# CPSC 440: Advanced Machine Learning

## Learning Gaussians

Mark Schmidt

University of British Columbia

Winter 2022

# Last Time: Inference in Multivariate Gaussian

- The multivariate normal/Gaussian distribution models PDF of vector $x^i$ as

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

  where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$ is symmetric with $\Sigma \succ 0$.
  - The density for a linear transformation of a product of independent Gaussians.
- Models correlations between non-zero elements of $\Sigma$.
  - If $\Sigma$ is diagonal then it assumes all variables are independent.
- We discussed affine property that linear transformation of Gaussians is Gaussian.
  - Can be used to generate samples from a Gaussian.
- We discussed how marginals and conditionals are Gaussian.
- We often draw a graph of the non-zero elements of precision $\Theta = \Sigma^{-1]}$.
  - Variables are conditionally independent if conditioning set blocks paths in the graph.

# Discussion of Independence in Gaussians

- If $\Sigma$ is diagonal then $\Theta$ is diagonal.
  - This gives a disconnected graph: all variables are independent.
- If $\Theta$ is a full matrix, graph does not imply any conditional independences.
  - "Everything depends on everything, no matter how many of the $x_j$ you know."
- The value $\Theta_{ij}$ is related to the partial correlation which is $-\Theta_{ij}/\sqrt{\Theta_{ii}\Theta_{jj}}$.
  - The "remaining correlation when we know all other variables".
- Dependencies can exist if $\Theta_{ij} = 0$ due to correlations with other variables.
  - Only independent if all paths that correlation could go across are blocked.

# MLE for Multivariate Gaussian (Mean Vector)

- With a multivariate Gaussian we have

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

  so up to a constant our negative log-likelihood for $n$ examples $x^i$ is

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

- This is a convex quadratic in $\mu$, setting gradient to zero gives

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

  - MLE for $\mu$ is the mean along each dimension, and it does not depend on $\Sigma$.

# MLE for Multivariate Gaussians (Covariance Matrix)

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

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}|
\end{aligned}$$

- After some tedious linear algebra (in bonus slides) we obtain that this is equal to

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

where:
  - $S$ is the covariance of the data (if we subtract mean from all examples, $S=(1/n)X^TX$).
  - Trace operator $\mathrm{Tr}(A)$ is the sum of the diagonal elements of $A$.

# MLE for Multivariate Gaussians (Covariance Matrix)

- Gradient matrix of NLL with respect to $\Theta$ is (not obviously)

$$\nabla f(\Theta) = \frac{n}{2}S - \frac{n}{2}\Theta^{-1}.$$

- The MLE for a given $\mu$ is obtained by setting gradient matrix to zero, giving

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

- The constraint $\Sigma \succ 0$ means we need positive-definite sample covariance, $S \succ 0$.
  - If $S$ is not positive-definite, NLL is unbounded below and no MLE exists.
  - This is like requiring "not all values are the same" in univariate Gaussian.
    - In $d$-dimensions, you need $d$ linearly-independent $x^i$ values (no "collinearity")

- For most distributions, the MLEs are not the data's mean and covariance.

# MAP Estimation for Mean

- For fixed $\Sigma$, conjugate prior for mean is a Gaussian:

$$x^i \sim \mathcal{N}(\mu, \Sigma), \mu \sim \mathcal{N}(\mu_0, \Sigma_0), \rightarrow \mu \mid X, \Sigma \sim \mathcal{N}(\mu^+, \Sigma^+),$$

where (using product of Gaussians property we are about to cover)

$$\Sigma^+ = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1},$$
$$\mu^+ = \Sigma^+(n\Sigma^{-1}\mu_{\text{MLE}} + \Sigma_0^{-1}\mu_0).$$

MAP estimate of $\mu$

- In special case of $\Sigma = \sigma^2 I$ and $\Sigma_0 = (1/\lambda)I$ we get

$$\Sigma^+ = ((n/\sigma^2)I + \lambda I)^{-1},$$
$$\mu^+ = \Sigma^+((n/\sigma^2)\mu_{\text{MLE}} + \lambda\mu_0).$$

