Learning and Intelligent Systems

Supervised learning via probabilistic modeling

Prof. Gunnar Rätsch
Biomedical Informatics (bmi.inf.ethz.ch)
Material: Andreas Krause
Motivation

- We have seen how we can fit prediction models (linear, non-linear) for regression and classification.
- So far, these models do not have any statistical interpretation.
- Often we would like to **statistically model** the data:
  - Quantify uncertainty
  - Express prior knowledge / assumptions about the data
- In the following, we will see how many of the approaches we have discussed can be interpreted as **fitting probabilistic models**.
- This view will allow us to derive new methods.
Recall: Bayes-optimal predictor for squared loss

- Have shown that

\[ h^*(\mathbf{x}) = \mathbb{E}[Y \mid X = \mathbf{x}] \]

- Thus, one strategy for estimating a predictor from training data is to estimate the conditional distribution

\[ \hat{P}(Y \mid X) \]

and then, for test point \( \mathbf{x} \), predict label

\[ \hat{y} = \hat{\mathbb{E}}[Y \mid X = \mathbf{x}] = \int \hat{P}(y \mid X = \mathbf{x}) \, y \, dy \]
Estimating conditional distributions

- Common approach: Parametric estimation
  - Choose a particular parametric form $\hat{P}(Y \mid X, \theta)$
  - Then optimize the parameters. How?

Maximum (conditional) Likelihood Estimation

$$\theta^* = \arg \max_{\theta} \hat{P}(y_1, \ldots, y_n \mid x_1, \ldots, x_n, \theta)$$

\[
= \arg \max_{\theta} \prod_{i=1}^{n} \hat{P}(y_i \mid x_i, \theta)
\]

\[
= \arg \max_{\theta} \log \prod_{i=1}^{n} \hat{P}(y_i \mid x_i, \theta)
\]

\[
= \arg \min_{\theta} -\sum_{i=1}^{n} \log \hat{P}(y_i \mid x_i, \theta)
\]
Least-squares regression = Gaussian MLE

- The **Maximum Likelihood Estimate (MLE)** is given by the **least squares solution**, assuming that the noise is iid Gaussian with constant variance.

- This is useful since MLE satisfies several nice statistical properties (not formally defined here):
  - **Consistency** (parameter estimate converges to true parameters in probability)
  - **Asymptotic efficiency** (smallest variance among all „well-behaved“ estimators for large $n$)
  - **Asymptotic normality**

- However, all these properties are asymptotic (hold as $n \to \infty$). For finite $n$, we must avoid overfitting!
Summary: Bias Variance Tradeoff

\[ \text{Prediction error} = \text{Bias}^2 + \text{Variance} + \text{Noise} \]

**Bias:** Excess risk of best model considered compared to minimal achievable risk knowing \( P(X,Y) \) (i.e., given infinite data)

**Variance:** Risk incurred due to estimating model from limited data

**Noise:** Risk error incurred by optimal model (i.e., irreducible error)

Trade bias and variance via model selection / regularization
Bias and variance in regression

- The maximum likelihood estimate (= least-squares fit) for linear regression is unbiased (if $h^*$ in class $H$)
- Furthermore, it is the minimum variance estimator among all unbiased estimators (Gauss-Markov Theorem, not explained further here)
- However, we have already seen that the least-squares solution can overfit

Thus, trade (a little bit of) bias for a (potentially dramatic) reduction in variance
- Regularization (e.g., ridge regression, Lasso, ...)

7
Introducing bias through Bayesian modeling

- Can introduce bias by expressing assumptions on parameters through a Bayesian prior.
- For example, let’s assume $\mathbf{w} \sim \mathcal{N}(0, \beta^2 \mathbf{I})$
  $$w_i \sim \mathcal{N}(0, \beta^2)$$
- Then, the posterior distribution of $\mathbf{w}$ is given using Bayes’ rule by

$$P(\mathbf{w} | x_{1:n}, y_{1:n}) = \frac{P(\mathbf{w}) \cdot P(y_{1:n} | x_{1:n}, \mathbf{w})}{P(y_{1:n} | x_{1:n})}$$

Bayes’ rule is applied to conditional distribution (conditioned on $x_{1:n}$).

