Classification methods

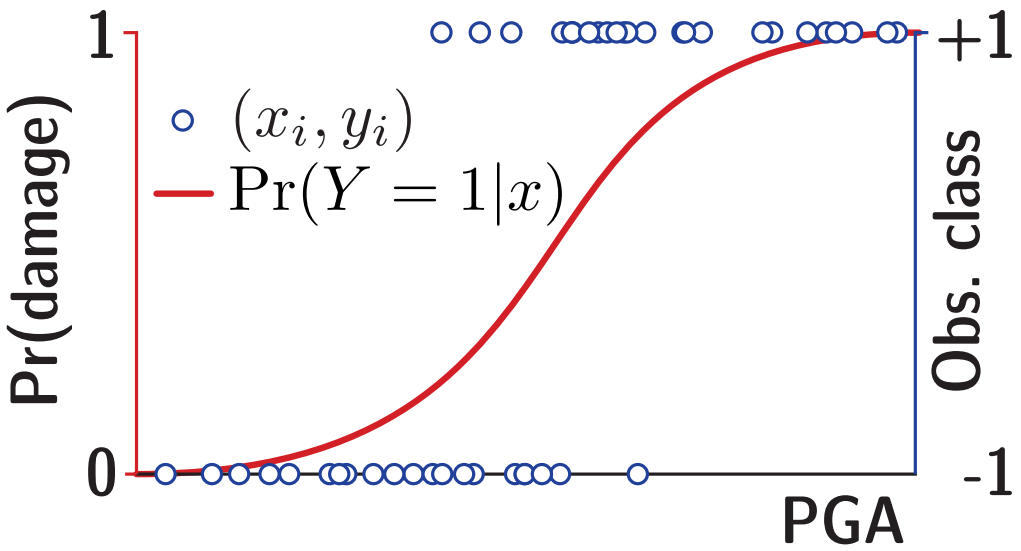
(CIV6540 - Probabilistic Machine Learning for Civil Engineers)

Professor: James-A. Goulet

Département des génies civil, géologique et des mines
Polytechnique Montréal

Chapter 9 – Goulet (2020)
Probabilistic Machine Learning for Civil Engineers
MIT Press

What is classification?



Data

$\mathcal{D} = \{(x_i, y_i), \forall i = 1 : \mathtt{D}\}$

$x_i \in \mathbb{R}$: Covariate / attribute / regressor

$y_i \in \{-1, 1\}$: Observation

Classification methods: mathematical models for $\Pr(Y|x)$

2 Types of classification methods

There are two types of classification methods:

1. **Generative**

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

2. **Discriminative**

$$p(y|x) = g(x)$$

Module #6 Outline

Introduction
Generative classifier
Logistic Regression
GPC
Neural Networks
Summary

Topics organization

Background
- 1 Revision probability & linear algebra
- 2 Probability distributions

Machine Learning Basics
- 0 Introduction
- 3 Bayesian Estimation $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$
- 4 MCMC sampling & Newton

Supervised learning
- 5 Regression
- 6 Classification
- 7 LSTM networks for time series

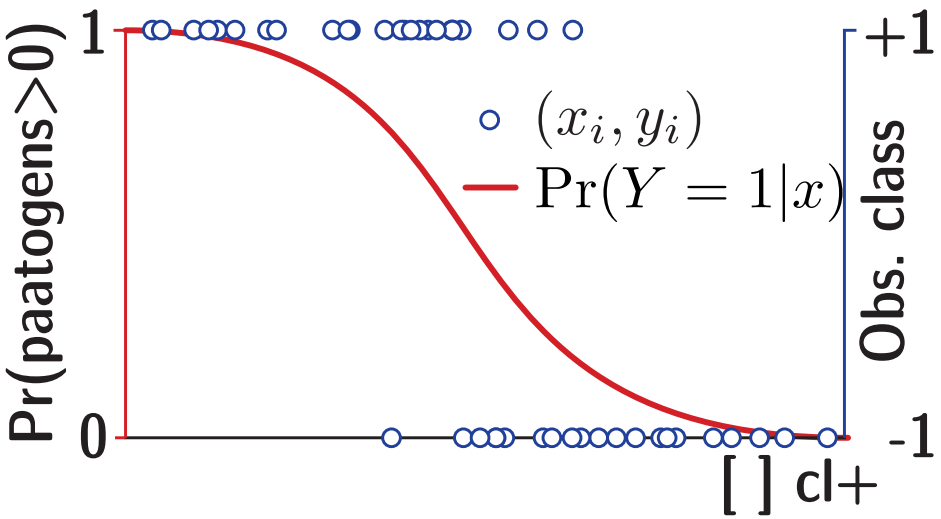
Unsupervised learning
- 7 State-space model for time-series

Decision Making & RL
- 8 Decision Theory
- 9 AI & Sequential decision problems

Section Outline

Generative classifier

Classification and Bayes rule



Data

$$\mathcal{D} = \{(x_i, y_i), \forall i = 1 : \mathtt{D}\}$$

$$x_i \in \mathbb{R} : \begin{cases} \text{Covariate} \\ \text{attribute} \\ \text{regressor} \end{cases}$$

$$y_i \in \{-1, 1\} : \text{Observation}$$

$$f(y, x) = p(y|x)f(x) = f(x|y)p(y) \rightarrow p(y|x) = \frac{f(x|y) \cdot p(y)}{f(x)}$$

$$\boxed{\Pr(Y = i|x) = \frac{f(x|y = i) \cdot p(y = i)}{f(x) \sum_j f(x|y = j) \cdot p(y = j)}}$$

Formulation - Binary classification $y \in \{-1, 1\}$

$$\overbrace{p(y|x)}^{\text{posterior}} = \frac{\overbrace{f(x|y)}^{\text{likelihood}} \cdot \overbrace{p(y)}^{\text{prior}}}{\underbrace{f(x)}_{\text{norm. cte.}}}$$

Posterior probability of a class

$$p(y|x) = \begin{cases} \Pr(Y = -1|x) \\ \Pr(Y = +1|x) = 1 - \Pr(Y = -1|x) \end{cases}$$

Prior probability of a class

$$p(y) = \begin{cases} \Pr(Y = -1) \\ \Pr(Y = +1) \end{cases}$$

Likelihood of x_i given a class

$$f(x|y) = \begin{cases} f(x|y = -1) \\ f(x|y = +1) \end{cases}$$

Normalization constant

$$f(x) = \sum_{y \in \{-1,1\}} f(x|y) \cdot p(y) = cte.$$

Learning parameters - Binary classification

$$\overbrace{p(y|x)}^{\text{posterior}} = \frac{\overbrace{f(x|y)}^{\text{likelihood}} \cdot \overbrace{p(y)}^{\text{prior}}}{\underbrace{f(x)}_{\text{norm. cte.}}}$$

We need to lean the PDF/PMF parameters from $\mathcal{D}$
MLE/Bayes

Formulation - Special case $f(x|y) = \mathcal{N}(x; \mu, \sigma^2)$

If we assume that $f(x|y=j) = \mathcal{N}(x; \mu^*_{x|y_j}, \sigma^{2*}_{x|y_j})$, and that the number of available data $\mathtt{D}$ is large

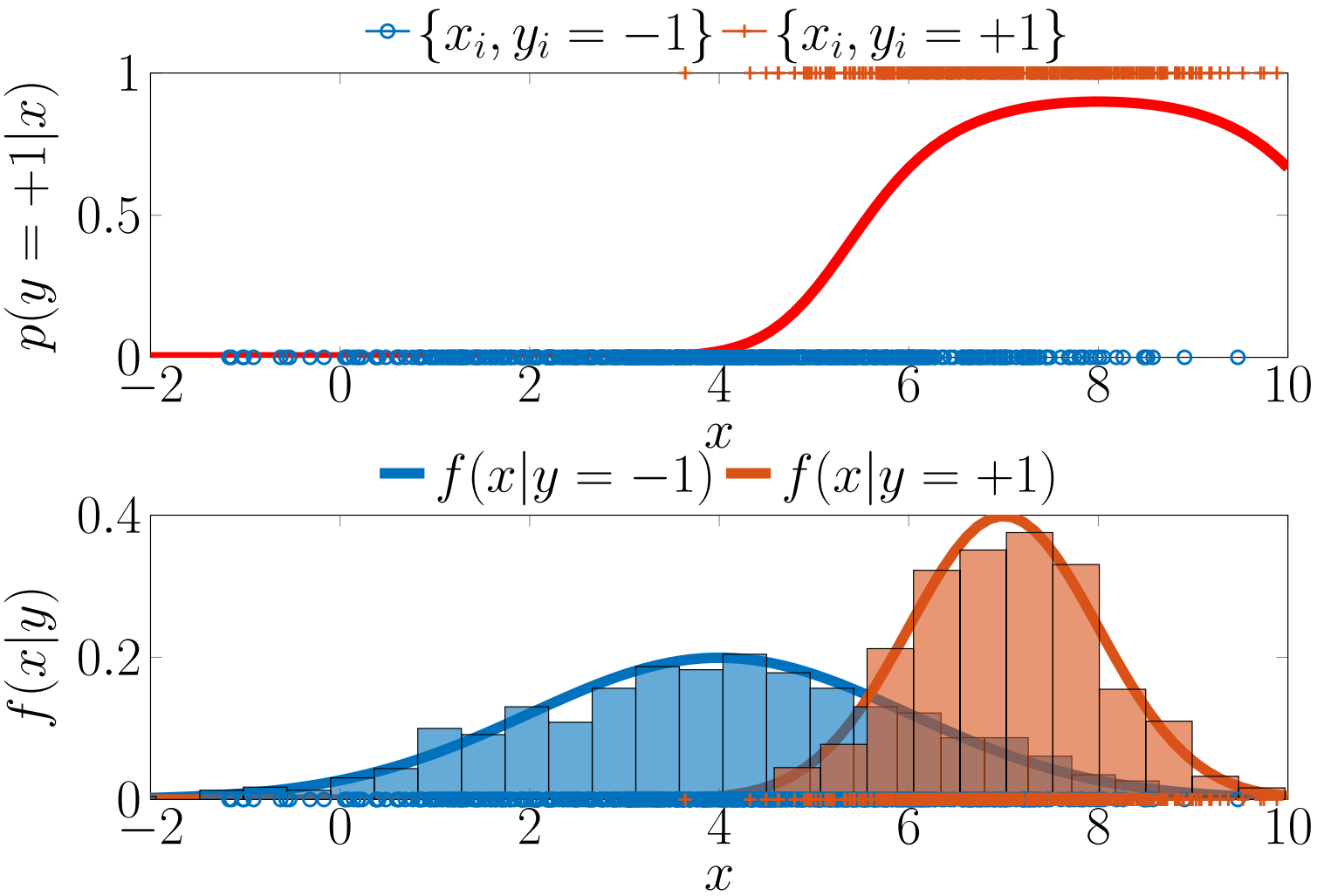
MLE approximation of $\mu_{x|y_j}$ & $\sigma^2_{x|y_j}$

$$\begin{aligned} \mu^*_{x|y_j} &= \frac{1}{\#\{i:y_i=j\}} \sum_i x_i, \ \forall\{i : y_i = j\} \\ \sigma^{2*}_{x|y_i} &= \frac{1}{\#\{i:y_i=j\}} \sum_i (x_i - \mu^*_{x|y_j})^2, \ \forall\{i : y_i = j\} \end{aligned}$$

MLE approximation of $p(y)$

$$p^*(y) = \left\{ \Pr^*(y=j) = \frac{\#\{i:y_i=j\}}{\mathtt{D}} \right.$$

Example – Binary classification [.m]



[CIV_ML/Classification_example_1.m]

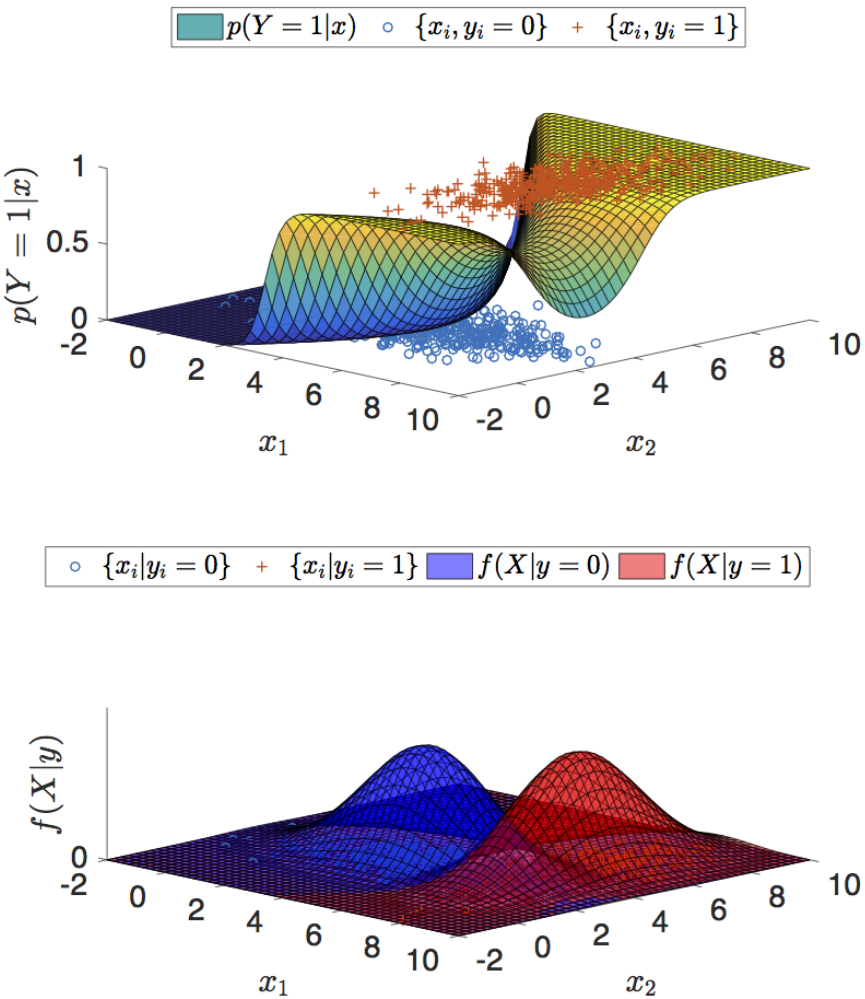
Multiattribute classification

What if there is **more than one attribute** $x = [x_1, x_2, \cdots, x_{\mathrm{X}}]$**?**

Employ joint PDFs: $f(\mathbf{x}|y = j)$

$$f(\mathbf{x}|y = j) = \mathcal{N}(\mathbf{x}; \underbrace{\boldsymbol{\mu}_{\mathbf{x}|y_j}, \boldsymbol{\Sigma}_{\mathbf{x}|y_j}}_{\text{MLE/Bayes}})$$

Example – Multiattribute classification [.m]



[CIV_ML/Classification_example_1_2D.pdf]

Formulation - Multi-classes classification

$$y \in \{1, 2, \cdots, \mathtt{Y}\}$$

$$\overbrace{p(y|x)}^{\text{posterior}} = \frac{\overbrace{f(x|y)}^{\text{likelihood}} \cdot \overbrace{p(y)}^{\text{prior}}}{\underbrace{f(x)}_{\text{norm. cte.}}}$$

Posterior probability of a class

$$p(y|x) = \{ \Pr(Y = j|x)$$

Prior probability of a class

$$p(y) = \{ \Pr(Y = j)$$

Likelihood of x_i given a class

$$f(x|y) = \{ f(x|y = j)$$

Normalization constant

$$f(x) = \sum_j f(x|y = j) \cdot p(y = j)$$

Context

Societal challenge

Right after an earthquake

– which building is safe?

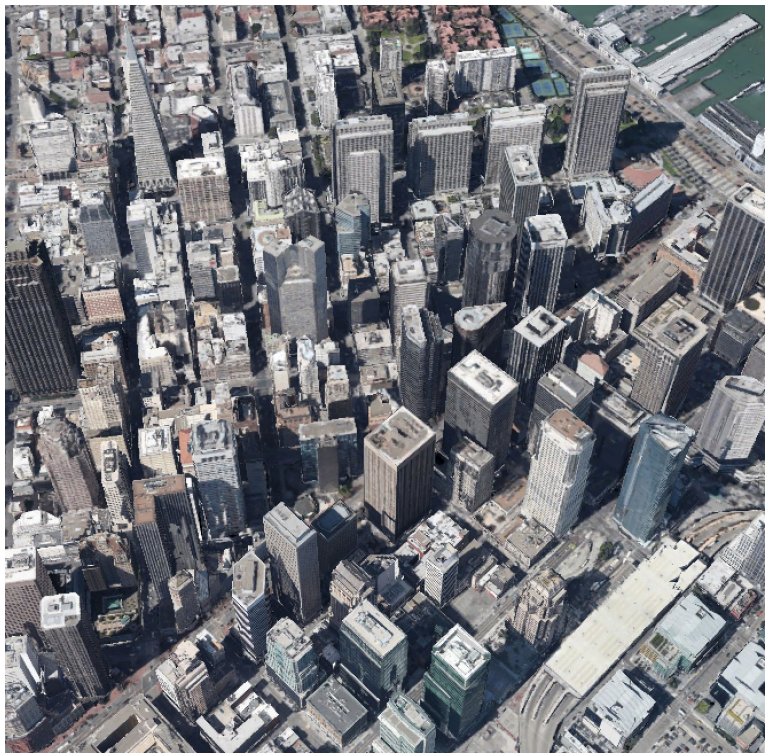
– which one should be evacuated?

Solution

Monitor structures

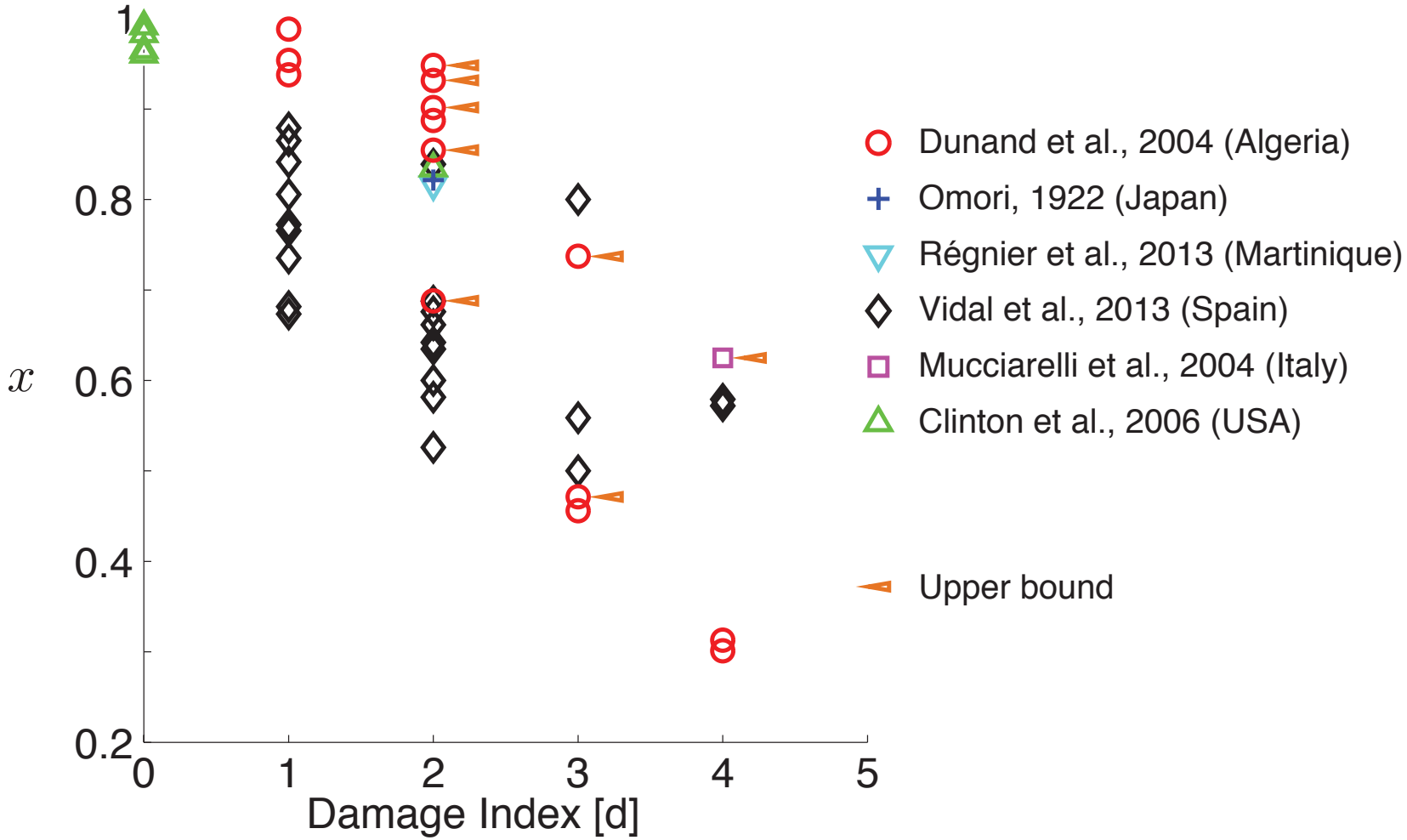
Scientific challenges

Real-time data-driven learning

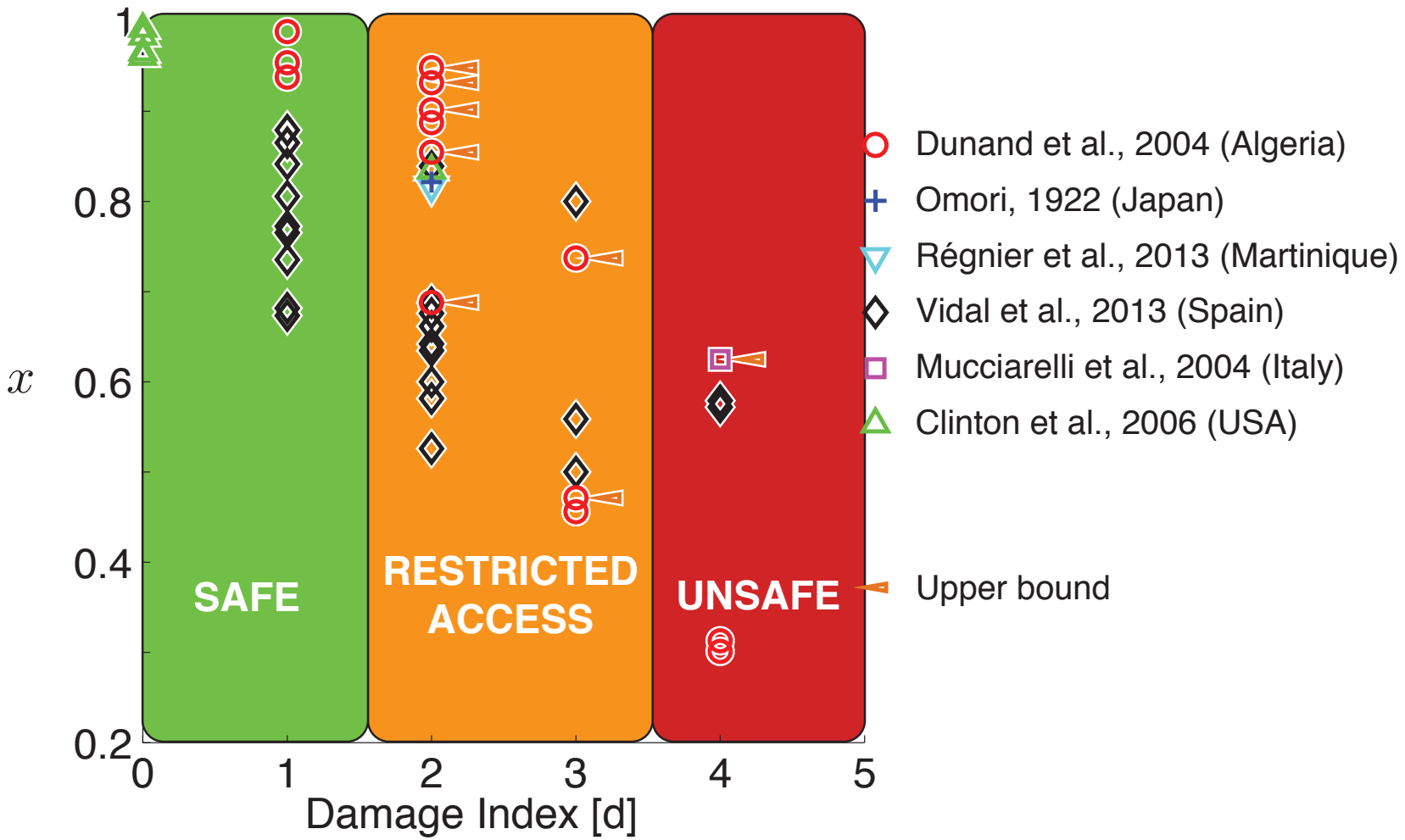


[Google images]