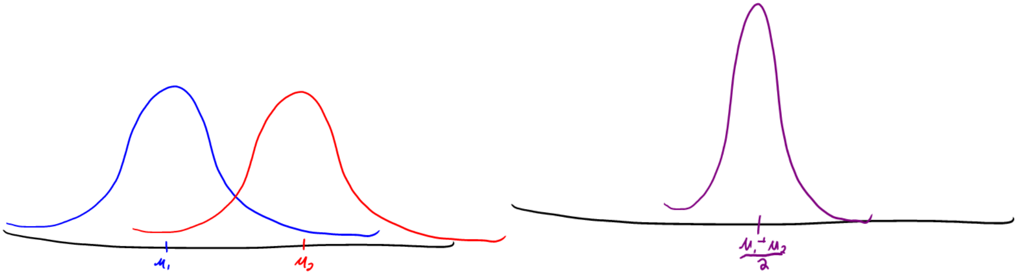
- Posterior predictive is $\mathcal{N}(\mu^+, \Sigma + \Sigma^+)$ (product of $(n+2)$ then marginalize).
    - Many Bayesian inference tasks have closed form, or Monte Carlo is easy.

# Product of Gaussian Densities Property

- Consider variable $x$ whose PDF is written as product of two Gaussians,

$$p(x) = f_1(x)f_2(x)$$

  where:
  - $f_1$ is proportional to a Gaussian with mean $\mu_1$ and covariance $I$.
  - $f_2$ is proportional to a Gaussian with mean $\mu_2$ and covariance $I$.
- Then this product of Gaussian PDFs is a Gaussian with $\mu = \frac{\mu_1+\mu_2}{2}$ and $\Sigma = \frac{1}{2}$.



# Product of Gaussian Densities Property

- If $p(x) \propto f_1(x)f_2(x)$ $f_1$ and $f_2$ with
  - $f_1$ proportional to a Gaussian with mean $\mu_1$ and covariance $\Sigma_1$.
  - $f_2$ proportional to a Gaussian with mean $\mu_2$ and covariance $\Sigma_2$.
- Then $p$ is a Gaussian with (see textbook)

$$\text{covariance of} \quad \Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}.$$

$$\text{mean of } \mu = \Sigma\Sigma_1^{-1}\mu_1 + \Sigma\Sigma_2^{-1}\mu_2,$$

- How we do we use this to derive the posterior distribution for the mean?

$$\begin{aligned} p(\mu \mid X, \Sigma, \mu_0, \Sigma_0) &\propto p(\mu \mid \mu_0, \Sigma_0) \prod_{i=1}^{n} p(x^i \mid \mu, \Sigma) && \text{(Bayes rule)} \\ &= p(\mu \mid \mu_0, \Sigma_0) \prod_{i=1}^{n} p(\mu \mid x^i, \Sigma) && (\text{symmetry of } x^i \text{ and } \mu) \\ &= \text{(product of } (n+1) \text{ Gaussians).} \end{aligned}$$

# MAP Estimation in Multivariate Gaussian (Trace Regularization)

- A common MAP estimate for $\Sigma$ is

$$\hat{\Sigma} = S + \lambda I,$$

where $S$ is the covariance of the data.
  - Key advantage: $\hat{\Sigma}$ is postiive-definite (eigenvalues are at least $\lambda$).

- This minimizes NLL plus L1-regularization of precision diagonals (see bonus)

$$f(\Theta) = \underbrace{\text{Tr}(S\Theta) - \log|\Theta|}_{\text{NLL}} + \lambda \sum_{j=1}^{d} |\Theta_{jj}|.$$

although it does not set $\Theta_{jj}$ values to exactly zero.
  - Log-determinant term becomes arbitrarily steep as the $\Theta_{jj}$ approach 0.

# Graphical LASSO

- A popular generalization called the graphical LASSO,

$$f(\Theta) = \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{i=1}^{d}\sum_{j=1}^{d} |\Theta_{ij}|,$$

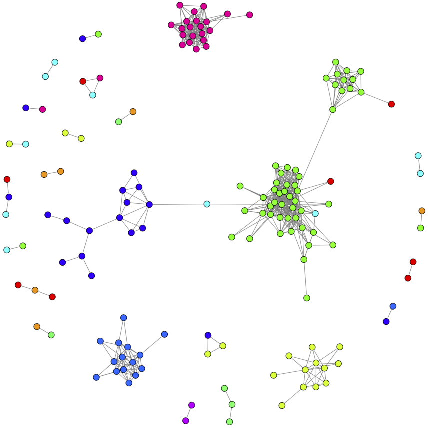
  where we apply L1-regularization to all elements of $\Theta$.

- With large enough $\lambda$, gives sparse off-diagonals in $\Theta$.
  - Though need specialized optimization algorithms to solve this problem.

- Recall that sparsity of $\Theta$ determines conditional independence.
  - When we set a $\Theta_{ij} = 0$ it remove an edges from the graph.
    - Makes the graph simpler, and can make computations cheaper.

