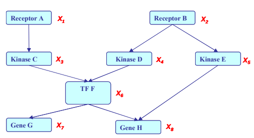
School of Computer Science
Carnegie Mellon

Statistical learning with basic graphical models

Probabilistic Graphical Models (10-708)

Lecture 7, part II
Oct 10, 2007



Eric Xing

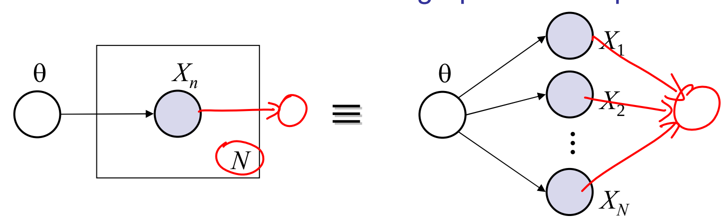
Reading: J-Chap. 5,6,7 KF-Chap. 8, 15

Announcements

- Condensed set of slides used in this lecture
  - Expanded set posted on the class web site: please read it
  - Some topics may be elaborated in recitation: please do attend
- Project Descriptions due by 12:00am tonight
- Homework 2 out: due next Wednesday
- Feedback on Homeworks 1 and 2 at the end of the class
  - Difficulty?
  - Time?

Before we start: A note on Plates notation

- A plate is a "macro" that allows subgraphs to be replicated



  - We can represent this as a Bayes net with *N* nodes.
    - The rules of plates are simple: repeat every structure in a box a number of times given by the integer in the corner of the box (e.g. N), updating the plate index variable (e.g. n) as you go.
    - Duplicate every arrow going into the plate and every arrow leaving the plate by connecting the arrows to each copy of the structure.

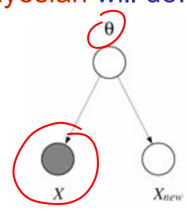
Last time we discussed…

- One node GMs
  - Parameter estimation for the Bernoulli distribution
    - Frequentist: Maximum Likelihood
    - Bayesian: MAP, Posterior mean

ML, MAP vs Full Bayesian estimation

- $\hat{\theta}_{MAP}$ is not Bayesian (even though it uses a prior) since it is a point estimate.
- Consider predicting the future. A sensible way is to combine predictions based on all possible values of θ, weighted by their posterior probability, this is what a Bayesian will do:

$$
\begin{aligned}
p(x_{new} \mid \mathbf{x}) &= \int p(x_{new}, \theta \mid \mathbf{x}) d\theta \\
&= \int p(x_{new} \mid \theta, \mathbf{x}) p(\theta \mid \mathbf{x}) d\theta \\
&= \int p(x_{new} \mid \theta) p(\theta \mid \mathbf{x}) d\theta
\end{aligned}
$$



- A frequentist will typically use a "plug-in" estimator such as ML/MAP:

$$p(x_{new} \mid \mathbf{x}) = p(x_{new} \mid \hat{\theta}_{ML}), \quad \text{or,} \quad p(x_{new} \mid \mathbf{x}) = p(x_{new} \mid \hat{\theta}_{MAP})$$

  - The Bayesian estimate will collapse to MAP for concentrated posterior

Eric Xing 5

Frequentist vs. Bayesian

- This is a "theological" war.
- Advantages of Bayesian approach:
  - Mathematically elegant.
  - Works well when amount of data is much less than number of parameters
  - Easy to do incremental (sequential) learning.
  - Can be used for model selection (max likelihood will always pick the most complex model).
- Advantages of frequentist approach:
  - Mathematically/ computationally simpler.
  - "objective", unbiased, invariant to reparameterization
- As $|D| \to \infty$, the two approaches become the same:

$$p(\theta \mid D) \to \delta(\theta, \hat{\theta}_{ML})$$

Eric Xing 6

Discrete Distributions

- Bernoulli distribution: Ber(p)

$$P(x) = \begin{cases} 1-p & \text{for } x = 0 \\ p & \text{for } x = 1 \end{cases} \quad \Rightarrow \quad P(x) = p^x (1-p)^{1-x}$$

$p \quad (1-p)$

- Multinomial distribution: Mult(1, θ)
  - Multinomial (indicator) variable:

$$X = \begin{bmatrix} X^1 \\ X^2 \\ X^3 \\ X^4 \\ X^5 \\ X^6 \end{bmatrix} \quad \text{where} \quad \begin{array}{l} X^j = [0,1], \quad \text{and} \quad \sum_{j \in [1,\dots,6]} X^j = 1 \\ X^j = 1 \text{ w.p. } \theta_j, \quad \sum_{j \in [1,\dots,6]} \theta_j = 1 \;. \end{array}$$

$$p(x(j)) = P(\{X_j = 1, \text{where } j \text{ index the dice-face}\})$$

$$= \theta_j = \theta_1^{x^1} \times \theta_2^{x^2} \times \theta_3^{x^3} \times \theta_4^{x^4} \times \theta_5^{x^5} \times \theta_6^{x^6} = \prod_k \theta_k^{x^k} = \theta^x$$