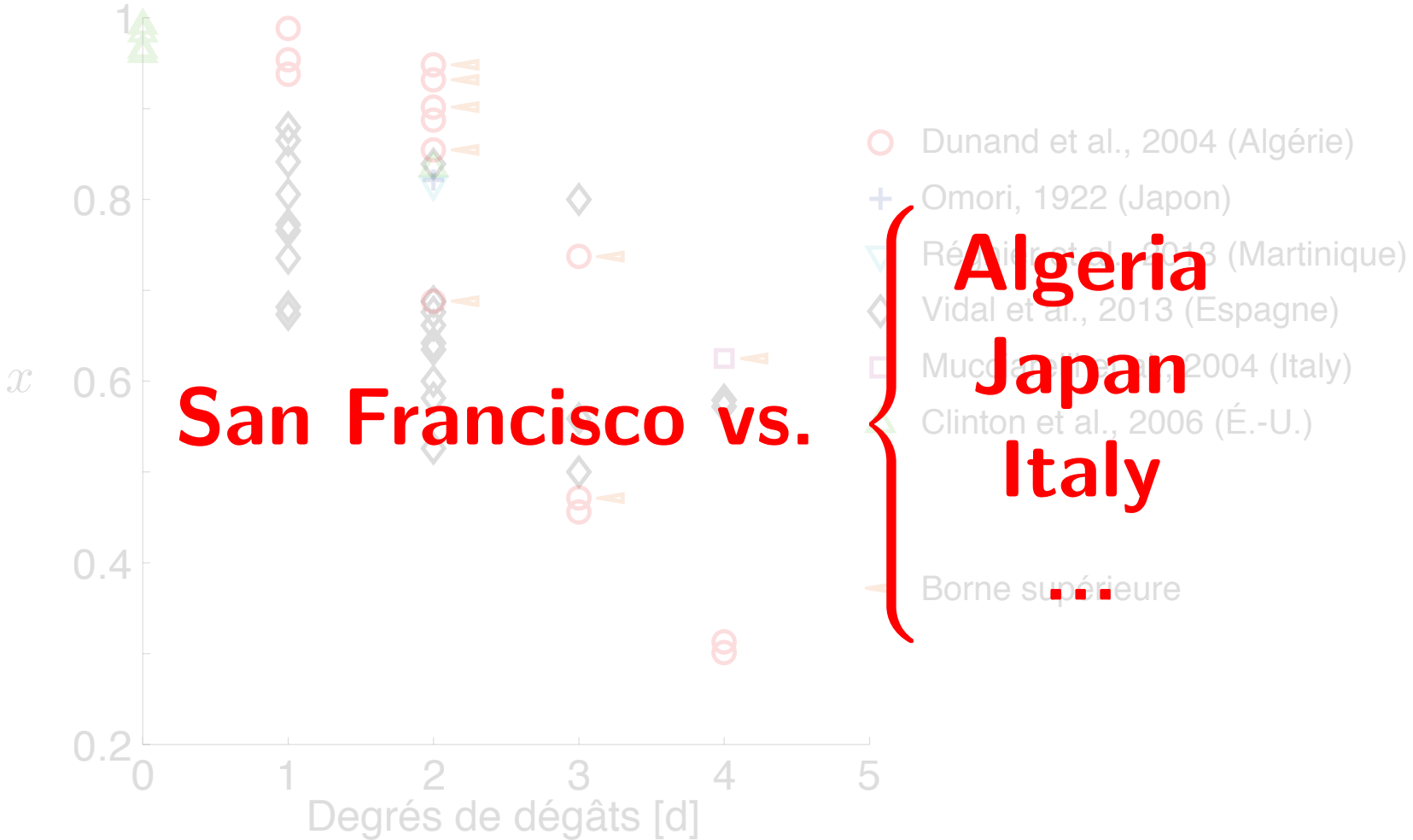
Frequency ratio: $x = \text{fq}_{\cdot\text{post-earthquake}} / \text{fq}_{\cdot\text{pre-earthquake}}$



Frequency ratio: $x = \text{fq}_{\cdot\text{post-earthquake}} / \text{fq}_{\cdot\text{pre-earthquake}}$



Frequency ratio: $x = \text{fq}_{\cdot\text{post-earthquake}} / \text{fq}_{\cdot\text{pre-earthquake}}$



Frequency ratio: $x = \text{fq}_{\cdot\text{post-earthquake}} / \text{fq}_{\cdot\text{pre-earthquake}}$

We need to learn as inspections are realized after an earthquake

Step 1 - Probabilistic learning

Given: $\mathcal{D} = \{ \underbrace{x_i}_{\in(0,1)}, \underbrace{d_i}_{\in\{0:5\}} \}, \ \forall i = 1 : \mathtt{D}$

Goal: Learn what is the conditional PDF (predictive) of X given D

$$f(x; \boldsymbol{\theta}(d)) = \mathcal{B}(x; \underbrace{\mu(d), \sigma(d)}_{\boldsymbol{\theta}})$$

$$\overbrace{f(\boldsymbol{\theta}(d)|\mathcal{D})}^{\text{posterior}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta}(d))}^{\text{likelihood}} \cdot \overbrace{f(\boldsymbol{\theta}(d))}^{\text{prior}}}{\underbrace{f(\mathcal{D})}_{\text{cte.}}}$$

$$f(x|d, \mathcal{D}) = \int f(x; \boldsymbol{\theta}(d)) \cdot f(\boldsymbol{\theta}(d)|\mathcal{D}) d\boldsymbol{\theta}$$

Step 1 - Probabilistic learning (likelihood & prior)

$$\overbrace{f(\mu(d),\sigma(d)|\mathcal{D})}^{\text{posterior}} \propto \overbrace{f(\mathcal{D}|\mu(d),\sigma(d))}^{\text{likelihood}} \cdot \overbrace{f(\mu(d),\sigma(d))}^{\text{prior}}$$

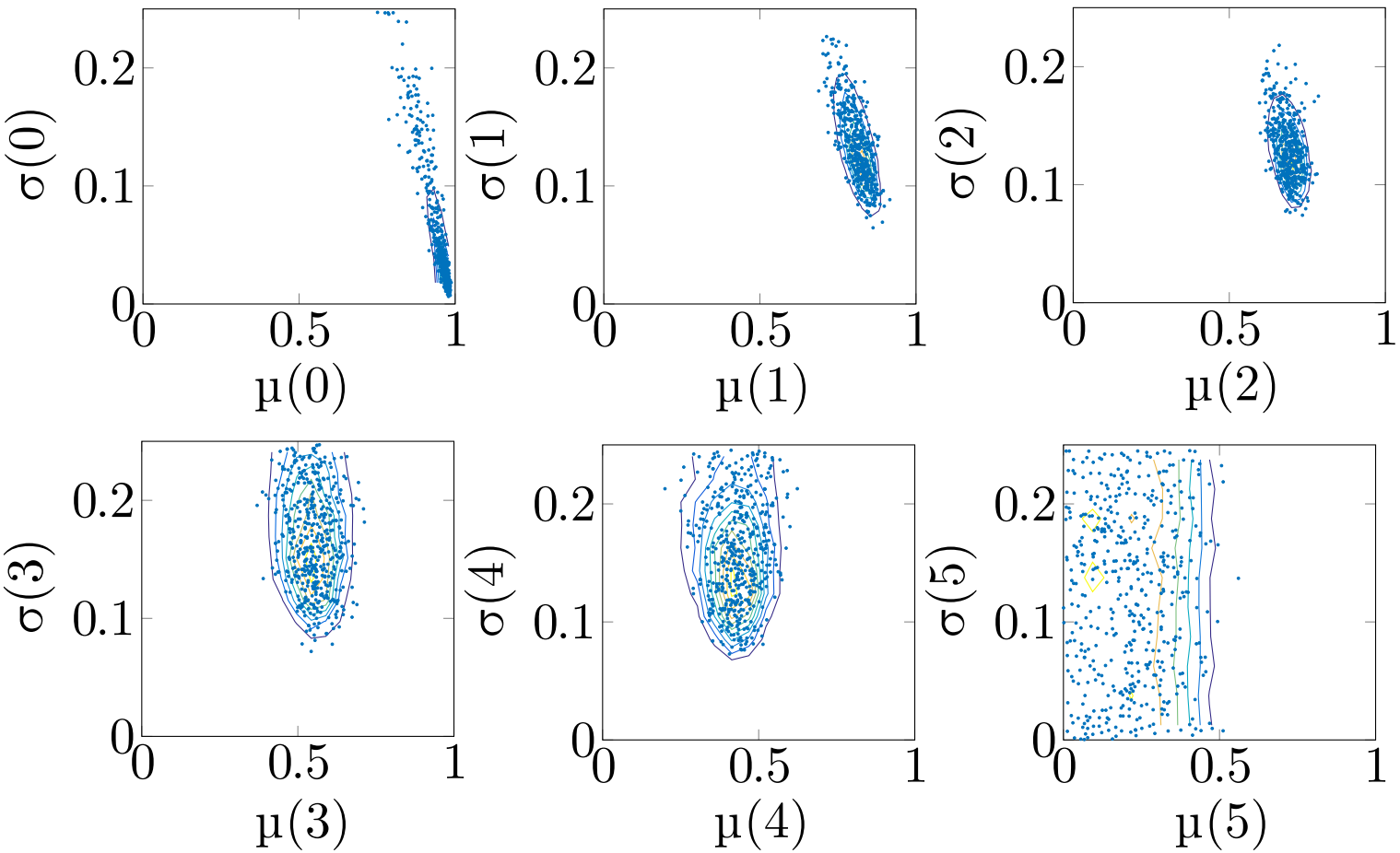
$$f(\mathcal{D}|\boldsymbol{\theta}) = \prod_{\{i:d_i=d\}} \mathcal{L}(x_i|\mu(d),\sigma(d))$$

$$\mathcal{L}(x_i|\mu(d),\sigma(d)) = \begin{cases} f(x_i|\mu(d),\sigma(d)), & \text{if } x_i \text{ direct obs.} \\ F(x_i|\mu(d),\sigma(d)), & \text{if } x_i \text{ censored obs.} \end{cases}$$

We assume no prior information except for the constraint that

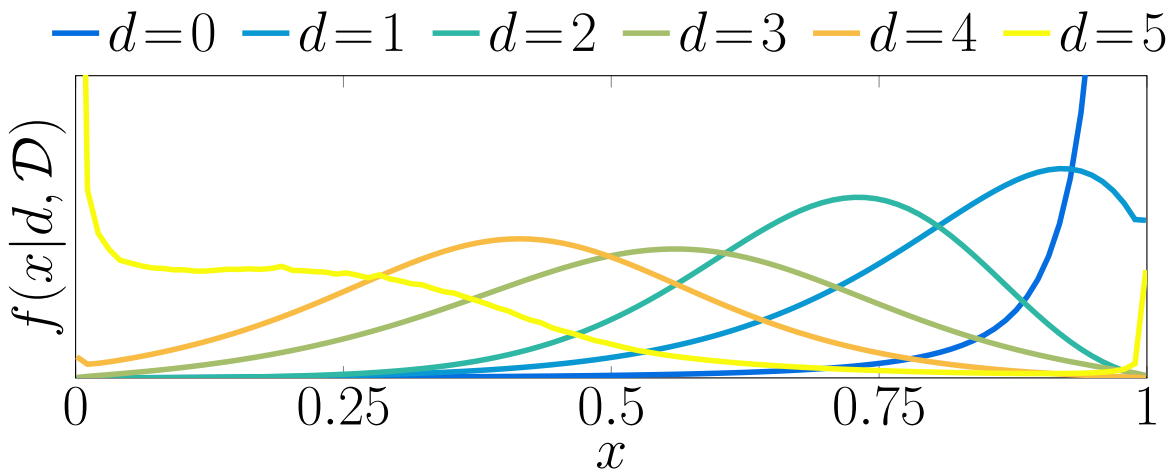
$$\mu(0) > \mu(1) > \cdots > \mu(5)$$

Application - $f(\mu(d), \sigma(d)|\mathcal{D})$ & $f(x|d)$



Application - $f(\mu(d), \sigma(d)|\mathcal{D})$ & $f(x|d)$

$$f(x|d,\mathcal{D}) = \int\int \mathcal{B}(x;\mu(d),\sigma(d)) f(\mu(d),\sigma(d)|\mathcal{D}) d\mu(d) d\sigma(d)$$

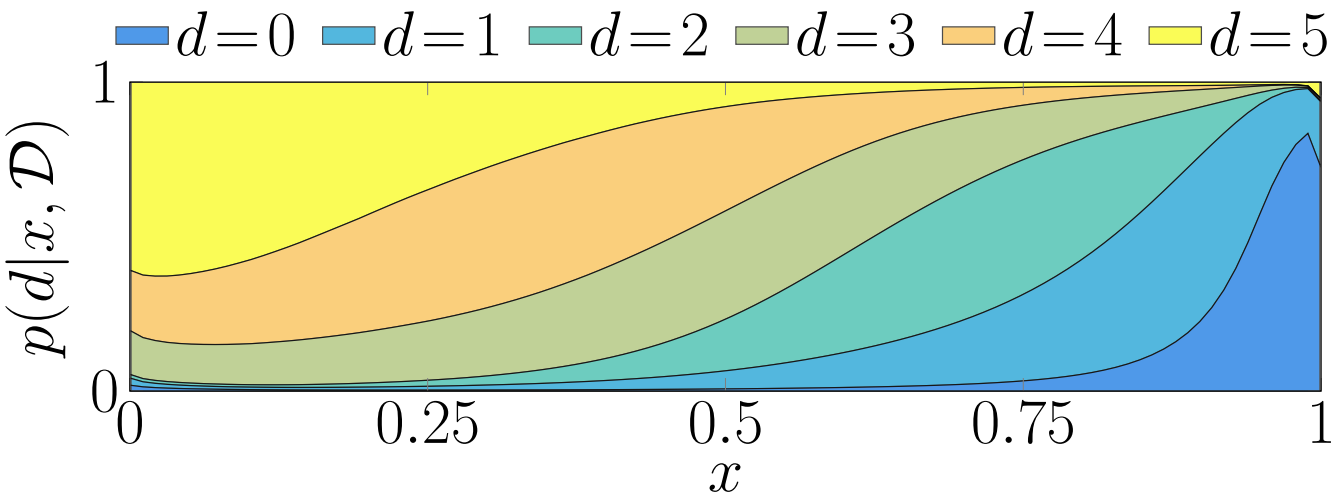


$$f(x|\boldsymbol{\theta}(d)) = \mathcal{B}(x;\mu(d),\sigma(d))$$

We have $f(x|d,\mathcal{D})$, we want $p(d|x,\mathcal{D})$

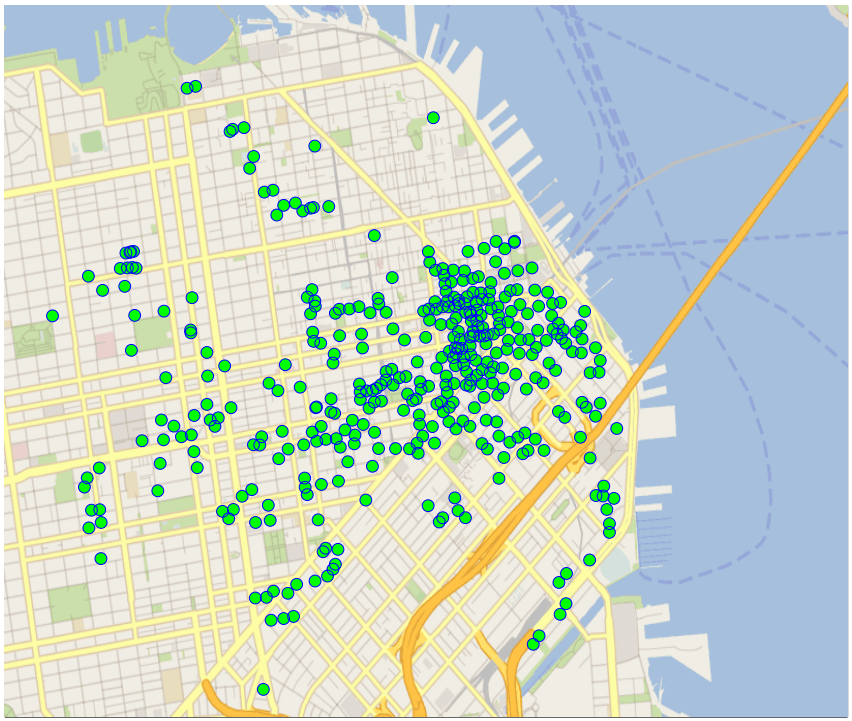
Step 2- Probabilistic prognosis

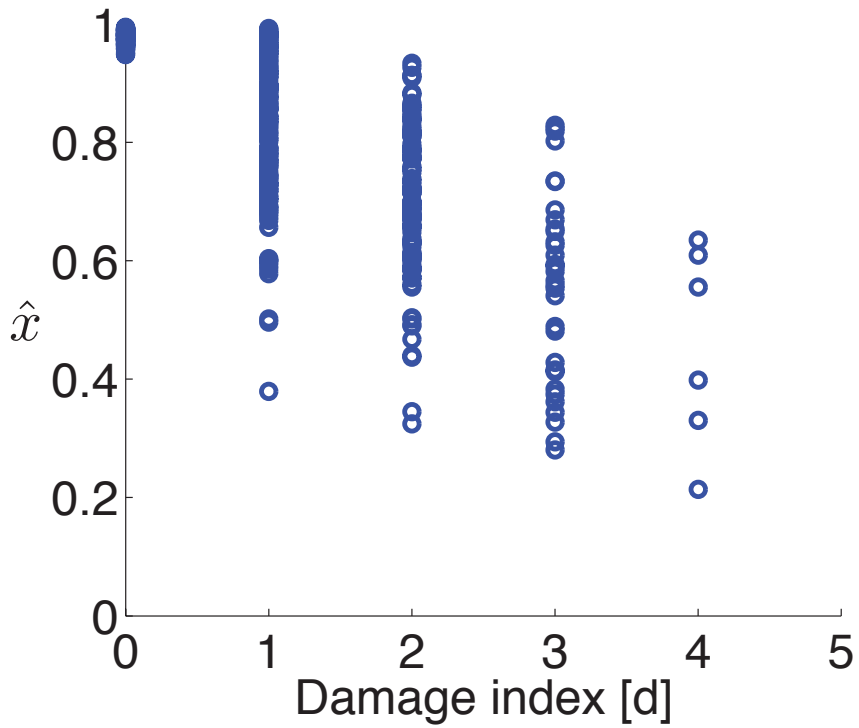
$$p(d|x,\mathcal{D}) = \int \frac{f(x;\boldsymbol{\theta}(d)) \cdot p(d)}{\sum_{d'=0}^{5} f(x;\boldsymbol{\theta}(d')) \cdot p(d')} \cdot f(\boldsymbol{\theta}(d)|\mathcal{D}) d\boldsymbol{\theta}$$

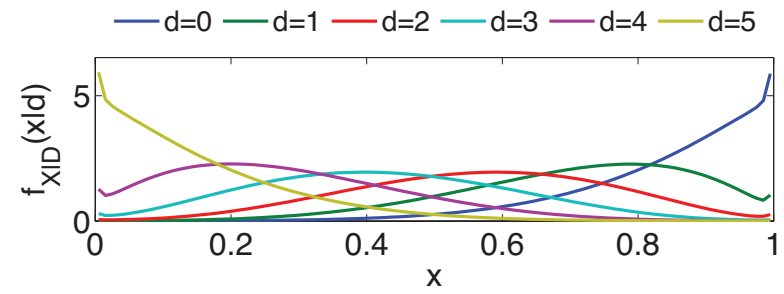


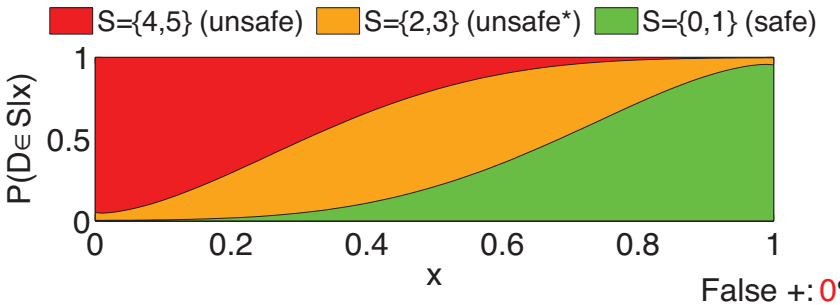
Post-earthquake damage simulation for San Francisco