Which parameters $\mathbf{w}$ are most likely a posteriori?
Introducing bias through Bayesian modeling

- Can introduce bias by expressing assumptions on parameters through a Bayesian prior

- For example, let’s assume \( w \sim \mathcal{N}(0, \beta^2 I) \)

- Then, the posterior distribution of \( w \) is given using Bayes’ rule by

- Which parameters \( w \) are most likely a posteriori?
Maximum a posteriori estimate

\[ P(w \mid x_1, \ldots, x_n, y_1, \ldots, y_n) = \frac{P(w) P(y_1, \ldots, y_n \mid x_1, \ldots, x_n, w)}{P(y_1, \ldots, y_n \mid x_1, \ldots, x_n)} \]

\[ \arg \max_w P(w \mid x_1, m, y_1, m) = \arg \min_w -\log P(w) - \log P(y_1, m \mid x_1, m, w) + \log P(y_1, m \mid x_1, m) \]

\[ -\log P(w) = -\log \prod_{j=1}^{d} P(w_j) = -\sum_{j=1}^{d} \log \left( \frac{1}{\sqrt{2\pi\beta^2}} \exp \left( -\frac{w_j^2}{2\beta^2} \right) \right) \]

\[ = \frac{d}{2} \log (2\pi\beta^2) + \sum_{j=1}^{d} \frac{w_j^2}{2\beta^2} = \frac{d}{2} \log (2\pi\beta^2) + \frac{1}{2\beta^2} \|w\|^2 \]

\[ \arg \min_w \frac{d}{2} \log (2\pi\beta^2) + \frac{1}{2\beta^2} \|w\|^2 + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \text{const} \]

\[ = \arg \min_w 2\|w\|^2 + \sum_{i=1}^{n} (y_i - w^T x_i)^2 \quad \text{where} \quad \lambda = \frac{d^2}{\beta^2} \]
Maximum a posteriori estimate

\[ P(w \mid x_1, \ldots, x_n, y_1, \ldots, y_n) = \frac{P(w)P(y_1, \ldots, y_n \mid x_1, \ldots, x_n, w)}{P(y_1, \ldots, y_n \mid x_1, \ldots, x_n)} \]
Ridge regression = MAP estimation

- Ridge regression can be understood as finding the Maximum A Posteriori (MAP) parameter estimate for a linear regression problem, assuming that
  - The noise $P(y|x, \mathbf{w})$ is iid Gaussian and
  - The prior $P(\mathbf{w})$ on the model parameters $\mathbf{w}$ is Gaussian

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n}(y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|^2_2 \equiv \arg\max_{\mathbf{w}} P(\mathbf{w}) \prod_i P(y_i | \mathbf{x}_i, \mathbf{w})$$
Regularization vs. MAP inference

More generally, regularized estimation can often be understood as MAP inference

\[
\arg\min_w \sum_{i=1}^n \ell(w^T x_i; x_i, y_i) + C(w) = \arg\max_w \prod_i P(y_i \mid x_i, w)P(w)
\]

\[
= \arg\max_w P(w \mid D)
\]

where \( C(w) = -\log P(w) \)

and \( \ell(w^T x_i; x_i, y_i) = -\log P(y_i \mid x_i, w) \)

This perspective allows changing priors (=regularizers) and likelihoods (=loss functions)
Example: \( l_1 \)-regularization

- Is there a prior that corresponds to \( l_1 \)-regularization?
- \textbf{Answer:} The Laplace prior

The Laplace prior

\[
p(x; \mu, b) = \frac{1}{2b} \exp \left( -\frac{|x - \mu|}{b} \right)
\]

Compare with Gaussian

\[p(x_i | \alpha, \beta)\]
Example: student-t likelihood

- Can introduce robustness by changing the likelihood (=loss) function

**Example:** (non-standardized) Student’s-t likelihood

\[ P(y \mid x, w, \nu, \sigma^2) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi \nu \sigma^2 \Gamma\left(\frac{\nu}{2}\right)}} \left(1 + \frac{(y - w^T x)^2}{\nu \sigma^2}\right)^{-\frac{\nu+1}{2}} \]

\[ P(1 | y - w^T x > t) = O(t^{-\frac{1}{2}}) \quad \alpha > 0 \]

Gaussian:

\[ P(1 | y - w^T x > t) = O(\exp(-\alpha t^2)) \quad \alpha > 0 \]
Example: student-t likelihood

- Can introduce robustness by changing the likelihood (=loss) function

- **Example**: (non-standardized) Student’s-t likelihood

\[
P(y \mid x, w, \nu, \sigma^2) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi \nu \sigma^2} \Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(y - w^T x)^2}{\nu \sigma^2}\right)^{-\frac{\nu+1}{2}}
\]

\[
P(y \mid x, w, \nu)
\]

\[
\nu = 1
\]

Student-t likelihood:
\[P(\lvert y - w^T x \rvert > t) = O(t^{\frac{\nu}{2}}) \quad \nu > 0\]

Gaussian:
\[P(\lvert y - w^T x \rvert > t) = O(e^{-\alpha t}) \quad \alpha > 0\]
Example fits