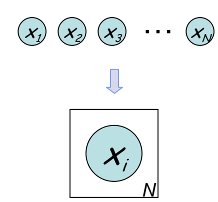
Example: multinomial model

- Data:
  - We observed N ***iid*** die rolls (K-sided): D={5, 1, K, …, 3}
- The likelihood of dataset $D=\{x_1, \dots, x_N\}$:

$$P(x_1, x_2, \dots, x_N \mid \theta) = \prod_{n=1}^{N} P(x_n \mid \theta) = \prod_{n=1}^{N} \left(\prod_k \theta_k^{x_n^k} \right)$$

$$= \prod_k \theta_k^{\sum_n^N x_n^k} = \prod_k \theta_k^{n_k}$$

GM:



MLE: constrained optimization with Lagrange multipliers

- Objective function:

$$\ell(\theta; D) = \log P(D \mid \theta) = \log \prod_k \theta_k^{n_k} = \sum_k n_k \log \theta_k$$

- We need to maximize this subject to the constraint $\sum_{k=1}^{K} \theta_k = 1$
- Constrained cost function with a Lagrange multiplier

$$\tilde{\ell} = \sum_k n_k \log \theta_k + \lambda \left(1 - \sum_{k=1}^{K} \theta_k \right)$$

- Take derivatives wrt θ_k

$$\frac{\partial \tilde{\ell}}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0$$

$$n_k = \lambda \theta_k \Rightarrow \sum_k n_k = N = \lambda \sum_k \theta_k = \lambda$$

$$\Longrightarrow \quad \hat{\theta}_{k,MLE} = \frac{n_k}{N} \quad \text{or} \quad \hat{\theta}_{MLE} = \frac{1}{N} \sum_n x_n$$

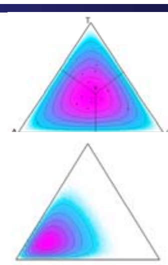
Frequency as sample mean

- Sufficient statistics
  - The counts, $\vec{n} = (n_1, \cdots, n_K),\ n_k = \sum_n x_n^k$, are **sufficient statistics** of data D

Bayesian estimation:

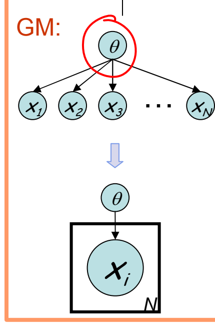
- Dirichlet distribution:

$$P(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} = C(\alpha) \prod_k \theta_k^{\alpha_k - 1}$$



- Posterior distribution of θ:

$$P(\theta \mid x_1, \ldots, x_N) = \frac{p(x_1, \ldots, x_N \mid \theta) p(\theta)}{p(x_1, \ldots, x_N)} \propto \prod_k \theta_k^{n_k} \prod_k \theta_k^{\alpha_k - 1} = \prod_k \theta_k^{\alpha_k + n_k - 1}$$



  - Notice the isomorphism of the posterior to the prior,
  - such a prior is called a **conjugate prior**

- Posterior mean estimation:

$$\frac{n_k}{N} \cdot \frac{N}{N+|\alpha|} + \frac{\alpha_k}{|\alpha|} \cdot \frac{|\alpha|}{N+|\alpha|}$$

$$\theta_k = \int \theta_k p(\theta \mid D) d\theta = C \int \theta_k \prod_k \theta_k^{\alpha_k + n_k - 1} d\theta = \frac{n_k + \alpha_k}{N + |\alpha|}$$

Dirichlet parameters can be understood as pseudo-counts

More in HW!

Sequential Bayesian updating

- Start with Dirichlet prior $P(\vec{\theta} \mid \vec{\alpha}) = \mathrm{Dir}(\vec{\theta} : \vec{\alpha})$
- Observe N' samples with sufficient statistics $\vec{n}'$. Posterior becomes:

$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}')$$

- Observe another N'' samples with sufficient statistics $\vec{n}''$. Posterior becomes:

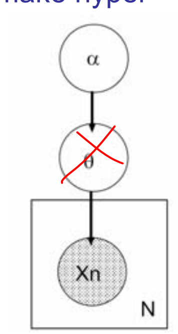
$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}', \vec{n}'') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}' + \vec{n}'')$$

- So sequentially absorbing data in any order is equivalent to batch update.

