Test Two Review

David Rosenberg

New York University

October 29, 2016

Kernels

- The kernel trick

Trees

- doesn't depend on actual values of features – just ordering within each feature

Bagging

- T/F: In bagging, as we increase the number of bootstrap samples, we expect that we will eventually overfit.
  - False.

Bagging

- T/F: In bagging, as we increase the number of bootstrap samples, we expect that we will eventually overfit.
  - False.
- With bagging, how can we get an estimate of test perfomance while still using all our data for training?
  - "out-of-bag" error

Random Forest

- T/F: Random forest is just bagging with trees.
  - False

Random Forest

- T/F: Random forest is just bagging with trees.
  - False
- T/F: Generating too many trees in a random forest will probably lead to overfitting.
  - False

Random Forest

- T/F: Random forest is just bagging with trees.
  - False
- T/F: Generating too many trees in a random forest will probably lead to overfitting.
  - False

AdaBoost

- T/F: We can use regression trees as the base classifier for AdaBoost.
  - False. AdaBoost is for hard classifiers.

AdaBoost

- T/F: We can use regression trees as the base classifier for AdaBoost.
  - False. AdaBoost is for hard classifiers.
- T/F: We can use SVM as a base classifier for AdaBoost.
  - True, if you map the output to$\{-1, 1\}$ with a modified sign function.

AdaBoost

- T/F: We can use regression trees as the base classifier for AdaBoost.
  - False. AdaBoost is for hard classifiers.
- T/F: We can use SVM as a base classifier for AdaBoost.
  - True, if you map the output to $\{-1, 1\}$ with a modified sign function.
- T/F: We can view AdaBoost a method for minimizing the exponential loss using forward stagewise additive modeling.
  - True

Gradient Boosting

- Know how to do gradient boosting with a new loss function and a black box regression algorithm.

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$
  - feature map: $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$
  - feature map: $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$
  - linear score function $(x, y) \mapsto \langle w, \Psi(x, y) \rangle$, parameterized by $w \in \mathbf{R}^d$

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$
  - feature map: $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$
  - linear score function $(x, y) \mapsto \langle w, \Psi(x, y) \rangle$, parameterized by $w \in \mathbf{R}^d$
  - final prediction function $x \mapsto \arg\max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$
  - loss functions:

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$
  - feature map: $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$
  - linear score function $(x, y) \mapsto \langle w, \Psi(x, y) \rangle$, parameterized by $w \in \mathbf{R}^d$
  - final prediction function $x \mapsto \arg\max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$
  - loss functions:
    - multiclass hinge / SVM loss

Multiclass Classification

- Understand the key pieces
  - class sensitive loss function $\Delta(y, y')$
  - feature map: $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$
  - linear score function $(x, y) \mapsto \langle w, \Psi(x, y) \rangle$, parameterized by $w \in \mathbf{R}^d$
  - final prediction function $x \mapsto \arg\max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$
  - loss functions:
    - multiclass hinge / SVM loss
    - multinomial logistic regression loss

Likelihood, MLE, Conditional Likelihood

- go through examples in slides (Poisson regression, Gaussian regression, binomial, multinomial)

Likelihood, MLE, Conditional Likelihood

- go through examples in slides (Poisson regression, Gaussian regression, binomial, multinomial)
- note that you can use these same losses for gradient boosting

Conditional Exponential Distribution

- Input: x gives location and time
- Output: y gives waiting time for taxi pickup

Conditional Exponential Distribution

- Input: x gives location and time
- Output: y gives waiting time for taxi pickup
- Exponential distributions are a natural candidate:

$$\text{ExpDists} = \left\{ p_\lambda(y) = \lambda e^{-\lambda y} 1(y \in [0, \infty)) \mid \lambda \in (0, \infty) \right\}.$$

Conditional Exponential Distribution

- Input: x gives location and time
- Output: y gives waiting time for taxi pickup
- Exponential distributions are a natural candidate:

$$\text{ExpDists} = \left\{ p_\lambda(y) = \lambda e^{-\lambda y} 1(y \in [0,\infty)) \mid \lambda \in (0,\infty) \right\}.$$

- For input x, we want to give back λ, the exponential distribution **parameter**.

Conditional Exponential Distribution

- Input: x gives location and time
- Output: y gives waiting time for taxi pickup
- Exponential distributions are a natural candidate:

$$\text{ExpDists} = \left\{ p_\lambda(y) = \lambda e^{-\lambda y} 1(y \in [0, \infty)) \mid \lambda \in (0, \infty) \right\}.$$

- For input x, we want to give back λ, the exponential distribution **parameter**.
- Let's make a generalized linear model.
- So we'll predict $x \mapsto f(w^T x)$ for some x.

