CPSC 540: Machine Learning

Density Estimation, Multivariate Gaussian

Mark Schmidt

University of British Columbia

Winter 2017

Admin

- Assignment 2:
  - Due Monday.
  - 1 late day to hand it in next Wednesday.
  - 2 late days to hand it in the Wednesday after that.
- Office hours this week:
  - On Thursday in ICICS 193 from 4-5:30 with me.
  - On Friday in usual place at usual time with Robbie.
  - Typos in Assignment 2 Question 1, please check the updates.
- Class cancelled next Wednesday, February 8th:
  - So you can go to the TensorFlow lecture at the same time in (check website)

Multiple Kernel Learning

- Last time we discussed kernelizing L2-regularized linear models,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} f(Xw, y) + \frac{\lambda}{2}\|w\|^2 \Leftrightarrow \operatorname*{argmin}_{v \in \mathbb{R}^n} f(Kv, y) + \frac{\lambda}{2}\|v\|_K^2,$$

under fairly general conditions.

Multiple Kernel Learning

- Last time we discussed kernelizing L2-regularized linear models,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} f(Xw, y) + \frac{\lambda}{2}\|w\|^2 \Leftrightarrow \operatorname*{argmin}_{v \in \mathbb{R}^n} f(Kv, y) + \frac{\lambda}{2}\|v\|_K^2,$$

  under fairly general conditions.
- What if we have multiple kernels and dont' know which to use?
  - Cross-validation.

Multiple Kernel Learning

- Last time we discussed kernelizing L2-regularized linear models,

$$\underset{w \in \mathbb{R}^d}{\text{argmin}}\, f(Xw, y) + \frac{\lambda}{2}\|w\|^2 \Leftrightarrow \underset{v \in \mathbb{R}^n}{\text{argmin}}\, f(Kv, y) + \frac{\lambda}{2}\|v\|_K^2,$$

  under fairly general conditions.
- What if we have multiple kernels and dont' know which to use?
  - Cross-validation.
- What if we have multiple potentially-relevant kernels?

Multiple Kernel Learning

- Last time we discussed kernelizing L2-regularized linear models,

$$\underset{w\in\mathbb{R}^d}{\text{argmin}}\, f(Xw,y) + \frac{\lambda}{2}\|w\|^2 \Leftrightarrow \underset{v\in\mathbb{R}^n}{\text{argmin}}\, f(Kv,y) + \frac{\lambda}{2}\|v\|_K^2,$$

  under fairly general conditions.
- What if we have multiple kernels and dont' know which to use?
  - Cross-validation.
- What if we have multiple potentially-relevant kernels?
  - Multiple kernel learning:

$$\underset{v_1\in\mathbb{R}^n, v_2\in\mathbb{R}^n,\dots,v_k\in\mathbb{R}^n}{\text{argmin}}\, f\left(\sum_{c=1}^{k} K_c v_c, y\right) + \frac{1}{2}\sum_{c=1}^{k}\lambda_c\|v\|_{K_c}.$$

  - Defines a valid kernel and is convex if f is convex.

Multiple Kernel Learning

- Last time we discussed kernelizing L2-regularized linear models,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} f(Xw, y) + \frac{\lambda}{2}\|w\|^2 \Leftrightarrow \operatorname*{argmin}_{v \in \mathbb{R}^n} f(Kv, y) + \frac{\lambda}{2}\|v\|_K^2,$$

  under fairly general conditions.
- What if we have multiple kernels and dont' know which to use?
  - Cross-validation.
- What if we have multiple potentially-relevant kernels?
  - Multiple kernel learning:

$$\operatorname*{argmin}_{v_1 \in \mathbb{R}^n, v_2 \in \mathbb{R}^n, \dots, v_k \in \mathbb{R}^n} f\left(\sum_{c=1}^{k} K_c v_c, y\right) + \frac{1}{2}\sum_{c=1}^{k} \lambda_c \|v\|_{K_c}.$$

  - Defines a valid kernel and is convex if f is convex.
  - Group L1-regularization of parameters associated with each kernel.
    - Selects a sparse set of kernels.
  - Hiearchical kernel learning:
    - Use structured sparsity to search through exponential number of kernels.

Last Time: Kernel Methods and Fenchel Duality

- We discussed valid kernels: functions k that define inner product.
  - Need $K \succeq 0$ for all inputs.

Last Time: Kernel Methods and Fenchel Duality

- We discussed valid kernels: functions k that define inner product.
  - Need $K \succeq 0$ for all inputs.
- We discussed how L2-regularized linear models,

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} f_i(w^T x_i) + \frac{\lambda}{2}\|w\|^2,$$

have Fenchel duals of the form

$$\underset{z \in \mathbb{R}^n}{\operatorname{argmax}} - \underbrace{\sum_{i=1}^{n} f_i^*(z_i)}_{\text{separable}} - \frac{1}{2\lambda} \underbrace{\|X^T z\|^2}_{z^T X X^T z}.$$

where f_i^* are convex conjugates.

Last Time: Kernel Methods and Fenchel Duality

- We discussed valid kernels: functions k that define inner product.
  - Need $K \succeq 0$ for all inputs.
- We discussed how L2-regularized linear models,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^n f_i(w^T x_i) + \frac{\lambda}{2}\|w\|^2,$$

  have Fenchel duals of the form

$$\operatorname*{argmax}_{z \in \mathbb{R}^n} - \underbrace{\sum_{i=1}^n f_i^*(z_i)}_{\text{separable}} - \frac{1}{2\lambda} \underbrace{\|X^T z\|^2}_{z^T X X^T z}.$$

  where f_i^* are convex conjugates.
- Dual problem allows kernels, is smooth, and allows coordinate optimization.
- We also discussed large-scale kernel methods,
  - Kernels with special structure, subsampling methods, explicit feature construction.

Unconstrained and Smooth Optimization

- For typical unconstrained/smooth optimization of ML problems,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w^T x_i) + \frac{\lambda}{2} \|w\|^2.$$

  we discussed several methods:
  - Gradient method:
    - Linear convergence but $O(nd)$ iteration cost.
    - Faster versions like Nesterov/Newton exist.

Unconstrained and Smooth Optimization

- For typical unconstrained/smooth optimization of ML problems,

$$\underset{w \in \mathbb{R}^d}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} f_i(w^T x_i) + \frac{\lambda}{2} \|w\|^2.$$

  we discussed several methods:
  - Gradient method:
    - Linear convergence but $O(nd)$ iteration cost.
    - Faster versions like Nesterov/Newton exist.
  - Coordinate optimization:
    - Faster than gradient method if iteration cost is $O(n)$.
  - Stochastic subgradient:
    - Iteration cost is $O(d)$ but sublinear convergence rate.
    - SAG/SVRG improve to linear rate for finite datasets.