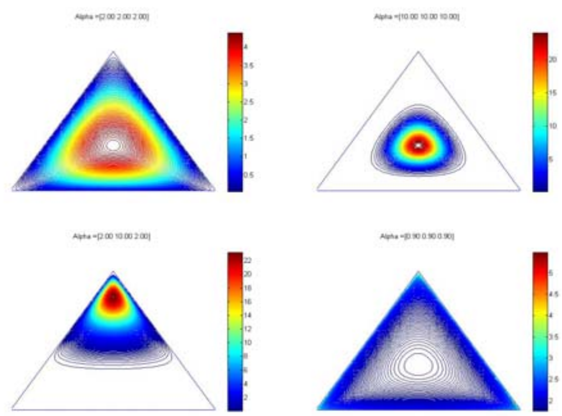
Hierarchical Bayesian Models

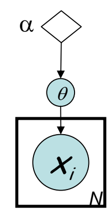
- θ are the parameters for the likelihood $p(x \mid \theta)$
- α are the parameters for the prior $p(\theta \mid \alpha)$.
- We can have hyper-hyper-parameters, etc.
- We stop when the choice of hyper-parameters makes no difference to the marginal likelihood; typically make hyper-parameters constants.
- Where do we get the prior?
  - Intelligent guesses
  - Empirical Bayes (Type-II maximum likelihood)
    → computing point estimates of α:

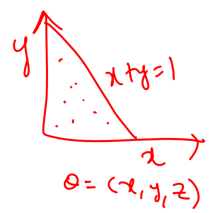
$$\hat{\vec{\alpha}}_{MLE} = \arg\max_{\vec{\alpha}} = p(\vec{n} \mid \vec{\alpha})$$



Limitation of Dirichlet Prior:







The Logistic Normal Prior

$\theta \sim LN_K(\mu, \Sigma)$

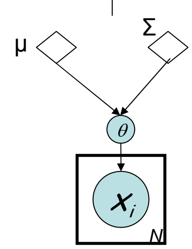
$\gamma \sim N_{K-1}(\mu, \Sigma) \qquad \gamma_K = 0$

$\theta_i = \exp\left\{\gamma_i - \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)\right\}$

$C(\gamma) = \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)$

- Log Partition Function
- Normalization Constant

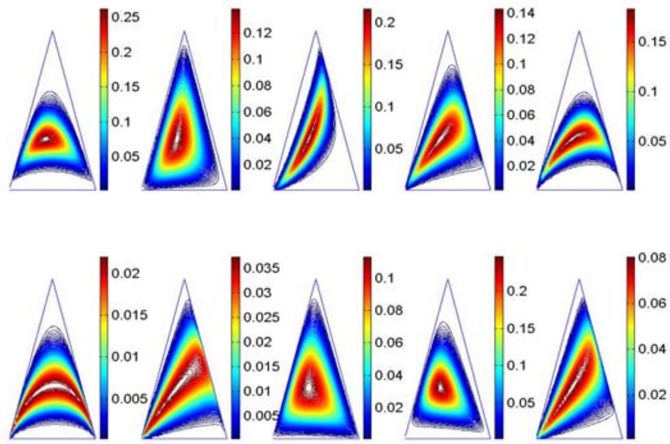
Problem



- Pro: co-variance structure
- Con: non-conjugate (we will discuss how to solve this later)

Logistic Normal Densities

Logistic Normal



Example 2: univariate-Gaussian

- Data:
  - We observed N ***iid*** real samples: D={-0.1, 10, 1, -5.2, …, 3}
- Model: $P(x) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left\{-(x-\mu)^2/2\sigma^2\right\}$
- Log likelihood:

$$\ell(\theta; D) = \log P(D \mid \theta) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{n=1}^{N}\frac{(x_n-\mu)^2}{\sigma^2}$$

- MLE: take derivative and set to zero:

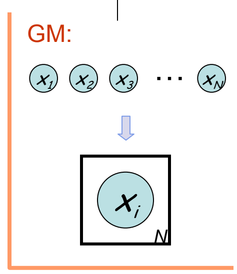
$$\frac{\partial \ell}{\partial \mu} = (1/\sigma^2)\sum_n (x_n - \mu)$$

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_n (x_n-\mu)^2$$

$$\Rightarrow$$

$$\mu_{MLE} = \frac{1}{N}\sum_n (x_n)$$

$$\sigma^2_{MLE} = \frac{1}{N}\sum_n (x_n - \mu_{ML})^2$$



MLE for a multivariate-Gaussian

- It can be shown that the MLE for μ *and* Σ is

$$x_n = \begin{pmatrix} x_n^1 \\ x_n^2 \\ \vdots \\ x_n^K \end{pmatrix}$$

$$X = \begin{pmatrix} ---x_1^T--- \\ ---x_2^T--- \\ \vdots \\ ---x_N^T--- \end{pmatrix}$$

$$\mu_{MLE} = \frac{1}{N}\sum_n (x_n)$$

$$\Sigma_{MLE} = \frac{1}{N}\sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \frac{1}{N} S$$

where the scatter matrix is

$$S = \sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum_n x_n x_n^T\right) - N\mu_{ML}\mu_{ML}^T$$

- The sufficient statistics are $\Sigma_n x_n$ and $\Sigma_n x_n x_n^T$.
- Note that $X^T X = \Sigma_n x_n x_n^T$ may not be full rank (eg. if $N < D$), in which case Σ_{ML} is not invertible

