Tuesday, 10/6/2020

EE 660

MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins
Presenter: Fernando Valladares Monteiro

Lecture 13

Lecture 13	EE 660	Oct 6, 2020

Announcements

- Homework 5 is due Friday
- Due to the shortened semester, our mid-semester project has been cancelled
  - Aspects of the mid-semester project will be included in the final project
  - Grading weights for each of the following categories will be increased by 5 percentage points:
    - Homeworks
    - End-of-semester quiz
    - Final project
- Details of the final project will be posted soon

Today's Lecture

- Regularization and complexity (AML view) (part 2)
- MAP and L1 regularization for feature selection

EE660 Lecture 13 Oct 6, 2020

Regularization and Complexity (AML view)
- part 2

To solve use Lagrangian optimization
with ineq. constraints
ineq. $[\cdot] \geq 0$

$$L(\underline{w}, \lambda_c) = E_D(h_{\underline{w}}) - \lambda_c [C - \underline{w}^T \underline{w}]$$

where $\lambda_c \geq 0$

$$\nabla_{\underline{w}} L = \nabla_w \{ E_D(h_{\underline{w}}) - \lambda_c [\cancel{C} - \underline{w}^T \underline{w}] \}$$

$$\nabla_{\underline{w}} L \Big|_{w = w_{reg}} = 0$$

$$\nabla_w \{ E_D(h_{\underline{w}}) + \lambda_c \underline{w}^T \underline{w} \} \Big|_{\underline{w} = \underline{w}_{reg}} = 0$$

$\Rightarrow \underline{w}_{reg}$ (locally) minimizes:
$E_{avg}(h_{\underline{w}}) = E_D(h_{\underline{w}}) + \lambda_c \underline{w}^T w, \quad \lambda_c \geq 0$

We also have λ_c^*:

$$\lambda_c^* = \arg\max_{\lambda_c} \left[\cancel{E_D} - \lambda_c (C - \underline{w}_{reg}^T \underline{w}_{reg}) \right], \quad \lambda_c \geq 0$$

$\Rightarrow$ either $\lambda_c^* = 0$

or $\underline{w}_{reg}^T \underline{w}_{reg} = C$

Thus:

(iii) $h_g = \arg\min_{h \in H} \{ E_D(h_{\underline{w}}) + \lambda_c \underline{w}^T \underline{w} \}, \quad \lambda_c \geq 0$

and treat λ_c as a parameter

$\underline{w}^T \underline{w}$ is a regularizer (l_2)

also called "weight decay"

Comment: C is a parameter in (i), (ii) [Lecture 12]

λ_c is a parameter in (iii)

Thus

$$E_{aug}(h_{\underline{w}}) = E_D(h_{\underline{w}}) + \lambda \underline{w}^T \underline{w}, \quad \lambda \geq 0$$

(unconstrained optimization problem)

And more generally:

$$E_{aug}(h_{\underline{w}}) = E_D(h_{\underline{w}}) + \frac{\lambda'}{N} \underbrace{\Omega(h_{\underline{w}})}$$

$$\lambda' \triangleq N\lambda \geq 0$$

general regularization penalty term

<u>Ex. 1</u>: $f(\underline{x})$ = 2nd order polynomial + small amount of noise

H: $\hat{f}(\underline{x})$ = 4th order polynomial

$N = 5$

[AML fig. 4.5]

Ex.2: $\hat{f} = \sum_{q=0}^{d} w_q x^q$

high q $\Rightarrow$ most volatile effect on $\hat{f}$

Instead of:

$$\Omega_{unif}(\underline{w}) = \sum_{q=0}^{15} w_q^2$$

Try: $\Omega_{low}(\underline{w}) = \sum_{q=0}^{15} q w_q^2$

[AML fig. 4.6]

Ex 3: Q_f order polynomial

$f(x) = \sum_{q=0}^{Q_f} a_q L_q(x)$ where $L_q(x)$ is an order q Legendre polynomial

$y_n = f(x_n) + \sigma \varepsilon_n$

↳ std. normal

Fit with H_{15} = 15th order polynomial

(a) Target $Q_f = 15$, vary σ^2

(b) Target $\sigma^2 = 0$, vary Q_f

[AML fig 4.7]

MAP and L1 regularization for feature selection [Murphy Sec. 13.3]

Why ~~sel~~ feature selection?

- Reduce VC dim (or complexity of $\mathcal{H}$) thereby reducing ε_{VC} (generalization bound on E_{out})
- Reduce overfitting
- Reduce computation time
- Remove redundant or correlated features

- Discover which features are most important for prediction (e.g. risk factors for heart ~~di~~ attack)

Ex.: housing price prediction:

x_1 = living area

x_2 = no. of rooms

x_3 = age of the house

If the features are standardized:

we can use $|w_i|$ as an indicator of relevance of feature i to predict y

Pose as a (MAP) estimation problem

$$(1)\begin{cases} p(\underline{w}|D) = \frac{1}{K} p(D|\underline{w})p(\underline{w}) \\ = \frac{1}{K}\left[\prod_{i=1}^{N} p(y_i|x_i,\underline{w})\right] p(\underline{w}) \\ = \frac{1}{K} p(\underline{y}|\underline{\underline{X}},\underline{w})\, p(\underline{w}) \end{cases}$$

Let $\underline{w} = \underline{w}^{(0)}$ (non-augmented)

If linear regression with usual assumption:

$p(y|\underline{x},\underline{w}) = \mathcal{N}(y|\underline{w}^T\underline{x}+w_0, \sigma^2)$

Now for prior: $p(\underline{w}) = p(\underline{w}|\lambda)$

$$p(\underline{w}|\lambda) = \prod_{j=1}^{D} \mathrm{Lap}(w_j|0, 1/\lambda)$$

$$= \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|} \propto \prod_{j=1}^{D} e^{-\lambda|w_j|}$$

and $p(w_0) \propto 1$

Laplacian prior favors $|w_i|$ close to $\emptyset$, (zero) and $|w_i|$ large (far from $\emptyset$) relative to Gaussian prior

$$(1) \Rightarrow J(\underline{w}) = -\log p(y \mid \underline{X}, \underline{w}) - \log p(\underline{w})$$

$$= NLL(\underline{w}) + \lambda \|\underline{w}\|_1, \quad \|w\|_1 = \sum_{j=1}^{D} |w_j|$$

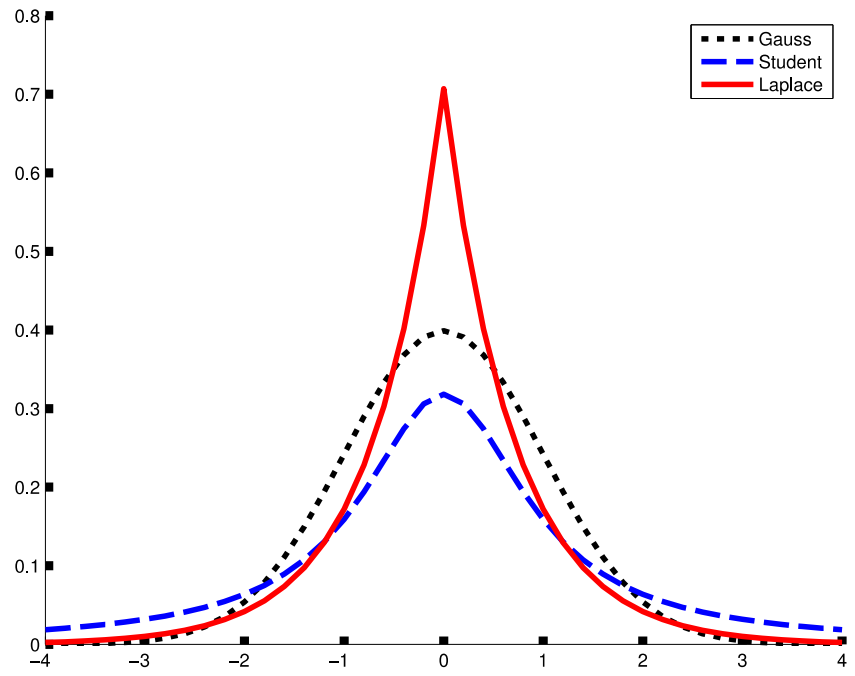
For linear model:

$$= \sum_{i=1}^{N} \frac{1}{2\sigma^2} [y_i - (w_0 + \underline{w}^T \underline{x})]^2 + \lambda \|\underline{w}\|_1$$

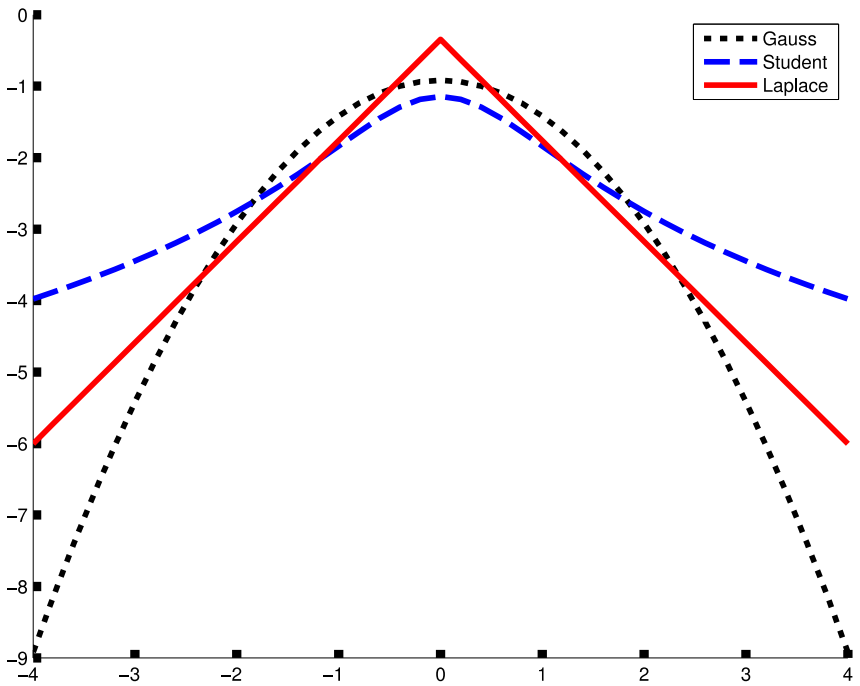
$$J(\underline{w}) \propto RSS(\underline{w}) + \lambda' \|\underline{w}\|, \quad \lambda' = 2\lambda\sigma^2$$

Basis Pursuit Denoising (BPDN)
[Murphy 13.3.0]

<u>Lasso</u> minimizes this J using subgradients
[Murphy 13.3.2]



Murphy Fig. 2.7(a). pdf's.



Murphy Fig. 2.7(b). log pdf's.