This Lecture: Advanced Machine Learning

- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Support Vector Regression
  - Gaussian Processes
- Learning with Latent Variables
  - EM and Generalizations
  - Dirichlet Processes
- Structured Output Learning
  - Large-margin Learning

Topics of This Lecture
- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Recap: Maximum Likelihood Regression

\[ \nabla_w \log p(t|X, w, \beta) = -\beta \sum_{n=1}^{N} (t_n - w^T \phi(x_n)) \phi(x_n) \]

- Setting the gradient to zero:
  \[ 0 = -\beta \sum_{n=1}^{N} (t_n - w^T \phi(x_n)) \phi(x_n) \]
  \[ \Leftrightarrow \phi t = \Phi \phi \Phi^{-1} w \]
  \[ \phi = \phi(x_1), \ldots, \phi(x_n) \]
  \[ \Phi = \left[ \phi(x_1) \ldots \phi(x_n) \right] \]
  \[ w_{ML} = (\Phi \Phi^{-1})^{-1} \phi t \]

- Least-squares regression is equivalent to Maximum Likelihood under the assumption of Gaussian noise.

Recap: Probabilistic Regression

- First assumption:
  - Our target function values \( t \) are generated by adding noise to the ideal function estimate:
  \[ t = y(X, w) + \epsilon + \text{Noise} \]

- Second assumption:
  - The noise is Gaussian distributed:
  \[ p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1}) \]

Recap: Probabilistic View on Regression

- First assumption:
  - Our target function values \( t \) are generated by adding noise to the ideal function estimate:
  \[ t = y(X, w) + \epsilon \]

- Second assumption:
  - The noise is Gaussian distributed:
  \[ p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1}) \]

Mean
Variance
(\beta^{-1})
Target function value
Regression function
Input value
Weights or parameters
Recap: Role of the Precision Parameter

- Also use ML to determine the precision parameter $\beta$:
  \[
  \log p(t|X, w, \beta) = -\frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2 + \frac{N}{2} \log \beta - \frac{N}{2} \log(2\pi)
  \]

- Gradient w.r.t. $\beta$:
  \[
  \nabla_\beta \log p(t|X, w, \beta) = -\frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right) \phi(x_n)^T + \frac{N}{2} \frac{1}{\beta} \nabla_\beta \left( \frac{1}{\beta} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2 \right)
  \]

$\Rightarrow$ The inverse of the noise precision is given by the residual variance of the target values around the regression function.

Recap: Predictive Distribution

- Having determined the parameters $w$ and $\beta$, we can now make predictions for new values of $x$.
  \[
  p(t|X, w_{ML}, \beta_{ML}) = N(t|p(x|w_{ML}), \beta_{ML}^{-1})
  \]

- This means
  - Rather than giving a point estimate, we can now also give an estimate of the estimation uncertainty.

Recap: MAP Solution

- Minimize the negative logarithm
  \[
  -\log p(w|X, t, \beta, \alpha) \propto -\log p(t|X, w, \beta) - \log p(w|\alpha)
  \]

  \[
  -\log p(t|X, w, \beta) = \frac{\beta}{2} \sum_{n=1}^{N} \left( y_n(x_n, w) - t_n \right)^2 + \text{const}
  \]

  \[
  -\log p(w|\alpha) = \frac{\alpha}{2} w^T w + \text{const}
  \]

- The MAP solution is therefore
  \[
  \operatorname{arg min}_w \frac{1}{2} \sum_{n=1}^{N} \left( y_n(x_n, w) - t_n \right)^2 + \frac{\alpha}{2} w^T w
  \]

$\Rightarrow$ Maximizing the posterior distribution is equivalent to
minimizing the regularized sum-of-squares error (with $\lambda = \frac{\alpha}{\beta}$).

Recap: Bayesian Curve Fitting

- Given
  - Training data points: $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$
  - Associated function values: $t = [t_1, \ldots, t_n]^T$
  - Our goal is to predict the value of $t$ for a new point $x$.

- Evaluate the predictive distribution
  \[
  p(t|x, X, t) = \int p(t|x, w)p(w|X, t)dw
  \]

Effect of regularization:
- Keeps the inverse well-conditioned

Effect of regularization:
- Keeps the inverse well-conditioned

Recap: Maximum-A-Posteriori Estimation

- Introduce a prior distribution over the coefficients $w$.
  - For simplicity, assume a zero-mean Gaussian distribution
    \[
    p(w|\alpha) = N(w|0, \alpha^{-1}) = \left( \frac{\alpha}{\pi} \right)^{d/2} \exp \left( -\frac{\alpha}{2} w^T w \right)
    \]
  - New hyperparameter $\alpha$ controls the distribution of model parameters.