Bayesian parameter estimation for a Gaussian

- There are various reasons to pursue a Bayesian approach
  - We would like to update our estimates sequentially over time.
  - We may have prior knowledge about the expected magnitude of the parameters.
  - The MLE for Σ may not be full rank if we don't have enough data.
- We will restrict our attention to conjugate priors.
- Various cases, in order of increasing complexity:
  - Known σ, unknown μ
  - Known μ, unknown σ
  - Unknown μ and σ

Bayesian estimation: unknown μ, known σ

- Normal Prior:

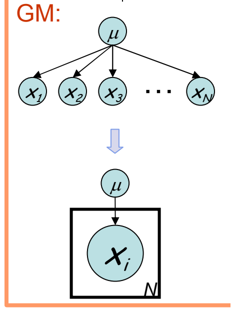
$$P(\mu) = \left(2\pi\tau^2\right)^{-1/2} \exp\left\{-(\mu-\mu_0)^2/2\tau^2\right\}$$

- Joint probability:

$$P(x,\mu) = \underbrace{\left(2\pi\sigma^2\right)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}}_{} \quad P(x|\mu)$$

$$\times \underbrace{\left(2\pi\tau^2\right)^{-1/2} \exp\left\{-(\mu-\mu_0)^2/2\tau^2\right\}}_{} \quad P(\mu|\mu_0,\tau)$$

GM:



- Posterior:

$$P(\mu|x) = \left(2\pi\tilde{\sigma}^2\right)^{-1/2} \exp\left\{-(\mu-\tilde{\mu})^2/2\tilde{\sigma}^2\right\}$$

where $\tilde{\mu} = \frac{N/\sigma^2}{N/\sigma^2+1/\tau^2}\bar{x} + \frac{1/\tau^2}{N/\sigma^2+1/\tau^2}\mu_0$, and $\frac{1}{\tilde{\sigma}^2} = \left(\frac{N}{\sigma^2}+\frac{1}{\tau^2}\right)$

Sample mean

Bayesian estimation: unknown μ, known σ

$$\mu_N = \frac{N/\sigma^2}{N/\sigma^2+1/\sigma_0^2}\bar{x} + \frac{1/\sigma_0^2}{N/\sigma^2+1/\sigma_0^2}\mu_0 \qquad \frac{1}{\tilde{\sigma}^2} = \left(\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}\right)$$

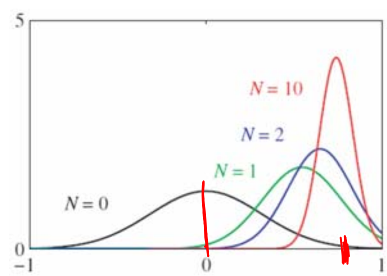
- The posterior mean is a convex combination of the prior and the MLE, with weights proportional to their respective relative precisions.
- The precision of the posterior $1/\sigma^2_N$ is the precision of the prior $1/\sigma^2_0$ plus one contribution of data precision $1/\sigma^2$ for each observed data point.

- Sequentially updating the mean
  - $\mu* = 0.8$ (unknown), $(\sigma^2)* = 0.1$ (known)
  - Effect of single data point

$$\mu_1 = \mu_0 + (x-\mu_0)\frac{\sigma_0^2}{\sigma^2+\sigma_0^2}$$

  - Uninformative (vague/ flat) prior, $\sigma^2_0 \to \infty$

$$\mu_N \to x$$



Summary

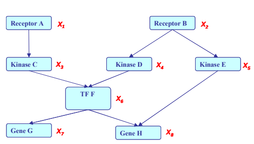
- Learning scenarios:
  - Objective function
  - Frequentist and Bayesian
- Learning single-node GM – density estimation
  - Typical discrete distribution
  - Typical continuous distribution
  - Conjugate priors

School of Computer Science
Carnegie Mellon

Learning two-node GMs

Probabilistic Graphical Models (10-708)

Lecture 8, Oct 10, 2007

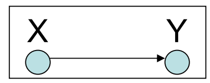


Eric Xing

Reading: J-Chap. 5,6,7 KF-Chap. 8,15

Two node GMs

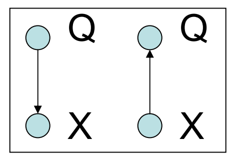
Conditional mixtures



Linear Regression

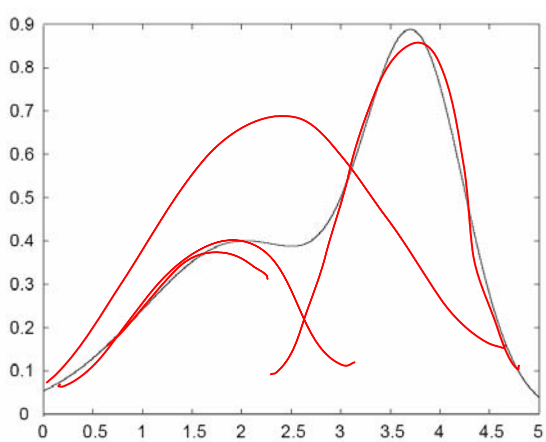
Classification

Generative and discriminative approaches



Multimodal models

- A bimodal probability density:



Conditional Gaussian

- The data:

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_N, y_N)\}$$

- Both nodes are observed:
  - Y is a class indicator vector

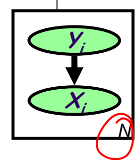
$$p(y_n) = \mathrm{multi}(y_n : \pi) = \prod_k \pi_k^{y_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean

$$p(x_n \mid y_n^k = 1, \mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\tfrac{1}{2\sigma^2} (x_n - \mu_k)^2 \right\}$$

$$p(x \mid y, \mu, \sigma) = \prod_n \left(\prod_k N(x_n : \mu_k, \sigma)^{y_n^k} \right)$$

GM:

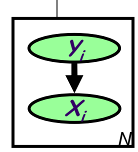


MLE of conditional Gaussian

- Data log-likelihood

$$\ell(\boldsymbol{\theta}; D) = \log \prod_n p(x_n, y_n) = \log \prod_n p(y_n \mid \pi) p(x_n \mid y_n, \mu, \sigma)$$

GM:



- MLE

$$\hat{\pi}_{k,MLE} = \arg\max_{\pi} \ell(\boldsymbol{\theta}; D), \qquad \hat{\pi}_{k,MLE} = \sum_n y_n^k \Big/ N = n_k \big/ N$$

the fraction of samples of class m

$$\hat{\mu}_{k,MLE} = \arg\max \ell(\boldsymbol{\theta}; D), \qquad \hat{\mu}_{k,MLE} = \frac{\sum_n y_n^k x_n}{\sum_n y_n^k} = \frac{\sum_n y_n^k x_n}{n_k}$$

the average of samples of class m

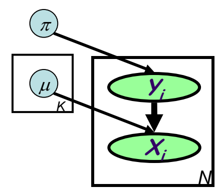
Bsyesian estimation of conditional Gaussian

- Prior:

$$P(\vec{\pi} \mid \vec{\alpha}) = \mathrm{Dir}(\vec{\pi} : \vec{\alpha})$$

$$P(\mu_k \mid \nu) = \mathrm{Normal}(\mu_k : \nu, \tau)$$

GM:



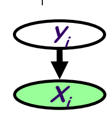
- Posterior mean (Bayesian est.)

$$\pi_{k,Bayes} = \frac{N}{N+|\alpha|}\widehat{\pi}_{k,ML} + \frac{|\alpha|}{N+|\alpha|}\frac{\alpha_k}{|\alpha|} = \frac{n_k + \alpha_k}{N+|\alpha|}$$

$$\mu_{k,Bayes} = \frac{n_k/\sigma^2}{n_k/\sigma^2 + 1/\tau^2}\widehat{\mu}_{k,ML} + \frac{1/\tau^2}{n_k/\sigma^2 + 1/\tau^2}\nu, \quad \text{and} \quad \sigma^2_{Bayes} = \left(\frac{N}{\sigma^2} + \frac{1}{\tau^2}\right)^{-1}$$

Classification

- From conditional density modeling to classification:
  - The joint probability of a datum and it label is:



$$p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times p(x_n \mid y_n^k = 1, \mu, \sigma)$$ $P(y_n)$.

$$= \pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\tfrac{1}{2\sigma^2}(x_n - \mu_k)^2 \right\}$$ $P(x_n|y_n)$

$P(y_n|x_n) = ?$

  - Given a datum x_n, we predict its label using the conditional probability of the label given the datum:

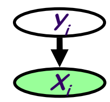
$$p(y_n^k = 1 \mid x_n, \mu, \sigma) = \frac{\pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\tfrac{1}{2\sigma^2}(x_n - \mu_k)^2 \right\}}{\sum_{k'} \pi_{k'} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\tfrac{1}{2\sigma^2}(x_n - \mu_{k'})^2 \right\}}$$

- This is basic inference
  - introduce evidence, and then normalize

Naïve Bayes Classifier

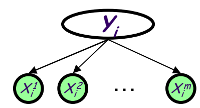
- When X is multivariate-Gaussian vector:
  - The joint probability of a datum and it label is:

$$p(\vec{x}_n, y_n^k = 1 \mid \vec{\mu}, \Sigma) = p(y_n^k = 1) \times p(\vec{x}_n \mid y_n^k = 1, \vec{\mu}, \Sigma)$$
$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\vec{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\vec{x}_n - \vec{\mu}_k)\right\}$$

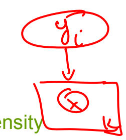


  - The naïve Bayes simplification

$$p(x_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times \prod_j p(x_n^j \mid y_n^k = 1, \mu_{k,j}, \sigma_{k,j})$$
$$= \pi_k \prod_j \frac{1}{(2\pi\sigma_{k,j}^2)^{1/2}} \exp\left\{-\tfrac{1}{2\sigma_{k,j}^2}(x_n^j - \mu_{k,j})^2\right\}$$

