Generative and Discriminative Models

CSci 5525: Machine Learning

Instructor: Paul Schrater

Generative Models and Bayes Rule

- Bayes rule states that

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

- For 2-class problem, posterior probability for C_1

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} = \frac{\exp(a)}{\exp(a)+1}$$

- Here a is the log-odds ratio:

$$a = \log\left(\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}\right)$$

- The class posterior can be written as

$$P(C_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

- Need to estimate (model): for $j = 1, 2$

Prior: $p(C_j)$ Conditional: $p(\mathbf{x}|C_j)$

Continuous Inputs: Multi-variate Gausians

- Assume class conditionals are Gaussian: different μ_j, same Σ

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k)\right\}$$

- Class labels: $y_n \in \{0, 1\}$, classes C_1, C_2:

$$\mathbf{x}_n \in C_1 \Rightarrow y_n = 1, \qquad \mathbf{x}_n \in C_2 \Rightarrow y_n = 0$$

- Class priors: $P(y_n = 1) = \pi, P(y_n = 0) = 1 - \pi$
- Likelihood of one data point $(y_n, \mathbf{x}_n)$

$$\begin{aligned} p(y_n, \mathbf{x}_n) &= p(y_n)p(\mathbf{x}_n|y_n) \\ &= \left\{\pi p(\mathbf{x}_n|\mu_1, \Sigma)\right\}^{y_n}\left\{(1-\pi)p(\mathbf{x}_n|\mu_2, \Sigma)\right\}^{1-y_n} \end{aligned}$$

Continuous Inputs: Multi-variate Gausians (Contd.)

- Likelihood of the data, assuming independence

$$
\begin{aligned}
p((\mathbf{y}, X)|\pi, \mu_1, \mu_2, \Sigma) &= p((y_1, \mathbf{x}_1), \ldots, (y_N, \mathbf{x}_N)|\pi, \mu_1, \mu_2, \Sigma) \\
&= \prod_{n=1}^{N} p(y_n, \mathbf{x}_n|\pi, \mu_1, \mu_2, \Sigma) \\
&= \prod_{n=1}^{N} \left\{ \pi p(\mathbf{x}_n|\mu_1, \Sigma) \right\}^{y_n} \left\{ (1-\pi) p(\mathbf{x}_n|\mu_2, \Sigma) \right\}^{1-y_n}
\end{aligned}
$$

- Estimate parameters by maximizing log-likelihood

$$
\begin{aligned}
&\log p((\mathbf{y}, X)|\pi, \mu_1, \mu_2, \Sigma) \\
&= \sum_{n=1}^{N} \left\{ y_n \log(\pi p(\mathbf{x}_n|\mu_1, \Sigma)) + (1-y_n) \log((1-\pi) p(\mathbf{x}_n|\mu_2, \Sigma)) \right\}
\end{aligned}
$$

Maximum Likelihood Estimation

- Log-likelihood of the data

$$\log p((\mathbf{y}, X)|\pi, \mu_1, \mu_2, \Sigma)$$
$$= \sum_{n=1}^{N} \left\{ y_n \log(\pi p(\mathbf{x}_n|\mu_1, \Sigma)) + (1 - y_n) \log((1 - \pi) p(\mathbf{x}_n|\mu_2, \Sigma)) \right\}$$

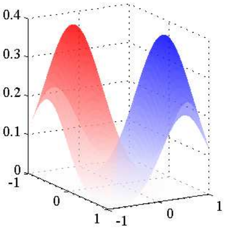
- Optimizing over the parameters $(\pi, \{\mu_1, \mu_2\}, \Sigma)$

$$\pi = \frac{1}{N} \sum_{n=1}^{N} y_n = \frac{N_1}{N_1 + N_2}$$
$$\mu_k = \frac{1}{N_k} \sum_{\mathbf{x}_n \in C_k} \mathbf{x}_n \ , k = 1, 2$$
$$\Sigma = \sum_{k=1}^{2} \frac{N_k}{N} \left(\frac{1}{N_k} \sum_{\mathbf{x} \in C_k} (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T \right)$$

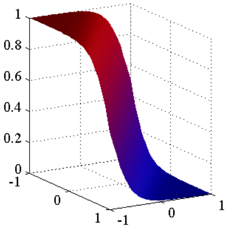
Prediction: 2-class problems

- For 2-class problem

$$
\begin{aligned}
p(C_1|\mathbf{x}) &= \sigma(\mathbf{w}^T\mathbf{x} + w_0) \\
\mathbf{w} &= \Sigma^{-1}(\mu_1 - \mu_2) \\
w_0 &= -\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \log\frac{p(C_1)}{p(C_2)}
\end{aligned}
$$