- Express the posterior distribution over $w$.
  - Using Bayes’ theorem:
    \[
    p(w|X, t, \beta, \alpha) \propto p(t|X, w, \beta)p(w|\alpha)
    \]
  - We can now determine $w$ by maximizing the posterior.
  - This technique is called maximum-a-posteriori (MAP).

Recap: MAP Solution (2)

- Setting the gradient to zero:
  \[
  \theta = -\frac{\beta}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right) \phi(x_n)^T + \frac{\alpha}{2} w
  \]

- We have:
  \[
  \sum_{n=1}^{N} t_n \phi(x_n) = \left( \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T \right) w + \frac{\alpha}{\beta} w
  \]

- Effect of regularization:
  - Keeps the inverse well-conditioned

Recap: Bayesian Curve Fitting

- Given
  - Training data points: $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$
  - Associated function values: $t = [t_1, \ldots, t_n]^T$
  - Our goal is to predict the value of $t$ for a new point $x$.

- Evaluate the predictive distribution
  \[
  p(t|x, X, t) = \int p(t|x, w)p(w|X, t)dw
  \]

- Noise distribution - again assume a Gaussian here
  \[
  p(t|x, w) = N(t|p(x|w), \beta^{-1})
  \]

- Assume that parameters $\alpha$ and $\beta$ are fixed and known for now.
Bayesian Curve Fitting

- Under those assumptions, the posterior distribution is a Gaussian and can be evaluated analytically:
  \[ p(t|x, X, t) = \mathcal{N}(t|m(x), s^2(x)) \]
  - where the mean and variance are given by
    \[ m(x) = \beta \phi(x)^T S \sum_{n=1}^{N} \phi(x_n) f_n \]
    \[ s^2(x) = \beta^{-1} + \phi(x)^T S \phi(x) \]
  - and \( S \) is the regularized covariance matrix
    \[ S^{-1} = \alpha I + \beta \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T \]

Analyzing the result

- Analyzing the variance of the predictive distribution
  \[ s(x)^2 = \beta^{-1} + \phi(x)^T S \phi(x) \]

Discussion

- We now have a better understanding of regression
  - Least-squares regression: Assumption of Gaussian noise
    \[ \Rightarrow \text{We can now also plug in different noise models and explore how they affect the error function.} \]
  - L2 regularization as a Gaussian prior on parameters \( w \).
    \[ \Rightarrow \text{We can now also use different regularizers and explore what they mean.} \]
    \[ \Rightarrow \text{This lecture...} \]
  - General formulation with basis functions \( \phi(x) \).
    \[ \Rightarrow \text{We can now also use different basis functions.} \]

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition
Loss Functions for Regression

- Given $p(y, x, w, \beta)$, how do we actually estimate a function value $y$, for a new point $x$?
- We need a loss function, just as in the classification case
  \[ L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+ \quad (t_n, y(x_n)) \rightarrow L(t_n, y(x_n)) \]
- Optimal prediction: Minimize the expected loss
  \[ E[L] = \int \int L(t, y(x)) p(x, t) \, dx \, dt \]

Optimal least-squares predictor given by the conditional mean

Intrinsic variability of target data \Rightarrow irreducible minimum value of the loss function

Visualization of Mean Prediction

• The squared loss is not the only possible choice
  - Poor choice when conditional distribution $p(x | t)$ is multimodal.

• Simple generalization: Minkowski loss
  \[ L(t, y(x)) = |y(x) - t|^q \]
  - Expectation
    \[ E[L_q] = \int \int |y(x) - t|^q p(x, t) \, dx \, dt \]
  - Minimum of $E[L_q]$ is given by
    - Conditional mean for $q = 2$,
    - Conditional median for $q = 1$,
    - Conditional mode for $q = 0$.
Minkowski Loss Functions

- \( \psi(x, w) = \sum_{j=0}^{M-1} \psi_j \phi_j(x) = w^T \phi(x) \)
- \( \phi_j(x) \) are known as basis functions.
- Typically, \( \phi_j(x) = 1 \), so that \( w_0 \) acts as a bias.
- In the simplest case, we use linear basis functions: \( \phi_j(x) = x_j \).

Linear Basis Function Models

- Generally, we consider models of the following form
- Let's take a look at some other possible basis functions...

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Linear Basis Function Models (2)

- Polynomial basis functions
- Properties
  - Global
  - A small change in \( x \) affects all basis functions.

Linear Basis Function Models (3)

- Gaussian basis functions
- Properties
  - Local
  - A small change in \( x \) affects only nearby basis functions.
  - \( \mu_j \) and \( \sigma \) control location and scale (width).