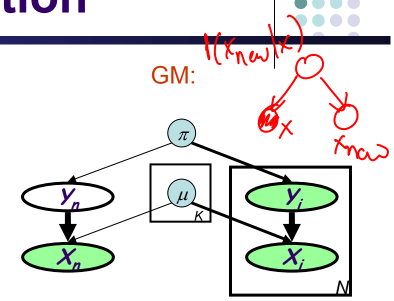


  - More generally: $p(x_n, y_n \mid \eta, \pi) = p(y_n \mid \pi) \times \prod_{j=1}^{m} p(x_n^j \mid y_n, \eta)$
    - Where $p(.\,|\,.)$ is an arbitrary conditional (discrete or continuous) 1-D density



Transductive classification

- Given X_n, what is its corresponding Y_n when we know the answer for a set of training data?



- Frequentist prediction:
  - we fit π, μ and σ from data first, and then …

$$p(y_n^k = 1 \mid x_n, \mu, \sigma, \pi) = \frac{p(y_n^k = 1, x_n \mid \mu, \sigma, \pi)}{p(x_n \mid \mu, \sigma, \pi)} = \frac{\pi_k N(x_n, \mid \mu_k, \sigma)}{\sum_j \pi_j N(x_n, \mid \mu_j, \sigma)}$$

- Bayesian:
  - we compute the posterior dist. of the parameters first …

The predictive distribution

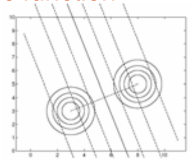
- Understanding the predictive distribution

$$p(y_n^k = 1 \mid x_n, \mu, \sigma, \pi) = \frac{p(y_n^k = 1, x_n \mid \mu, \sigma, \pi)}{p(x_n \mid \mu, \sigma)} = \frac{\pi_k N(x_n, \mid \mu_k, \sigma)}{\sum_j \pi_j N(x_n, \mid \mu_j, \sigma)}$$

- For two class (i.e., *K*=2), * turns out to be the logistic function

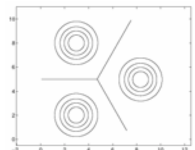
$$p(y_n^1 = 1 \mid x_n) = \frac{1}{1 + \frac{\pi_2 \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x_n - \mu_2)^2\right\}}{\pi_1 \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x_n - \mu_1)^2\right\}}} = \frac{1}{1 + \exp\left\{-x_n \frac{1}{\sigma^2}(\mu_1 - \mu_2) + \log \frac{\pi_2}{\pi_1}\right\}}$$

$$= \frac{1}{1 + e^{-\theta^T x_n}}$$



- For multiple class (i.e., *K*>2), * correspond to a softmax function

$$p(y_n^k = 1 \mid x_n) = \frac{e^{-\theta_k^T x_n}}{\sum_j e^{-\theta_j^T x_n}}$$

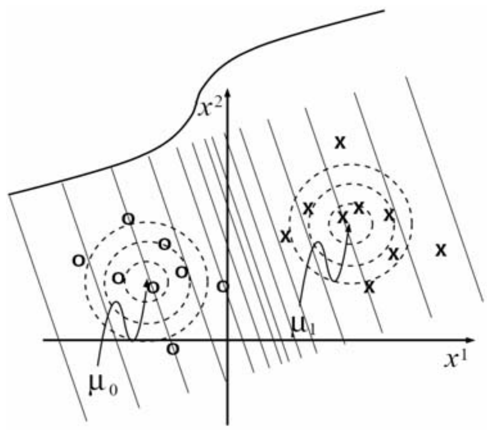


Discussion

- We've seen how to learning two-node model $p(y_n, x_n)$, but in certain problems the goal is to learning $p(y_n \mid x_n)$
- Can we model $p(y_n \mid x_n)$ directly?
- How?

Generative and discriminative classifiers

- Generative:
  - Modeling the joint distribution of all data
- Discriminative:
  - How?



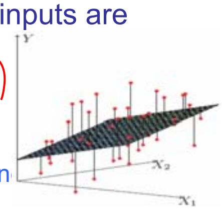
Linear Regression: A discriminative model

- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

$p(y|x)$

N



where ε is an error term of unmodeled effects or random n

- Now assume that ε follows a Gaussian $N(0,\sigma)$, then we have:

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

- By independence assumption:

$$L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Linear regression

- Hence the log-likelihood is:

$$l(\theta) = n\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta^T \mathbf{x}_i)^2$$

- Do you recognize the last term?

Yes it is: $$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$

- It is same as the MSE!