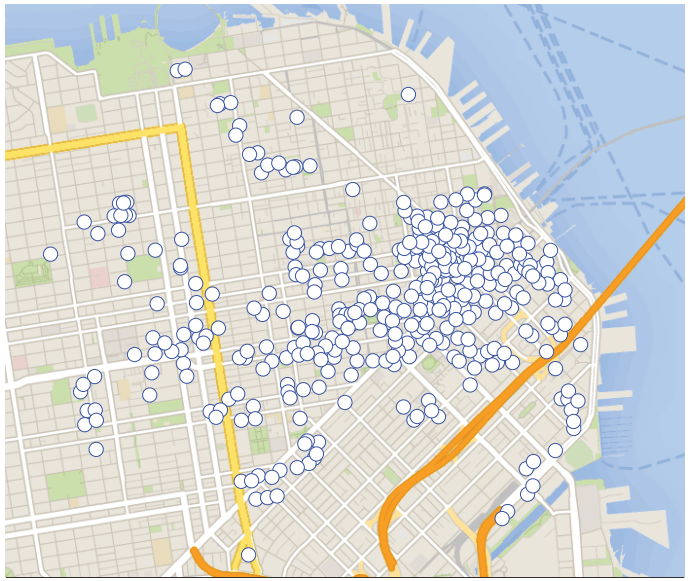
389 buildings $\geq$ 10 story

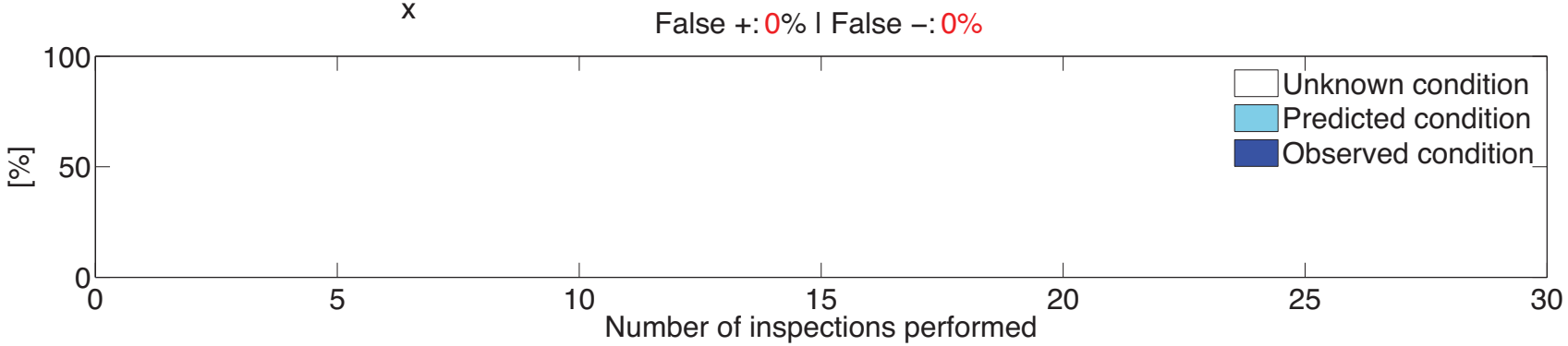


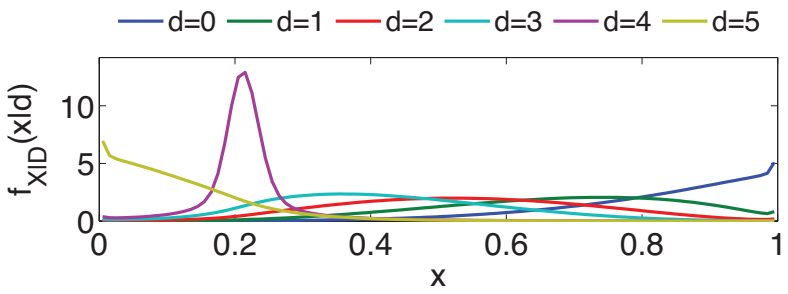


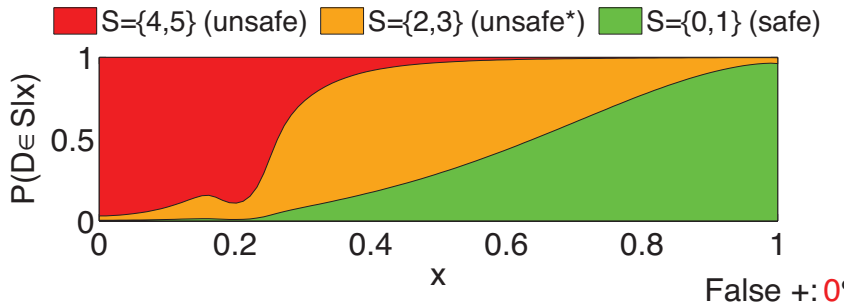

d=0
d=1
d=2
d=3
d=4
d=5
$f_{X|D}(x|d)$
5
0
0
0.2
0.4
0.6
0.8
1
x

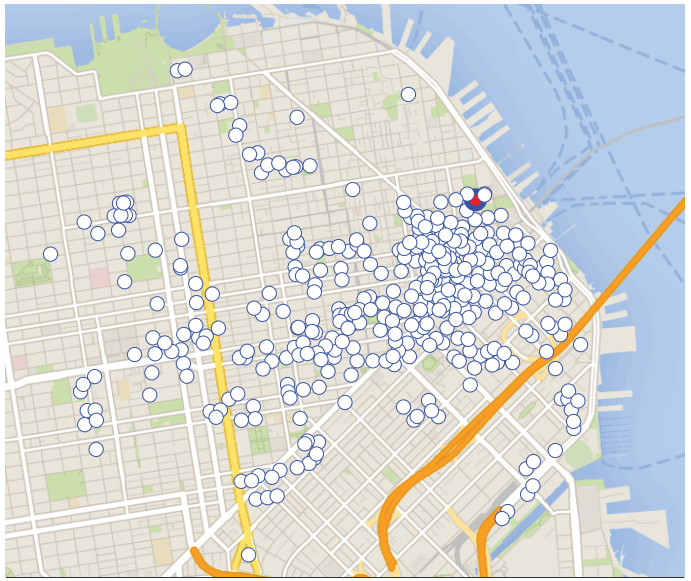

S={4,5} (unsafe)
S={2,3} (unsafe*)
S={0,1} (safe)
P(D∈S|x)
1
0.5
0
0
0.2
0.4
0.6
0.8
1
x



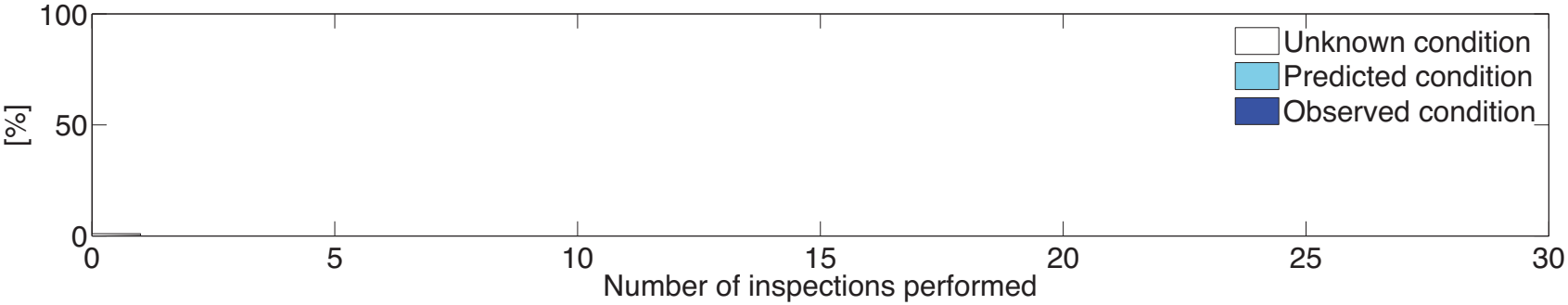

False +: 0% | False −: 0%
100
50
0
[%]
0
5
10
15
20
25
30
Number of inspections performed
Unknown condition
Predicted condition
Observed condition

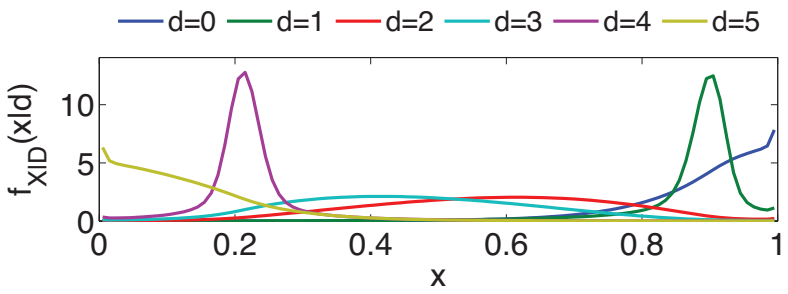


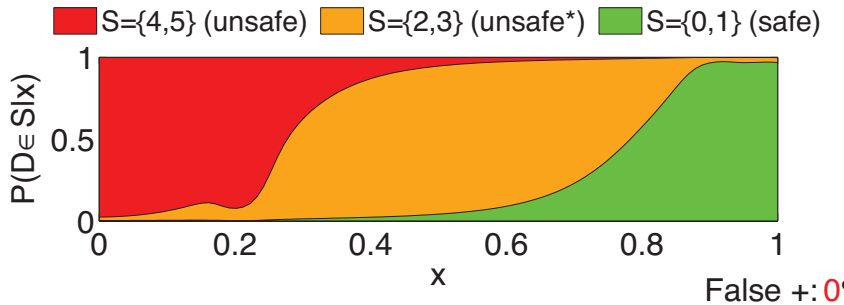


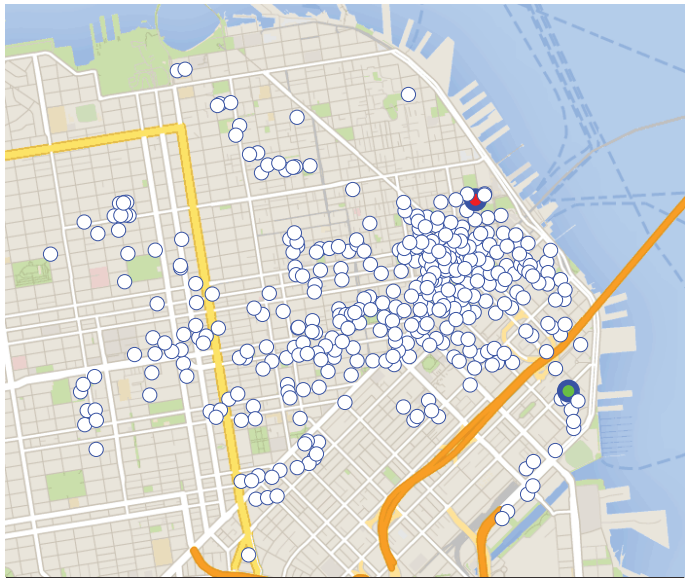


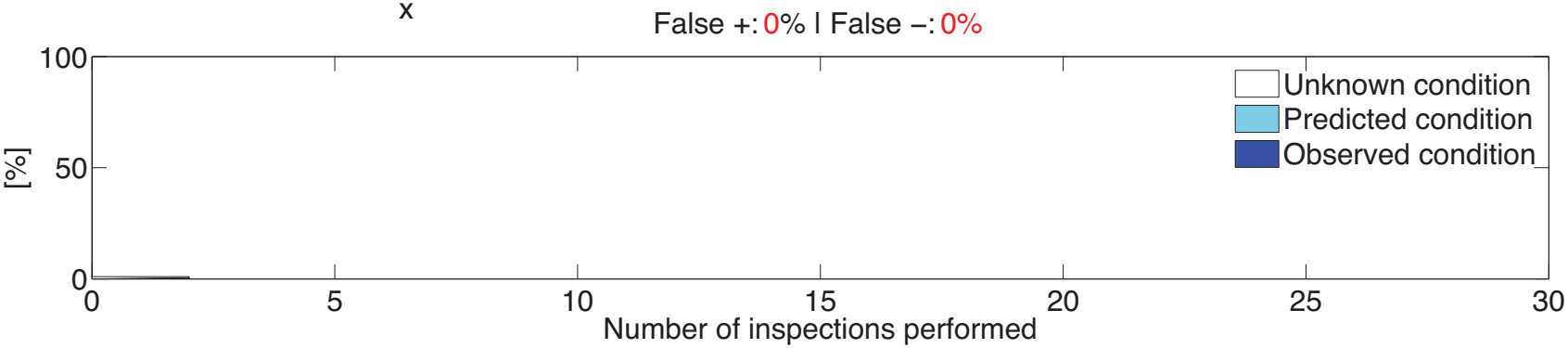
False +: 0% | False −: 0%

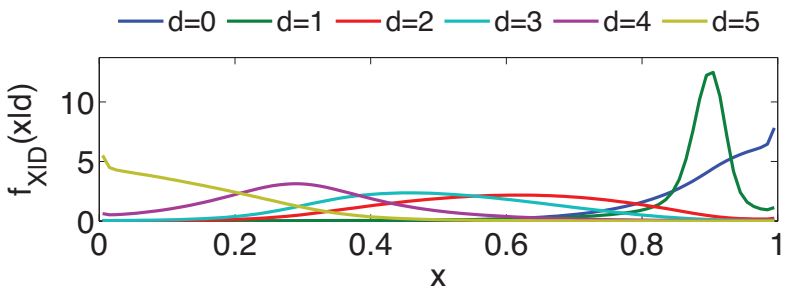


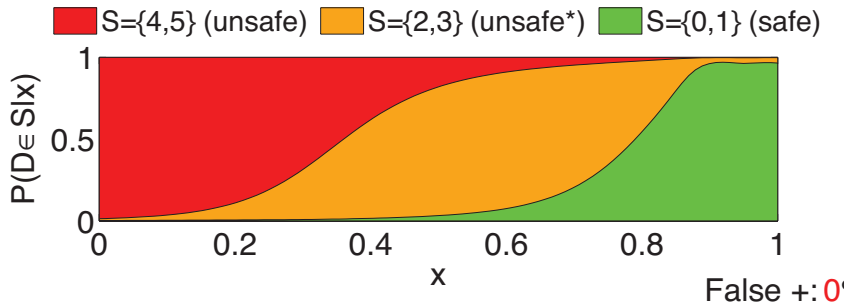

d=0
d=1
d=2
d=3
d=4
d=5
$f_{X|D}(x|d)$
10
5
0
0
0.2
0.4
0.6
0.8
1
x

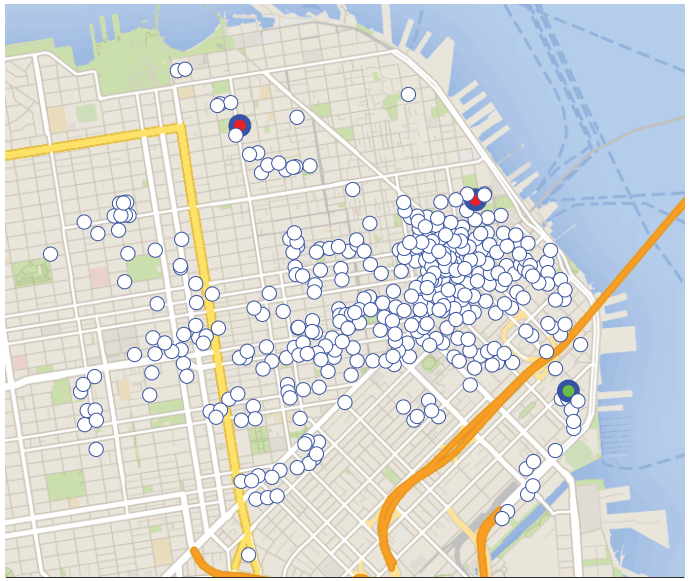

S={4,5} (unsafe)
S={2,3} (unsafe*)
S={0,1} (safe)
P(D∈S|x)
1
0.5
0
0
0.2
0.4
0.6
0.8
1
x



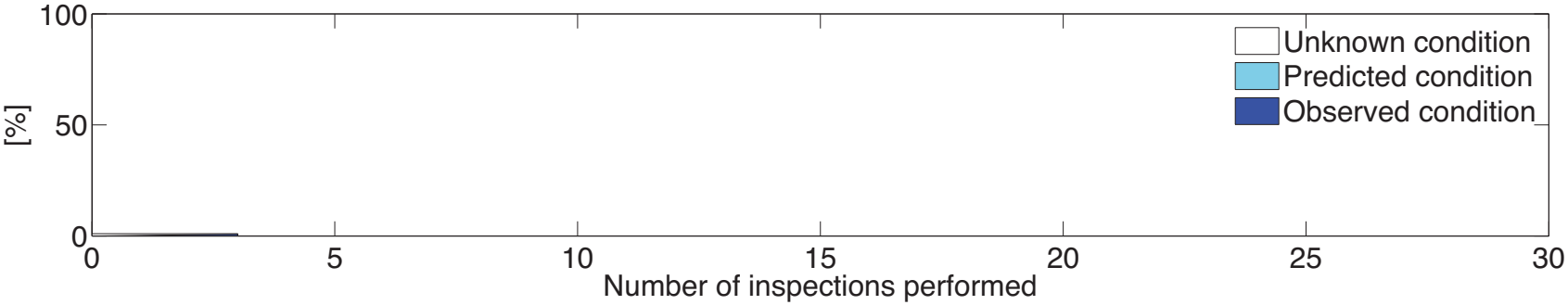

False +: 0% | False −: 0%
100
50
0
[%]
0
5
10
15
20
25
30
Number of inspections performed
Unknown condition
Predicted condition
Observed condition

