = $\{s_u : \alpha_u > 0\}$

compute $V_t = \arg \max_{S \in \text{active-set}(x_t)} \langle \nabla f(x_t), S \rangle$

$$d_{FW} \triangleq S_t - x_t$$

$$d_A \triangleq x_t - V_t$$



AFW picks the direction with best inner-product

i.e. pick d_A if $\langle d_A, -\nabla f(x_t) \rangle > \langle d_{FW}, -\nabla f(x_t) \rangle$

If use d_A , let $x_{t+1} = \arg \min_{x \in M} f(x_t + \delta d_A)$

$$\delta \in [0, \delta_{\max}]$$

δ_{\max} is determined from α_u 's

$$x_{t+1} = x_t + \delta_t(x_t - v_t)$$

$$x_{t+1} = \sum_u (1 + \delta_t) \alpha_u s_u - \delta_t v_t$$

$$\delta - \delta\alpha = \alpha$$

let α be coefficient of v_t ; $(1 + \delta_{\max})\alpha - \delta_{\max} = 0$

$$\boxed{\delta_{\max} = \frac{\alpha}{1 - \alpha}}$$

when $\delta_t = \delta_{\max}$

"drop stop" \rightarrow removed v_t from expansion

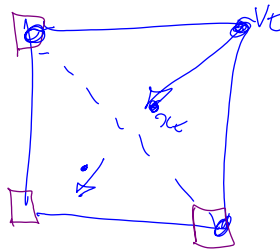
run AFW: either you maintain some expansion of $x_t = \sum_u \alpha_u \tilde{s}_u$

or you have a feasibility oracle + away step oracle

[see NIPS 2006 paper]
Moshir & Gordon

here
 $\text{corners}(M) \subseteq \{0, 1\}^d$

$M \rightarrow x \geq 0$ and $Ax = b$

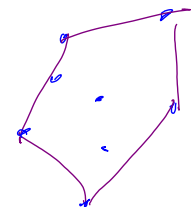


AFW has linear convergence rate on polytopes when f is strongly convex

(*) other comment: if $M = \text{conv}(A)$ where A is some finite set "atoms"

$$\text{LMO}(r) : \min_{s \in M = \text{conv}(A)} \langle s, r \rangle = \min_{a \in A} \langle a, r \rangle$$

\neg . Oats of modification of this in M_1 .



$\{M\} = \{M \mid \exists x \in A, y \in B, M = \text{conv}(A, B, x, y)\}$

$S \in M = \text{conv}(A)$

$a \in A$

\rightarrow lots of applications of this in ML

e.g. $A \rightarrow$ integer flows
 $\text{conv}(A) \rightarrow$ flow polytope

$\text{LMO} \rightarrow$ min cost network flow

$A \rightarrow$ degree assignment in graph
 $\text{conv}(A) \rightarrow$ marginal polytope

$\text{LMO} \rightarrow$ max-product
 or graph cut alg.
 for submodular potential
 (see later)