Constrained and Non-Smooth Optimization

- For typical constrained/non-smooth optimization of ML problems, the "optimal" method for large d is subgradient methods.

Constrained and Non-Smooth Optimization

- For typical constrained/non-smooth optimization of ML problems, the "optimal" method for large d is subgradient methods.
- But we discussed better methods for specific cases:
  - Smoothing which doesn't work quite as well as we would like.

Constrained and Non-Smooth Optimization

- For typical constrained/non-smooth optimization of ML problems, the "optimal" method for large d is subgradient methods.
- But we discussed better methods for specific cases:
  - Smoothing which doesn't work quite as well as we would like.
  - Coordinate optimization if g is separable.
    - L1-regularization problems.
  - Projected-gradient for "simple" constraints.
    - Non-negative constraints or sum-to-1 constraints.
  - Proximal-gradient if g is "simple".
    - Group L1-regulaization and structure sparsity.

Constrained and Non-Smooth Optimization

- For typical constrained/non-smooth optimization of ML problems, the "optimal" method for large d is subgradient methods.
- But we discussed better methods for specific cases:
  - Smoothing which doesn't work quite as well as we would like.
  - Coordinate optimization if g is separable.
    - L1-regularization problems.
  - Projected-gradient for "simple" constraints.
    - Non-negative constraints or sum-to-1 constraints.
  - Proximal-gradient if g is "simple".
    - Group L1-regulaization and structure sparsity.
  - Projected- and Proximal-Newton for expensive f_i and simple constraints or g.
    - Density estimation (coming next).

Constrained and Non-Smooth Optimization

- For typical constrained/non-smooth optimization of ML problems, the "optimal" method for large d is subgradient methods.
- But we discussed better methods for specific cases:
  - Smoothing which doesn't work quite as well as we would like.
  - Coordinate optimization if g is separable.
    - L1-regularization problems.
  - Projected-gradient for "simple" constraints.
    - Non-negative constraints or sum-to-1 constraints.
  - Proximal-gradient if g is "simple".
    - Group L1-regulaization and structure sparsity.
  - Projected- and Proximal-Newton for expensive f_i and simple constraints or g.
    - Density estimation (coming next).
- With a few more tricks, you can almost always beat subgradient methods:
  - Dual optimzation for smoothing strongly-convex problems.
  - ADMM: for "simple" regularized composed with affine function like $\|Ax\|_1$.
  - Frank-Wolfe: for nuclear-norm regularization.
  - Mirror descent: for probability-simplex constraints.

Even Bigger Problems?

- What about datasets that don't fit on one machine?
  - We need to consider parallel and distributed optimization.

Even Bigger Problems?

- What about datasets that don't fit on one machine?
  - We need to consider parallel and distributed optimization.
- Major issues:
  - Synchronization: we can't wait for the slowest machine.
  - Communication: it's expensive to transfer across machines.

Even Bigger Problems?

- What about datasets that don't fit on one machine?
  - We need to consider parallel and distributed optimization.
- Major issues:
  - Synchronization: we can't wait for the slowest machine.
  - Communication: it's expensive to transfer across machines.

- "Embarassingly" parallel solution:
  - Split data across machines, each machine computes gradient of their subset.

- Fancier methods (key idea is usually that you just make step-size smaller):
  - Asyncronous stochastic gradient.
  - Parallel coordinate optimization.
  - Decentralized gradient.

Outline

Unsupervised Learning

- Supervised learning:
  - We have instances of features x^i and class labels y^i.
  - Want a program that gives y^i from corresponding x^i.
- Unsupervised learning:
  - We only have x^i values, but no explicit target labels.
  - You want to do "something" with them.

Unsupervised Learning

- Supervised learning:
  - We have instances of features x^i and class labels y^i.
  - Want a program that gives y^i from corresponding x^i.
- Unsupervised learning:
  - We only have x^i values, but no explicit target labels.
  - You want to do "something" with them.
- Some unsupervised learning tasks from CPSC 340:
  - Clustering: what types of x^i are there?
  - Association rules: which x_j and x_k occur together?
  - Outlier detection: is this a "normal" x^i?
  - Latent-factors: what "parts" are x^i made from?
  - Data visualization: what do the high-dimensional x^i look like?
  - Ranking: which are the most important x^i?

Density Estimation

- We're going to focus on the task of density estimation:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

  - What is probability of x^i for a generic feature vector x^i?

Density Estimation

- We're going to focus on the task of density estimation:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

  - What is probability of x^i for a generic feature vector x^i?

- For the training data this is easy:
  - Set $p(x^i)$ to "number of times x^i is in the training data" divided by n.

Density Estimation

- We're going to focus on the task of density estimation:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \hat{X} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

  - What is probability of x^i for a generic feature vector x^i?

- For the training data this is easy:
  - Set $p(x^i)$ to "number of times x^i is in the training data" divided by n.
- We're interested in the probability of test data,
  - What is probability of seeing feature vector $\hat{x}^i$ for a new example i.
- We're also interested in continuous x^i and estimating probability density.

Density Estimation Applications

- Density estimation could be called a “master problem” in machine learning.
  - Solving this problem lets you solve a lot of other problems.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.
  - Missing data in x^i can be "filled in" based on $p(x^i)$.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
    - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
    - Outliers could be cases where $p(x^i)$ is small.
    - Missing data in x^i can be "filled in" based on $p(x^i)$.
    - Association rules can be computed from conditionals $p(x^i_j | x^i_k)$.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.
  - Missing data in x^i can be "filled in" based on $p(x^i)$.
  - Association rules can be computed from conditionals $p(x^i_j | x^i_k)$.
  - Vector quantization can be achieved by assigning shorter code to high $p(x^i)$ values.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.
  - Missing data in x^i can be "filled in" based on $p(x^i)$.
  - Association rules can be computed from conditionals $p(x^i_j | x^i_k)$.
  - Vector quantization can be achieved by assigning shorter code to high $p(x^i)$ values.

- We can also do density estimation on (x^i, y^i) jointly:
  - Supervised learning can be done using conditional $p(y^i | x^i)$.

Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.
  - Missing data in x^i can be "filled in" based on $p(x^i)$.
  - Association rules can be computed from conditionals $p(x^i_j | x^i_k)$.
  - Vector quantization can be achieved by assigning shorter code to high $p(x^i)$ values.

- We can also do density estimation on (x^i, y^i) jointly:
  - Supervised learning can be done using conditional $p(y^i | x^i)$.
  - Feature relevance can be analyzed by looking at $p(x^i | y^i)$.
- Above, notice that y^i could be a set of variables or could have structure.
  - This is where we're going...

