CS489/698
Lecture 7: Jan 25, 2017

Classification by Logistic Regression, Generalized linear models

[RN] Sec 18.6.4, [B] Sec. 4.3, [M] Chapt. 8, [HTF] Sec. 4.4

Beyond Mixtures of Gaussians

- Mixture of Gaussians:
  - Restrictive assumption: each class is Gaussian
  - Picture:

- Can we consider other distributions than Gaussians?

Exponential Family

- More generally, when $\Pr(\boldsymbol{x}|c_k)$ are members of the exponential family (e.g., Gaussian, exponential, Bernoulli, categorical, Poisson, Beta, Dirichlet, Gamma, etc.)

$$\Pr(\boldsymbol{x}|\boldsymbol{\theta}_k) = \exp(\boldsymbol{\theta}_k^T T(\boldsymbol{x}) - A(\boldsymbol{\theta}_k) + B(\boldsymbol{x}))$$

where $\boldsymbol{\theta}_k$: parameters of class k
$T(\boldsymbol{x}), A(\boldsymbol{\theta}_k), B(\boldsymbol{x})$: arbitrary fns of the inputs and params

- the posterior is a sigmoid logistic linear in $\boldsymbol{x}$

$$\Pr(c_k|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

Probabilistic Discriminative Models

- Instead of learning $\Pr(c_k)$ and $\Pr(\boldsymbol{x}|c_k)$ by maximum likelihood and finding $\Pr(c_k|\boldsymbol{x})$ by Bayesian inference, why not learn $\Pr(c_k|\boldsymbol{x})$ directly by maximum likelihood?

- We know the general form of $\Pr(c_k|\boldsymbol{x})$:
  - **Logistic sigmoid** (binary classification)
  - **Softmax** (general classification)

Logistic Regression

- Consider a single data point $(\boldsymbol{x}, y)$:

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}}\, \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}})$$

- Similarly, for an entire dataset $(\boldsymbol{X}, \boldsymbol{y})$:

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} \prod_n \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n)^{y_n} \big(1 - \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n)\big)^{1-y_n}$$

Objective: negative log likelihood (minimization)

$$L(\boldsymbol{w}) = -\sum_n y_n \ln \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n) + (1 - y_n) \ln(1 - \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n))$$

Tip: $\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1 - \sigma(a))$

Logistic Regression

- NB: Despite the name, logistic regression is a form of classification.

- However, it can be viewed as regression where the goal is to estimate the posterior $\Pr(c_k|\boldsymbol{x})$, which is a continuous function

Maximum likelihood

- Convex loss: set derivative to 0

$$0 = \frac{\partial L}{\partial \boldsymbol{w}} = -\sum_n y_n \frac{\cancel{\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)}\left(1-\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)\right)\bar{\boldsymbol{x}}_n}{\cancel{\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)}}$$

$$-\sum_n (1-y_n) \frac{\left(\cancel{1-\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)}\right)\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_{\boldsymbol{n}})(-\bar{\boldsymbol{x}}_n)}{\cancel{1-\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)}}$$

$$\Longrightarrow 0 = -\sum_n y_n \bar{\boldsymbol{x}}_n - \cancel{\sum_n y_n \sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)\bar{\boldsymbol{x}}_n}$$

$$+\sum_n \sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)\bar{\boldsymbol{x}}_n + \cancel{\sum_n y_n \sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n)\bar{\boldsymbol{x}}_n}$$

$$\Longrightarrow 0 = \sum_n [\sigma(\boldsymbol{w}^T\bar{\boldsymbol{x}}_n) - y_n]\bar{\boldsymbol{x}}_n$$

- Sigmoid prevents us from isolating $\boldsymbol{w}$, so we use an iterative method instead