Class conditionals



Class posteriors

Generative Models and Bayes Rule: K-class

- Recall that

$$p(\mathbf{x}) = \sum_{j=1}^{K} p(\mathbf{x}, C_j) = \sum_{j=1}^{K} p(C_j)p(\mathbf{x}|C_j)$$

- For K-class problem, posterior probability for C_k

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_{j=1}^{K} p(\mathbf{x}|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}$$

- Here, a_k is given by

$$a_k = \log p(\mathbf{x}|C_k)p(C_k) = \log p(\mathbf{x}|C_k) + \log p(C_k)$$

- Need to estimate (model): for $k = 1, 2, \ldots, K$

Prior: $p(C_k)$ Conditional: $p(\mathbf{x}|C_k)$

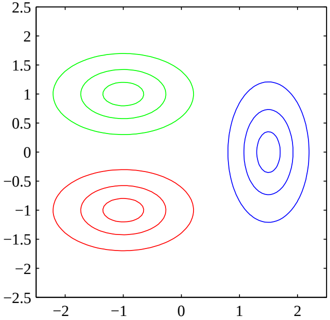
- Make "parametric" assumptions about the conditional $p(\mathbf{x}|C_k)$

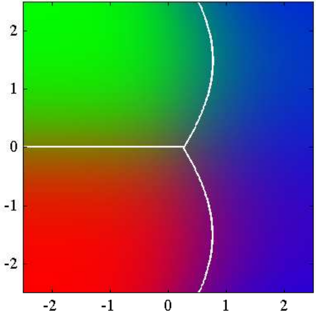
Prediction: K-class problems

- For K-class problem

$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$
$$\mathbf{w}_k = \Sigma^{-1} \mu_k$$
$$w_{k0} = -\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log p(C_k)$$

- If Σ is the same for each class: Linear Discriminant
- If Σ is not the same for each class: Quadratic Discriminant





Naive Bayes: Conditional Independence of Features

- Generative models need to specify $p(\mathbf{x}|C_k)$
- Conditional independence (CI) simplifies the specification

$$p(\mathbf{x}|C_k) = p(x_1, \ldots, x_D|C_k) = \prod_{i=1}^{D} P(x_i|C_k)$$

- Factorized form for $p(\mathbf{x}|C_k)$
- Sufficient to specify marginal distributions $p(x_i|C_k)$
- Examples:
  - Binary $x_i \in \{0, 1\}$, Bernoulli distribution $p(x_i|C_k) = \mu_{ik} \in [0, 1]$
  - Count $x_i \in \{0, 1, 2, \ldots\}$, multinomial, Poisson, etc.
  - Real $x_i \in \mathbb{R}$, univariate Gaussian $p(x_i|C_k) = \mathcal{N}(\mu_{ik}, \sigma^2_{ik})$

Naive Bayes: Binary Features, Bernoulli Marginals

- Assume binary features $x_i \in \{0, 1\}$
- Bernoulli marginals: for feature i, class k, $\mu_{ik} \in [0, 1]$

$$p(x_i = 1|C_k) = \mu_{ik} \qquad p(x_i = 0|C_k) = 1 - \mu_{ik}$$

- Assume conditional independence of features

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{D} p(x_i|C_k) = \prod_{i=1}^{D} \mu_{ik}^{x_i}(1 - \mu_{ik})^{1-x_i}$$

- For K-classes, we have

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \{x_i \log \mu_{ik} + (1 - x_i) \log(1 - \mu_{ik})\} + \log p(C_k)$$

Naive Bayes: Count Features

- Assume count features $x_i \in \{0, 1, 2, \ldots\}$
- Probability of x_i occurring $p(x_i|C_k) = \pi_{ik} \in [0, 1]$
- Probability of x_i occurring n_{ik} times, assuming CI

$$p(\underbrace{x_i, \ldots, x_i}_{n_{ik} \text{ times}}|C_k) = \prod_{j=1}^{n_{ik}} p(x_i|C_k) = \pi_{ik}^{n_{ik}}$$

- Naive-Bayes model for text classification
  - $\pi_{ik} = p(x_i|C_k)$: probability of word x_i is class C_k
- Assume W words total, $n_{\mathbf{x}}$ words in $\mathbf{x}$
- Assuming conditional independence

$$p(\mathbf{x}|C_k) = p(x_1, \ldots, x_{n_{\mathbf{x}}}|C_k) = \prod_{i=1}^{W} \pi_{ki}^{n_{ik}}$$