Bernoulli Distribution on Binary Variables

- Let's start with the simplest case: $x^i \in \{0, 1\}$ (e.g., coin flips),

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Bernoulli Distribution on Binary Variables

- Let's start with the simplest case: $x^i \in \{0, 1\}$ (e.g., coin flips),

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

- For IID data the only choice is the Bernoulli distribution:

$$p(x = 1 \mid \theta) = \theta, \quad p(x = 0 \mid \theta) = 1 - \theta.$$

Bernoulli Distribution on Binary Variables

- Let's start with the simplest case: $x^i \in \{0, 1\}$ (e.g., coin flips),

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

- For IID data the only choice is the Bernoulli distribution:

$$p(x = 1 \mid \theta) = \theta, \quad p(x = 0 \mid \theta) = 1 - \theta.$$

- We can write both cases

$$p(x|\theta) = \theta^{\mathcal{I}[x=1]}(1-\theta)^{\mathcal{I}[x=0]}, \text{ where } \mathcal{I}[y] = \begin{cases} 1 & \text{if } y \text{ is true} \\ 0 & \text{if } y \text{ is false} \end{cases}.$$

Maximum Likelihood with Bernoulli Distribution

- MLE for Bernoulli likelihood is

$$\underset{0 \leq \theta \leq 1}{\operatorname{argmax}}\, p(X|\theta) = \underset{0 \leq \theta \leq 1}{\operatorname{argmax}} \prod_{i=1}^{n} p(x^i|\theta)$$

Maximum Likelihood with Bernoulli Distribution

- MLE for Bernoulli likelihood is

$$\begin{aligned} \underset{0 \leq \theta \leq 1}{\operatorname{argmax}}\, p(X|\theta) &= \underset{0 \leq \theta \leq 1}{\operatorname{argmax}} \prod_{i=1}^{n} p(x^i|\theta) \\ &= \underset{0 \leq \theta \leq 1}{\operatorname{argmax}} \prod_{i=1}^{n} \theta^{\mathcal{I}[x^i=1]}(1-\theta)^{\mathcal{I}[x^i=0]} \end{aligned}$$

Maximum Likelihood with Bernoulli Distribution

- MLE for Bernoulli likelihood is

$$\begin{aligned}
\underset{0\leq\theta\leq 1}{\operatorname{argmax}}\, p(X|\theta) &= \underset{0\leq\theta\leq 1}{\operatorname{argmax}} \prod_{i=1}^{n} p(x^i|\theta) \\
&= \underset{0\leq\theta\leq 1}{\operatorname{argmax}} \prod_{i=1}^{n} \theta^{\mathcal{I}[x^i=1]}(1-\theta)^{\mathcal{I}[x^i=0]} \\
&= \underset{0\leq\theta\leq 1}{\operatorname{argmax}}\, \theta^{N_1}(1-\theta)^{N_0},
\end{aligned}$$

where N_1 is count of number of 1 values and N_0 is the number of 0 values.

Maximum Likelihood with Bernoulli Distribution

- MLE for Bernoulli likelihood is

$$
\begin{aligned}
\operatorname*{argmax}_{0\leq\theta\leq 1} p(X|\theta) &= \operatorname*{argmax}_{0\leq\theta\leq 1} \prod_{i=1}^{n} p(x^i|\theta) \\
&= \operatorname*{argmax}_{0\leq\theta\leq 1} \prod_{i=1}^{n} \theta^{\mathcal{I}[x^i=1]}(1-\theta)^{\mathcal{I}[x^i=0]} \\
&= \operatorname*{argmax}_{0\leq\theta\leq 1} \theta^{N_1}(1-\theta)^{N_0},
\end{aligned}
$$

  where N_1 is count of number of 1 values and N_0 is the number of 0 values.
- If you equate the derivative of the log-likelihood with zero, you get $\theta = \frac{N_1}{N_1+N_0}$.
- So if you toss a coin 50 times and it lands heads 24 times, your MLE is $24/50$.

Multinomial Distribution on Categorical Variables

- Consider the multi-category case: $x \in \{1, 2, 3, \ldots, k\}$ (e.g., rolling di),

$$X = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}.$$

Multinomial Distribution on Categorical Variables

- Consider the multi-category case: $x \in \{1, 2, 3, \ldots, k\}$ (e.g., rolling di),

$$X = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}.$$

- The categorical distribution is

$$p(x = c | \theta_1, \theta_2, \ldots, \theta_k) = \theta_c,$$

  where $\sum_{c=1}^{k} \theta_c = 1$.

- We can write this for a generic x as

$$p(x | \theta_1, \theta_2, \ldots, \theta_k) = \prod_{c=1}^{k} \theta_c^{\mathcal{I}[x=c]}.$$

Multinomial Distribution on Categorical Variables

- Using Lagrange multipliers to add constraint to log-likelihood, the MLE is

$$\theta_c = \frac{N_c}{\sum_{c'} N_{c'}}.$$

Multinomial Distribution on Categorical Variables

- Using Lagrange multipliers to add constraint to log-likelihood, the MLE is

$$\theta_c = \frac{N_c}{\sum_{c'} N_{c'}}.$$

- If we never see category 4 in the data, should we assume $\theta_4 = 0$?
  - If we assume $\theta_4 = 0$ and we have a 4 in test set, our test set likelihood is 0.

Multinomial Distribution on Categorical Variables

- Using Lagrange multipliers to add constraint to log-likelihood, the MLE is

$$\theta_c = \frac{N_c}{\sum_{c'} N_{c'}}.$$

- If we never see category 4 in the data, should we assume $\theta_4 = 0$?
  - If we assume $\theta_4 = 0$ and we have a 4 in test set, our test set likelihood is 0.
- To leave room for this possibility we often use "Laplace smoothing",

$$\theta_c = \frac{N_c + 1}{\sum_{c'}(N_{c'} + 1)}.$$

- This is like adding a "fake" example to the training set for each class.

MAP Estimation with Bernoulli Distributions

- In the binary case, a generalization of Laplace smoothing is

$$\theta = \frac{N_1 + \alpha - 1}{(N_1 + \alpha - 1) + (N_0 + \beta - 1)},$$

- We get the MLE when $\alpha = \beta = 1$, and Laplace smoothing with $\alpha = \beta = 2$.