# Graphical LASSO Example

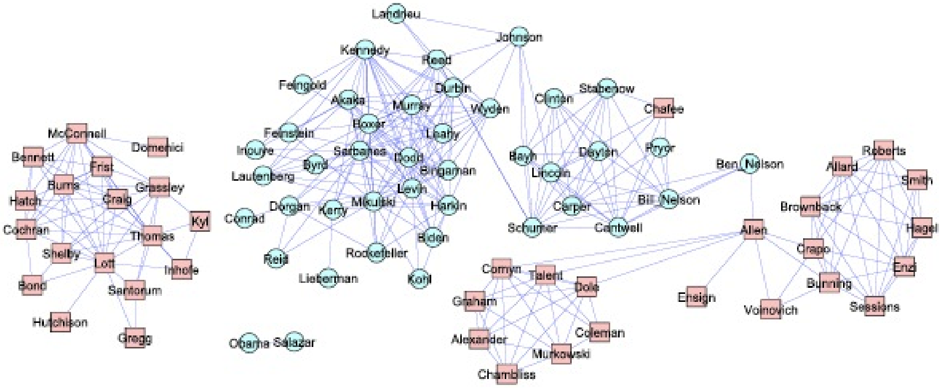
- Graphical LASSO applied to stocks data:



https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

# Graphical LASSO Example

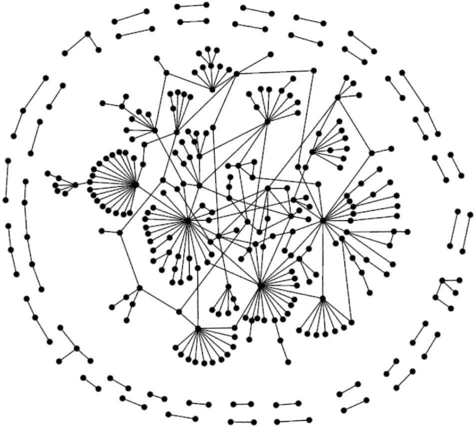
- Graphical LASSO applied to US senate voting data (Bush junior era):




https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

# Graphical LASSO Example

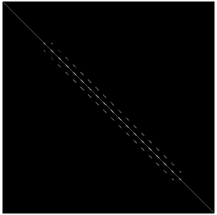
- Graphical LASSO applied to protein data:



https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

# Graphical LASSO on Digits

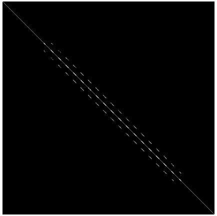
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- To understand this picture, first the size of the precision matrix:
  - The images of digits, which are $m \times m$ matrices ($m$ pixels by $m$ pixels)
    - This gives $d = m^2$ elements of $x^i$, which we'll assume are in "column-major" order.
    - Frist $m$ elements of $x^i$ are column 1, next $m$ elements are columm 2, and so on.
  - The picture above, which is $d \times d$ so will thus be $m^2 \times m^2$.

# Graphical LASSO on Digits

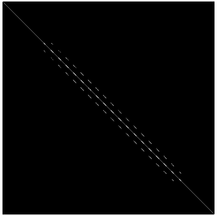
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- So what are the non-zeroes in the precision matrix?
  1. The diagonals $\Theta_{i,i}$ (positive-definite matrices must have positive diagonals).
  2. The first off-diagonals $\Theta_{i,i+1}$ and $\Theta_{i+1,i}$.
     - This represents the dependencies between adjacent pixels vertically.
  3. The $(m+1)$ off-diagonals $\Theta_{i,i+m}$ and $\Theta_{i+m,i}$.
     - This represents the dependencies between adjacent pixels horizontally.
     - Because in "column-major" order, you go "right" a pixel every $m$ indices.

# Graphical LASSO on Digits

- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- The edges in the graph are pixels next to each other in the image.
- Graphical Lasso is a special case of structure learning in graphical models.
  - We will discusss graphical models more later.

