CS142 Announcements

- Updates on grade-related queries: coming soon
- Midterm exam: Out today, due Friday Oct. 30 @ 5:00pm
- Midterm material: Lecture 11 (October 20) and earlier. 
  *This week’s lectures will appear on future homeworks.*
- Midterm formatting & submission: Like homeworks.
  *Shorter than homeworks, no programming.*
- Midterm collaboration: Not allowed!
  *Email TA list for questions & clarifications.*
  *We will still hold office hours this week.*
Gaussian Distribution

\[ \mathcal{N}(x | \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]

Why the Gaussian distribution?
- **Central limit theorem**: Property of (some) big datasets
- **Flexibility**: Can capture arbitrary mean and covariance
- **Convenience**: Quadratic log-probability easy to optimize

Why consider non-Gaussian likelihoods?
- **Data type**: Observations may not be continuous numbers
- **Outliers**: Increase robustness to non-typical data

Why consider non-Gaussian priors on parameters?
- **Sparsity**: Allow selection of most important model features
CS142 ML: Lecture 13 Outline

- Outliers & robust regression likelihoods
- Feature selection via combinatorial search
- $L_1$ regularization, sparsity, and the lasso
Relative to Gaussian distributions with equal variance:
- Many samples are near mean
- Occasional large-magnitude samples are far more likely
- Negative log probability density is \textit{convex but not smooth}

Probability Densities

\[
\text{Lap}(x \mid \mu, \lambda) = \frac{\lambda}{2} \exp(-\lambda |x - \mu|)
\]
Relative to Gaussian distributions with equal variance:

- Approaches Gaussian as *degrees of freedom* (DOF) approaches infinity
- For small DOF, large-magnitude samples are far more likely
- Negative log probability density is *smooth but not convex*
Outliers and ML Estimation of Mean

Maximum likelihood estimates of mean parameters:
- **Gaussian**: Sample mean of data
- **Laplacian**: Sample median of data
- **Student T**: No closed form, optimize via gradient methods
Outliers and Linear Regression

![Graph showing different regression models: least squares, laplace, and student with dof=0.409. Each model is represented by a line and data points.]
Huber Loss (Negative Log-Prob) Function

Relative to Gaussian distributions with equal variance:
- Behaves like Gaussian near origin ("non-outliers")
- Behaves like Laplacian far from origin (robustness)
- Negative log probability density is smooth and convex

Many similar functions studied in robust statistics literature.
CS142 ML: Lecture 13 Outline

- Outliers & robust regression likelihoods
- Feature selection via combinatorial search
- $L_1$ regularization, sparsity, and the lasso
Reminder: Linear Regression

\[ f(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \phi(x_n)^T w)^2 = \frac{1}{2} ||t - \Phi w||^2 \]

- \( N \rightarrow \) number of examples
- \( M \rightarrow \) number of features
- \( t_n \rightarrow \) output or response
- \( x_n \rightarrow \) input or covariates
- \( y(x_n, w) = \phi(x_n)^T w \)
- \( \phi(x_n) \in \mathbb{R}^M \quad w \in \mathbb{R}^M \)

\[
\begin{bmatrix}
  t_1 \\
  t_2 \\
  \vdots \\
  t_N
\end{bmatrix} \in \mathbb{R}^N \quad \Phi = \begin{bmatrix}
  \phi(x_1)^T \\
  \phi(x_2)^T \\
  \vdots \\
  \phi(x_N)^T
\end{bmatrix} \in \mathbb{R}^{N \times M}
\]
Reminder: Linear Regression

\[ y(x_n, w) = \phi(x_n)^T w \]

\[ f(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \phi(x_n)^T w)^2 = \frac{1}{2} ||t - \Phi w||^2 \]

- \( N \rightarrow \) number of examples
- \( M \rightarrow \) number of features
- \( t_n \rightarrow \) output or response
- \( x_n \rightarrow \) input or covariates

> With Gaussian prior on weight vector (L2 regularization):

\[ f(w) = \frac{1}{2} ||t - \Phi w||^2 + \frac{\lambda}{2} ||w||^2 \]

\( \phi(x_n) \in \mathbb{R}^M \quad w \in \mathbb{R}^M \)
Regularization in Regression

- Basic model selection: Coefficients are ordered, only first $M$ are non-zero
  - Classical example: polynomial regression
  - What if my features aren’t easy to interpret?
- Gaussian prior ($L_2$ regularization): Coefficients are small
  - Computation & storage: Expensive for many features
  - Interpretability: Doesn’t identify important features
- Many applications: Only some of my features are relevant, but I don’t know how many or which ones are relevant
Feature Selection Models

$\phi_j(x) \in \mathbb{R}$ is some possible feature of input data

$\gamma_j = 1$ if feature $j$ is relevant, 0 otherwise

➢ We can determine feature relevance from weight vector:

$\gamma_j = 1$ if $w_j \neq 0$, $\gamma_j = 0$ if $w_j = 0$

➢ The following $L_0$ regularization explicitly encourages sparsity:

$$f(w) = \|t - \Phi w\|_2^2 + \lambda \|w\|_0 \quad \lambda > 0$$

$$\|w\|_p = \left( \sum_{j=1}^{M} |w_j|^p \right)^{1/p} \quad \|w\|_0 = \text{number of nonzero entries}$$
Feature Selection Models