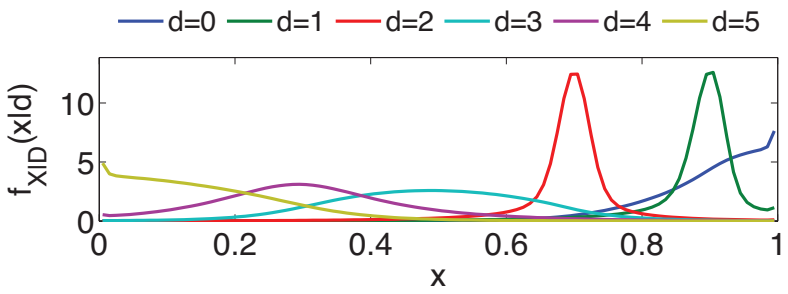


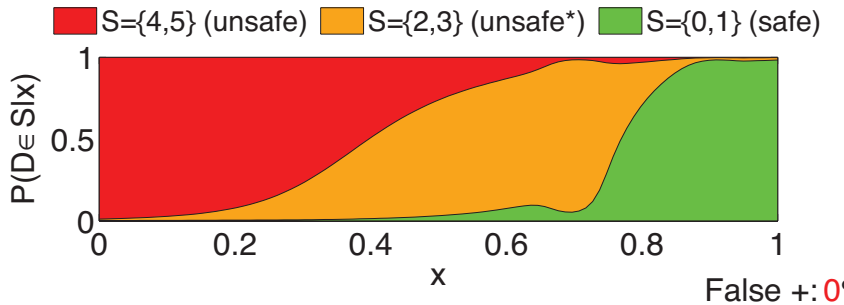


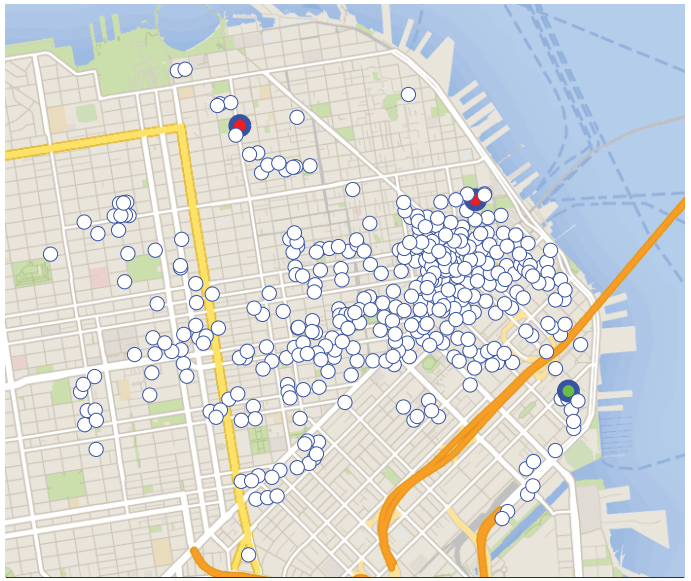


False +: 0% | False −: 0%

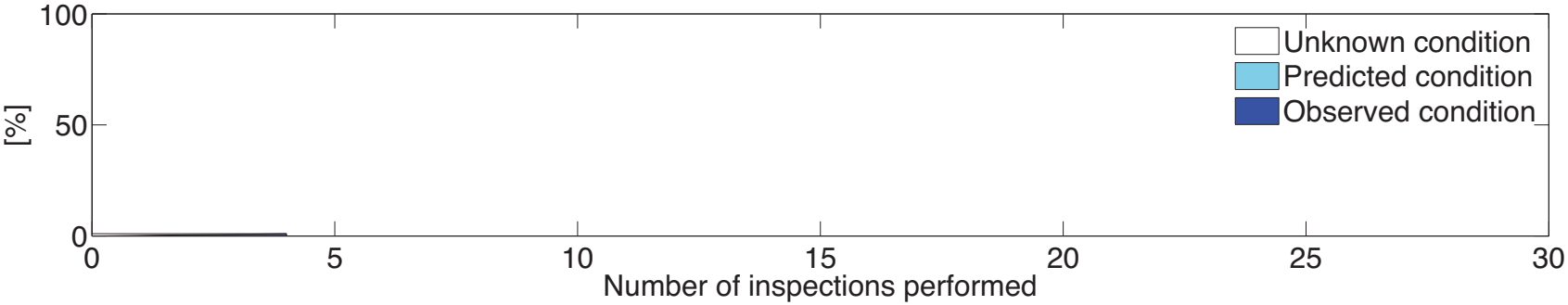


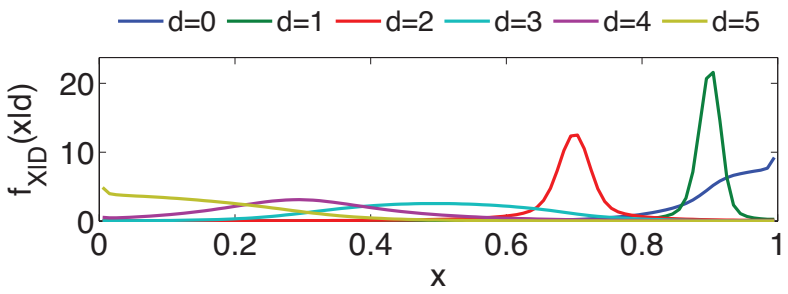


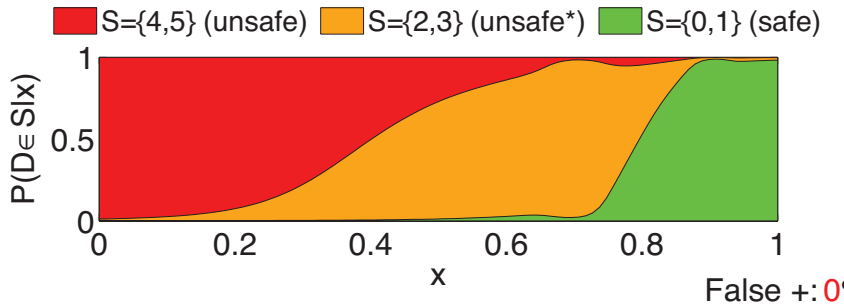


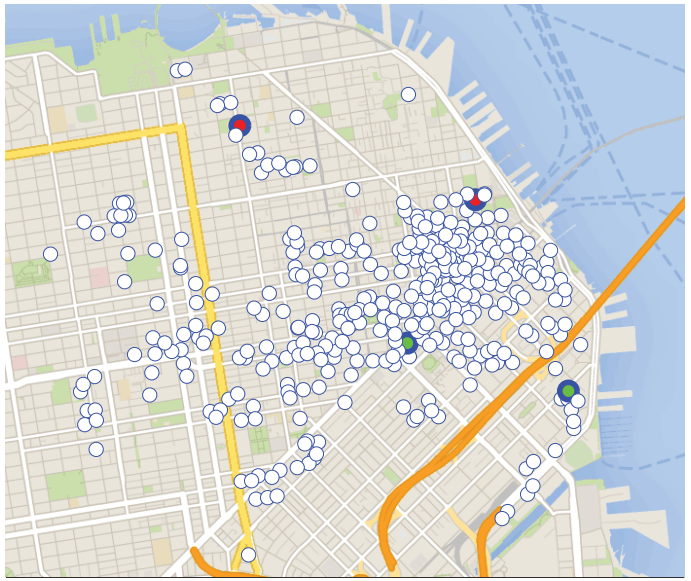


False +: 0% | False −: 0%

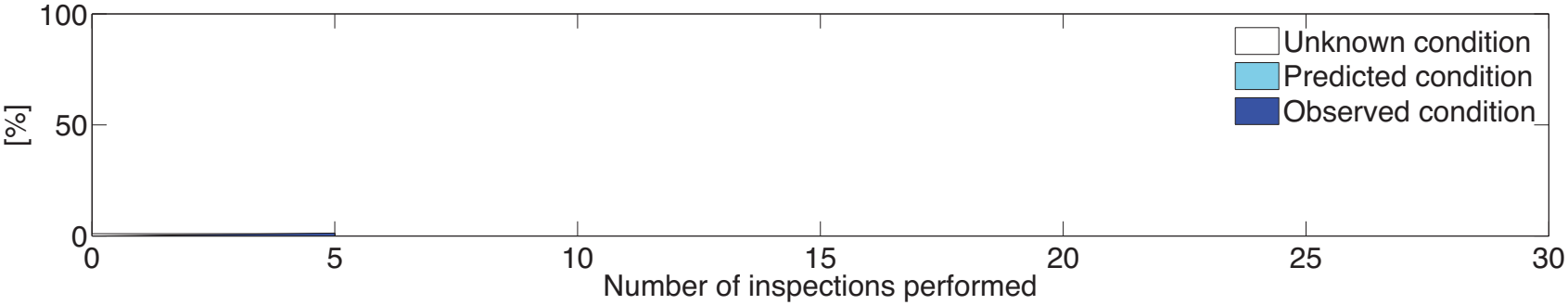


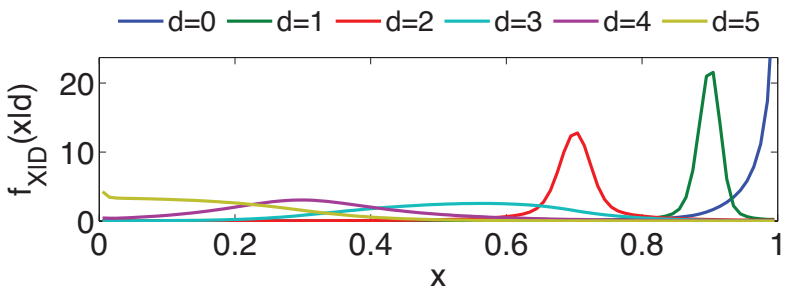


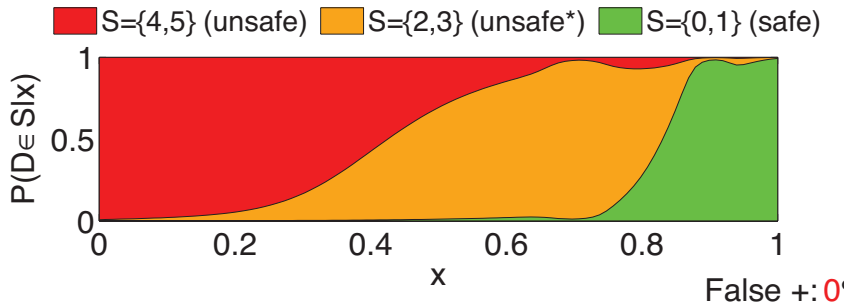


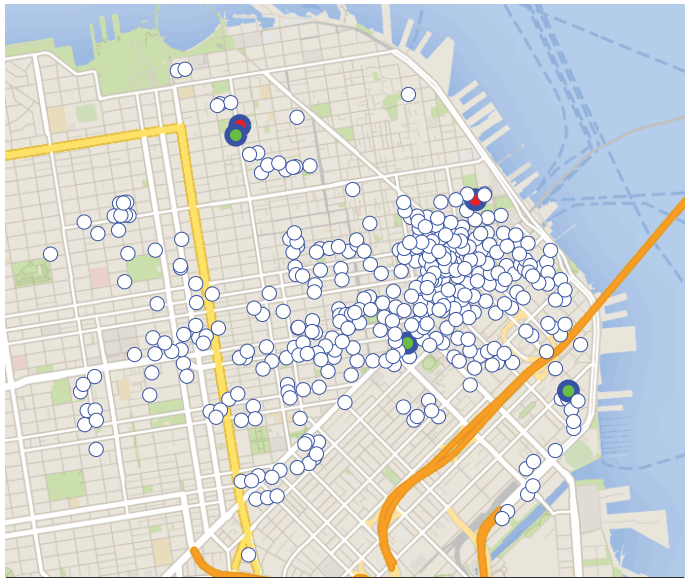


False +: 0% | False −: 0%

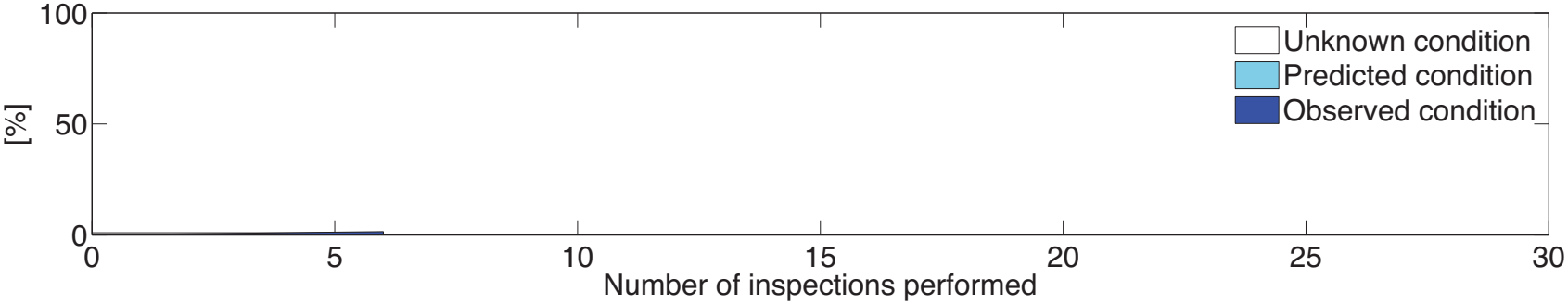


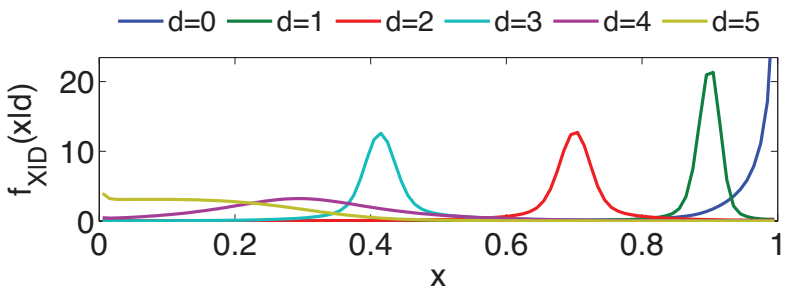


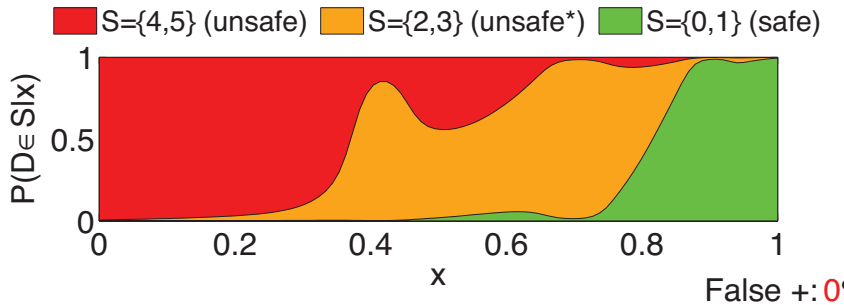


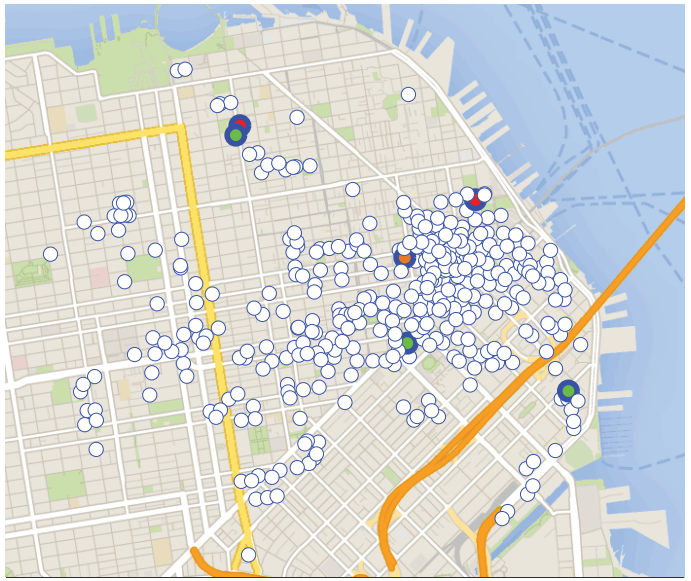


False +: 0% | False −: 0%

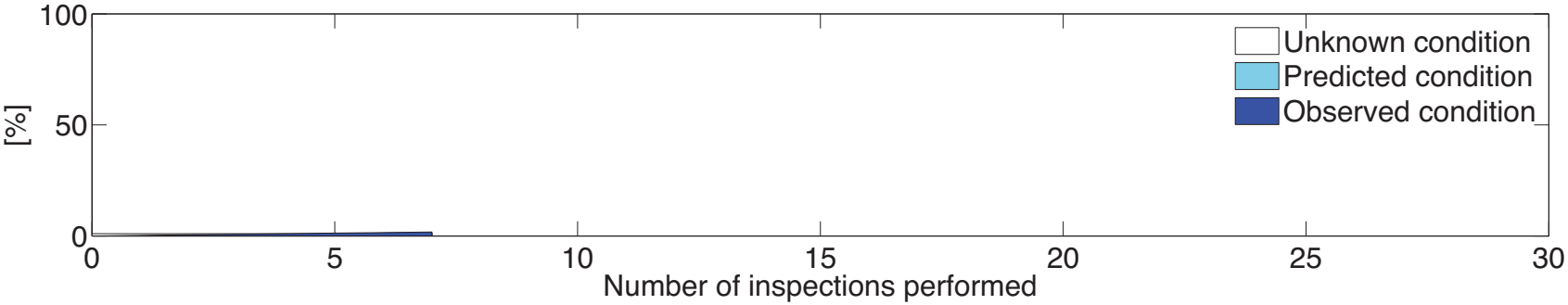


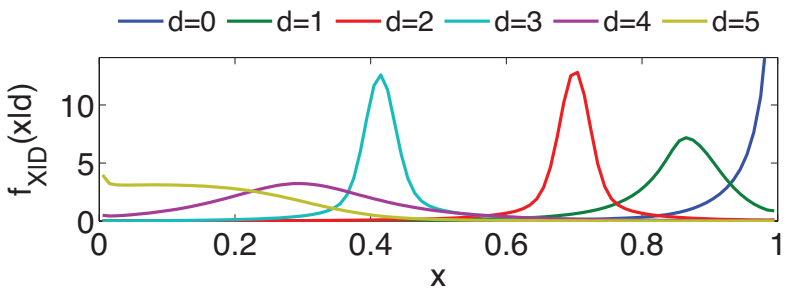


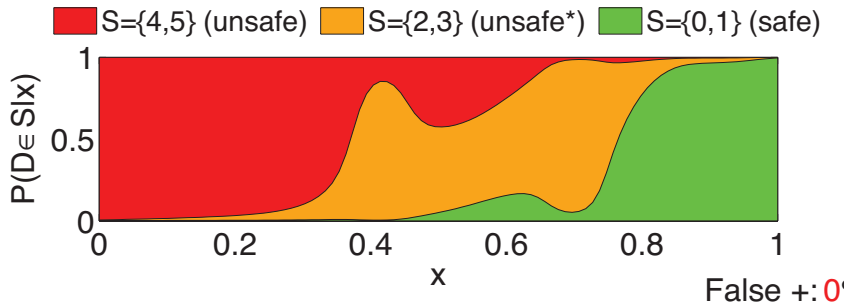


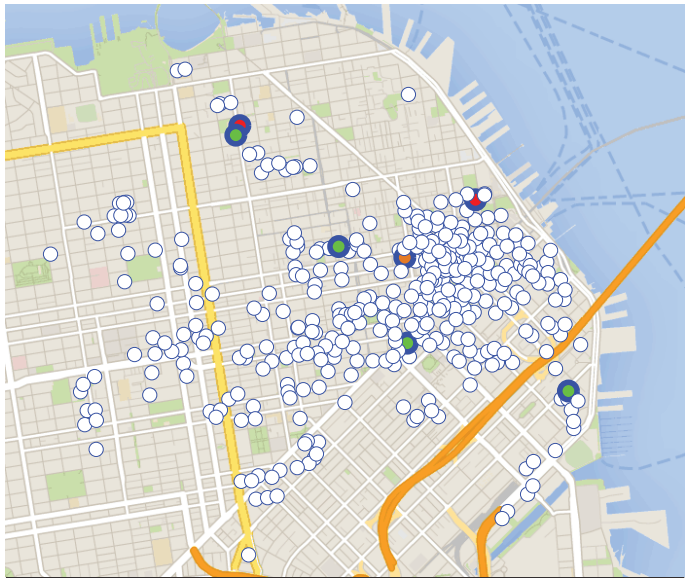


False +: 0% | False −: 0%

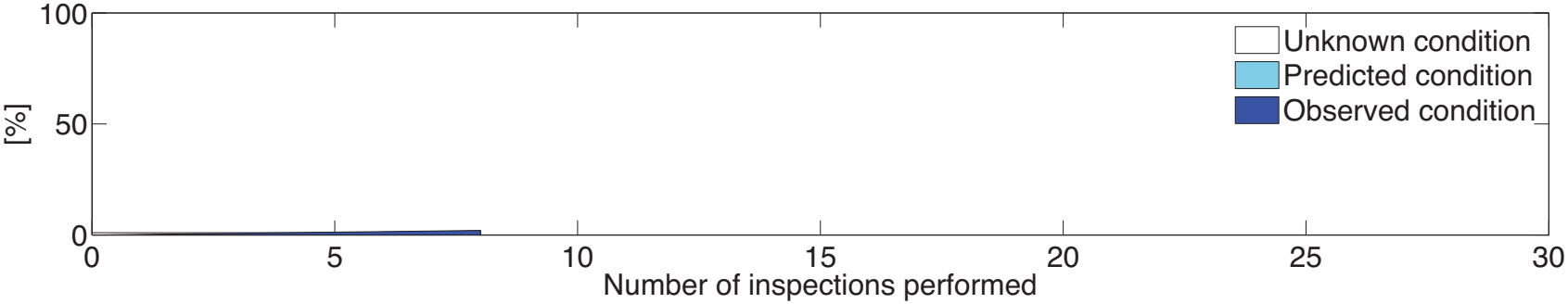


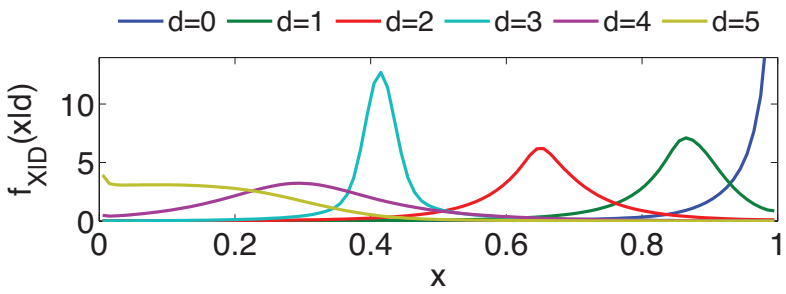


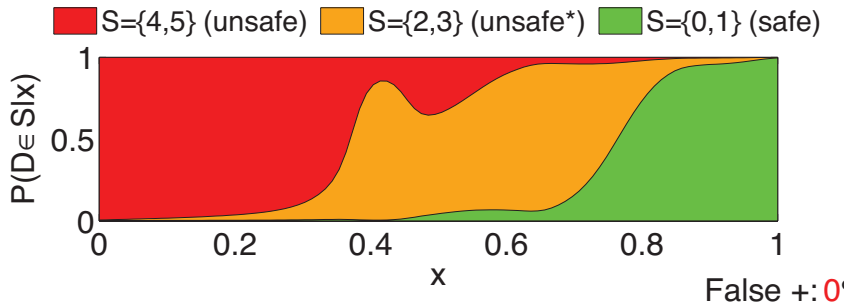


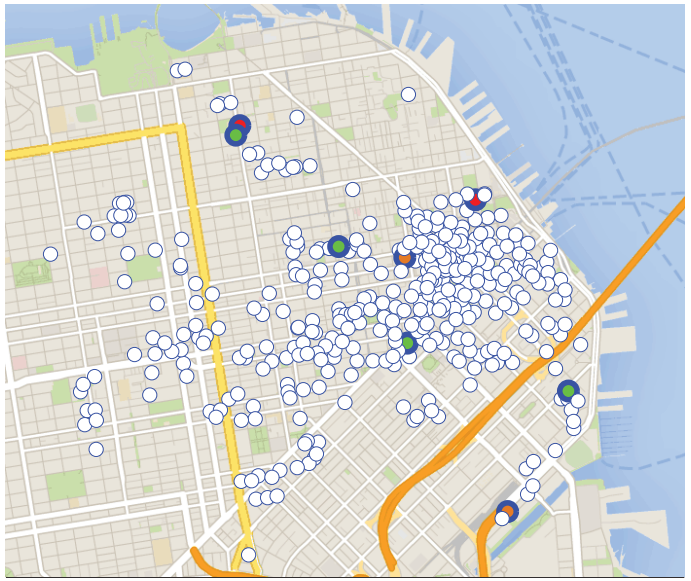


False +: 0% | False −: 0%

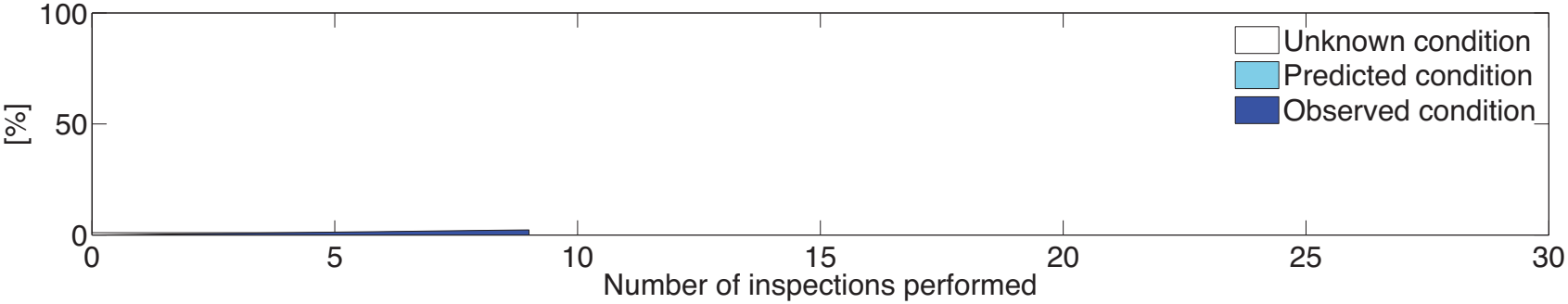


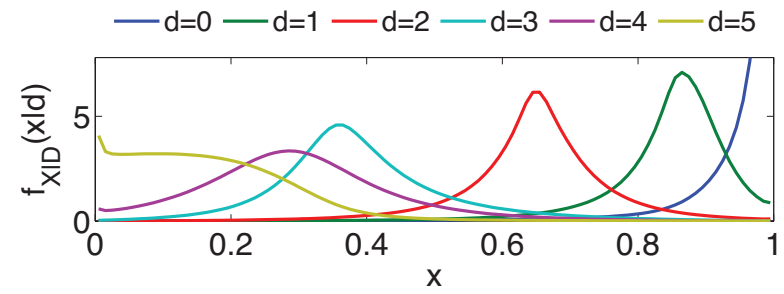


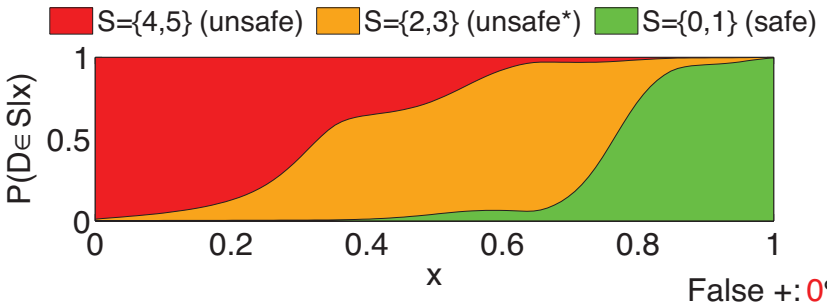


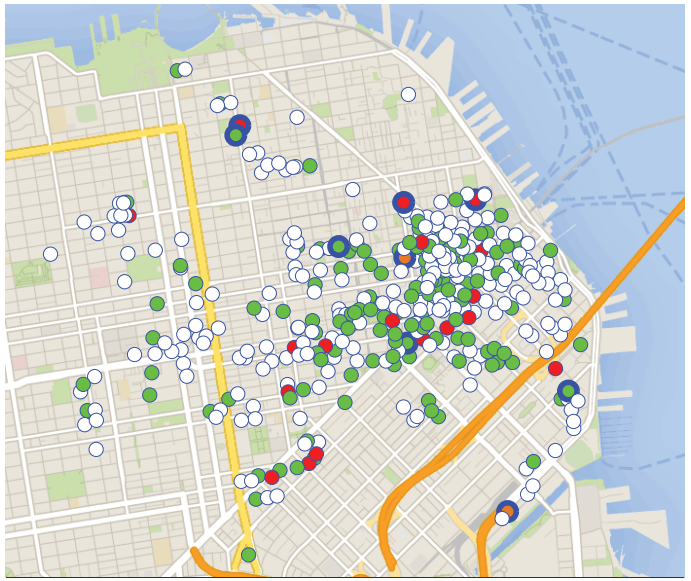


False +: 0% | False −: 0%

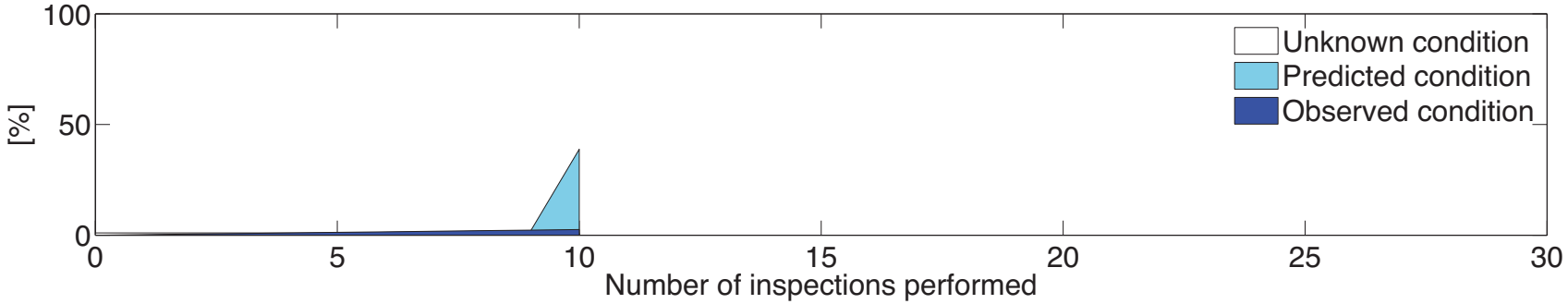


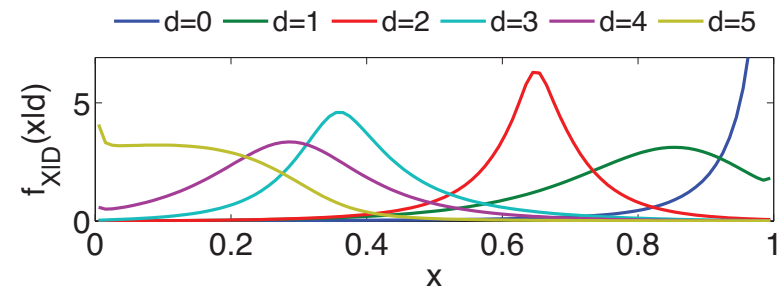


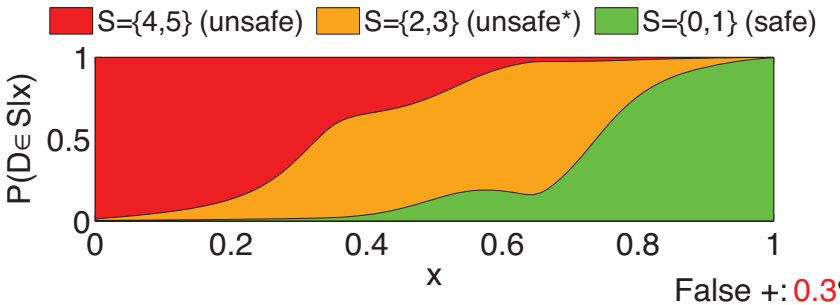


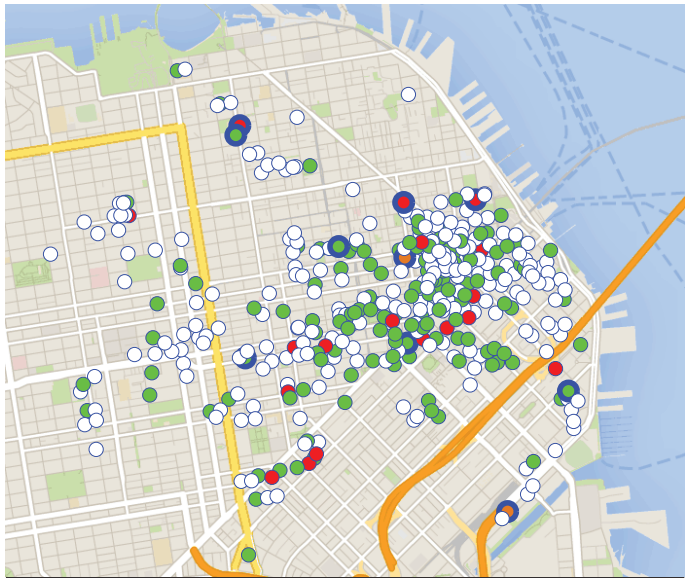


False +: 0% | False −: 3.9%

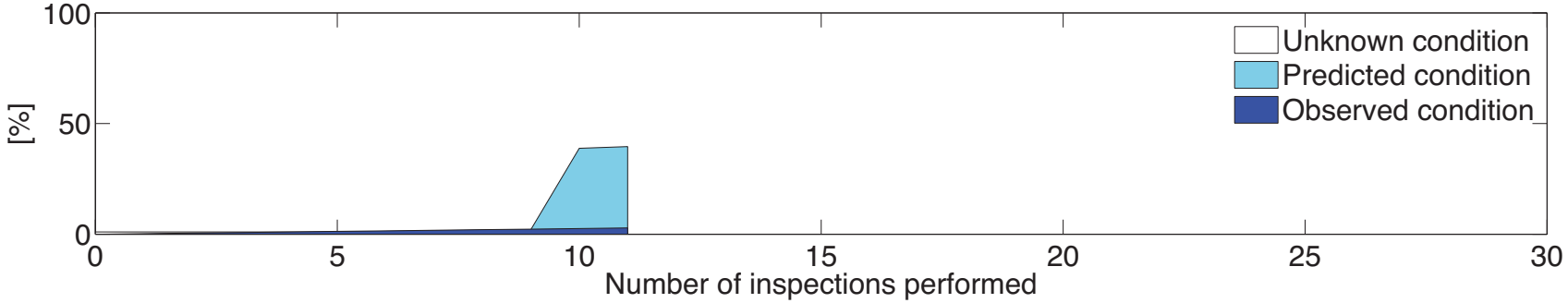


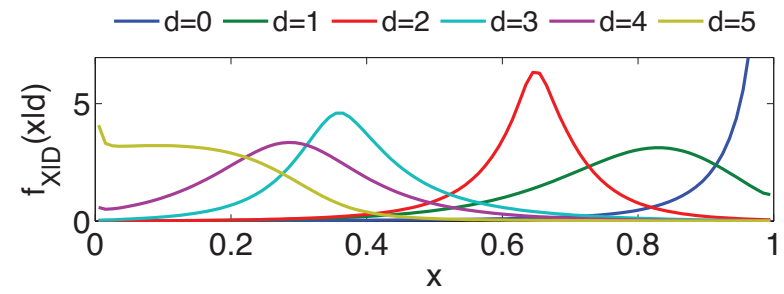


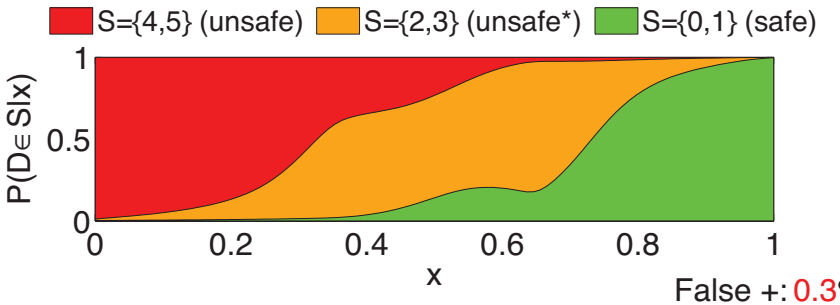