Linear Basis Function Models (4)

- Sigmoid basis functions
- Properties
  - Local
  - A small change in \( x \) affects only nearby basis functions.
  - \( \mu_j \) and \( \sigma \) control location and scale (slope).
Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Multiple Outputs

- Multiple Output Formulation
  - So far only considered the case of a single target variable $t$.
  - We may wish to predict $K > 1$ target variables in a vector $t$.
  - We can write this in matrix form
    \[ y(x, W) = W^T \phi(x) \]
    where
    \[ y = [y_1, \ldots, y_K]^T \]
    \[ \phi(x) = [1, \phi_1(x), \ldots, \phi_{M-1}(x)]^T \]
    \[ W = \begin{bmatrix}
    \beta_0, & \cdots, & \beta_K
    
    \vdots
    
    \beta_{M-1,1}, & \cdots, & \beta_{M-1,K}
    \end{bmatrix}^T \]

Multiple Outputs (2)

- Analogously to the single output case we have:
  \[ p(t|W, \beta) = \mathcal{N}(t; y(W, x), \beta^{-1}I) \]
  \[ = \mathcal{N}(t; W^T \phi(x), \beta^{-1}I). \]
- Given observed inputs, $X = \{x_1, \ldots, x_N\}$, and targets, $T = \{t_1, \ldots, t_N\}$, we obtain the log likelihood function
  \[ \ln p(T|X, W, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n; W^T \phi(x_n), \beta^{-1}I) \]
  \[ = \frac{NK}{2} \ln \left( \frac{\beta}{2\pi} \right) - \frac{\beta}{2} \sum_{n=1}^{N} \|t_n - W^T \phi(x_n)\|^2. \]

Multiple Outputs (3)

- Maximizing with respect to $W$, we obtain
  \[ W_{ML} = \left( \Phi^T \Phi \right)^{-1} \Phi^T T. \]
- If we consider a single target variable, $t_{k}$, we see that
  \[ w_k = \left( \Phi^T \Phi \right)^{-1} \Phi^T t_k = \Phi \hat{t}_k \]
  where $t_k = [t_{k1}, \ldots, t_{KN}]^T$, which is identical with the single output case.

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Sequential Learning

- Up to now, we have mainly considered batch methods
  - All data was used at the same time
  - Instead, we can also consider data items one at a time (a.k.a. online learning)
- Stochastic (sequential) gradient descent:
  \[ w^{(r+1)} = w^{(r)} - \eta \nabla E_n \]
  \[ = w^{(r)} + \eta(t_n - w^{(r)T} \phi(x_n)) \phi(x_n). \]
- This is known as the least-mean-squares (LMS) algorithm.
- Issue: how to choose the learning rate $\eta$?
**Topics of This Lecture**
- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

**Regularization Revisited**
- Consider the error function
  \[ E_D(w) + \lambda E_W(w) \]
  Data term + Regularization term
- With the sum-of-squares error function and a quadratic regularizer, we get
  \[ \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2 + \frac{\lambda}{2} w^T w \]
  which is minimized by
  \[ w = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T t \]
  \( \lambda \) is called the regularization coefficient.

**Recall: Lagrange Multipliers**
- Let’s look at more general regularizers
  \[ \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|^q \]
  “L_q norms”
  - “Lasso”
  - “Ridge Regression”

**Regularized Least-Squares**
- We want to minimize
  \[ \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|^q \]
  This is equivalent to minimizing
  \[ \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2 \]
  subject to the constraint
  \[ \sum_{j=1}^{M} |w_j|^q \leq \eta \]
  (for some suitably chosen \( \eta \))

**Effect: Sparsity**
- For \( q \leq 1 \).
  - Minimization tends to set many coefficients to zero

**Why is this good?**
- Why don’t we always do it, then? Any problems?
The Lasso

- Consider the following regressor
  \[ w_{\text{Lasso}} = \arg \min_w \frac{1}{2} \sum_{n=1}^N (y_n - w^T \phi(x_n))^2 + \lambda \sum_{j=1}^M |w_j| \]
  - This formulation is known as the Lasso.

- Properties
  - \( L_1 \) regularization \( \Rightarrow \) The solution will be sparse (only few coefficients will be non-zero).
  - The \( L_1 \) penalty makes the problem non-linear.
  - There is no closed-form solution.
  - Need to solve a quadratic programming problem.
  - However, efficient algorithms are available with the same computational cost as for ridge regression.

