Bayes Decision Theory Cookbook

Karel Zimmermann

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Center for Machine Perception
http://cmp.felk.cvut.cz/~zimmerk, zimmerk@fel.cvut.cz

Minimum risk problem

- $\mathbf{x}$ features
- ω discrete states of the nature, categories, classes
- $P(\omega|\mathbf{x})$ conditional probability of being in state ω

Minimum risk problem

- $\mathbf{x}$ features
- ω discrete states of the nature, categories, classes
- $P(\omega|\mathbf{x})$ conditional probability of being in state ω
- α discrete actions, decisions
- $\lambda(\alpha|\omega)$ loss-function (loss caused by doing action α while being in state ω)

Minimum risk problem

- $\mathbf{x}$ features
- ω discrete states of the nature, categories, classes
- $P(\omega|\mathbf{x})$ conditional probability of being in state ω
- α discrete actions, decisions
- $\lambda(\alpha|\omega)$ loss-function (loss caused by doing action α while being in state ω)
- $R(\alpha|\mathbf{x}) = \sum_{\omega} \lambda(\alpha|\omega) P(\omega|\mathbf{x})$ conditional risk.

Minimum risk problem

- $\mathbf{x}$ features
- ω discrete states of the nature, categories, classes
- $P(\omega|\mathbf{x})$ conditional probability of being in state ω
- α discrete actions, decisions
- $\lambda(\alpha|\omega)$ loss-function (loss caused by doing action α while being in state ω)
- $R(\alpha|\mathbf{x}) = \sum_{\omega} \lambda(\alpha|\omega) P(\omega|\mathbf{x})$ conditional risk.
- $\alpha^*(\mathbf{x}) = \arg\min_{\alpha} R(\alpha|\mathbf{x})$ optimal strategy.

Classification problem

- Actions α_1, α_2 corresponds to classes ω_1, ω_2 (e.g. α_1 means that we classify the object to class ω_1)
- Only two classes/actions.]pause
- $\lambda(\alpha_i|\omega_j) = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}$ loss-function is 2×2 table.

Classification problem

- Actions α_1, α_2 corresponds to classes ω_1, ω_2 (e.g. α_1 means that we classify the object to class ω_1)
- Only two classes/actions.]pause
- $\lambda(\alpha_i|\omega_j) = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}$ loss-function is 2×2 table.
- $R(\alpha_1|\mathbf{x}) = \lambda_{11} P(\omega_1|\mathbf{x}) + \lambda_{12} P(\omega_2|\mathbf{x})$.
- $R(\alpha_2|\mathbf{x}) = \lambda_{21} P(\omega_1|\mathbf{x}) + \lambda_{22} P(\omega_2|\mathbf{x})$.

Classification problem

- Actions α_1, α_2 corresponds to classes ω_1, ω_2 (e.g. α_1 means that we classify the object to class ω_1)
- Only two classes/actions.]pause
- $\lambda(\alpha_i|\omega_j) = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}$ loss-function is 2×2 table.
- $R(\alpha_1|\mathbf{x}) = \lambda_{11} P(\omega_1|\mathbf{x}) + \lambda_{12} P(\omega_2|\mathbf{x})$.
- $R(\alpha_2|\mathbf{x}) = \lambda_{21} P(\omega_1|\mathbf{x}) + \lambda_{22} P(\omega_2|\mathbf{x})$.
- Optimal strategy:

$$\alpha^*(\mathbf{x}) = \arg\min_{\alpha \in \{\alpha_1, \alpha_2\}} R(\alpha|\mathbf{x}) = \begin{cases} \alpha_1 & \text{if } R(\alpha_1|\mathbf{x}) < R(\alpha_2|\mathbf{x}) \\ \alpha_2 & \text{otherwise} \end{cases}$$

Classifier

- Given observed feature $\mathbf{x}$, we decide for α_1 if:

Classifier

- Given observed feature $\mathbf{x}$, we decide for α_1 if:

$$\lambda_{11}P(\omega_1|\mathbf{x}) + \lambda_{12}P(\omega_2|\mathbf{x}) < \lambda_{21}P(\omega_1|\mathbf{x}) + \lambda_{22}P(\omega_2|\mathbf{x})$$

Classifier

- Given observed feature $\mathbf{x}$, we decide for α_1 if:
  $\lambda_{11} P(\omega_1|\mathbf{x}) + \lambda_{12} P(\omega_2|\mathbf{x}) < \lambda_{21} P(\omega_1|\mathbf{x}) + \lambda_{22} P(\omega_2|\mathbf{x})$
- if $\lambda_{11} < \lambda_{12}$ and $\lambda_{22} < \lambda_{21}$ then the condition is rewritten as follows:

$$\underbrace{\frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}}_{\text{likelihood ratio}} > \underbrace{\frac{\lambda_{12} - \lambda_{21}}{\lambda_{21} - \lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}}_{\text{constant } \theta}$$

- θ is hard to define explicitly $\Rightarrow$ function $f(\mathbf{x})$ converging to the likelihood ratio is trained:

$$f(\mathbf{x}) = \frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}$$

Classifier

- Given observed feature $\mathbf{x}$, we decide for α_1 if:
  $\lambda_{11}P(\omega_1|\mathbf{x}) + \lambda_{12}P(\omega_2|\mathbf{x}) < \lambda_{21}P(\omega_1|\mathbf{x}) + \lambda_{22}P(\omega_2|\mathbf{x})$
- if $\lambda_{11} < \lambda_{12}$ and $\lambda_{22} < \lambda_{21}$ then the condition is rewritten as follows:

$$\underbrace{\frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}}_{\text{likelihood ratio}} > \underbrace{\frac{\lambda_{12} - \lambda_{21}}{\lambda_{21} - \lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}}_{\text{constant } \theta}$$