LS (normal)

Student-t (v=2)
Statistical models for classification

- So far, we have focused on *regression*
- Are there natural statistical models for *classification*?
Logistic regression

**Idea:** Use (generalized) linear model for the class probability

\[ P(Y = +1 | x, w) \approx 1 \]

\[ P(Y = +1 | x, w) \approx 0.5 \]

\[ P(Y = +1 | x, w) \approx 0 \]

\[ P(Y = +1 | x, w) = \sigma(w^T x) \]

\[ \sigma(z) = \frac{1}{1 + e^{\exp(-z)}} \]

[logistic function]
Link function for logistic regression

\[ P(Y = +1 \mid x, w) = \sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)} \]
Logistic regression

- Logistic regression (a classification method) replaces the assumption of Gaussian noise (squared loss) by iid Bernoulli noise:

\[
P(y \mid x, w) = \text{Ber}(y; \sigma(w^T x))
\]

\[
= \begin{cases} 
\frac{1}{1 + \exp(-w^T x)} & \text{if } y = +1 \\
\frac{1}{1 - \frac{1}{1 + \exp(-w^T x)}} & \text{if } y = -1 
\end{cases}
\]

- How can we estimate the parameters \( w \)?

\( \rightarrow \) Maximum Likelihood Estimation / MAP estimation
Logistic regression

Logistic regression (a classification method) replaces the assumption of Gaussian noise (squared loss) by iid Bernoulli noise:

$$P(y | x, w) = \text{Ber}(y; \sigma(w^T x))$$

$$= \begin{cases} \frac{1}{1 + \exp(-w^T x)} & \text{if } y = +1 \\ \frac{1}{1 - \frac{1}{1 + \exp(-w^T x)}} & \text{if } y = -1 \end{cases}$$

How can we estimate the parameters $w$?

Maximum Likelihood Estimation / MAP estimation
MLE for logistic regression

\[
\arg\max_w P(y_i = 1|x_i; w) = \ln -\sum_{i=1}^{n} \log P(y_i = 1|x_i; w)
\]

\[
-\log P(y_i = 1|x_i; w) = -\log \frac{1}{1 + \exp(-y_i w^T x_i)}
\]

\[
= \underbrace{\log (1 + \exp(-y_i w^T x_i))}_{L_{\text{logistic}}(w; x_i, y_i)}
\]
MLE for logistic regression

\[
\begin{align*}
\operatorname{argmax}_w \ P(y_{i,n} | x_{i,n}, w) &= \arg \min_w \ -\sum_{i=1}^{n} \log P(y_i | x_i, w) \\
-\log P(y_i | x_i, w) &= -\log \frac{1}{1 + \exp(-y_i w^T x_i)} \\
&= \log \left(1 + \exp(-y_i w^T x_i)\right)
\end{align*}
\]
MLE for logistic regression

- Negative log likelihood (=objective) function given by

\[ L(w) = \sum_{i=1}^{n} \log \left( 1 + \exp(-y_i w^T x_i) \right) \]

- The logistic loss is convex
  - Can use convex optimization techniques (e.g., SGD)
Logistic loss vs hinge loss
Gradient for logistic regression

- Loss for data point \((x, y\))

\[
\ell(w) = \log\left(1 + \exp(-yw^T x)\right)
\]

\[
\nabla_w \ell(w) = \frac{1}{1 + \exp(-yw^T x)} \exp(-yw^T x) \cdot (-yx)
\]

\[
= \frac{1}{1 + \exp(yw^T x)} \cdot (-yx)
\]

\[
= \hat{p}(y = -y \mid x, w) \cdot (-yx)
\]
Gradient for logistic regression

Loss for data point \((x, y)\)

\[
\ell(w) = \log \left( 1 + \exp(-y w^T x) \right)
\]

\[
\nabla_w \ell(w) = \frac{1}{1 + \exp(-y w^T x)} \exp(-y w^T x) (-y x)
\]

\[
= \frac{1}{1 + \exp(y w^T x)} \cdot (-y x)
\]

\[
= \hat{P}(Y = -y \mid x, w) \cdot (-y x)
\]
SGD for logistic regression

- Initialize $w$
- For $t = 1, 2, ...$
  - Pick data point $(x, y)$ uniformly at random from data D
  - Compute probability of misclassification with current model
    $$\hat{P}(Y = -y \mid w, x) = \frac{1}{1 + \exp(yw^T x)}$$
- Take gradient step
  $$w \leftarrow w + \eta_t \ y \ x \ \hat{P}(Y = -y \mid w, x)$$
Logistic regression and regularization

- Similar to SVMs and linear regression, want to use regularizer to control model complexity.
- Thus, instead of solving MLE

\[
\min_{\mathbf{w}} \sum_{i=1}^{n} \log (1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i))
\]

estimate MAP/solve regularized problem.