Lasso as Bayes Estimation

- Interpretation as Bayes Estimation
  \[ w = \arg \min_w \frac{1}{2} \sum_{n=1}^N (y_n - w^T \phi(x_n))^2 + \lambda \sum_{j=1}^M |w_j| \]
  - We can think of \(|w|\) as the log-prior density for \( w \).

- Prior for Lasso \((q = 1)\): Laplacian distribution
  \[ p(w) = \frac{1}{2\tau} \exp \left\{-\frac{|w|}{\tau} \right\} \quad \text{with} \quad \tau = \frac{1}{\lambda} \]

Analysis

- Equicontours of the prior distribution
  \[ q = 4 \quad q = 2 \quad q = 1 \quad q = 0.5 \quad q = 0.1 \]

- Analysis
  - For \( q \leq 1 \), the prior is not uniform in direction, but concentrates more mass on the coordinate directions.
  - The case \( q = 1 \) (lasso) is the smallest \( q \) such that the constraint region is convex.
  - Non-convexity makes the optimization problem more difficult.
  - Limit for \( q = 0 \): regularization term becomes \( \sum_{j=1}^M 1 = M \).
  - This is known as Best Subset Selection.

Discussion

- Bayesian analysis
  - Lasso, Ridge regression and Best Subset Selection are Bayes estimates with different priors.
  - However, derived as maximizers of the posterior.
  - Should ideally use the posterior mean as the Bayes estimate!
  - Ridge regression solution is also the posterior mean, but Lasso and Best Subset Selection are not.

  - We might also try using other values of \( q \) besides \( 0, 1, 2 \)...
    - However, experience shows that this is not worth the effort.
    - Values of \( q \in (1,2) \) are a compromise between lasso and ridge.
    - However, \(|w|\) with \( q > 1 \) is differentiable at 0.
    - Loses the ability of lasso for setting coefficients exactly to zero.

Topics of This Lecture

- Recap: Probabilistic View on Regression
- Properties of Linear Regression
  - Loss functions for regression
  - Basis functions
  - Multiple Outputs
  - Sequential Estimation
- Regularization revisited
  - Regularized Least-squares
  - The Lasso
  - Discussion
- Bias-Variance Decomposition

Bias-Variance Decomposition

- Recall the expected squared loss,
  \[ E[L] = \int \int \left\{ p(x) - h(x) \right\}^2 p(x) dx \right\} dx + \int \int \left\{ h(x) - t \right\}^2 p(x,t) dx dt \]
  - where
    \[ h(x) = E[t|x] = \int p(t|x) dt. \]

  - The second term of \( E[L] \) corresponds to the noise inherent in the random variable \( t \).

  - What about the first term?
Bias-Variance Decomposition

• Suppose we were given multiple data sets, each of size $N$. Any particular data set $D$ will give a particular function $y(x; D)$. We then have

$$\{y(x; D) - h(x)\}^2 = \{y(x; D) - E_D[y(x; D)] + E_D[y(x; D)] - h(x)\}^2$$

$$= \{y(x; D) - E_D[y(x; D)]\}^2 + \{E_D[y(x; D)] - h(x)\}^2$$

$$+ 2[y(x; D) - E_D[y(x; D)]][E_D[y(x; D)] - h(x)].$$

• Taking the expectation over $D$ yields

$$E_D[\{y(x; D) - h(x)\}^2] = [E_D[y(x; D)] - h(x)]^2 + E_D[\{y(x; D) - E_D[y(x; D)]\}^2].$$

• Thus we can write

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

$$(\text{bias})^2 = \int (E_D[y(x; D)] - h(x))^2 p(x) \, dx$$

$$\text{variance} = \int [E_D[y(x; D)] - E_D[y(x; D)]^2] p(x) \, dx$$

$$\text{noise} = \int (h(x) - t)^2 p(x, t) \, dx \, dt$$

Example

• 25 data sets from the sinusoidal, varying the degree of regularization, $\lambda$.

The Bias-Variance Trade-Off

• We can compute an estimate for the generalization capability this way (magenta curve)!

$\Rightarrow$ Computation is based on average w.r.t. ensembles of data sets.

$\Rightarrow$ Unfortunately of little practical value...

- An over-regularized model (large $\lambda$) will have a high bias.
- An under-regularized model (small $\lambda$) will have a high variance.
References and Further Reading

- More information on linear regression, including a discussion on regularization can be found in Chapters 1.5.5 and 3.1-3.2 of the Bishop book.
  
  Christopher M. Bishop  
  Pattern Recognition and Machine Learning  
  Springer, 2006

- Additional information on the Lasso, including efficient algorithms to solve it, can be found in Chapter 3.4 of the Hastie book.

  T. Hastie, R. Tibshirani, J. Friedman  
  Elements of Statistical Learning  