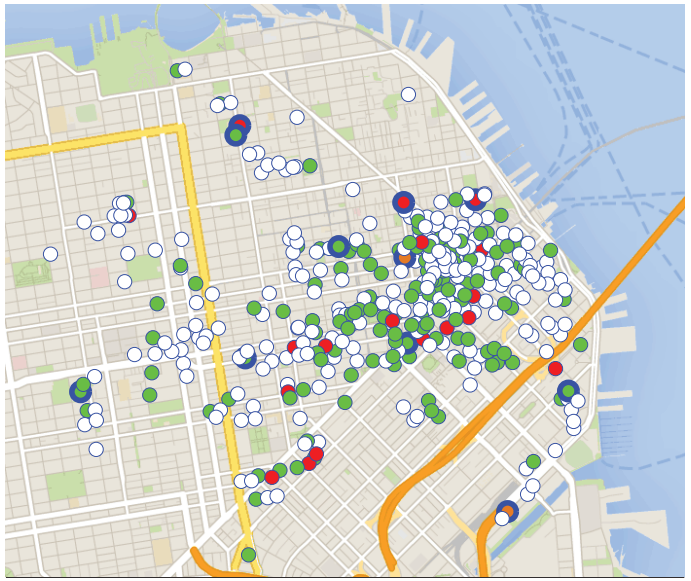


False +: 0.3% | False −: 3.9%

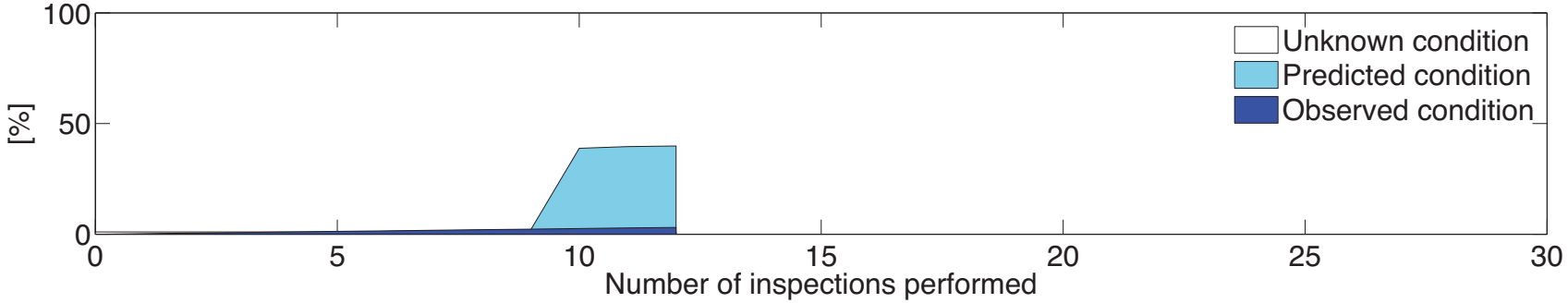


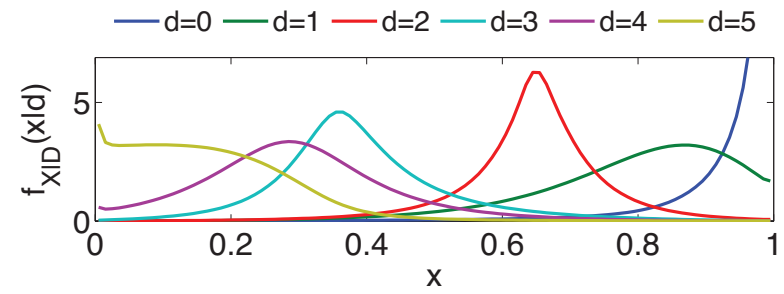


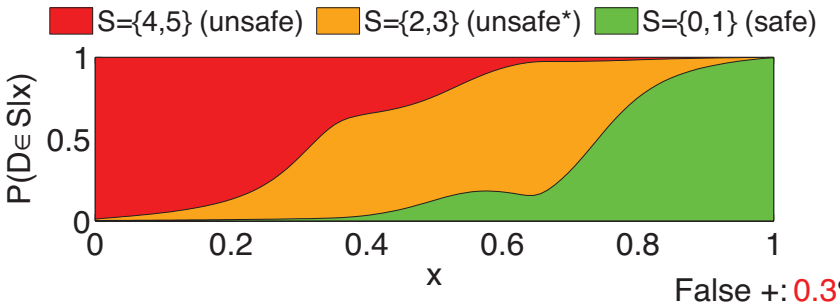


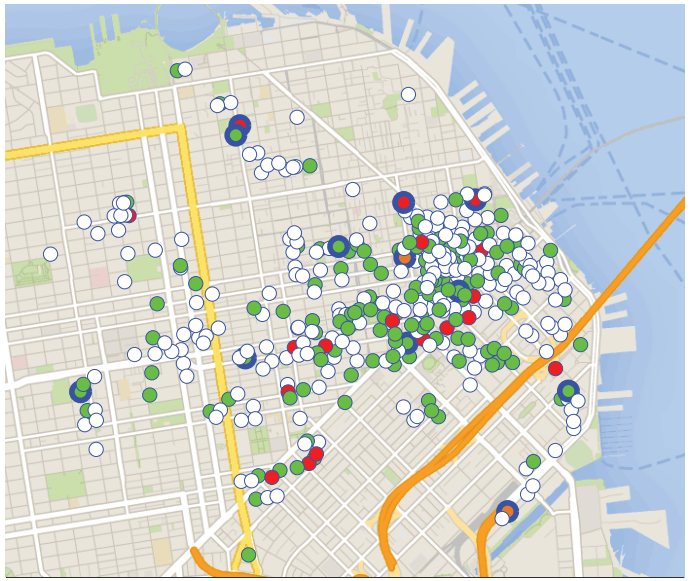


False +: 0.3% | False −: 3.9%

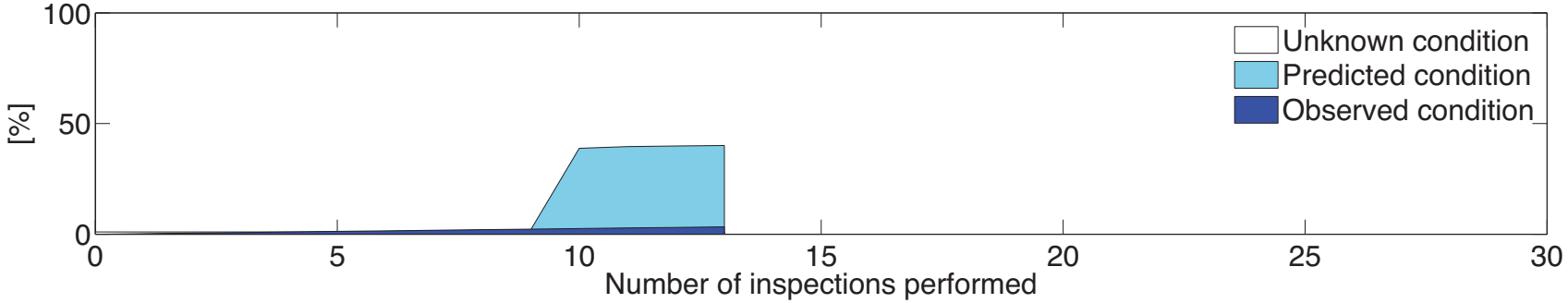


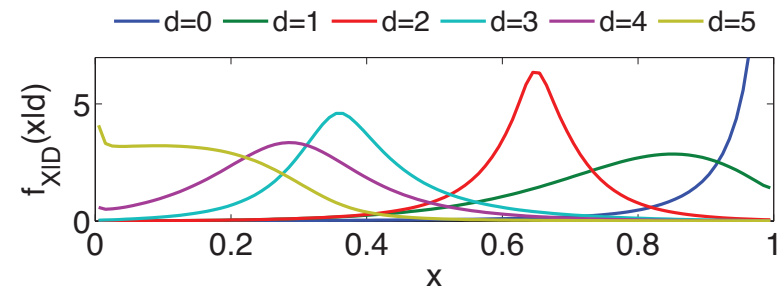


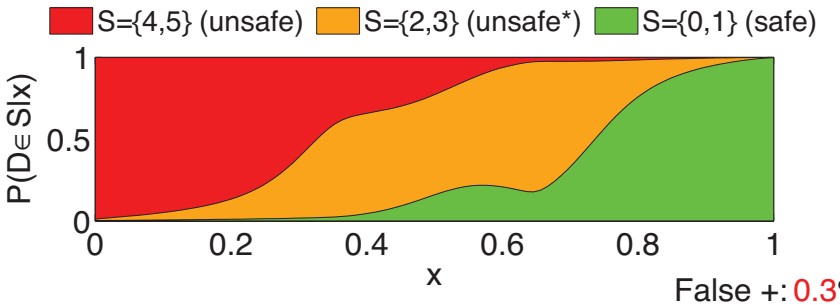


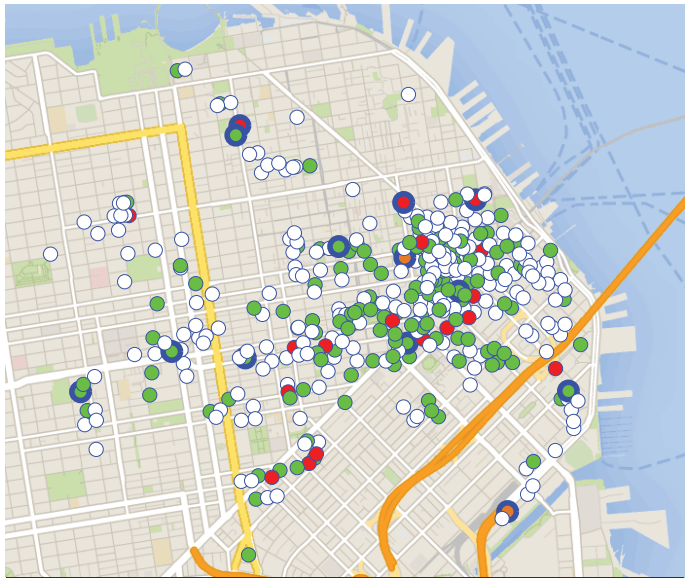


False +: 0.3% | False −: 3.9%

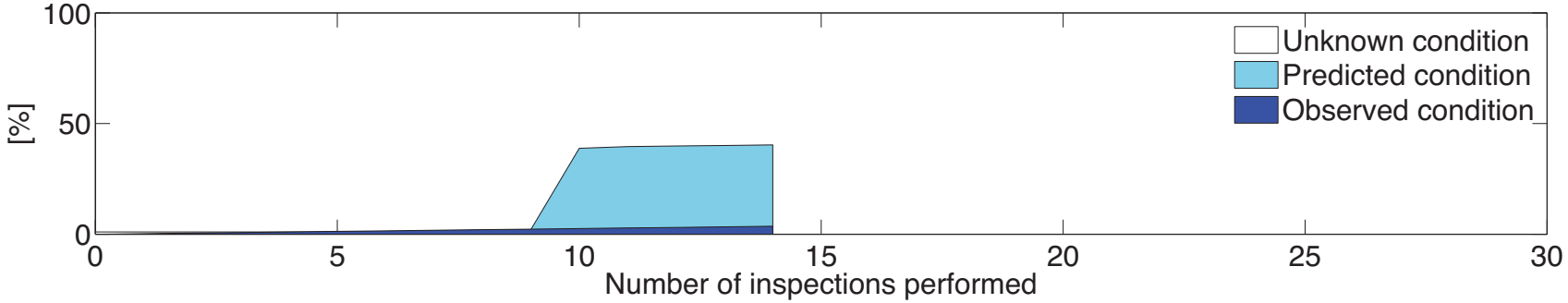


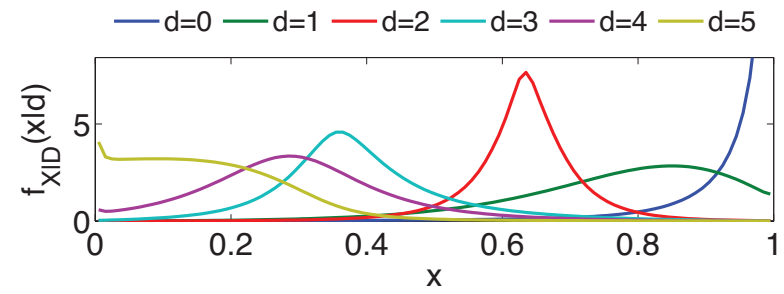


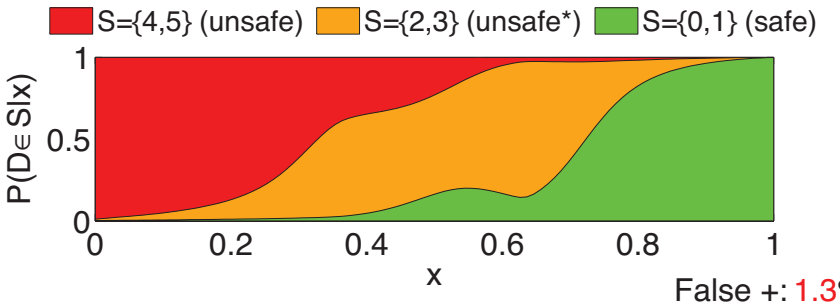


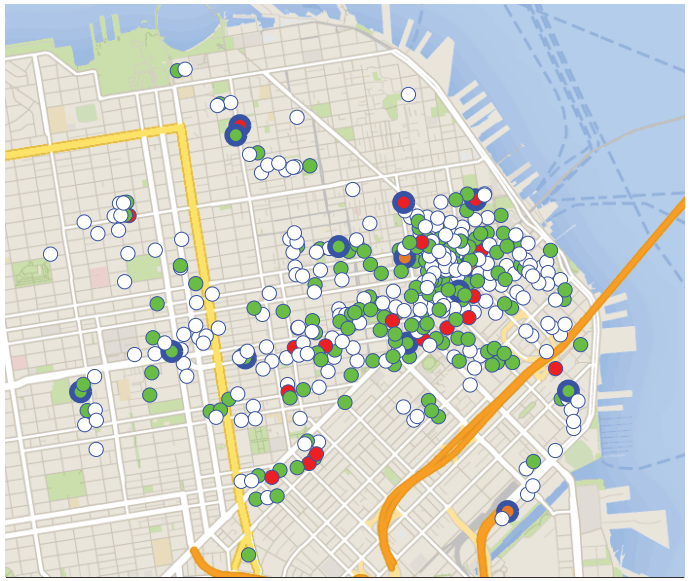


False +: 0.3% | False −: 3.9%

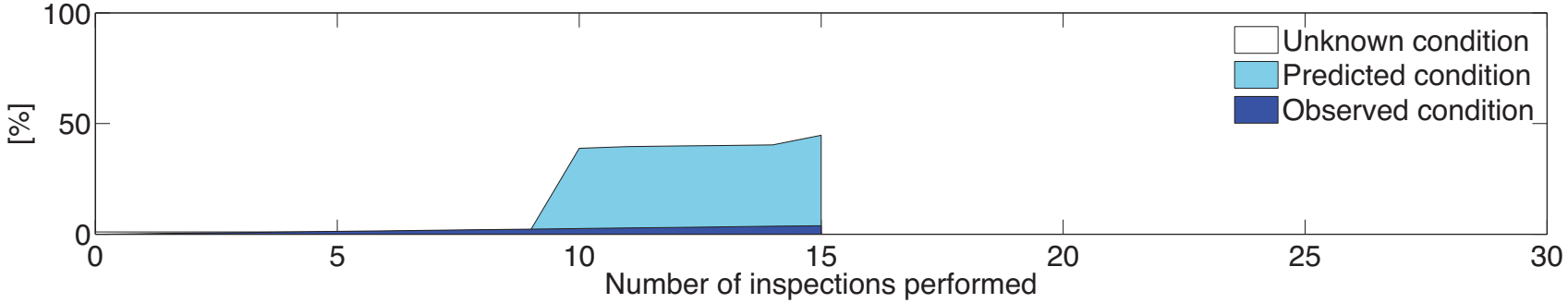


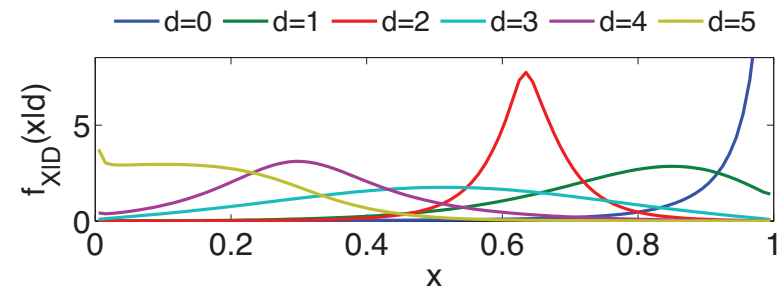


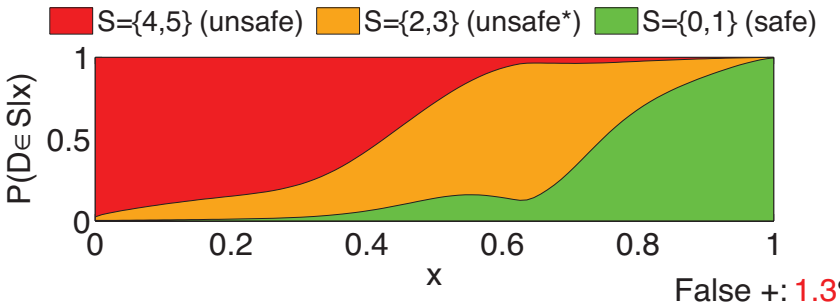


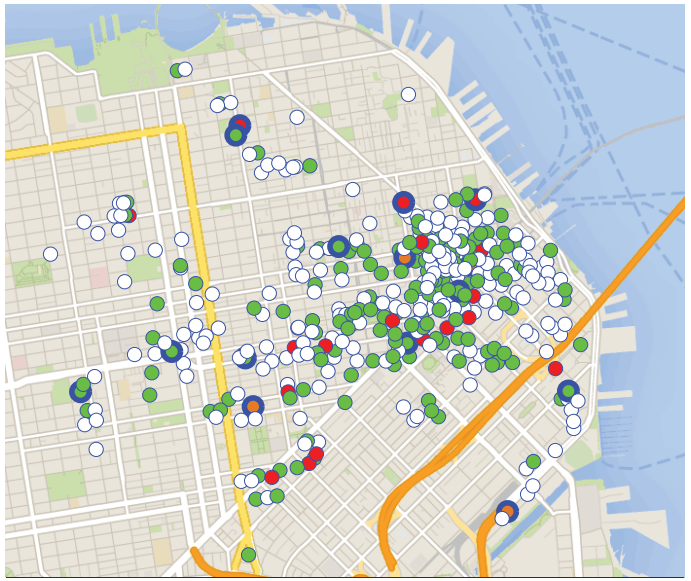


False +: 1.3% | False −: 3.9%

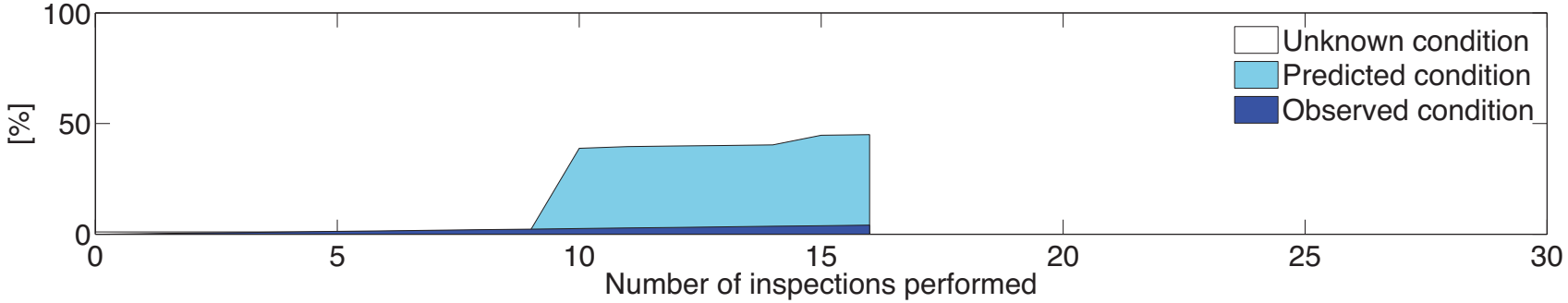


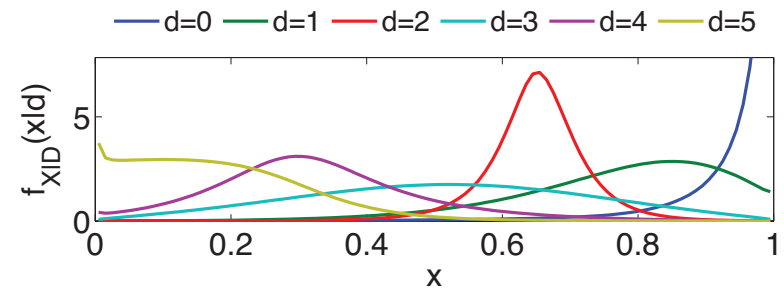


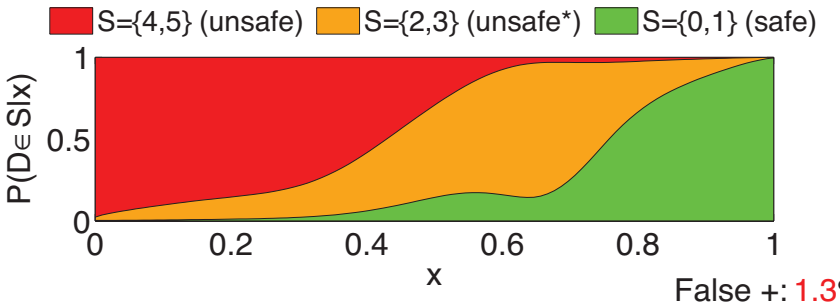


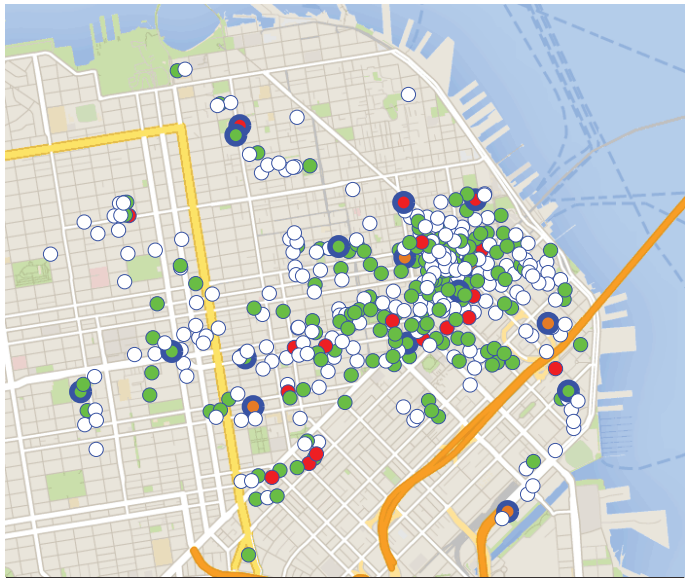


False +: 1.3% | False −: 3.9%

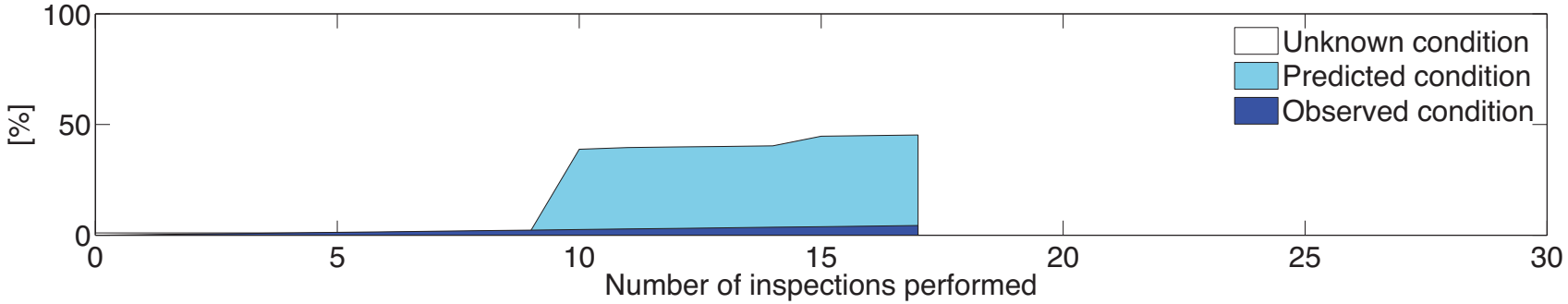


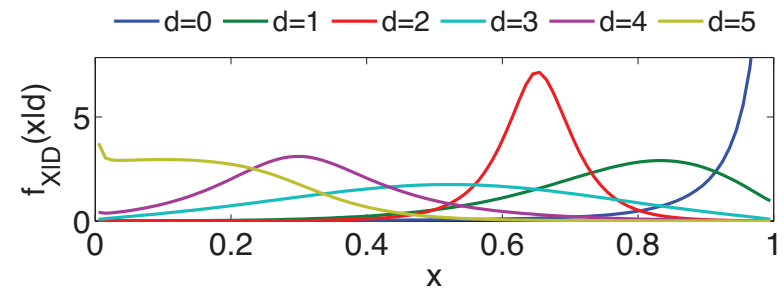


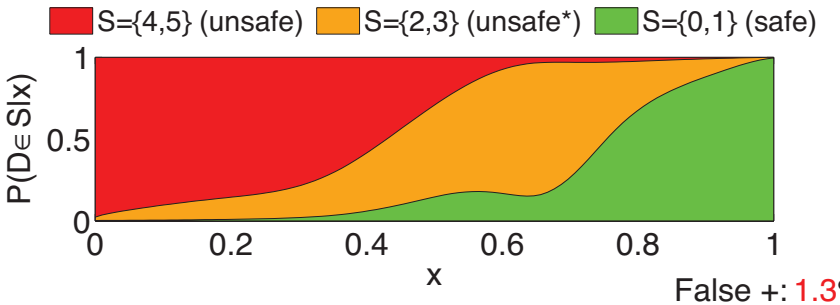


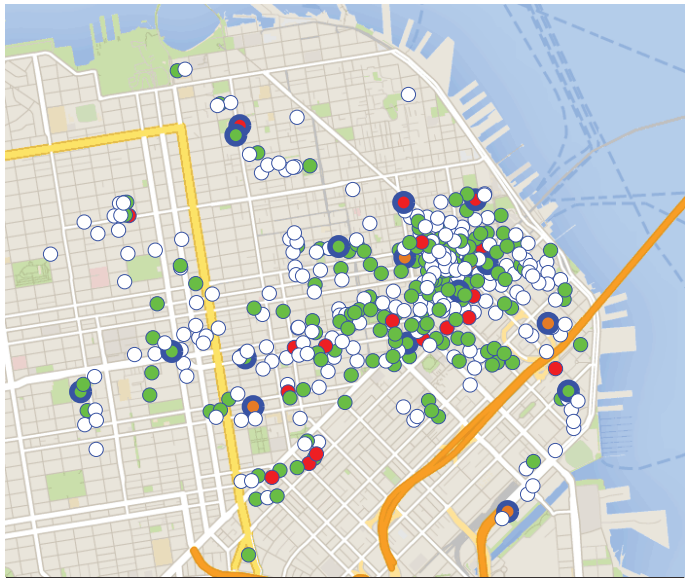


False +: 1.3% | False −: 3.9%

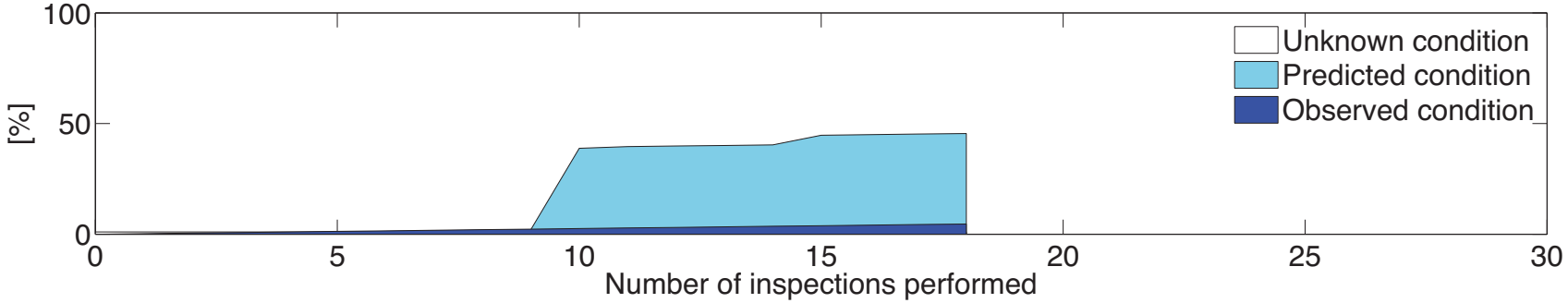


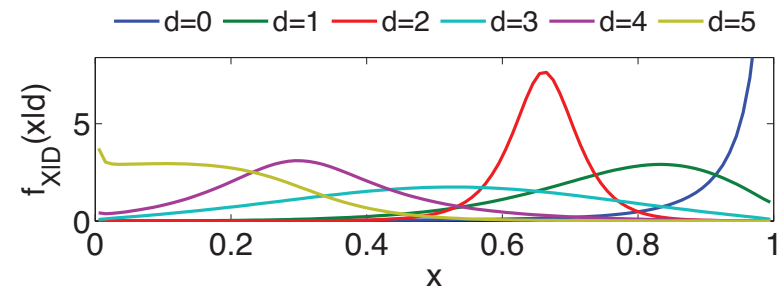


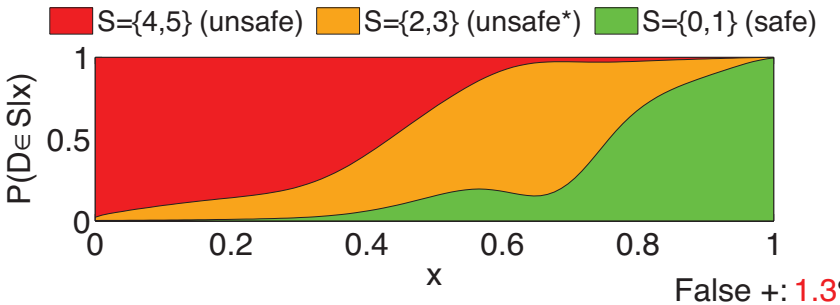


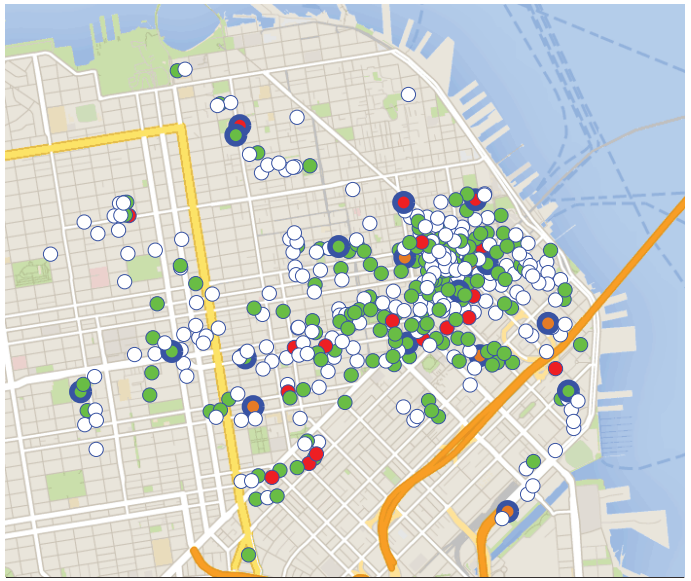


False +: 1.3% | False −: 3.9%

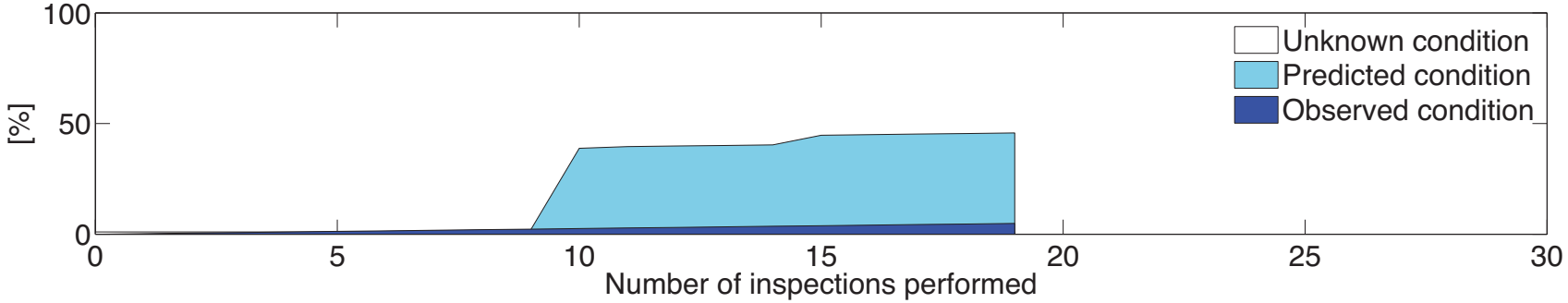


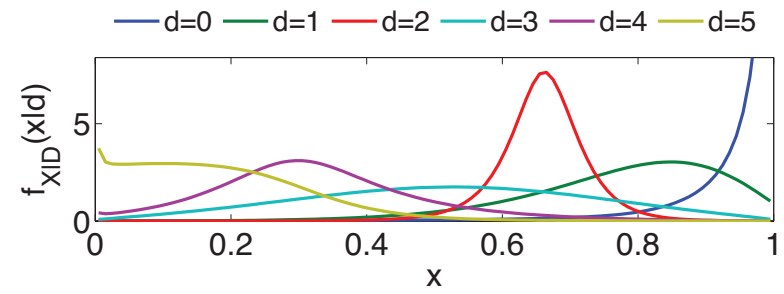


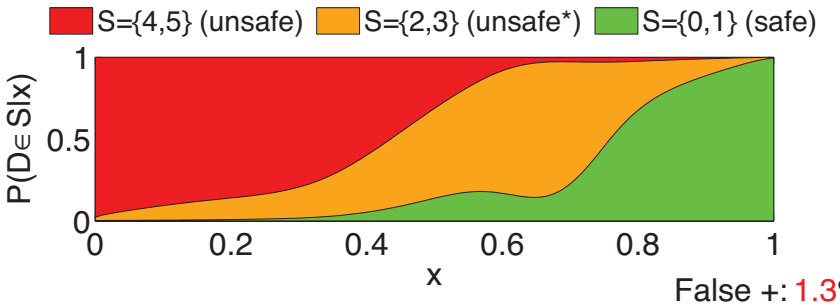


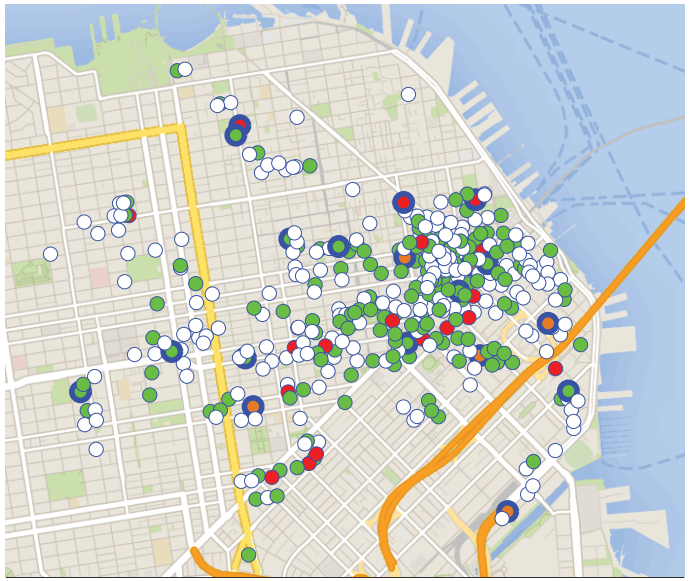