MAP Estimation with Bernoulli Distributions

- In the binary case, a generalization of Laplace smoothing is

$$\theta = \frac{N_1 + \alpha - 1}{(N_1 + \alpha - 1) + (N_0 + \beta - 1)},$$

- We get the MLE when $\alpha = \beta = 1$, and Laplace smoothing with $\alpha = \beta = 2$.
- This is a MAP estimate under a beta prior,

$$p(\theta|\alpha, \beta) = \frac{1}{B(\alpha, \beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1},$$

where the beta function B makes the probability integrate to one,

MAP Estimation with Bernoulli Distributions

- In the binary case, a generalization of Laplace smoothing is

$$\theta = \frac{N_1 + \alpha - 1}{(N_1 + \alpha - 1) + (N_0 + \beta - 1)},$$

- We get the MLE when $\alpha = \beta = 1$, and Laplace smoothing with $\alpha = \beta = 2$.
- This is a MAP estimate under a beta prior,

$$p(\theta|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

where the beta function B makes the probability integrate to one,

$$B(\alpha, \beta) = \int_\theta \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta$$

MAP Estimation with Bernoulli Distributions

- In the binary case, a generalization of Laplace smoothing is

$$\theta = \frac{N_1 + \alpha - 1}{(N_1 + \alpha - 1) + (N_0 + \beta - 1)},$$

- We get the MLE when $\alpha = \beta = 1$, and Laplace smoothing with $\alpha = \beta = 2$.
- This is a MAP estimate under a beta prior,

$$p(\theta|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1},$$

where the beta function B makes the probability integrate to one,

$$B(\alpha,\beta) = \int_\theta \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta \quad \Rightarrow \quad \int_\theta p(\theta|\alpha,\beta)d\theta = 1.$$

- Note that $B(\alpha,\beta)$ is constant in terms of θ, it doesn't affect MAP estimate.

MAP Estimation with Categorical Distributions

- In the categorical case, a generalization of Laplace smoothing is

$$\theta_c = \frac{N_c + \alpha_c - 1}{\sum_{c'=1}^{k}(N_{c'} + \alpha_{c'} - 1)},$$

which is a MAP estimate under a Dirichlet prior,

$$p(\theta_1, \theta_2, \dots, \theta_k | \alpha_1, \alpha_2, \dots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{c=1}^{k} \theta_c^{\alpha_c - 1},$$

MAP Estimation with Categorical Distributions

- In the categorical case, a generalization of Laplace smoothing is

$$\theta_c = \frac{N_c + \alpha_c - 1}{\sum_{c'=1}^k (N_{c'} + \alpha_{c'} - 1)},$$

which is a MAP estimate under a Dirichlet prior,

$$p(\theta_1, \theta_2, \dots, \theta_k | \alpha_1, \alpha_2, \dots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{c=1}^k \theta_c^{\alpha_c - 1},$$

where B makes the multivariate distribution integrate to 1,

$$\int_{\theta_1} \int_{\theta_2} \cdots \int_{\theta_{k-1}} \int_{\theta_k} \prod_{c=1}^k \left[\theta_c^{\alpha_c - 1}\right] d\theta_k d\theta_{k-1} \cdots d\theta_2 d\theta_1.$$

- Because of MAP-regularization connection, Laplace smoothing is regularization.

General Discrete Distribution

- Now consider the case where $x \in \{0,1\}^d$ (e..g, words in e-mails):

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

General Discrete Distribution

- Now consider the case where $x \in \{0,1\}^d$ (e..g, words in e-mails):

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

- Now there are 2^d possible values of x.
  - Can't afford to even store a θ for each possible x.

General Discrete Distribution

- Now consider the case where $x \in \{0,1\}^d$ (e..g, words in e-mails):

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

- Now there are 2^d possible values of x.
  - Can't afford to even store a θ for each possible x.
  - With n training examples we see at most n unique x^i values.
  - But unless we have a small number of repeated x values, we'll hopelessly overfit.
- With finite dataset, we'll need to make assumptions...

Product of Independent Distributions

- A common assumption is that the variables are independent:

$$p(x_1, x_2, \ldots, x_d|\Theta) = \prod_{j=1}^{d} p(x_j|\theta_j).$$

Product of Independent Distributions

- A common assumption is that the variables are independent:

$$p(x_1, x_2, \ldots, x_d|\Theta) = \prod_{j=1}^{d} p(x_j|\theta_j).$$

- Now we just need to model each column of X as its own dataset:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}. \quad X_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \ldots$$

.

- A big assumption, but now you can fit Bernoulli for each variable.
  - We did this in CPSC 340 for naive Bayes.

Density Estimation and Fundamental Trade-off

- Product of independent distributions:
  - Easily estimate each θ_c but can't model many distributions.

Density Estimation and Fundamental Trade-off

- Product of independent distributions:
  - Easily estimate each θ_c but can't model many distributions.
- General discrete distribution:
  - Hard to estimate 2^d parameters but can model any distribution.

Density Estimation and Fundamental Trade-off

- Product of independent distributions:
  - Easily estimate each θ_c but can't model many distributions.
- General discrete distribution:
  - Hard to estimate 2^d parameters but can model any distribution.

- An unsupervised version of the fundamental trade-off:
  - Simple models often don't fit the data well but don't overfit much.
  - Complex models fit the data well but often overfit.

Density Estimation and Fundamental Trade-off

- Product of independent distributions:
  - Easily estimate each θ_c but can't model many distributions.
- General discrete distribution:
  - Hard to estimate 2^d parameters but can model any distribution.

- An unsupervised version of the fundamental trade-off:
  - Simple models often don't fit the data well but don't overfit much.
  - Complex models fit the data well but often overfit.

- We'll consider models that lie between these extremes:
  1. Mixture models.
  2. Graphical models.
  3. Boltzmann machines.

Outline

Univariate Gaussian

- Consider the case of a continuous variable $x \in \mathbb{R}$:

$$X = \begin{bmatrix} 0.53 \\ 1.83 \\ -2.26 \\ 0.86 \end{bmatrix}.$$

Univariate Gaussian

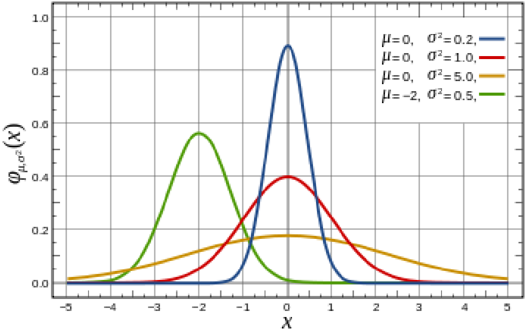
- Consider the case of a continuous variable $x \in \mathbb{R}$:

$$X = \begin{bmatrix} 0.53 \\ 1.83 \\ -2.26 \\ 0.86 \end{bmatrix}.$$

- Even with 1 variable there are many possible distributions.
- Most common is the Gaussian (or "normal") distribution:

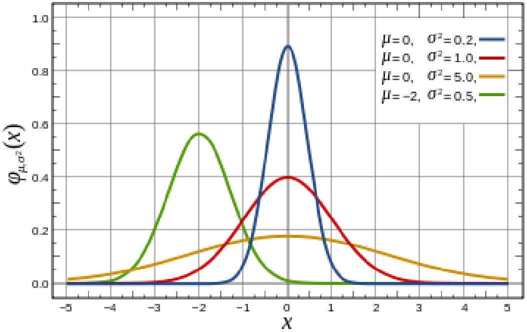
$$p(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{or} \quad x \sim \mathcal{N}(\mu,\sigma^2).$$

Univariate Gaussian



https://en.wikipedia.org/wiki/Gaussian_function

Univariate Gaussian



https://en.wikipedia.org/wiki/Gaussian_function

Negative log-likelihood for IID x^i is

$$-\log p(X|\mu,\sigma^2) = -\sum_{i=1}^{n} \log p(x^i|\mu,\sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x^i-\mu)^2 + n\log(\sigma) + \text{const.}$$

Univariate Gaussian

- Why the Gaussian distribution?
    - Central limit theorem: mean estimators converges in distribution to Gaussian.
        - Bad justification: doesn't imply data distribution converges to Gaussian.

Univariate Gaussian

- Why the Gaussian distribution?
  - Central limit theorem: mean estimators converges in distribution to Gaussian.
    - Bad justification: doesn't imply data distribution converges to Gaussian.
  - Data might actually follow Gaussian (good justification if true, but usually false).

Univariate Gaussian

- Why the Gaussian distribution?
  - Central limit theorem: mean estimators converges in distribution to Gaussian.
    - Bad justification: doesn't imply data distribution converges to Gaussian.
  - Data might actually follow Gaussian (good justification if true, but usually false).

  - Closed-form MLEs.
    - By setting derivative of log-likelihood equal to 0, MLE for mean is

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x^i.$$

and MLE for variance given $\hat{\mu}$ is

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x^i - \hat{\mu})^2,$$

for $n > 1$.

Univariate Gaussian

- Why the Gaussian distribution?
  - Central limit theorem: mean estimators converges in distribution to Gaussian.
    - Bad justification: doesn't imply data distribution converges to Gaussian.
  - Data might actually follow Gaussian (good justification if true, but usually false).

  - Closed-form MLEs.
    - By setting derivative of log-likelihood equal to 0, MLE for mean is

      $$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x^i.$$

      and MLE for variance given $\hat{\mu}$ is

      $$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x^i - \hat{\mu})^2,$$

      for $n > 1$.

  - Distribution with maximum entropy that fits mean and variance of data.
    - Beyond fitting mean/variance, it makes fewest assumptions about the data.
    - Proved via the convex conjugate of the log-likelihood.

Alternatives to Univariate Gaussian

- Why not the Gaussian distribution?
  - Negative log-likelihood is a quadratic function of μ,

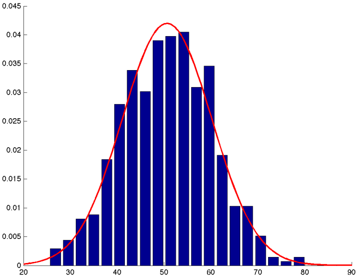
$$-\log p(X|\mu,\sigma^2) = -\sum_{i=1}^{n} \log p(x^i|\mu,\sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x^i-\mu)^2 + n\log(\sigma) + \text{const.}$$

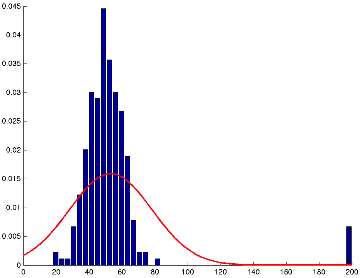
Alternatives to Univariate Gaussian

- Why not the Gaussian distribution?
  - Negative log-likelihood is a quadratic function of μ,

$$-\log p(X|\mu,\sigma^2) = -\sum_{i=1}^{n} \log p(x^i|\mu,\sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x^i-\mu)^2 + n\log(\sigma) + \text{const.}$$

so as with least squares the Gaussian is not robust to outliers.



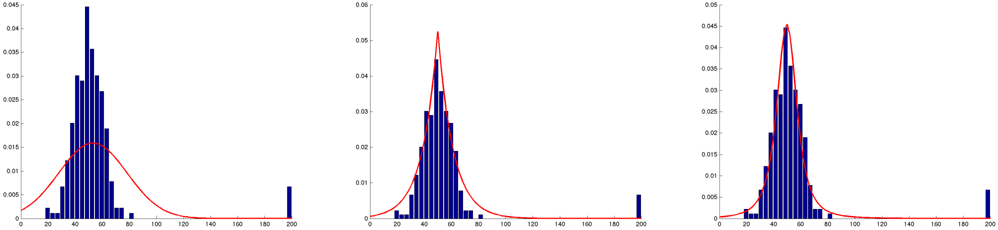


Alternatives to Univariate Gaussian

- Why not the Gaussian distribution?
  - Negative log-likelihood is a quadratic function of μ,

$$-\log p(X|\mu,\sigma^2) = -\sum_{i=1}^{n} \log p(x^i|\mu,\sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x^i-\mu)^2 + n\log(\sigma) + \text{const.}$$

    so as with least squares the Gaussian is not robust to outliers.
  - More robust: Laplace distribution or student's t-distribution

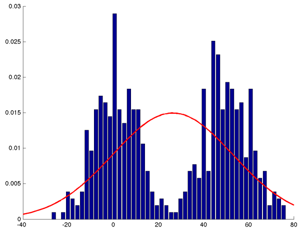


Alternatives to Univariate Gaussian

- Why not the Gaussian distribution?
  - Negative log-likelihood is a quadratic function of μ,

$$-\log p(X|\mu,\sigma^2) = -\sum_{i=1}^{n} \log p(x^i|\mu,\sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x^i-\mu)^2 + n\log(\sigma) + \text{const.}$$

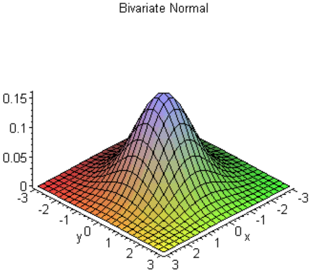
    so as with least squares distribution is not robust to outliers.
  - More robust: Laplace distribution or student's t-distribution
  - Gaussian distribution is unimodal.



Outline

Multivariate Gaussian Distribution

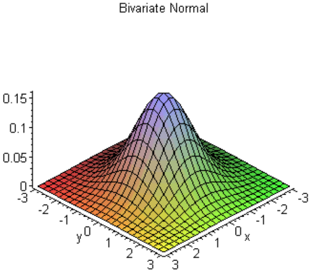
- The generalization to multiple variables is the multivariate normal/Gaussian,



http://personal.kenyon.edu/hartlaub/MellonProject/Bivariate2.html

Multivariate Gaussian Distribution

- The generalization to multiple variables is the multivariate normal/Gaussian,



http://personal.kenyon.edu/hartlaub/MellonProject/Bivariate2.html

- The probability density is given by

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \qquad \text{or } x \sim \mathcal{N}(\mu,\Sigma),$$

where $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d\times d}$ and $\Sigma \succ 0$, and $|\Sigma|$ is the determinant.