- For K-classes, we have

$$a_k(\mathbf{x}) = \sum_{i=1}^{W} n_{ik} \log \pi_{ik} + \log p(C_k)$$

Naive Bayes: Real-valued features

- Assume count features $x_i \in \mathbb{R}$
- Marginal Gaussian distribution $p(x_i|C_k) = \mathcal{N}(\mu_{ik}, \sigma_{ik}^2)$
- Joint distribution is multivariate Gaussian, $\Sigma_k = \mathrm{diag}(\sigma_{ik}^2)$

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{D} p(x_i|C_k) = \frac{1}{(2\pi)^{D/2}\left(\prod_{i=1}^{D}\sigma_{ik}\right)} \exp\left\{-\sum_{i=1}^{D}\frac{(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}\right\}$$

- For K-classes, we have

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \log p(x_i|C_k) + \log p(C_k)$$

Discriminative Models and Bayes Rule

- Bayes rule states that

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \ .$$

- Generative models make assumptions about $p(\mathbf{x}|y)$
- Discriminative models
  - Make assumptions about $p(y|\mathbf{x})$
  - There is no attempt to model $p(\mathbf{x})$
  - Does not solve a more general problem

Logistic Regression (2 Class)

- Assume a 2 class problem with $\mathbf{x} \in \mathbb{R}^D$ and $y \in \{0, 1\}$
- Logistic Regression assumes

$$\log\left(\frac{P(1|\mathbf{x})}{P(0|\mathbf{x})}\right) = \mathbf{w}^T\mathbf{x}$$

- The log-odds ratio is affine in $\mathbf{x}$
- A direct calculation gives

$$\begin{aligned} P(1|\mathbf{x}) &= \frac{\exp(\mathbf{w}^T\mathbf{x})}{1+\exp(\mathbf{w}^T\mathbf{x})} = \sigma(\mathbf{w}^T\mathbf{x}) \\ P(0|\mathbf{x}) &= \frac{1}{1+\exp(\mathbf{w}^T\mathbf{x})} = 1-\sigma(\mathbf{w}^T\mathbf{x}) \end{aligned}$$

Exponential Family and Logistic Regression

- An exponential family has density

$$p(\mathbf{x};\eta) = \exp(\eta^T \mathbf{x}) g(\eta) h(\mathbf{x})$$

  - Family is determined by "partition function" $g(\cdot)$
  - Specific distribution has parameter η
  - Example: Gaussian, Bernoulli, Multinomial, Beta, etc.
- Assume $p(\mathbf{x}|C_k)$ are exponential family distributions
  - Belong to the same family, i.e., same $g(\cdot)$
  - Each class has a different parameter $\eta_h, h = 1, \ldots, k$
- The log-odds ratio of the posterior

$$\log\left(\frac{P(C_h|\mathbf{x})}{P(C_k|\mathbf{x})}\right) = \mathbf{w}^T \mathbf{x} + w_0$$

- Exponential family assumption leads to affine log-odds
- Logistic regression models have lower "bias"

Logistic Regression as a Bernoulli Model

- Logistic regression: $P(1|\mathbf{x}) = \pi, P(0|\mathbf{x}) = (1-\pi)$
- Logistic regression as Bernoulli model with $\pi = \pi(\mathbf{w};\mathbf{x})$
- Likelihood of label y: sample from Bernoulli distribution

$$p(y;\pi) = \pi^y(1-\pi)^{1-y} = \exp\left\{\ln\left(\frac{\pi}{1-\pi}\right)y\right\}(1-\pi)$$

- Maximize likelihood of training set labels $\{y_1, \ldots, y_N\}$ w.r.t. π

$$\max_{\pi} \prod_{n=1}^{N} p(y_n;\pi)$$

- Recall that $\pi = \pi(\mathbf{w};\mathbf{x})$ with

$$\pi(\mathbf{w};\mathbf{x}) = \frac{\exp(\mathbf{w}^T\mathbf{x})}{1+\exp(\mathbf{w}^T\mathbf{x})} \qquad \text{and} \qquad \ln\left(\frac{\pi(\mathbf{w};\mathbf{x})}{1-\pi(\mathbf{w};\mathbf{x})}\right) = \mathbf{w}^T\mathbf{x}$$

Logistic Regression: Training

- Training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ with $y_n \in \{0, 1\}$
- Then

$$P(1|\mathbf{x}) = \frac{\exp(\mathbf{w}^T\mathbf{x})}{1 + \exp(\mathbf{w}^T\mathbf{x})} = \sigma\left(\mathbf{w}^T\mathbf{x}\right)$$