False +: 1.3% | False −: 3.9%

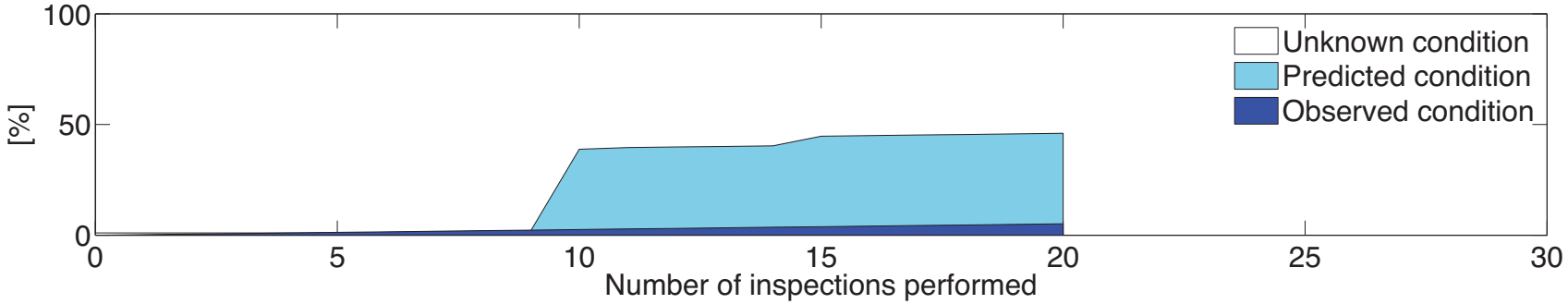


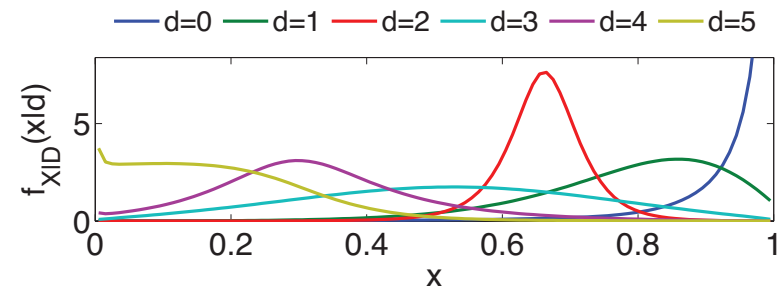


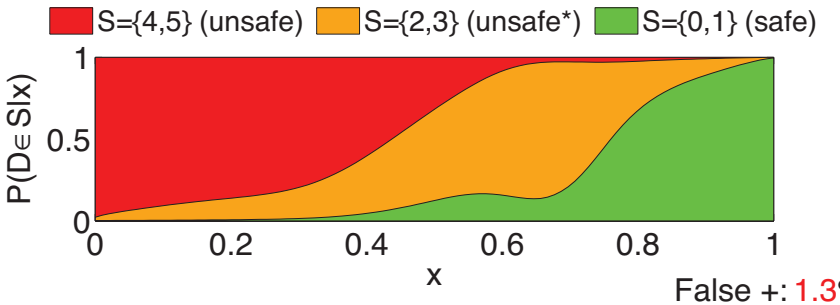


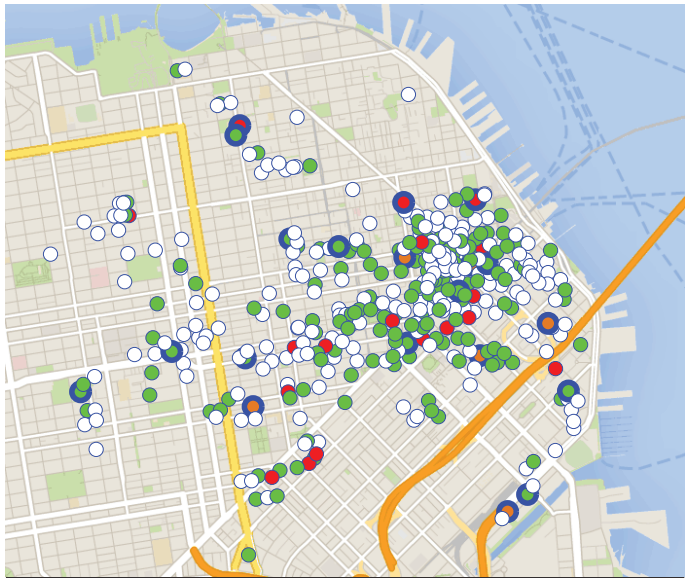


False +: 1.3% | False −: 3.9%

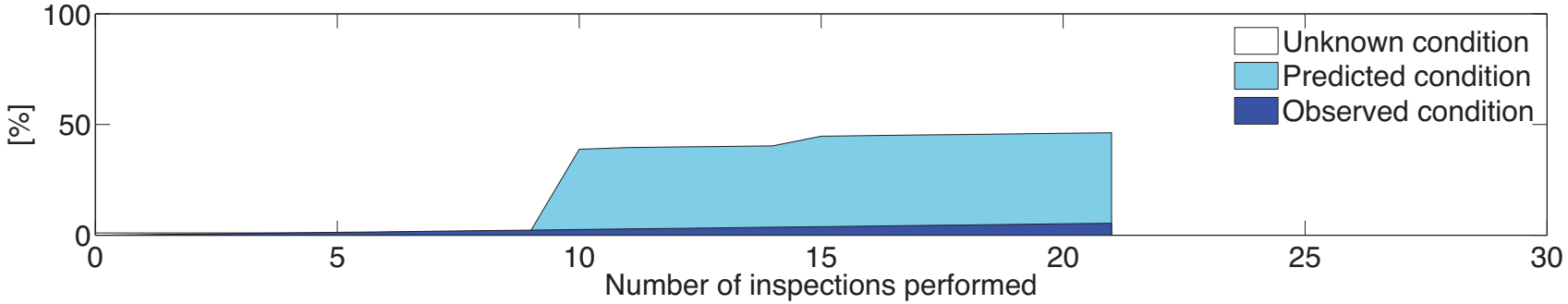


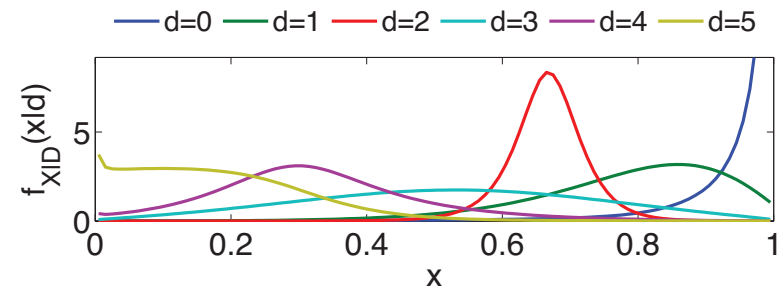


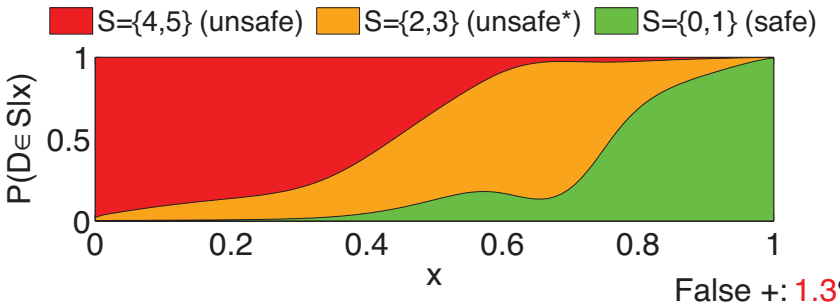


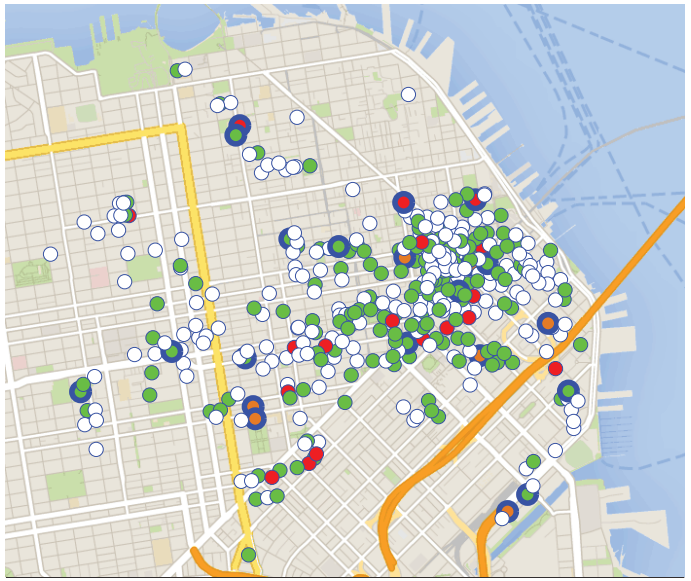


False +: 1.3% | False −: 3.9%

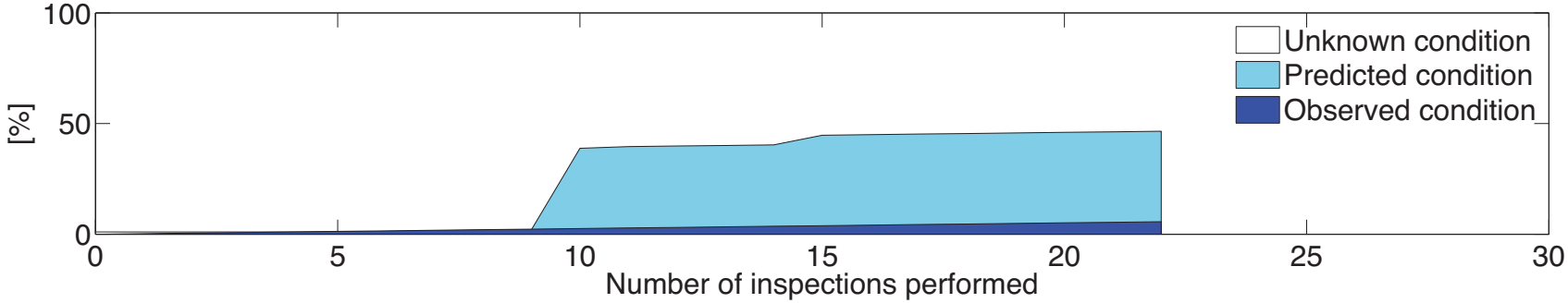


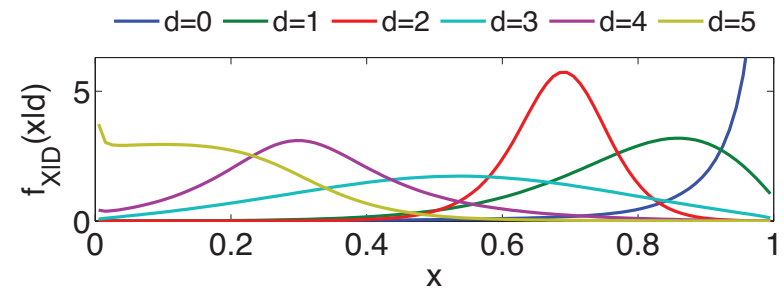


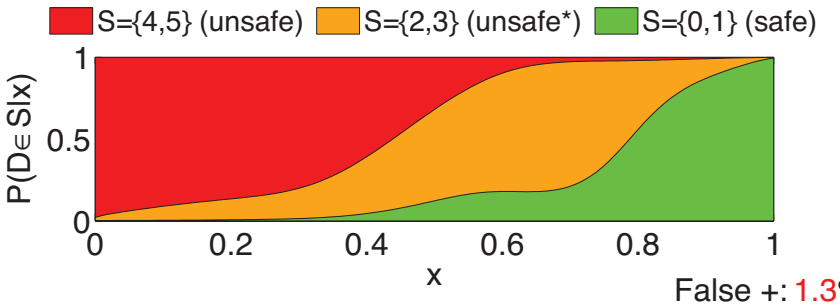


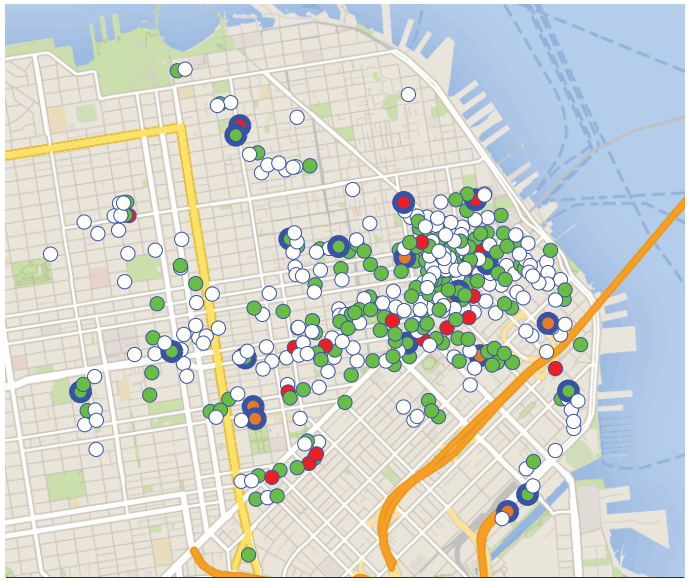


False +: 1.3% | False −: 3.9%

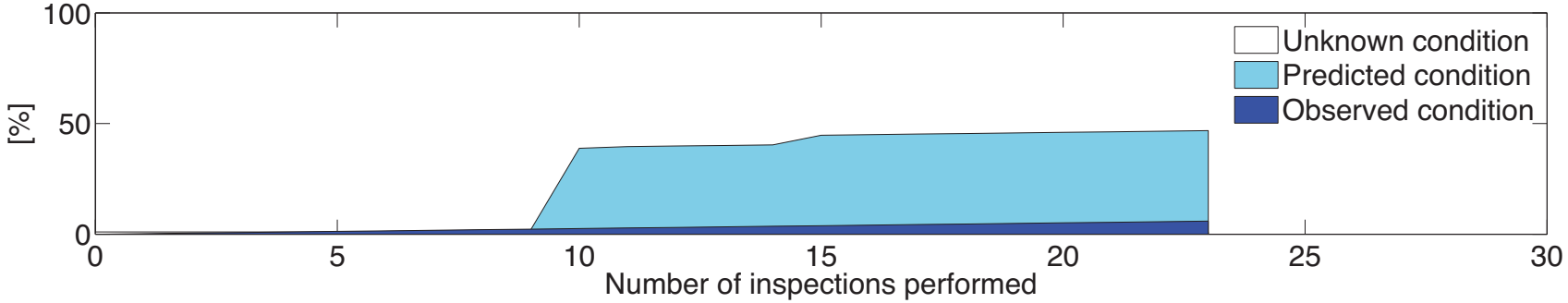


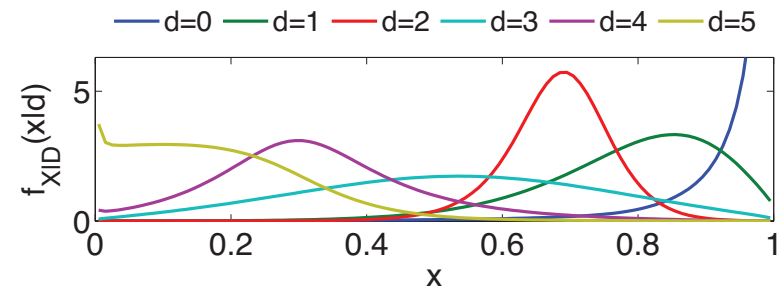


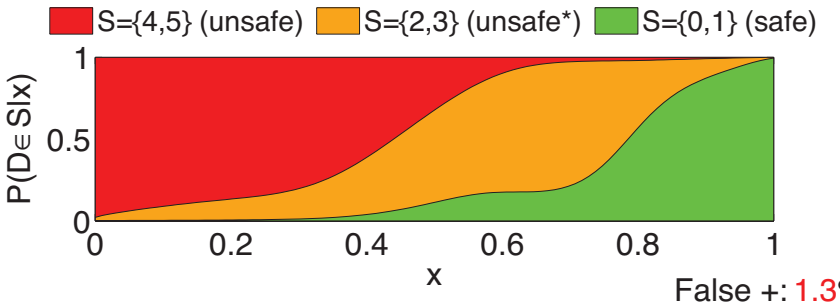


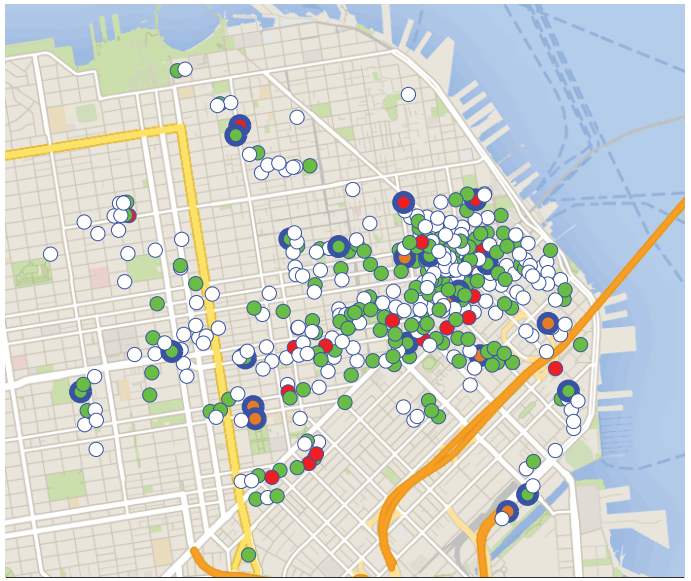


False +: 1.3% | False −: 3.9%

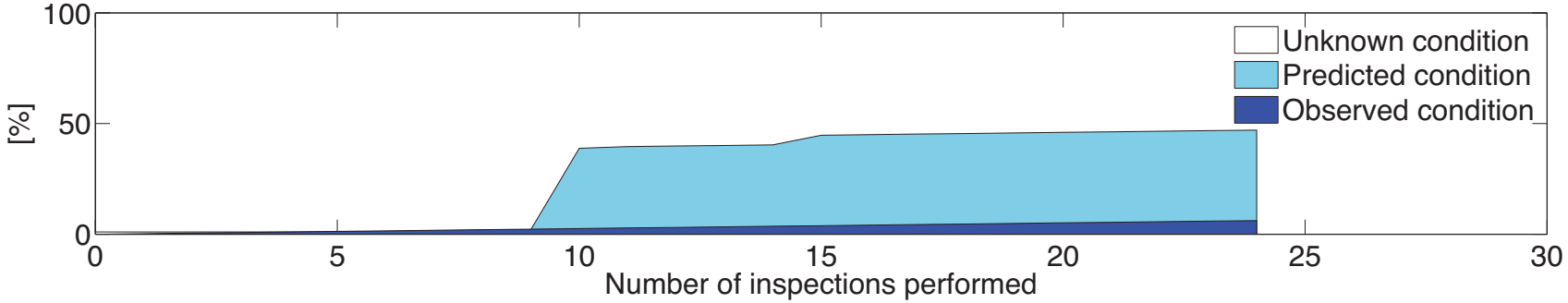


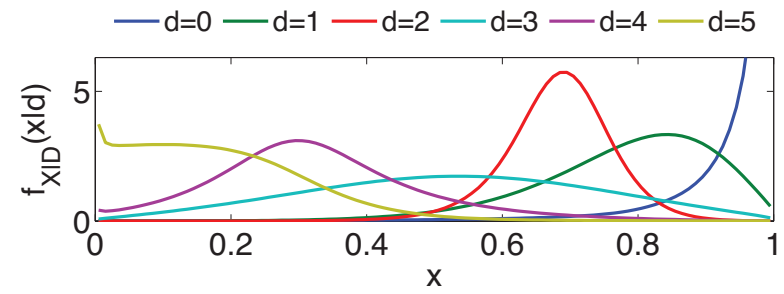


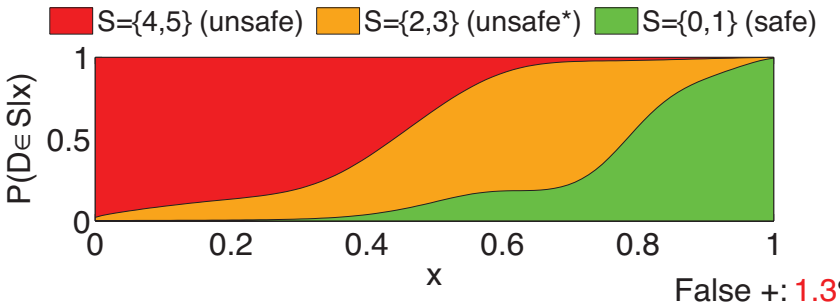


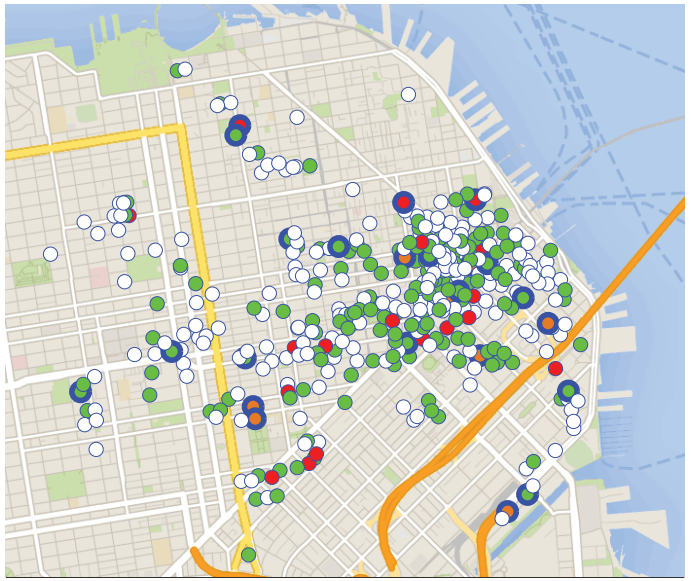


False +: 1.3% | False −: 3.9%

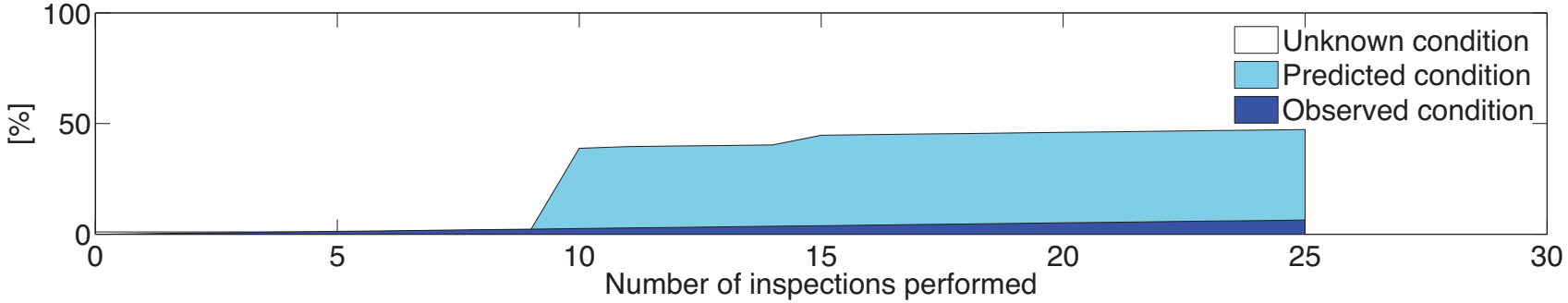


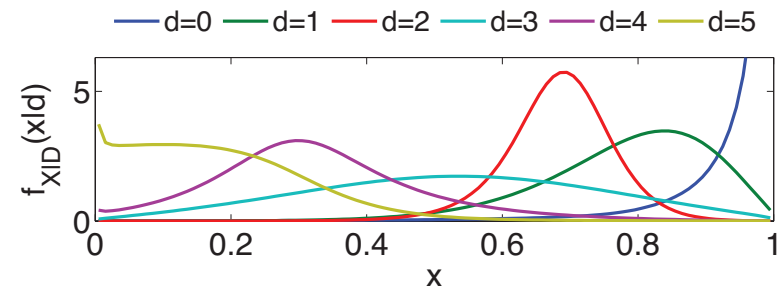


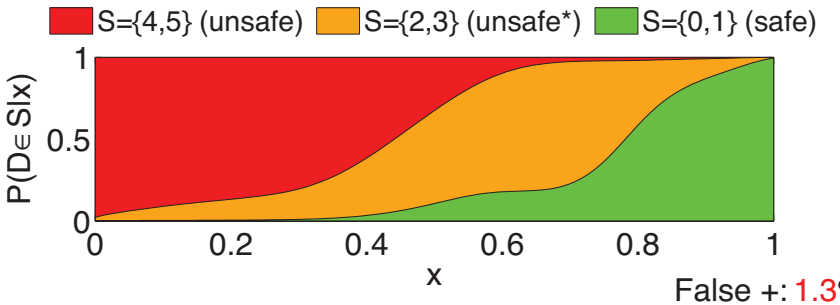


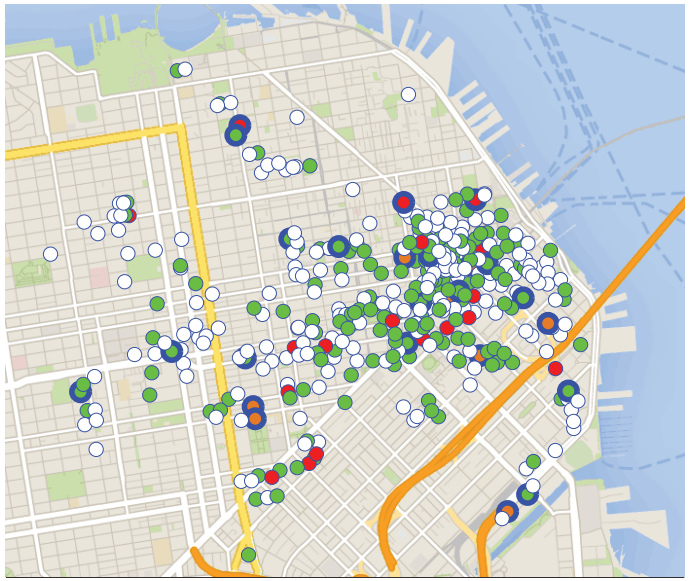


False +: 1.3% | False −: 3.9%

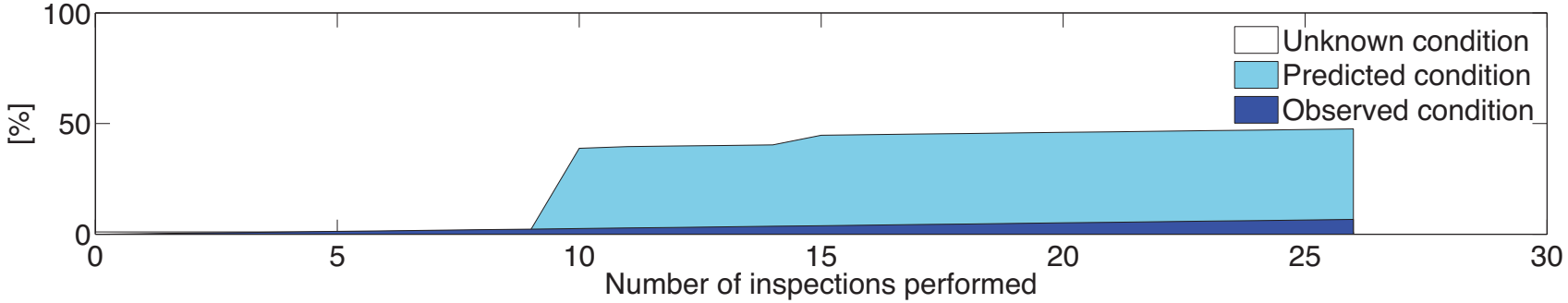


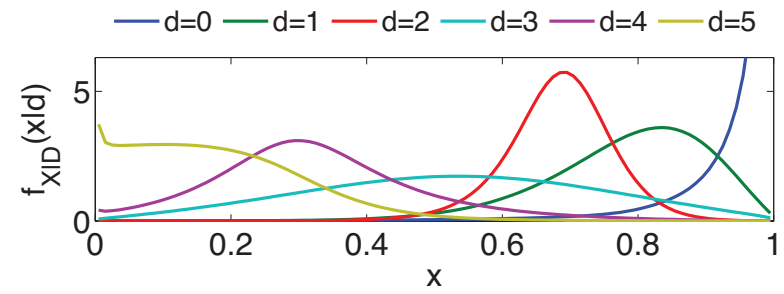


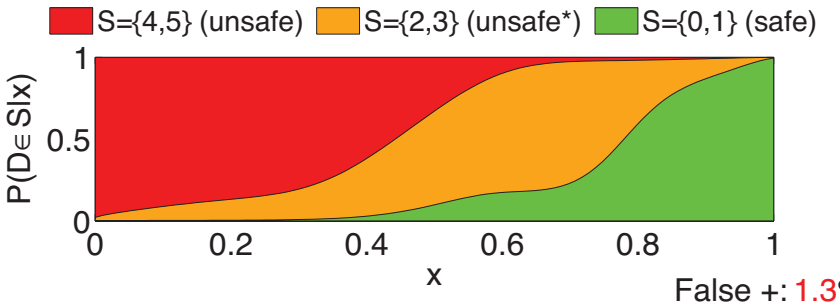


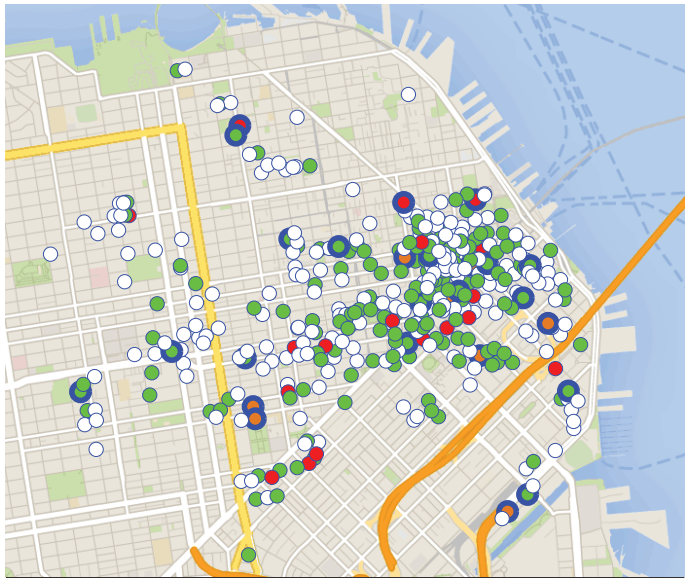


False +: 1.3% | False −: 3.9%

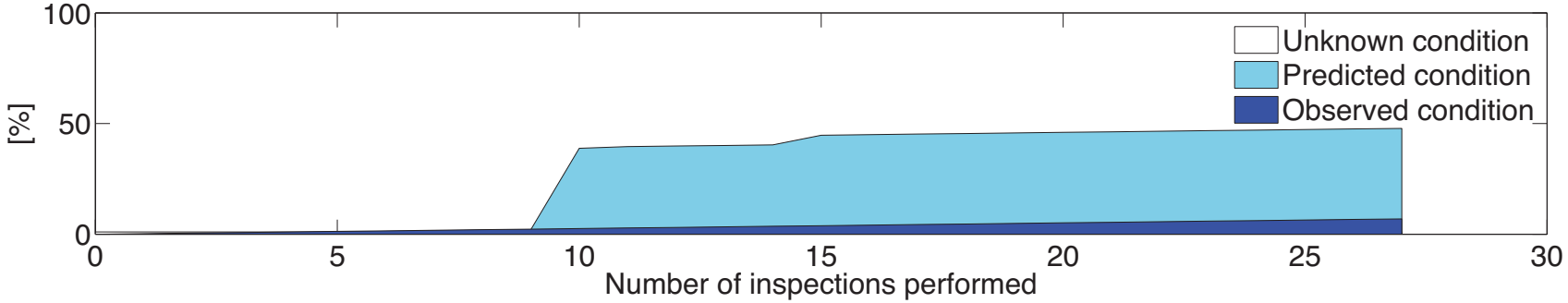


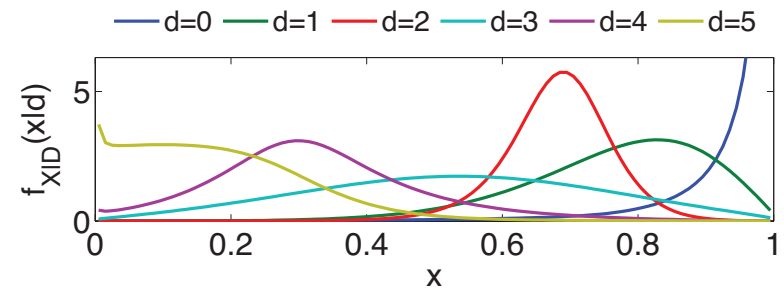


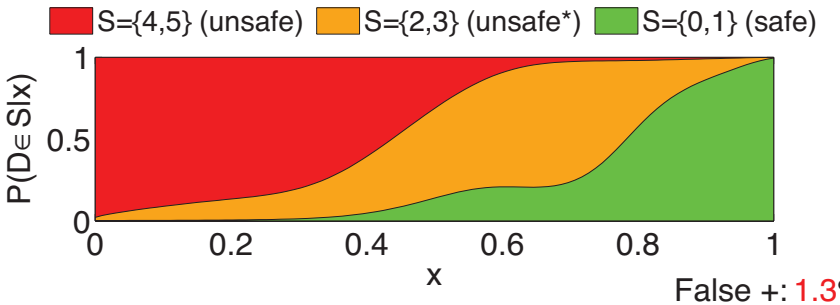


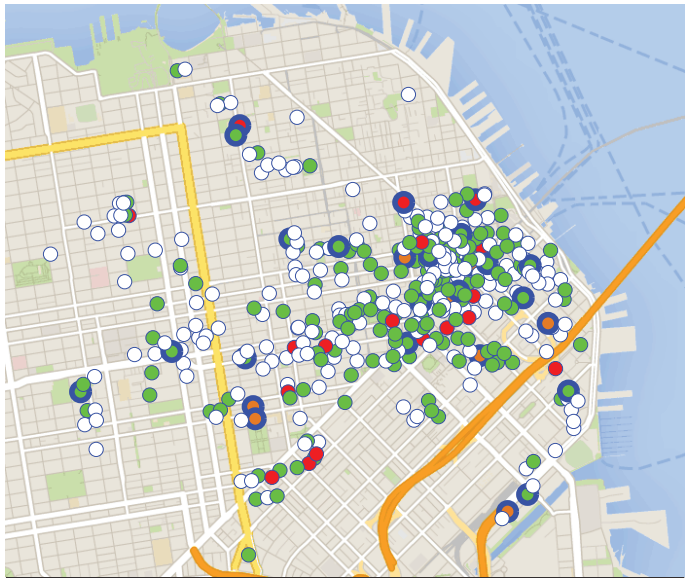


False +: 1.3% | False −: 3.9%

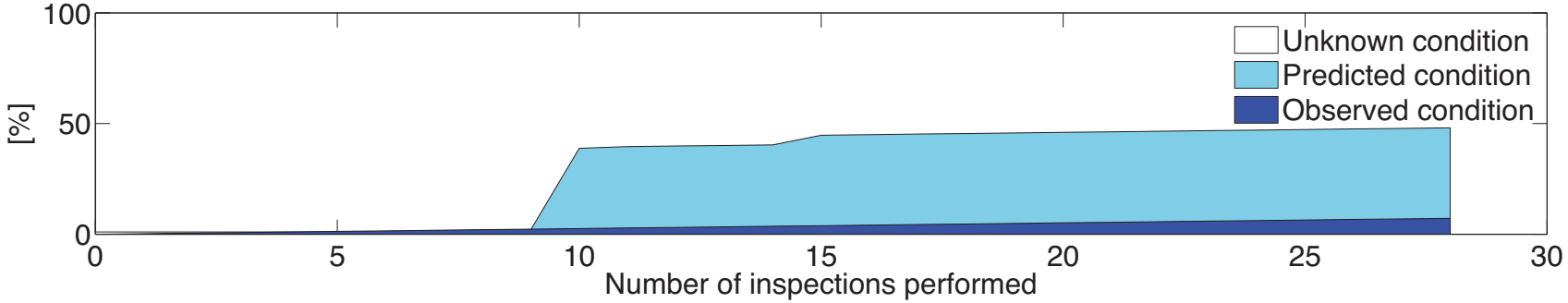