Product of Independent Gaussians

- Consider case where each variable follows independent Gaussian,

$$x_j \sim \mathcal{N}(\mu_j, \sigma_j^2),$$

Product of Independent Gaussians

- Consider case where each variable follows independent Gaussian,

$$x_j \sim \mathcal{N}(\mu_j, \sigma_j^2),$$

- In this case the joint density over all d variables is

$$\prod_{j=1}^{d} p(x_j | \mu_j, \sigma_j^2) \propto \prod_{j=1}^{d} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Product of Independent Gaussians

- Consider case where each variable follows independent Gaussian,

$$x_j \sim \mathcal{N}(\mu_j, \sigma_j^2),$$

- In this case the joint density over all d variables is

$$\begin{aligned}\prod_{j=1}^d p(x_j|\mu_j,\sigma_j^2) &\propto \prod_{j=1}^d \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right)\\ &= \exp\left(-\frac{1}{2}\sum_{j=1}^d -\frac{(x_j-\mu_j)^2}{\sigma_j^2}\right) \qquad (e^a e^b = e^{a+b})\end{aligned}$$

Product of Independent Gaussians

- Consider case where each variable follows independent Gaussian,

$$x_j \sim \mathcal{N}(\mu_j, \sigma_j^2),$$

- In this case the joint density over all d variables is

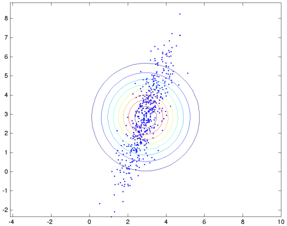
$$\begin{aligned}
\prod_{j=1}^{d} p(x_j|\mu_j, \sigma_j^2) &\propto \prod_{j=1}^{d} \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right) \\
&= \exp\left(-\frac{1}{2}\sum_{j=1}^{d} -\frac{(x_j-\mu_j)^2}{\sigma_j^2}\right) \qquad (e^a e^b = e^{a+b}) \\
&= \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right),
\end{aligned}$$

where $\mu = (\mu_1, \mu_2, \dots, \mu_d)$ and Σ is diagonal with elements σ_j^2.

- So it's a multivariate Gaussian with diagonal covariance.

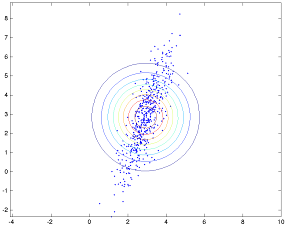
Product of Independent Gaussians

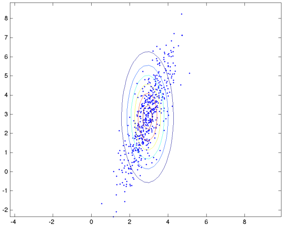
- The effect of a diagonal Σ on the multivariate Gaussian:
  - If $\Sigma = \alpha I$ the level curves are circles: 1 parameter.



Product of Independent Gaussians

- The effect of a diagonal Σ on the multivariate Gaussian:
  - If $\Sigma = \alpha I$ the level curves are circles: 1 parameter.
  - If $\Sigma = D$ (diagonal) then axis-aligned ellipses: d parameters.

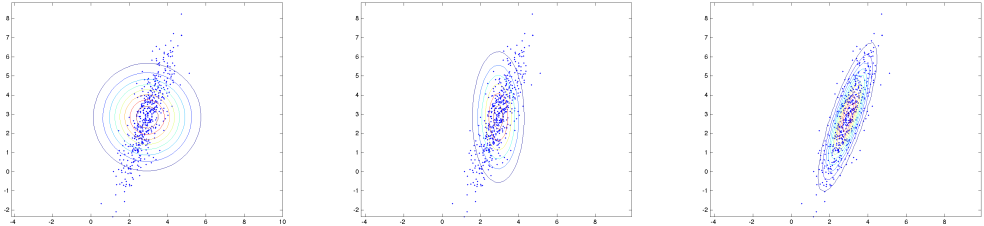




Product of Independent Gaussians

- The effect of a diagonal Σ on the multivariate Gaussian:
  - If $\Sigma = \alpha I$ the level curves are circles: 1 parameter.
  - If $\Sigma = D$ (diagonal) then axis-aligned ellipses: d parameters.
  - If Σ is dense they do not need to be axis-aligned: $d(d+1)/2$ parameters.

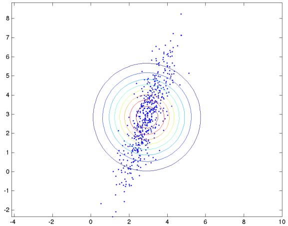
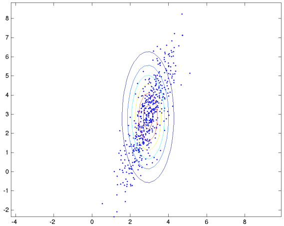
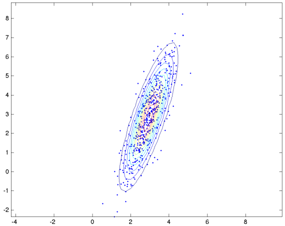
(by symmetry, we only need upper-triangular part of Σ)



Product of Independent Gaussians

- The effect of a diagonal Σ on the multivariate Gaussian:
  - If $\Sigma = \alpha I$ the level curves are circles: 1 parameter.
  - If $\Sigma = D$ (diagonal) then axis-aligned ellipses: d parameters.
  - If Σ is dense they do not need to be axis-aligned: $d(d+1)/2$ parameters.

(by symmetry, we only need upper-triangular part of Σ)

- As with the univariate Gaussian, multivariate Gaussian has closed-form MLE...

Maximum Likelihood Estimation in Multivariate Gaussians

- With a multivariate Gaussian we have

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right),$$

so up to a constant our negative log-likelihood is

$$\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu) + \frac{n}{2}\log|\Sigma|.$$

Maximum Likelihood Estimation in Multivariate Gaussians

- With a multivariate Gaussian we have

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right),$$

so up to a constant our negative log-likelihood is

$$\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu) + \frac{n}{2}\log|\Sigma|.$$

- This is quadratic in μ, taking the gradient and setting to zero gives

$$\mu = \frac{1}{n}\sum_{i=1}^{n}x^i,$$

using that $\Sigma \succ 0$ (so it's strongly-convex with unique solution).
  - MLE for μ is the average along each dimension, and it doesn't depend on Σ.

