5.1 To solve $\min \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2$ s.t. $\sum_{j=1}^{n}|w_j|^q \le \eta$ } ①

Use Lagrange multipliers,

$$L = \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \alpha\left(\sum_{j=1}^{n}|w_j|^q - \eta\right)$$

$$w^* = \min_w \left[\max_{\alpha:\alpha\ge 0} \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \alpha\left(\sum_{j=1}^{n}|w_j|^q - \eta\right)\right]$$

$$= \min_w \left[\frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \alpha^*\sum_{j=1}^{n}|w_j|^q - \alpha^*\eta\right]$$

$$= \min_w \left[\frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \alpha^*\sum_{j=1}^{n}|w_j|^q\right]$$

It means when $\lambda = 2\alpha^*$, the minimization of the regularized error function is equivalent to ①.

5.2 $\max_\theta \prod_{i=1}^{n} P(x_i|\theta)P(\theta)$

$$\Rightarrow \max_\theta \left[\log \prod_{i=1}^{n} P(x_i|\theta) + \log P(\theta)\right]$$

$$\Rightarrow \max_\theta \left[\log \prod_{i=1}^{n} P(x_i|\theta) + \log\frac{\lambda}{2} - \lambda|\theta|\right]$$

$$\Rightarrow \max_\theta \left[\log \prod_{i=1}^{n} P(x_i|\theta) - \lambda|\theta|\right]$$

5.3 (a). $MSE(\hat{x}) = \mathbb{E}[(\hat{x}-x)^2]$

$= \mathbb{E}[(\hat{x}-x)^T(\hat{x}-x)]$

$= \mathbb{E}[\hat{x}^T\hat{x}] - 2\mathbb{E}[\hat{x}^T x] + \mathbb{E}[x^T x]$

$Var(\hat{x}) = \mathbb{E}[(\hat{x}-\mathbb{E}[\hat{x}])^2]$

$= \mathbb{E}[(\hat{x}-\mathbb{E}[\hat{x}])^T(\hat{x}-\mathbb{E}[\hat{x}])]$

$= \mathbb{E}[\hat{x}^T\hat{x} - \hat{x}\mathbb{E}[\hat{x}] - \mathbb{E}[\hat{x}]\hat{x} + \mathbb{E}[\hat{x}]^T\mathbb{E}[\hat{x}]]$

$= \mathbb{E}[\hat{x}^T\hat{x} + \mathbb{E}[\hat{x}]^T\mathbb{E}[\hat{x}] - 2\mathbb{E}[\hat{x}]^T\hat{x}$

$[Bias(\hat{x})]^2 = (\mathbb{E}[\hat{x}]-x)^T(\mathbb{E}[\hat{x}]-x)$

$= \mathbb{E}[\hat{x}]^T\mathbb{E}[\hat{x}] - \mathbb{E}[\hat{x}]^T x - x^T\mathbb{E}[\hat{x}] + x^T x$

$Var(\hat{x}) + (Bias(\hat{x}))^2 = \mathbb{E}(\hat{x}^T\hat{x}) - 2\mathbb{E}[\hat{x}]^T x + x^T x$

Also, we have $\mathbb{E}[x] = x$ and $\mathbb{E}[\hat{x}^T x] = \mathbb{E}[\hat{x}]^T x$

$\therefore Var(\hat{x}) + (Bias(\hat{x}))^2 = MSE(\hat{x})$

(b) $\mathbb{E}[(\hat{x}-x-N)^2] = \mathbb{E}[(\hat{x}-x-N)^T(\hat{x}-x-N)]$

$= \mathbb{E}[(\hat{x}-x)^T(\hat{x}-x) - 2(\hat{x}-x)^T N + N^T N]$

$= MSE(\hat{x}) - 2\mathbb{E}(\hat{x}-x)^T\mathbb{E}(N) + \mathbb{E}[N^T N]$

$= MSE(\hat{x}) + \delta^2$

5.4 (a) $\min(k, |\mathcal{X}|-k)$

Since for hypothesis set $\mathcal{H}_k^{\mathcal{X}}$, there are exactly k elements in $\mathcal{X}$ are labeled as 1 and $(|\mathcal{X}|-k)$ elements are labeled as 0. So $\mathcal{H}$ can scatter any $\min(k, |\mathcal{X}|-k)$ points.

~~For any points~~ If there are $(k+1)$ points labeled 1 or $(|\mathcal{X}|-k+1)$ points labeled as 0, $\mathcal{H}_k^{\mathcal{X}}$ cannot scatter them.

It means $VC(\mathcal{H}_k^{\mathcal{X}}) < k+1$ and $VC(\mathcal{H}_k^{\mathcal{X}}) < |\mathcal{X}|-k+1$

So $VC(\mathcal{H}_k^{\mathcal{X}}) = \min(k, |\mathcal{X}|-k)$

(b). $\min(2k+1, |\mathcal{X}|)$

For $\mathcal{H}_{\leq k}^{\mathcal{X}}$, there are at most k elements of $\mathcal{X}$ to make $h(x)=1$ or $h(x)=0$. It means when we give a set of points in $\mathcal{X}$, and there always exists m points labeled 1 or 0, where $m \leq k$, then we can find a hypothesis from $\mathcal{H}_{\leq k}^{\mathcal{X}}$ to scatter them.

So for any points less than $(2k+1)$, $\mathcal{H}_{\leq k}^{\mathcal{X}}$ can scatter them.

For $(2k+2)$ points with $(k+1)$ labeled 1 and $(k+1)$ labeled 0, $\mathcal{H}_{\leq k}^{\mathcal{X}}$ cannot scatter them. $VC(\mathcal{H}_{\leq k}^{\mathcal{X}}) < (2k+2)$

Also, we ~~shol~~ should consider $|\mathcal{X}|$, so $VC(\mathcal{H}_{\leq k}^{\mathcal{X}}) = \min(2k+1, |\mathcal{X}|)$.

Large Scale Linear Programming Decoding via the Alternating Direction Method of Multipllers

The keynote speech of **Stark Draper**

Professor shared his application of the ADMM to solve the linear programming relaxation of maximum likelihood docoding for error-correction codes. ADMM, which is a topic I am interested about. I am curious about the algorithm but the area of error-correction codes is unfamiliar to me. So I was lost in the most important part of content. But the background knowledge part gave me some intuitions about the ADMM.

The objective function is separable in x and y. The dual update requires solving a proximity function in x and y at the same time; the ADMM technique allows this problem to be solved approximately by first solving for x with y fixed, and then solving for y with x fixed. Rather than iterate until convergence. the algorithm proceeds directly to updating the dual variable and then repeating the process in an efficient and paralleizable manner. Because of this approximation, the algorithm is distinct from the pure augmented Lagrangian method.

To sumarize, ADMM is an algorithm that solves convex optimization problems by breaking them into smaller pieces, each of which are then easier to handle. It has recently found wide application in a number of areas.