# Conjugate Priors for Covariance

- Graphical LASSO is not using a conjugate prior.
- Conjugate prior for $\Theta$ with known mean is Wishart distribution
  - A multi-dimensional generalization of the gamma distribution.
    - Gamma is a distribution over positive scalars.
    - Wishart is a distribution over positive-definite matrices.
  - Posterior predictive is a student $t$ distribution.
  - Conjugate prior for $\Sigma$ is inverse-Wishart (equivalent posterior).
- If both $\mu$ and $\Theta$ are variables, conjugate prior is normal-Wishart.
  - Normal times Wishart, with a particular dependency among parameters.
  - Posterior predictive is again a student $t$ distribution.
- Wikipedia has already done a lot of possible homework questions for you:
  - `https://en.wikipedia.org/wiki/Conjugate_prior`

# Outline

# Generative Classification with Gaussians

- We previously considerd the generative classifier naive Bayes.
  - Assumed $x_i \perp x_j \mid y$, which is strong/unrealistic.

- Consider a generative classifier with continuous features:

$$\begin{aligned} p(y^i \mid x^i) &\propto p(x^i, y^i) \\ &= \underbrace{p(x^i \mid y^i)}_{\text{continuous}} \underbrace{p(y^i)}_{\text{discrete}}. \end{aligned}$$

- In Gaussian discriminant analysis (GDA) we assume $x^i \mid y^i$ is a Gaussian.
  - It is classification so output $y^i$ is categorical.
  - Classifier asks "which Gaussian makes $x^i$ most likely?"
  - This can model pairwise correlations within each class.
    - Does not need naive Bayes assumption.

# Gaussian Discriminant Analysis (GDA) and Closed-Form MLE

- In Gaussian discriminant analysis we assume $x^i \mid y^i$ is a Gaussian.

$$p(x^i, y^i = c) = \underbrace{p(y^i)p(x^i \mid y^i = c)}_{\text{product rule}} = \underbrace{\pi_c}_{p(y^i=c)} \underbrace{p(x^i \mid \mu_c, \Sigma_c)}_{\text{Gaussian PDF}}.$$

- A special case is linear discriminant analysis (LDA):
  - In LDA we assume that $\Sigma_c$ is the same for all classes $c$.

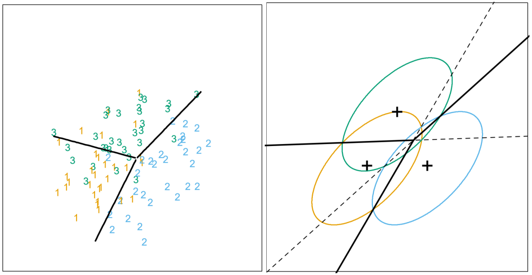
- In LDA the MLE has a simple closed-form expression:

$$\hat{\pi}_c = \frac{n_c}{n}, \quad \hat{\mu}_c = \frac{1}{n_c}\sum_{y^i=c} x^i.$$

  - $\hat{\pi}_c$ is fraction of times we are in class $c$, $\hat{\mu}$ is mean of class $c$.

# Linear Discriminant Analysis (LDA)

- Example of fitting linear discriminant analysis (LDA) to a 3-class problem:



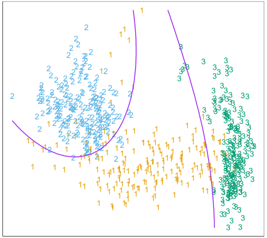
https://web.stanford.edu/~hastie/Papers/ESLII.pdf

- LDA is a linear classifier.
  - Unlike other linear classifiers like logistic regression, it has a closed-form MLE.
  - Though it may be less accurate if the classes do not look like Gaussians.
- If class proportions $\pi_c$ are equal, class label is determined by nearest mean.
  - Prediction is like a "1-nearest neighbour" or $k$-means clustering method.

# Gaussian Discriminant Analysis (GDA)

- We can also have a covariance $\Sigma_c$ for each class.
  - So the class will be determined by class proportions, means, and variances.
- The MLE for each each $\Sigma_c$ is the covariance of data in class $c$,

$$\hat{\Sigma}_c = \frac{1}{n_c} \sum_{y^i = c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T,$$

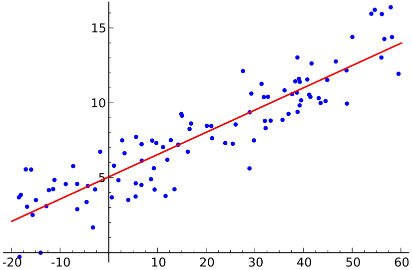


https://web.stanford.edu/~hastie/Papers/ESLII.pdf

- This leads to a quadratic classifier.
  - GDA is sometimes called quadratic discriminant analysis.

# Regression with Gaussians

- Regression is a variant on supervised learning where $y^i$ is continuous.