Newton's method

- Iterative reweighted least square:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \boldsymbol{H}^{-1} \nabla L(\boldsymbol{w})$$

where ∇L is the gradient (column vector)

and H is the Hessian (matrix)

$$H = \begin{bmatrix} \frac{\partial L}{\partial^2 w_0} & \cdots & \frac{\partial L}{\partial w_0 \partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial w_m \partial w_0} & \cdots & \frac{\partial L}{(\partial w_m)^2} \end{bmatrix}$$

Hessian

$$\boldsymbol{H} = \nabla(\nabla L(\boldsymbol{w}))$$
$$= \sum_{n=1}^{N} \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n)\big(1 - \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n)\big)\overline{\boldsymbol{x}}_n \overline{\boldsymbol{x}}_n^T$$
$$= \overline{\boldsymbol{X}} \boldsymbol{R} \overline{\boldsymbol{X}}^T$$

where $\boldsymbol{R} = \begin{bmatrix} \sigma_1(1-\sigma_1) & & \\ & \ddots & \\ & & \sigma_N(1-\sigma_N) \end{bmatrix}$

and $\sigma_1 = \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_1), \quad \sigma_N = \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_N)$

Case study

- Applications: recommender systems, ad placement

- Used by all major companies

- Advantages: logistic regression is **simple, flexible and efficient**

App Recommendation

- Flexibility: millions of features (binary & numerical)
  - Examples:

- Efficiency: classification by dot products

$$c^* = argmax_k\ \sigma(\boldsymbol{w}_k^T \overline{\boldsymbol{x}})$$
$$= argmax_k\ \boldsymbol{w}_k^T \overline{\boldsymbol{x}}$$

  - Sparsity:
  - Parallelization:

Numerical Issues

- Logistic Regression is subject to overfitting
  - When not enough data, linear regression can classify each data point arbitrarily well (i.e., $\Pr(correct\ class) \to 1$)
- Problems: $weights \to \pm\infty$
  Hessian → singular
- Picture

Regularization

- Solution: penalize large weights

- Objective: $\min_{\mathbf{w}} L(\boldsymbol{w}) + \frac{1}{2}\lambda\left|\left|\boldsymbol{w}\right|\right|_2^2$

$$= \min_{\boldsymbol{w}} - \sum_n y_n \ln \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n) + (1 - y_n) \ln(1 - \sigma(\boldsymbol{w}^T \overline{\boldsymbol{x}}_n)) + \frac{1}{2}\lambda \boldsymbol{w}^T \boldsymbol{w}$$

- Hessian

$$\boldsymbol{H} = \overline{\boldsymbol{X}}\boldsymbol{R}\overline{\boldsymbol{X}}^T + \lambda \boldsymbol{I}$$

where $R_{nn} = \sigma(\boldsymbol{w}^{\boldsymbol{T}} \overline{\boldsymbol{x}}_n)(1 - \sigma(\boldsymbol{w}^{\boldsymbol{T}} \overline{\boldsymbol{x}}_n)$

the term λI ensures that $\boldsymbol{H}$ is not singular (eigenvalues $\geq \lambda$)

Generalized Linear Models

- How can we do non-linear regression and classification while using the same machinery?

- Idea: map inputs to a different space and do linear regression/classification in that space

Example

- Suppose the underlying function is quadratic

Basis functions

- Use non-linear basis functions:
  - Let ϕ_i denote a basis function

$$\phi_0(x) = 1$$
$$\phi_1(x) = x$$
$$\phi_2(x) = x^2$$

  - Let the hypothesis space H be

$$H = \{x \rightarrow w_0\phi_0(x) + w_1\phi_1(x) + w_2\phi_2(x) | w_i \in \Re\}$$

- If the basis functions are non-linear in x, then a non-linear hypothesis can still be found by linear regression

Common basis functions

- Polynomial: $\phi_j(x) = x^j$
- Gaussian: $\phi_j(x) = e^{-\frac{(x-\mu_j)^2}{2s^2}}$
- Sigmoid: $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$

  where $\sigma(a) = \frac{1}{1+\mathrm{e}^{-\mathrm{a}}}$
- Also Fourier basis functions, wavelets, etc.