Maximum Likelihood Estimation in Multivariate Gaussians

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$\frac{1}{2}\sum_{i=1}^{n}(x^i - \mu)^T\Sigma^{-1}(x^i - \mu) + \frac{n}{2}\log|\Sigma|$$

Maximum Likelihood Estimation in Multivariate Gaussians

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$\frac{1}{2}\sum_{i=1}^{n}(x^i - \mu)^T\Sigma^{-1}(x^i - \mu) + \frac{n}{2}\log|\Sigma|$$

$$= \frac{1}{2}\sum_{i=1}^{n}\mathsf{Tr}\left((x^i - \mu)^T\Theta(x^i - \mu)\right) + \frac{n}{2}\log|\Theta^{-1}| \qquad (y^TAy = \mathsf{Tr}(y^TAy)$$

- Where the trace $\mathsf{Tr}(A)$ is the sum of the diagonal elements of A.
  - That $\mathsf{Tr}(AB) = \mathsf{Tr}(BA)$ when dimensions match is the "cyclic property".

Maximum Likelihood Estimation in Multivariate Gaussians

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu) + \frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}\text{Tr}\left((x^i-\mu)^T\Theta(x^i-\mu)\right) + \frac{n}{2}\log|\Theta^{-1}| \qquad (y^TAy = \text{Tr}(y^TAy)) \\
=&\frac{1}{2}\sum_{i=1}^{n}\text{Tr}((x^i-\mu)(x^i-\mu)^T\Theta) - \frac{n}{2}\log|\Theta| \qquad (\text{Tr}(AB) = \text{Tr}(BA))
\end{aligned}
$$

- Where the trace $\text{Tr}(A)$ is the sum of the diagonal elements of A.
    - That $\text{Tr}(AB) = \text{Tr}(BA)$ when dimensions match is the "cyclic property".

Maximum Likelihood Estimation in Multivariate Gaussians

- In terms of precision matrix Θ we have

$$= \frac{1}{2}\sum_{i=1}^{n} \mathsf{Tr}((x^i - \mu)(x^i - \mu)^T\Theta) - \frac{n}{2}\log|\Theta| \qquad (\mathsf{Tr}(AB) = \mathsf{Tr}(BA))$$

- We can exchange the sum and trace (which is also a sum) to get,

$$= \frac{1}{2}\mathsf{Tr}\left(\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T\Theta\right) - \frac{n}{2}\log|\Theta| \quad (\sum_i \mathsf{Tr}(A_iB) = \mathsf{Tr}(\sum_i A_iB))$$

Maximum Likelihood Estimation in Multivariate Gaussians

- In terms of precision matrix Θ we have

$$=\frac{1}{2}\sum_{i=1}^{n}\text{Tr}((x^i-\mu)(x^i-\mu)^T\Theta)-\frac{n}{2}\log|\Theta| \qquad (\text{Tr}(AB)=\text{Tr}(BA))$$

- We can exchange the sum and trace (which is also a sum) to get,

$$=\frac{1}{2}\text{Tr}\left(\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^T\Theta\right)-\frac{n}{2}\log|\Theta| \quad (\sum_i \text{Tr}(A_iB)=\text{Tr}(\sum_i A_iB))$$

$$=\frac{n}{2}\text{Tr}(S\Theta)-\frac{n}{2}\log|\Theta|,$$

where we've used $S=\frac{1}{n}\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^T$ is the sample covariance matrix.

Maximum Likelihood Estimation in Multivariate Gaussians

- So the NLL in terms of the precision matrix Θ is

$$\frac{n}{2}\mathsf{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T$$

Maximum Likelihood Estimation in Multivariate Gaussians

- So the NLL in terms of the precision matrix Θ is

$$\frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

    (it's the matrix version of an inner-product $s^T\theta$)
  - Negative log-determinant is strictly-convex and has $\nabla_\Theta \log|\Theta| = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).

Maximum Likelihood Estimation in Multivariate Gaussians

- So the NLL in terms of the precision matrix Θ is

$$\frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

    (it's the matrix version of an inner-product $s^T\theta$)
  - Negative log-determinant is strictly-convex and has $\nabla_\Theta \log|\Theta| = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).
- The MLE for a given μ is obtained by setting gradient matrix to zero, giving

$$\Theta = S^{-1} \quad \text{or} \quad \Sigma = S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T.$$

Maximum Likelihood Estimation in Multivariate Gaussians

- So the NLL in terms of the precision matrix Θ is

$$\frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^T$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.
    (it's the matrix version of an inner-product $s^T\theta$)
  - Negative log-determinant is strictly-convex and has $\nabla_\Theta \log|\Theta| = \Theta^{-1}$.
    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).
- The MLE for a given μ is obtained by setting gradient matrix to zero, giving

$$\Theta = S^{-1} \quad \text{or} \quad \Sigma = S = \frac{1}{n}\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^T.$$

- The constraint $\Sigma \succ 0$ means we need positive-definite sample covariance, $S \succ 0$.
  - If S is not invertible, NLL is unbounded below and no MLE exists.

MAP Estimation in Multivariate Gaussian

- We typically don't regularize μ, but you could add an L2-regularizer $\frac{\lambda}{2}\|\mu\|^2$.
- A classic "hack" for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ by construction.

MAP Estimation in Multivariate Gaussian

- We typically don't regularize μ, but you could add an L2-regularizer $\frac{\lambda}{2}\|\mu\|^2$.
- A classic "hack" for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ by construction.
  - This corresponds to a regularizer that penalizes diagonal of the precision,

$$f(\Theta) = \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) = \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta|.$$

MAP Estimation in Multivariate Gaussian

- We typically don't regularize μ, but you could add an L2-regularizer $\frac{\lambda}{2}\|\mu\|^2$.
- A classic "hack" for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ by construction.

  - This corresponds to a regularizer that penalizes diagonal of the precision,

$$f(\Theta) = \text{Tr}(S\Theta) - \log|\Theta| + \lambda\text{Tr}(\Theta) = \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta|.$$

- Recent substantial interest in generalization called the graphical LASSO,

$$f(\Theta) = \text{Tr}(S\Theta) - \log|\Theta| + \lambda\|\Theta\|_1.$$

where we are using the element-wise L1-norm.

  - Gives sparse Θ.

(we'll discuss "graphical" part later)

  - Can solve very large instances with proximal-Newton and other tricks ("QUIC").

Properties of Multivariate Gaussian and Product of Gaussians

- Multivariate Gaussian has nice properties of univariate Gaussian:
  - Central limit theorem: mean estimates of random variables converge to Gaussians.
  - Closed-form MLE for μ and Σ.
  - Maximizes entropy subject to fitting mean and covariance of data.