\(\phi_j(x) \in \mathbb{R}\) is some possible feature of input data

\(\gamma_j = 1\) if feature \(j\) is relevant, 0 otherwise

- Probabilistic “spike & slab” interpretation of \(L_0\) regularization:

\[
p(\gamma) = \prod_{j=1}^{M} \text{Ber}(\gamma_j \mid \mu)
\]

\(\mu < 0.5\)

\[
p(w_j \mid \gamma_j = 0) = \delta_0(w_j)
\]

\[
p(w_j \mid \gamma_j = 1) = \mathcal{N}(w_j \mid 0, \alpha^{-1}), \quad \alpha \to 0.
\]

\[
p(t_n \mid x_n, w) = \mathcal{N}(t_n \mid w^T \phi(x_n), \beta^{-1})
\]

- The following \(L_0\) regularization explicitly encourages sparsity:

\[
f(w) = \|t - \Phi w\|^2_2 + \lambda \|w\|_0 \quad \lambda = -2\beta^{-1} \log \left( \frac{\mu}{1 - \mu} \right)
\]

\[
\|w\|_p = \left( \sum_{j=1}^{M} |w_j|^p \right)^{1/p}
\]

\[
\|w\|_0 = \text{number of nonzero entries}
\]
Feature Selection: Example

All combinations of 10 features ($2^{10}=1024$ in Gray code order)

Dataset: $N=10$ samples from linear regression model with

$$\mathbf{w} = (0.00, -1.67, 0.13, 0.00, 0.00, 1.19, 0.00, -0.04, 0.33, 0.00)$$
Feature Selection: Example

Dataset: $N=10$ samples from linear regression model with

$$w = (0.00, -1.67, 0.13, 0.00, 0.00, 1.19, 0.00, -0.04, 0.33, 0.00)$$

Most likely models: $\{2\}$, $\{2,6\}$, $\{2,6,9\}$, $\{2,3,6\}$...

Marginal Inclusion Probabilities

$$\hat{\gamma} = \{j : p(\gamma_j = 1|\mathcal{D}) > 0.5\}$$
Greedy Deterministic Search

- Test all possible ways of adding *(forward selection)* or removing *(backward selection)* one feature.
- Add or remove the best feature, or stop if the current model is best.
- Wrapper method: Can be applied to any objective. *But no guarantees!!*
Too Many Models

Pascal’s Triangle (http://www.mathwarehouse.com/)
CS142 ML: Lecture 13 Outline

- Outliers & robust regression likelihoods
- Feature selection via combinatorial search
- $L_1$ regularization, sparsity, and the lasso
When used as a zero-mean prior on vectors of model parameters:
- Compared to Gaussian, stronger bias that many near zero
- When find MAP estimate, some weights are exactly zero
- Learning harder than for Gaussian, but still convex
Constrained Optimization

Laplacian prior
$L_1$ regularization
Lasso regression

where do level sets of the quadratic regression cost function first intersect the constraint set?

Gaussian prior
$L_2$ regularization
Ridge regression

$p(w) = \prod_{j=1}^{D} \text{Lap}(w_j | 0, \rho)$

$f(w) = ||y - \Phi w||_2^2 + \lambda ||w||_1$

$p(w) = \prod_{j=1}^{D} \text{Norm}(w_j | 0, \sigma^2)$

$f(w) = ||y - \Phi w||_2^2 + \lambda ||w||_2^2$
Gradient-Based Optimization

**Laplacian prior**

$L_1$ regularization

Lasso regression

**Gaussian prior**

$L_2$ regularization

Ridge regression

**Objective Function:**

$f(w) = -\log p(w)$

**Negative Gradient:**

$-f'(w)$

*(Informal intuition: Gradient of $L_1$ objective not defined at zero)*
Generalized Norms: Bridge Regression

\[ \text{NLL}(w) + \lambda \sum_{i} |w_i|^b \]

\[ \text{ExpPower}(w|\mu, a, b) := \frac{b}{2a\Gamma(1/b)} \exp(-\frac{|x-\mu|}{a})^b \]

- Convex objective function (true norm): \( b \geq 1 \)
- Encourages sparse solutions (cusp at zero): \( b \leq 1 \)
- Lasso/Laplacian (convex & sparsifying): \( b = 1 \)
- Ridge/Gaussian (classical, closed form solutions): \( b = 2 \)
- Sparsity via discrete feature count (greedy search): \( b \rightarrow 0 \)
Bayesian Regression: 0 Observations

\[ y = w_0 + w_1 x \]

Data Space

\[ y = w_0 + w_1 x \]
Bayesian Regression: 1 Observations

\[ y = w_0 + w_1 x \]
Regression Posteriors with Sparse Priors

\[ \text{NLL}(w) + \lambda \sum_{j} |w_j|^b \]

\[ \text{ExpPower}(w|\mu, a, b) := \frac{b}{2a\Gamma(1/b)} \exp\left( -\frac{|x - \mu|}{a} \right)^b \]

**Priors**

- \( b = 2 \)
- \( b = 1 \)
- \( b = 0.4 \)

**Posteriors**