Conditional Exponential Distribution

- Input: x gives location and time
- Output: y gives waiting time for taxi pickup
- Exponential distributions are a natural candidate:

$$\text{ExpDists} = \left\{ p_\lambda(y) = \lambda e^{-\lambda y} 1(y \in [0, \infty)) \mid \lambda \in (0, \infty) \right\}.$$

- For input x, we want to give back λ, the exponential distribution **parameter**.
- Let's make a generalized linear model.
- So we'll predict $x \mapsto f(w^T x)$ for some x.
- What can we use for f?

Conditional Exponential Distribution

- Taking $w^T x \mapsto \exp\left(w^T x\right)$ does the trick. Maps into $(0, \infty)$.

Conditional Exponential Distribution

- Taking $w^T x \mapsto \exp\left(w^T x\right)$ does the trick. Maps into $(0, \infty)$.
- The likelihood for observation $y \geqslant 0$ for

$$p_\lambda(y) = \lambda e^{-\lambda y}$$

Conditional Exponential Distribution

- Taking $w^T x \mapsto \exp\left(w^T x\right)$ does the trick. Maps into $(0, \infty)$.
- The likelihood for observation $y \geqslant 0$ for

$$p_\lambda(y) = \lambda e^{-\lambda y}$$

- For input x, predicted parameter is $\lambda = \exp\left(w^T x\right)$.

Conditional Exponential Distribution

- Taking $w^T x \mapsto \exp\left(w^T x\right)$ does the trick. Maps into $(0,\infty)$.
- The likelihood for observation $y \geqslant 0$ for

$$p_\lambda(y) = \lambda e^{-\lambda y}$$

- For input x, predicted parameter is $\lambda = \exp\left(w^T x\right)$.
- Likelihood of $y \mid x$ is then

$$p_w(y \mid x) = \exp\left(w^T x\right) e^{-\exp\left(w^T x\right) y}$$

Conditional Exponential Distribution

- Log-likelihood of $y \mid x$ is then

$$
\begin{aligned}
p_w(y|x) &= \exp\left(w^T x\right) e^{-\exp\left(w^T x\right) y} \\
\implies \log p_w(y \mid x) &= w^T x - y \exp\left(w^T x\right)
\end{aligned}
$$

Conditional Exponential Distribution

- Log-likelihood of $y \mid x$ is then

$$
\begin{aligned}
p_w(y|x) &= \exp\left(w^T x\right) e^{-\exp\left(w^T x\right) y} \\
\implies \log p_w(y \mid x) &= w^T x - y \exp\left(w^T x\right)
\end{aligned}
$$

- Log-likelihood of $(x_1, y_1), \ldots, (x_n, y_n)$ is

$$
\sum_{i=1}^{n} \left[w^T x_i - y_i \exp\left(w^T x_i\right) \right]
$$

- MLE is then

$$
\hat{w}_{\text{MLE}} = \underset{w \in \mathbf{R}^d}{\arg\max} \sum_{i=1}^{n} \left[w^T x_i - y_i \exp\left(w^T x_i\right) \right]
$$

Conditional Exponential Distribution with GBM?

- For linear version, we take parameter to be $f(w^T x)$.

$$\log p_w(y|x) = w^T x - y \exp\left(w^T x\right)$$

Conditional Exponential Distribution with GBM?

- For linear version, we take parameter to be $f(w^Tx)$.

$$\log p_w(y|x) \quad = \quad w^Tx - y\exp\left(w^Tx\right)$$

- Replace w^Tx by a general function $g(x)$ that we will learn with GBM.

Conditional Exponential Distribution with GBM?

- For linear version, we take parameter to be $f(w^T x)$.

$$\log p_w(y|x) \quad = \quad w^T x - y \exp\left(w^T x\right)$$

- Replace $w^T x$ by a general function $g(x)$ that we will learn with GBM.
- Log-likelihood objective function is

$$J(g) = \sum_{i=1}^{n} \left[g(x_i) - y_i \exp\left(g(x_i)\right)\right].$$

(we want to maximize it)

Conditional Exponential Distribution with GBM?

- For linear version, we take parameter to be $f(w^T x)$.

$$\log p_w(y|x) = w^T x - y \exp\left(w^T x\right)$$

- Replace $w^T x$ by a general function $g(x)$ that we will learn with GBM.
- Log-likelihood objective function is

$$J(g) = \sum_{i=1}^{n} \left[g(x_i) - y_i \exp\left(g(x_i)\right)\right].$$

  (we want to maximize it)
- Differentiate w.r.t. $g(x_i)$... etc...