Properties of Multivariate Gaussian and Product of Gaussians

- Multivariate Gaussian has nice properties of univariate Gaussian:
  - Central limit theorem: mean estimates of random variables converge to Gaussians.
  - Closed-form MLE for μ and Σ.
  - Maximizes entropy subject to fitting mean and covariance of data.
- Another notable property is that product of Gaussian PDFs is Gaussian PDF.
  - We saw that product of independent Gaussians yields a Gaussian.
  - A Gaussian likelihood with Gaussian prior on mean gives Gaussian posterior.

Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right),$$

so are dataset would be something like

$$X = \begin{bmatrix} | & | & | & | \\ x_1 & x_2 & z_1 & z_2 \\ | & | & | & | \end{bmatrix}.$$

Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right).$$

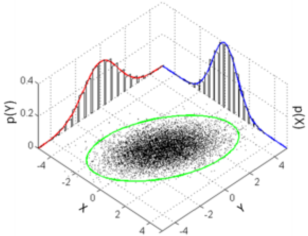
Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right).$$

- The marginal probabilities are Gaussian with parameters from partition,

$$x \sim \mathcal{N}(\mu_x, \Sigma_{xx}).$$



https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right).$$

- The marginal probabilities are Gaussian with parameters from partition,

$$x \sim \mathcal{N}(\mu_x, \Sigma_{xx}).$$

- This seemes less intuitive if you write it as

$$p(x) = \int_{z_1}\int_{z_2}\cdots\int_{z_d} \frac{1}{(2\pi)^{\frac{d}{2}}\left|\begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} x \\ z \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}\right)\begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}^{-1}\left(\begin{bmatrix} x \\ z \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}\right)\right) dz_d dz_{d-1} \dots dz_1.$$

Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right).$$

- The conditional probabilities are also Gaussian,

$$x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z}),$$

where

$$\mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(z - \mu_z), \quad \Sigma_{x|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}.$$

Marginalization and Conditioning in Gaussians

- Consider partitioning multivariate Gaussian variables into two sets,

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}\right).$$

- The conditional probabilities are also Gaussian,

$$x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z}),$$

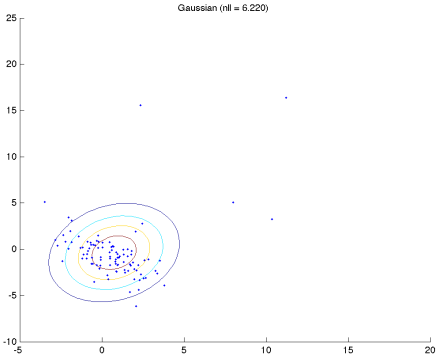
  where

$$\mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(z - \mu_z), \quad \Sigma_{x|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}.$$

- "Closedness" of Gaussians is not true for any other continuous distribution on $\mathbb{R}^d$.
  - Sometimes we'll use these to simplify computations.
  - Sometimes we use that non-Gaussian variables don't satisfy unique such properties.

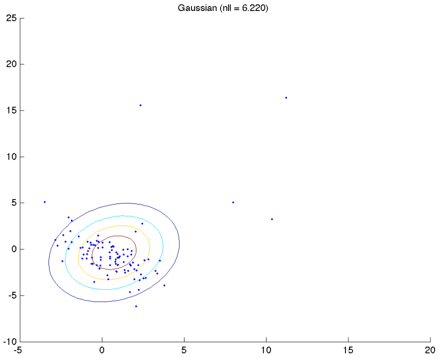
Problems with Multivariate Gaussian

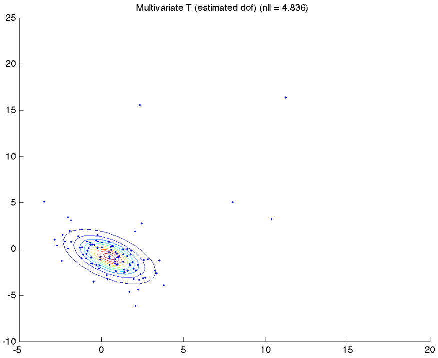
- Why not the multivariate Gaussian distribution?
  - Still not robust, may want to consider multivariate Laplace or multivariate T.
    - These require numerical optmization to compute MLE/MAP.



Problems with Multivariate Gaussian

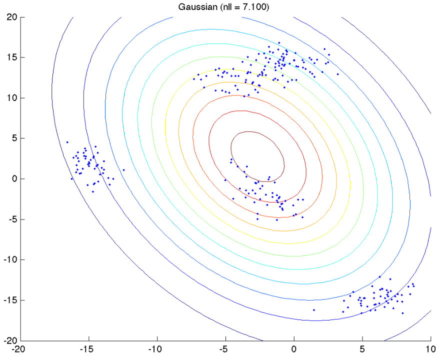
- Why not the multivariate Gaussian distribution?
  - Still not robust, may want to consider multivariate Laplace or multivariate T.
    - These require numerical optmization to compute MLE/MAP.





Problems with Multivariate Gaussian

- Why not the multivariate Gaussian distribution?
  - Still not robust, may want to consider multivariate Laplace of multivariate T.
  - Still unimodal, which often leads to very poor fit.



Summary

- Density estimation: unsupervised modelling of probability of feature vectors.

Summary

- Density estimation: unsupervised modelling of probability of feature vectors.
- Product of independent distributions is simple/crude density estimation method.

Summary

- Density estimation: unsupervised modelling of probability of feature vectors.
- Product of independent distributions is simple/crude density estimation method.
- Multivariate Gaussian generalizes univariate Gaussian for multiple variables.
  - Many analytic properties like closed-form MLE, products, marginals, conditionals.
  - But unimodal and not robust.

- Next time: missing data and the most cited paper in statistics.

Bonus Slide: Comments on Positive-Definiteness

- If we define centered vectors $\tilde{x}^i = x^i - \mu$ then empirical covariance is

$$S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^T = \sum_{i=1}^{n}\tilde{x}^i(\tilde{x}^i)^T = \tilde{X}^T\tilde{X} \succeq 0,$$

  so S is positive semi-definite but not positive-definite by construction.
- If data has noise, it will be positive-definite with n large enough.
- For $\Theta \succ 0$, note that for an upper-triangular T we have

$$\log|T| = \log(\text{prod}(eig(T))) = \log(\text{prod}(\text{diag}(T))) = \text{Tr}(\log(\text{diag}T)),$$

  where we've used Matlab notation.
- So to compute $\log|\Theta|$ for $\Theta \succ 0$, use Cholesky to turn into upper-triangular.
  - Bonus: Cholesky will fail if $\Theta \succ 0$ is not true, so it checks constraint.