The Least-Mean-Square (LMS) method

- The Cost Function:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$

- Consider a **gradient descent** algorithm:

$$\theta^{t+1} = \theta^t - \boxed{\alpha}\nabla J(\theta)\big|_t$$

$$\nabla J(\theta) = \frac{1}{2}\sum_{i=1}^{n} 2(\bar{x}_i^T\theta - y_i)\bar{x}_i$$

$$\theta^{t+1} = \theta^t + \alpha\sum_{i=1}^{n}(y_i - \bar{x}_i^T\theta)\bar{x}_i$$

The Least-Mean-Square (LMS) method

- Now we have the following descent rule:

$$\theta^{t+1} = \theta^t + \alpha \cancel{\sum_{i=1}^{n}} (y_n - \mathbf{x}_n^T \theta^t)\mathbf{x}_n$$

  - This is as a **batch** gradient descent algorithm

- For a single training point, we have:

$$\theta^{t+1} = \theta^t + \alpha (y_i - \mathbf{x}_i^T \theta^t)\mathbf{x}_i$$

  - This is known as the LMS update rule, or the Widrow-Hoff learning rule
  - This can be used as a **on-line** algorithm

Eric Xing 37

The normal equations

$x^T A x$ $= 1\times 1$
$1\times n$ $n\times n$ $n\times 1$

- Write the cost function in matrix form:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$$

$$= \frac{1}{2}(X\theta - \vec{y})^T (X\theta - \vec{y})$$

$\theta^T X^T y$

$$= \frac{1}{2}\left(\theta^T X^T X\theta - 2\theta^T X^T y - \cancel{y^T X\theta} + y^T y\right)$$

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2 & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n & -- \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} \quad X\theta$$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1^T\theta \\ x_2^T\theta \\ \vdots \\ x_n^T\theta \end{bmatrix}$$

$$\sum_{i=1}^{n}(x_i^T\theta - y_i)^2$$

- To minimize $J(\theta)$, take derivative and set to zero:

$$\nabla J(\theta) = \frac{1}{\cancel{2}}\left(\cancel{2}X^T X\theta - \cancel{2}X^T y\right) = 0$$

$$X^T X\theta = X^T y$$

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

$$\Rightarrow \boxed{X^T X\theta = X^T\vec{y}}$$

The normal equations

$$\Downarrow$$

$$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$

Eric Xing 38

A recap:

- LMS update rule

$$\theta^{t+1} = \theta^t + \alpha(y_n - \mathbf{x}_n^T \theta^t)\mathbf{x}_n$$

  - Pros: on-line, low per-step cost
  - Cons: coordinate, maybe slow-converging
- Steepest descent

$$\theta^{t+1} = \theta^t + \alpha \sum_{i=1}^{n} (y_n - \mathbf{x}_n^T \theta^t)\mathbf{x}_n$$

  - Pros: fast-converging, easy to implement
  - Cons: a batch,
- Normal equations

$$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$

  - Pros: a single-shot algorithm! Easiest to implement.
  - Cons: need to compute pseudo-inverse $(X^T X)^{-1}$, expensive, numerical issues (e.g., matrix is singular ..)

Multivariate Linear Regression

- Consider vector-valued input $X \in \mathbf{R}^k$ leading to vector-valued output $Y \in \mathbf{R}^d$ via regression matrix $A \in \mathbf{R}^{k \times d}$:

$$p(y \mid x) = \frac{1}{(2\pi)^{-d/2} |\Sigma|^{-1/2}} \exp\left\{ -\frac{1}{2}(y - Ax)^T \Sigma^{-1} (y - Ax) \right\}$$

- Log-(conditional-) likelihood

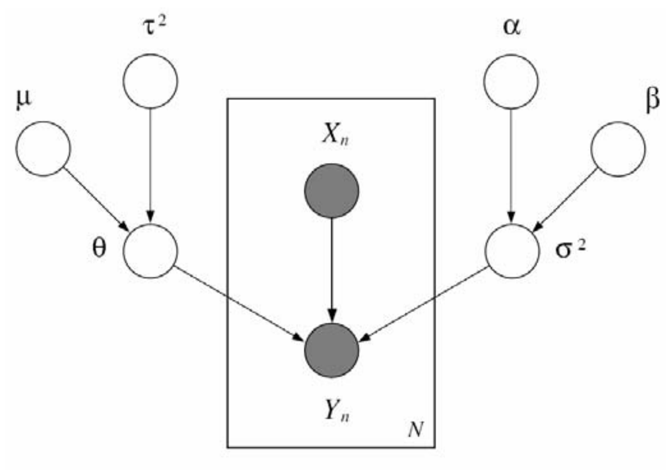
$$\ell = -\frac{1}{2}\sum_n |\Sigma| - \frac{1}{2}\sum_n (y_n - Ax_n)^T \Sigma^{-1} (y_n - Ax_n) + c$$

- To take derivatives wrt a matrix, we use the following identity

$$\frac{\partial\left((\mathrm{M}a + b)^T \mathrm{C}(\mathrm{M}a + b)\right)}{\partial \mathrm{M}} = (\mathrm{C} + \mathrm{C}^T)(\mathrm{M}a + b)a^T$$

$$\text{where } \mathrm{M} = A, a = -x_n, b = y_n \text{ and } \mathrm{C} = \Sigma^{-1}$$