- **L2 (Gaussian prior):**

\[
\min_{\mathbf{w}} \sum_{i=1}^{n} \log (1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)) + \lambda \|\mathbf{w}\|_2^2
\]

- **L1 (Laplace):**

\[
\min_{\mathbf{w}} \sum_{i=1}^{n} \log (1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)) + \lambda \|\mathbf{w}\|_1
\]
Initialize $w$

For $t = 1, 2, ...$
- Pick data point $(x, y)$ uniformly at random from data $D$
- Compute probability of misclassification with current model

$$\hat{P}(Y = -y \mid w, x) = \frac{1}{1 + \exp(yw^T x)}$$

- Take gradient step

$$w \leftarrow w(1 - 2\lambda\eta_t) + \eta_t y x \hat{P}(Y = -y \mid w, x)$$
Logistic regression

Learning:

- Find optimal weights by minimizing logistic loss + regularizer

\[ \hat{w} = \arg \min_w \sum_{i=1}^{n} \log \left( 1 + \exp \left( -y_i w^T x_i \right) \right) + \lambda \|w\|_2^2 \]

\[ = \arg \max_w P(w \mid x_1, \ldots, x_n, y_1, \ldots, y_n) \]

Classification:

- Use conditional distribution

\[ P(y \mid x, \hat{w}) = \frac{1}{1 + \exp(-y \hat{w}^T x)} \]

- E.g., predict more likely class label

\[ \approx \text{sign} \left( w^T x \right) \]
Multi-class logistic regression

- Can extend logistic regression to multi-class setting
- Maintain one weight vector per class and model

\[
P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp(w_i^T x)}{\sum_{j=1}^{c} \exp(w_j^T x)} = \hat{\rho}_i
\]

\[
\hat{\rho}_i > 0 \quad \forall i, \quad \sum_{i} \hat{\rho}_i = 1
\]

Sps. replace \( w_i \) by \( \hat{w}_i^1 = w_i + \Delta \quad \forall i \)

\[
\frac{\exp((\hat{w}_i + \Delta)^T x)}{\sum_{j} \exp((\hat{w}_j + \Delta)^T x)} \quad \Rightarrow \quad \frac{\exp(w_i^T x)}{\sum_{j} \exp(w_j^T x)} = \frac{\exp(w_i^T x)}{\sum_{j} \exp(w_j^T x)}
\]

where \( c = \exp(\Delta^T x) \)
Multi-class logistic regression

- Can extend logistic regression to multi-class setting
- Maintain one weight vector per class and model

\[
P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp(w_i^T x)}{\sum_{j=1}^{c} \exp(w_j^T x)} = \hat{\rho}_i
\]

\[\hat{\rho}_i > 0 \text{ } \forall i \text{, } \sum_{i} \hat{\rho}_i = 1\]

Sps. replace \( w_i \) by \( w_i' = w_i + \Delta \) \( \forall i \)

\[
\frac{\exp((w_i + \Delta)^T x)}{\sum_{j} \exp((w_j + \Delta)^T x)} = \frac{\sum_{j} \exp(w_j^T x)}{\sum_{j} \exp(w_j^T x)} \cdot \exp(w_i^T x)
\]

where \( c = \exp(\Delta^T x) \)
Multi-class logistic regression

- Maintain one weight vector per class and model

\[ P(Y = i \mid x, w_1, \ldots, w_c) = \frac{\exp(w_i^T x)}{\sum_{j=1}^c \exp(w_j^T x)} \]

- Not unique – can force uniqueness by setting \( w_c = 0 \) (this recovers logistic regression as special case)

- Corresponding loss function (cross-entropy loss):

\[ \ell(y; x, w_1, \ldots, w_c) = -\log P(Y = y \mid x, w_1, \ldots, w_c) \]
Training neural nets for multi-class

Loss: \[ \ell(Y = i; f_1, \ldots, f_c) = - \log \frac{\exp(f_i)}{\sum_{j=1}^{c} \exp(f_j)} \]
## SVM vs. Logistic regression

<table>
<thead>
<tr>
<th>Method</th>
<th>SVM / Perceptron</th>
<th>Logistic regression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>Sometimes higher classification accuracy; Sparse sol’s</td>
<td>Can obtain class probabilities</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>Can’t (easily) get class probabilities</td>
<td>Dense solutions</td>
</tr>
</tbody>
</table>
Probabilistic modeling big picture so far

- Least squares Regression
- Ridge Regression
- Logistic regression
- GP Regression
- GP Classification