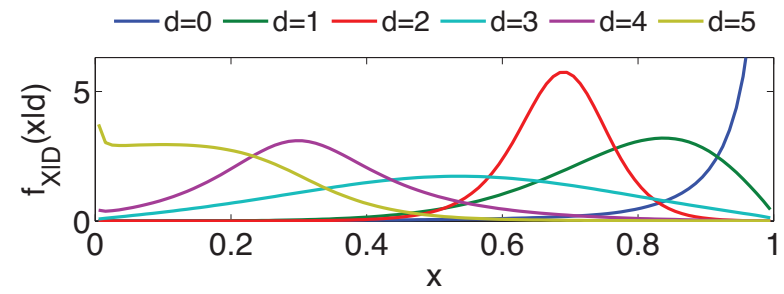


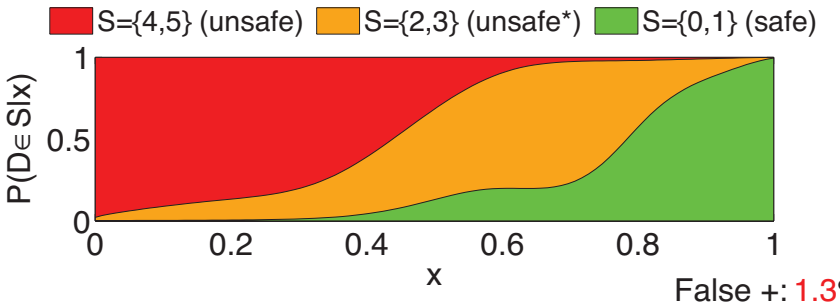


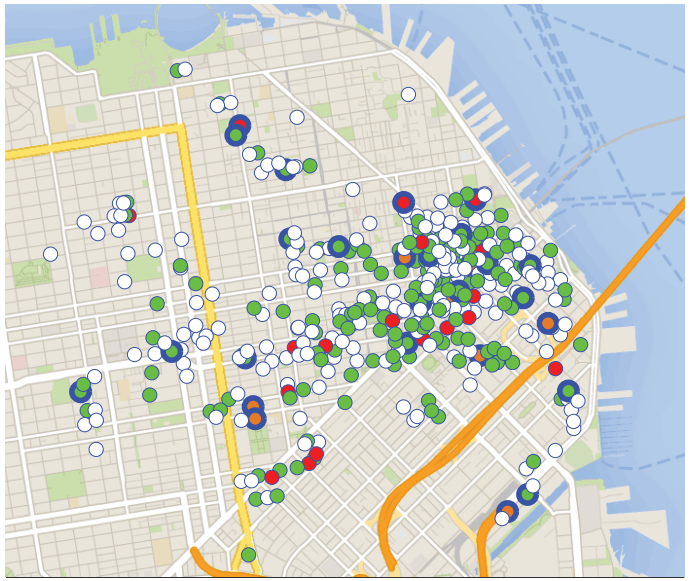


False +: 1.3% | False −: 3.9%

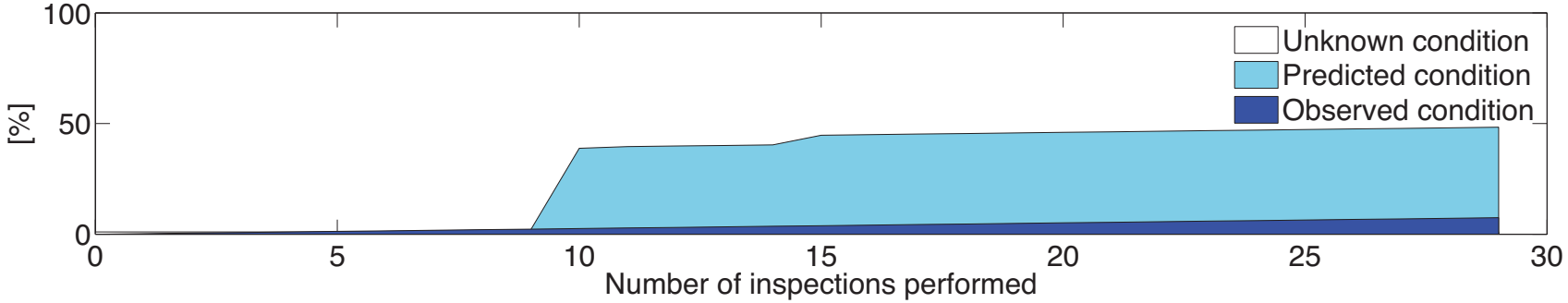


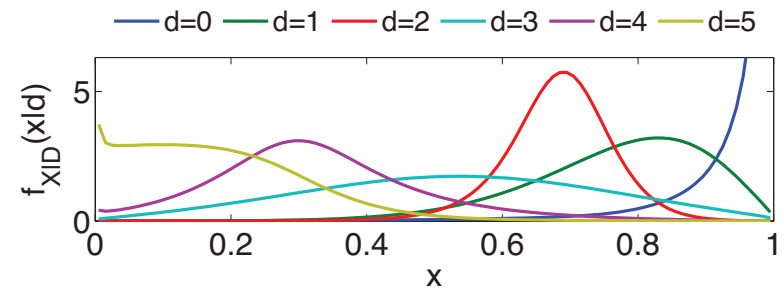


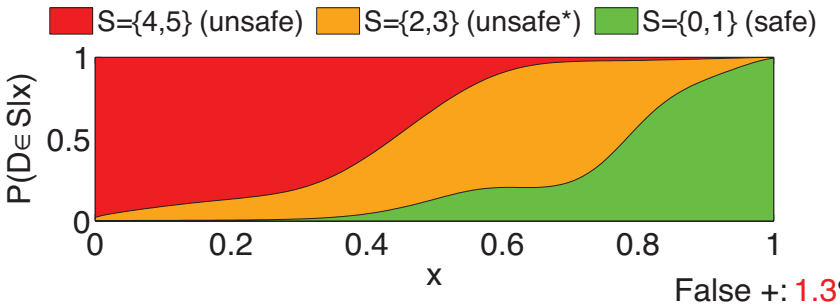


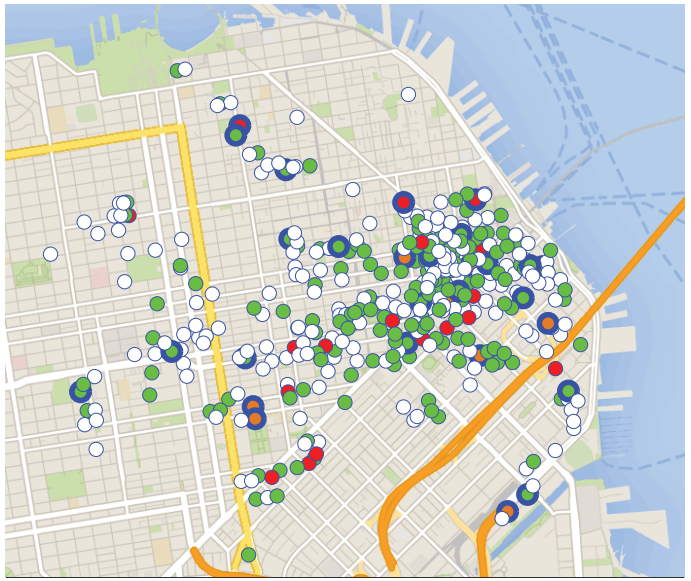


False +: 1.3% | False −: 3.9%

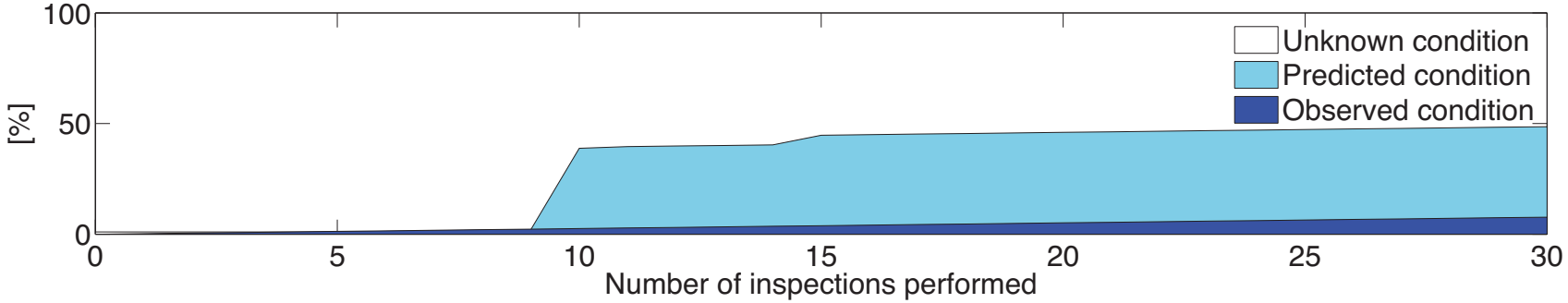




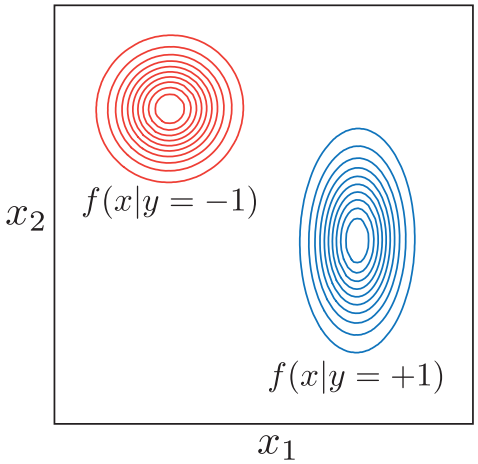




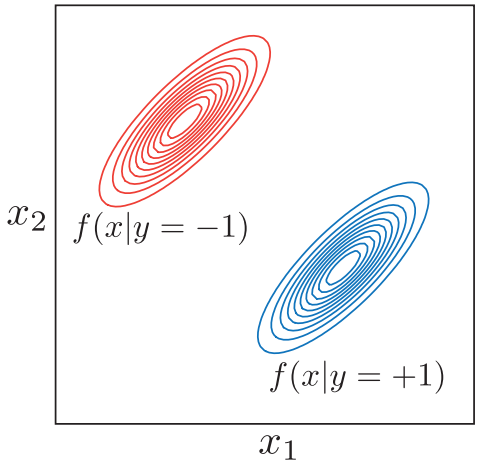
False +: 1.3% | False −: 3.9%



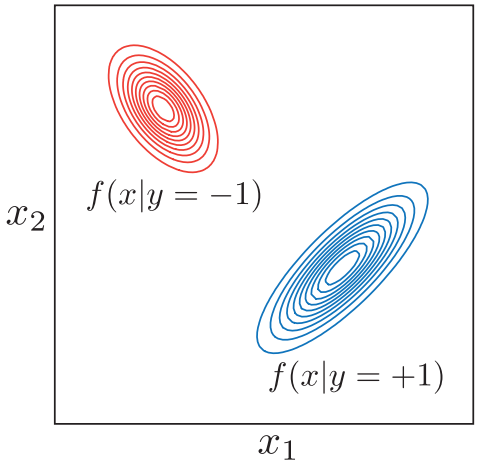
Classic generative methods



(a) Naive Bayes



(b) LDA



(c) QDA

Method	$f(\mathbf{x}\|y=j)$
Naive Bayes	$X_i \perp\!\!\!\perp X_k \| y, \forall i \neq k$ e.g. $\mathcal{B}(x_1;\alpha,\beta)\cdot\mathcal{N}(x_2,\mu,\sigma)$
LDA	$\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{\mathbf{x},j},\boldsymbol{\Sigma}_{\mathbf{x}})$
QDA	$\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{\mathbf{x},j},\boldsymbol{\Sigma}_{\mathbf{x},j})$

Summary - Generative classifier

MLE/MAP point estimation

$$\overbrace{p(y|x,\boldsymbol{\theta}^*)}^{\text{posterior}} = \frac{\overbrace{f(x|y,\boldsymbol{\theta}^*)}^{\text{likelihood}} \cdot \overbrace{p(y;\boldsymbol{\theta}^*)}^{\text{prior}}}{\underbrace{f(x;\boldsymbol{\theta}^*)}_{\text{norm. cte.}}}$$

Bayesian estimation

$$\overbrace{p(y|x,\mathcal{D})}^{\text{posterior predictive}} = \int \frac{\overbrace{f(x|y,\boldsymbol{\theta})}^{\text{likelihood}} \cdot \overbrace{p(y;\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{f(x;\boldsymbol{\theta})}_{\text{norm. cte.}}} \cdot \overbrace{f(\boldsymbol{\theta}|\mathcal{D})}^{\text{posterior}} d\boldsymbol{\theta}$$