- Likelihood, assuming independence

$$P(\mathbf{y}|X) = \frac{p(\mathbf{y}, X)}{p(X)} = \prod_{n=1}^{N} P(y_i|\mathbf{x}_i) = \prod_{n=1}^{N} P(1|\mathbf{x}_n)^{y_n}(1 - P(1|\mathbf{x}_n))^{(1-y_n)}$$

- Log-likelihood, to be maximized

$$\begin{aligned} G(\mathbf{w}) &= \sum_{n=1}^{N} \{y_n \log P(1|\mathbf{x}_n) + (1 - y_n) \log(1 - P(1|\mathbf{x}_n))\} \\ &= \sum_{n=1}^{N} \left\{y_n \mathbf{w}^T\mathbf{x}_n - \log(1 + \exp(\mathbf{w}^T\mathbf{x}_n))\right\} \end{aligned}$$

Logistic Regression: Training (Contd)

- Let

$$\pi_n = \pi(\mathbf{w}; \mathbf{x}_n) = \frac{\exp(\mathbf{w}^T \mathbf{x}_n)}{1 + \exp(\mathbf{w}^T \mathbf{x}_n)} = \sigma(\mathbf{w}^T \mathbf{x}_n)$$

- The negative log-likelihood, to be minimized

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \left\{ y_n \mathbf{w}^T \mathbf{x}_n - \log(1 + \exp(\mathbf{w}^T \mathbf{x}_n)) \right\}$$

- The gradient of the objective function

$$\nabla E(\mathbf{w}_t) = \sum_{n=1}^{N} (\pi(\mathbf{w}_t; \mathbf{x}_n) - y_n) \mathbf{x}_n = X^T (\pi(\mathbf{w}_t; X) - \mathbf{y})$$

  - Our notation: $X^T = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N]$ is $D \times N$

- Convex objective: Can use gradient descent, step-size α_t

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \nabla E(\mathbf{w}_t)$$

Iteratively Reweighted Least Squares (IRLS)

- We want to solve $\nabla E(\mathbf{w}) = X^T(\pi - \mathbf{y}) = 0$
- From Newton-Raphson iterative optimization

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - H^{-1}(\mathbf{w}^{\text{old}})\nabla E(\mathbf{w}^{\text{old}})$$

- The Hessian for logistic regression

$$H = \nabla^2 E(\mathbf{w}) = \sum_{n=1}^{N} \pi_n(1 - \pi_n)\mathbf{x}_n\mathbf{x}_n^T = X^T R X$$

  - R is a diagonal matrix with $R_{nn} = \pi_n(1 - \pi_n)$
- Hence, the Newton-Raphson updates

$$\begin{aligned}
\mathbf{w}^{\text{new}} &= \mathbf{w}^{\text{old}} - (X^T R X)^{-1} X^T(\pi - \mathbf{y}) \\
&= (X^T R X)^{-1}\left\{X^T R X \mathbf{w}^{\text{old}} - X^T(\pi - \mathbf{y})\right\} \\
&= (X^T R X)^{-1} X^T R \mathbf{z}
\end{aligned}$$

where $\mathbf{z} = X\mathbf{w}^{\text{old}} - R^{-1}(\pi - \mathbf{y})$

Iterative Reweighted Least Squares (IRLS) (Contd.)

- The update for logistic regression

$$\mathbf{w}^{\text{new}} = (X^T R X)^{-1} X^T R \mathbf{z}$$

- Recall the solution to the least squares regression

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}$$

- However, since $\pi = \sigma(\mathbf{w}^T \mathbf{x})$
  - An update of $\mathbf{w}$ updates π
  - An update of π updates R, $R_{nn} = \pi_n(1 - \pi_n)$
- We have to repeatedly solve the update equation for $\mathbf{w}^{\text{new}}$
- Convergence and scalability of IRLS

Multi-class Logistic Regression

- The class posteriors are given by:

$$p(C_k|\mathbf{x}) = \pi_k(\mathbf{w}_k; \mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \ , \quad a_k = \mathbf{w}_k^T \mathbf{x}$$

- The likelihood can be written using $\mathbf{y}_n$ (1-of-K coding)

$$p(\mathbf{y}|\mathbf{w}_1, \dots, \mathbf{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(C_k|\mathbf{x}_n)^{y_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_{nk}^{y_{nk}}$$

$$E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\log p(\mathbf{y}|\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} \log \pi_{nk}$$