- θ is hard to define explicitly $\Rightarrow$ function $f(\mathbf{x})$ converging to the likelihood ratio is trained:

$$f(\mathbf{x}) = \frac{P(\mathbf{x}|\omega_1)}{P(\mathbf{x}|\omega_2)}$$

- Then the threshold θ parametrize the decision function (classifier)

$$\alpha^*(\mathbf{x};\theta) = \begin{cases} \alpha_1 & \text{if } f(\mathbf{x}) > \theta \\ \alpha_2 & \text{otherwise} \end{cases}$$

Measuring classifier quality

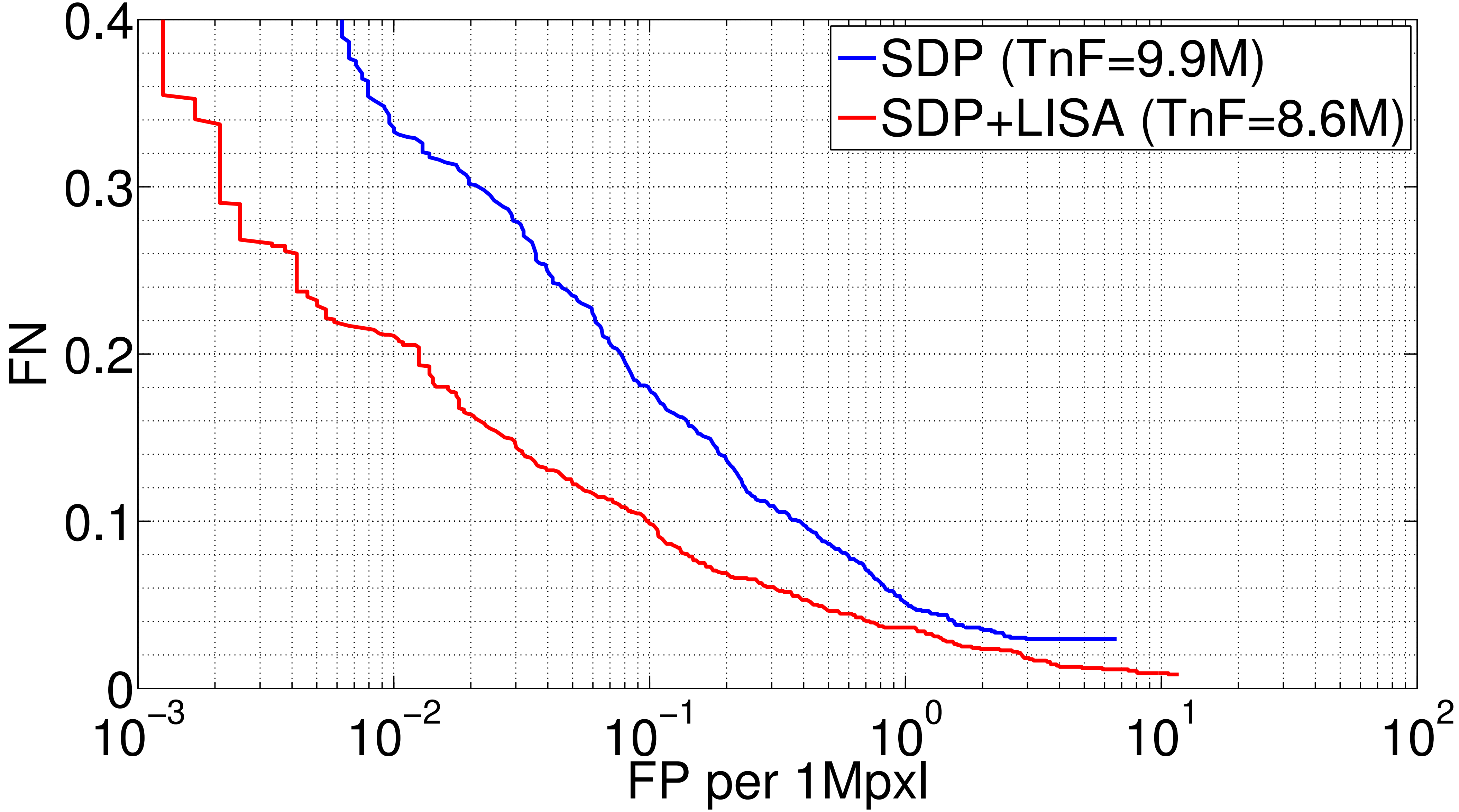
- $\omega_1/\alpha_1 \ldots$ unsafe class
  (e.g. ill patient, power-plant explosion, human detected in security camera).
- $\omega_2/\alpha_2 \ldots$ safe class
  (e.g. healthy patient, power-plant safe state, no human in security camera)
- $X_i = \{\mathbf{x} \mid \alpha^*(\mathbf{x};\theta) = \alpha_i\}$ set of all features which we classify to ω_i.
- **False negative ratio:** Probability of missing a dangerous situation (i.e the case where object is in unsafe class ω_1 and we report the safe class ω_2).

$$FN(\theta) = \sum_{\mathbf{x} \in X_2} p(\mathbf{x}|\omega_1) \approx \frac{\#\text{ of } \omega_1\text{-objects classified to } \omega_2}{\#\text{ of } \omega_1\text{-objects}}$$

- **False positive ratio:** Probability of false alarm (i.e the object is in safe class ω_2 ans we report for unsafe class α_1.

$$FP(\theta) = \sum_{\mathbf{x} \in X_1} p(\mathbf{x}|\omega_2) \approx \frac{\#\text{ of } \omega_2\text{-objects classified to } \omega_1}{\#\text{ of } \omega_2\text{-objects}}$$

Measuring classifier quality - ROC



- Which one is better?