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

- It is possible to use generative regression models.
  - For example, we could model $p(x, y)$ as a multivariate Gaussian.
    - Then use that the conditional $p(y \mid x)$ is Gaussian for prediction.

- But we usually treat features as fixed (as in discriminative classification models).
  - And to start, we will consider models that make linear predictions, $\hat{y}^i = w^T x^i$.

# Review: L2-Regularized Least Squares and Gaussians

- A common linear regression model is L2-regularized least squares,

$$\underset{w}{\text{argmin}} \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2,$$

- This corresponds to MAP estimation with a Gaussian likelihood and prior,

$$y^i \sim \mathcal{N}(w^T x^i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda^{-1}).$$

- By setting the gradient to zero, the unique solution is given by:

$$\hat{w} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}X^T X + \lambda I\right)^{-1} X^T y.$$

- In 340 we fixed $\sigma^2 = 1$ (since changing is $\sigma^2$ equivalent to changing $\lambda$).
  - In Bayesian inference, both $\sigma^2$ and $\lambda$ affect the predictions.
- To predict on new example $\tilde{x}$ with MAP estimate, we use $\hat{y} = \hat{w}^T \tilde{x}$.

# Summary

- MLE for multivariate Gaussian:
  - MLE for $\mu$ is mean of data, MLE for $\Sigma$ is covariance of data (if positive definite).
- Posterior and posterior predictive under Gaussian prior on mean is Gaussian.
  - Can be shown using that product of Gaussians is Gaussian.
- Graphical Lasso uses L1-regularization of precision matrix.
  - Leads to a sparse graph structure representing conditional independences.
- Supervised learning with Gaussians
  - Generative classifier with Gaussian classes is Gaussian discriminant analysis (GDA).
  - L2-regularized least squares is obtained using a Gaussian likelihood and prior.
    - Regression model assuming features fixed/non-random as in discriminative classifiers.

- Next time: linear regression plus empirical Bayes to cheat on your stats homework.

# MLE for Multivariate Gaussians (Covariance Matrix)

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

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}| && \text{(ok because } \Sigma \text{ is invertible)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\mathrm{Tr}\left((x^i-\mu)^\top\Theta(x^i-\mu)\right)+\frac{n}{2}\log|\Theta|^{-1} && \text{(scalar } y^\top Ay=\mathrm{Tr}(y^\top Ay)\text{)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\mathrm{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta| && (\mathrm{Tr}(ABC)=\mathrm{Tr}(CAB))
\end{aligned}$$

- Where the trace $\mathrm{Tr}(A)$ is the sum of the diagonal elements of $A$.
  - That $\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB)$ when dimensions match is the cyclic property of trace.

# MLE for Multivariate Gaussians (Covariance Matrix)

- From the last slide we have in terms of precision matrix $\Theta$ that

$$= \frac{1}{2}\sum_{i=1}^{n} \mathrm{Tr}((x^i - \mu)(x^i - \mu)^\top \Theta) - \frac{n}{2}\log|\Theta|$$

- We can exchange the sum and trace (trace is a linear operator) to get,

$$= \frac{1}{2}\mathrm{Tr}\left(\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top \Theta\right) - \frac{n}{2}\log|\Theta| \qquad \sum_i \mathrm{Tr}(A_i B) = \mathrm{Tr}\left(\sum_i A_i B\right)$$

$$= \frac{n}{2}\mathrm{Tr}\left(\left(\underbrace{\frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top}_{\text{sample covariance '}S\text{'}}\right)\Theta\right) - \frac{n}{2}\log|\Theta|. \qquad \left(\sum_i A_i B\right) = \left(\sum_i A_i\right) B$$

# MLE for Multivariate Gaussians (Covariance Matrix)

- So the NLL in terms of the precision matrix $\Theta$ and sample covariance $S$ is

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

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

    (it's the matrix version of an inner-product $s^\top\theta$)
  - Negative log-determinant is strictly-convex and has $\nabla_\Theta \log|\Theta| = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).
- Using these two properties the gradient matrix has a simple form:

$$\nabla f(\Theta) = \frac{n}{2}S - \frac{n}{2}\Theta^{-1}.$$

# Trace Regularization and L1-regularization

- A classic regularizer for $\Sigma$ is to add a diagonal matrix to $S$ and use

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

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least $\lambda$).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned}
f(\Theta) &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)} \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0) \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) && \text{(Definition of trace)} \\
&= \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta| && \text{(Linearity of trace)} \\
&= \text{Tr}((S + \lambda I)\Theta) - \log|\Theta| && \text{(Distributive law)}
\end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.