Bayesian linear regression



Bayesian Linear regression: L2 regularization

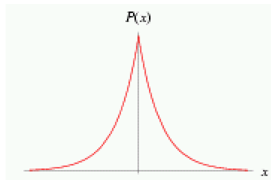
- Let
$$p(\theta \mid \lambda) = \left(\frac{\lambda}{\pi}\right)^{N/2} \exp\left(-\lambda(\theta - 0)^T(\theta - 0)^T\right)$$
- The joint likelihood:
$$p(y_i, \theta \mid x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right) \times \left(\frac{\lambda}{\pi}\right)^{N/2} \exp\left(-\lambda|\theta|_2^2\right)$$
- The "regularized" regression cost function
$$J(\theta) = (y_i - \theta^T \mathbf{x}_i)^2 + \lambda|\theta|_2^2$$
  - Regularization term restricts large value components
  - Smooth and convex,
  - Can be computed directly (O(n³))
  - Or can use iterative methods (e.g. conjugate gradients method)

Bayesian Linear regression: Laplace Prior and Sparsity

- The Laplace prior:

$$p(\theta_k \mid \lambda) = \frac{\lambda}{2}\exp\left(-\lambda|\theta_k|\right)$$

$$p(\theta \mid \lambda) = \frac{\lambda}{2}\exp\left(-\lambda|\theta|_1\right)$$



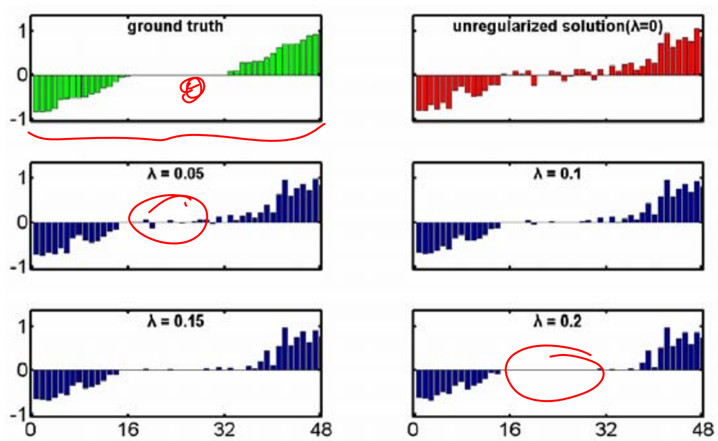
- The joint likelihood:

$$p(y_i, \theta \mid x_i) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right) \times \frac{\lambda}{2}\exp\left(-\lambda|\theta|_1\right)$$

- The "regularized" regression cost function

$$J(\theta) = (y_i - \theta^T \mathbf{x}_i)^2 + \lambda|\theta|_1$$

Effects of L1-Regularization



Select λ by cross-validation

Recall the condition-Gaussian classifier

- So we have seen a new scheme based on LMS (ML) to learn two node GM: $p(y \mid x;\theta) = \mathcal{N}\left(y;\theta^T x,\sigma^2\right)$ discriminatively
  - Gradient descent
  - Normal equation

$p(y=1|x) = \mu(x)$ $= \frac{1}{1+e^{\theta^T x}}$ $p(y|x)$

$\bigcirc y$

$\uparrow$

$\bigcirc x$

- How can we use this scheme to learning the conditional Gaussian classifier discriminatively?

$p(y=0|x) = 1-\mu(x).$

  - Recall that $\underbrace{p(y \mid x)} = \mu(x)^y (1-\mu(x))^{1-y}$ $p^y(1-p)^{1-y}$

where $\mu(x) = \underbrace{\frac{1}{1+e^{-\theta^T x}}}$ $= \frac{e^{\theta_k^T x}}{\sum_j e^{\theta_j^T x}}$

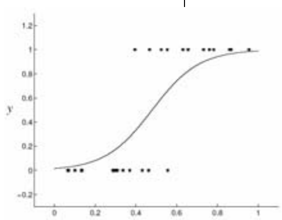
Logistic regression (sigmoid classifier)

- The condition distribution: a Bernoulli

$$p(y \mid x) = \mu(x)^y (1-\mu(x))^{1-y}$$

where μ is a logistic function

$$\mu(x) = \frac{1}{1+e^{-\theta^T x}}$$



- We can used the brute-force gradient method as in Linear Regression
- But we can also apply generic laws by observing the $p(y|x)$ is an exponential family function, more specifically, a generalized linear model (see next lecture!)

Summary

- Conditional Density Est.
- Classification
  - Generative classifier
  - Discriminative classifier
- Linear Regression
  - Algorithms
    - LMS
    - Steepest descent
    - Normal equation
  - Regularized regression vs. Bayesian regression

- Feedback on Homeworks 1 and 2
  - Difficulty?
  - Time?