- We can similarly compute gradient, Hessian, and do updates

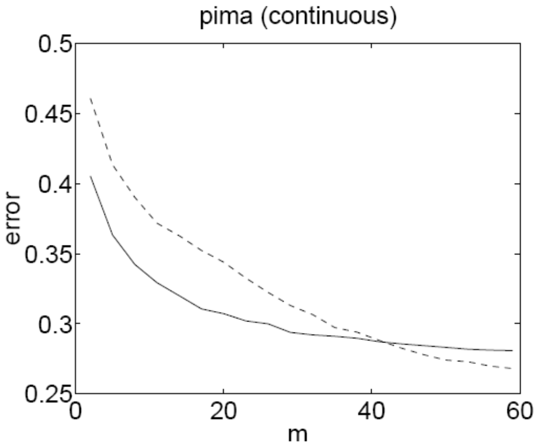
$$\nabla_{\mathbf{w}_k} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = \sum_{n=1}^{N} (\pi_{nk} - y_{nk}) \mathbf{x}_n$$

$$\nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N} \pi_{nk} (I_{kj} - \pi_{nk}) \mathbf{x}_n \mathbf{x}_n^T$$

Generative Vs Discriminative

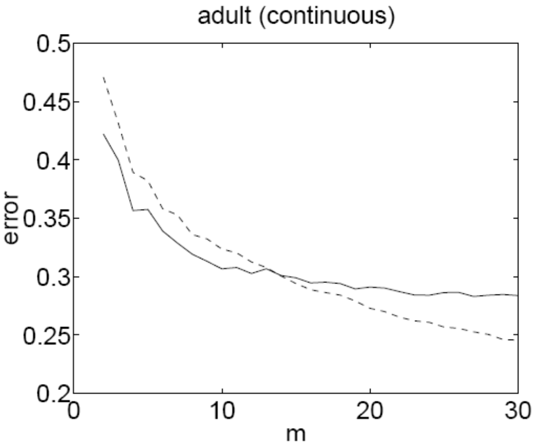
- Generative models make explicit assumptions on $p(\mathbf{x}|y)$
  - Solves a more general problem, finds $p(\mathbf{x})$
  - Has higher "bias" (focuses on a smaller set of models)
  - Converges faster to asymptotic performance
  - There are consistent estimation algorithms
  - True error rate may be high if assumptions are not appropriate
- Logistic regression makes assumptions on $p(y|\mathbf{x})$
  - Does not solve a more general problem
  - Has "lower bias" (focuses on a bigger set of models)
  - Convergence to asymptotic performance is slower
  - Careful consistency analysis is required
  - True error rate may be lower due to low bias

Results: Pima



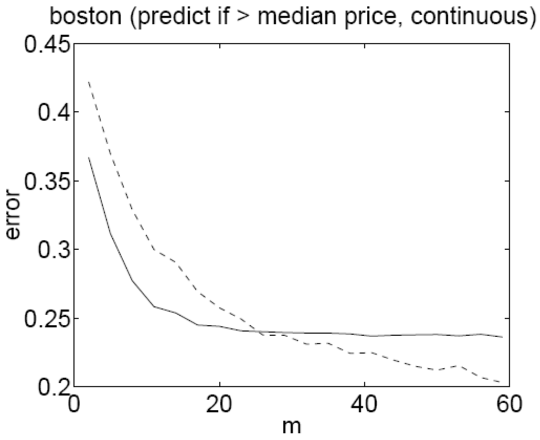
Bold = Naive Bayes, Dashed = Logistic Regression

Results: Adult



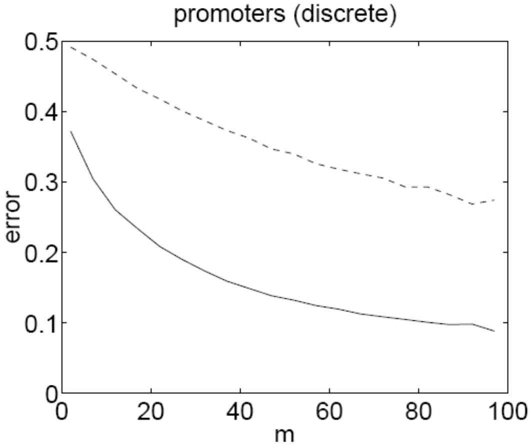
Bold = Naive Bayes, Dashed = Logistic Regression

Results: Boston



Bold = Naive Bayes, Dashed = Logistic Regression

Results: Promoters



Bold = Naive Bayes, Dashed = Logistic Regression