Summary - Strength & Limitations

$$\overbrace{p(y|x)}^{\text{posterior}} = \frac{\overbrace{f(x|y)}^{\text{likelihood}} \cdot \overbrace{p(y)}^{\text{prior}}}{\underbrace{f(x)}_{\text{norm. cte.}}}$$

Strength

- Compatible with any type of observation (error type/observation type)
- Small dataset $\rightarrow$ Bayesian inference
- Can handle unbalanced observation classes

Limitations

- Complex to set-up tailored model structures
- Simple structures (e.g. Naive Bayes, LDA & QDA) are outperformed by other *discriminative* methods

Section Outline

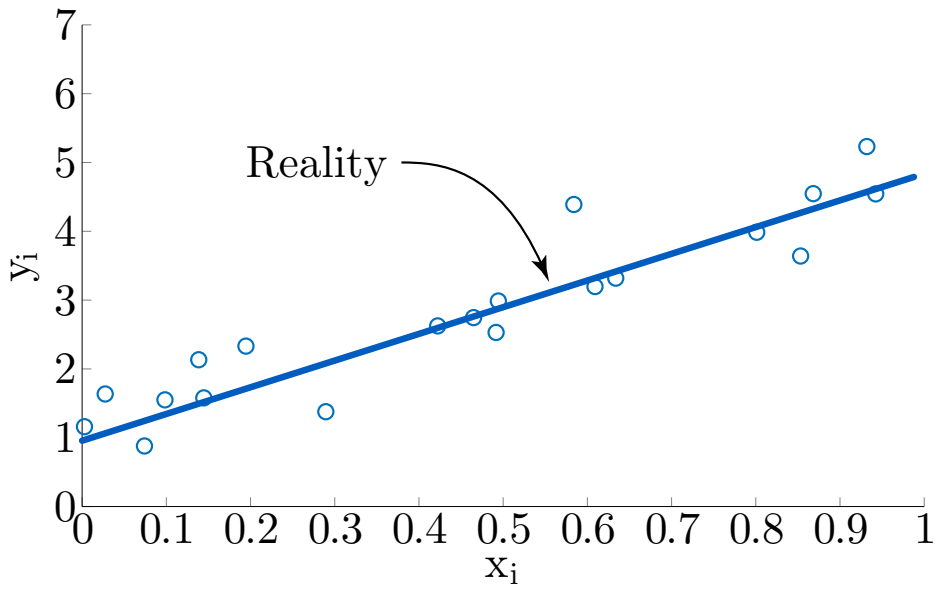
Logistic Regression

3.1 Revision – Linear Regression

3.2 Linear $\rightarrow$ Logistic Regression

3.3 Exemple

Revision – Linear Regression [ ]



Data

$$\mathcal{D} = \{(x_i, y_i), \forall i = 1 : \mathtt{D}\}$$

$$x_i \in \mathbb{R} : \begin{cases} \text{Covariate} \\ \text{attribute} \\ \text{regressor} \end{cases}$$

$$y_i \in \mathbb{R} : \text{Observation}$$

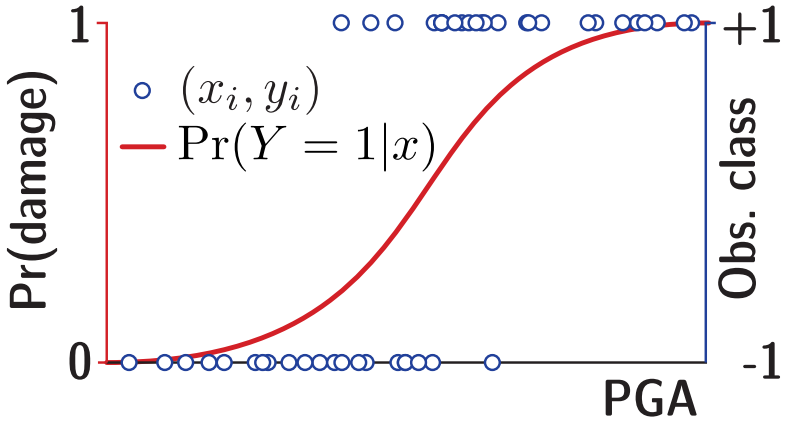
Model

$$g(x) \equiv \text{fct}(x)$$

Hypothesis: $g(x) \equiv$ reality, i.e. no variability

$$y = g(x) + v, \;\; v : V \sim \mathcal{N}(v; 0, \sigma_V^2)$$

Introduction to Logistic regression



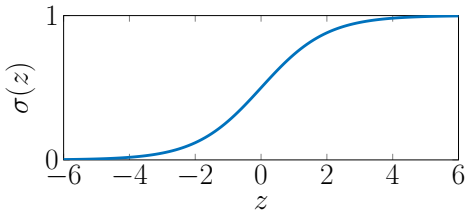
Data

$$\mathcal{D} = \{(x_i, y_i), \forall i = 1 : \mathtt{D}\}$$

$$x_i \in \mathbb{R} : \begin{cases} \text{Covariate} \\ \text{attribute} \\ \text{regressor} \end{cases}$$

$$y_i \in \{-1, 1\} : \text{Observation}$$

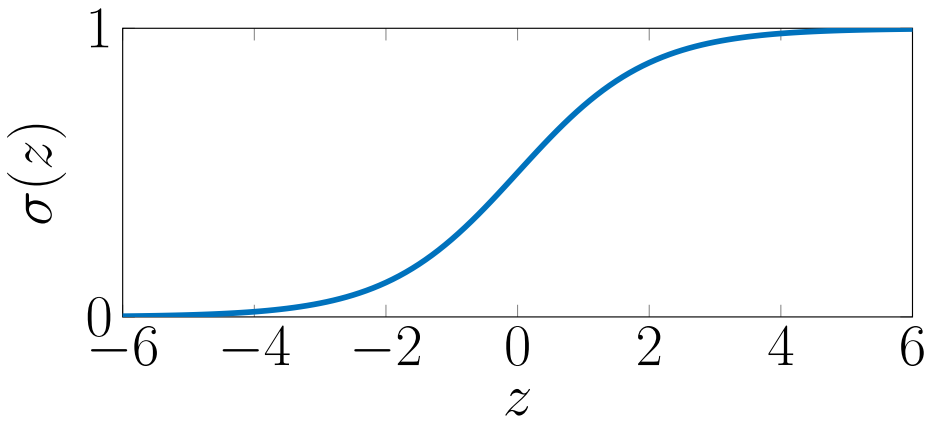
$$\begin{aligned} \Pr(Y = 1|x) &= \sigma(\overbrace{b_0 + b_1 x}^{z}) \\ &= \frac{1}{1 + \exp(-z)} \\ &= \end{aligned}$$



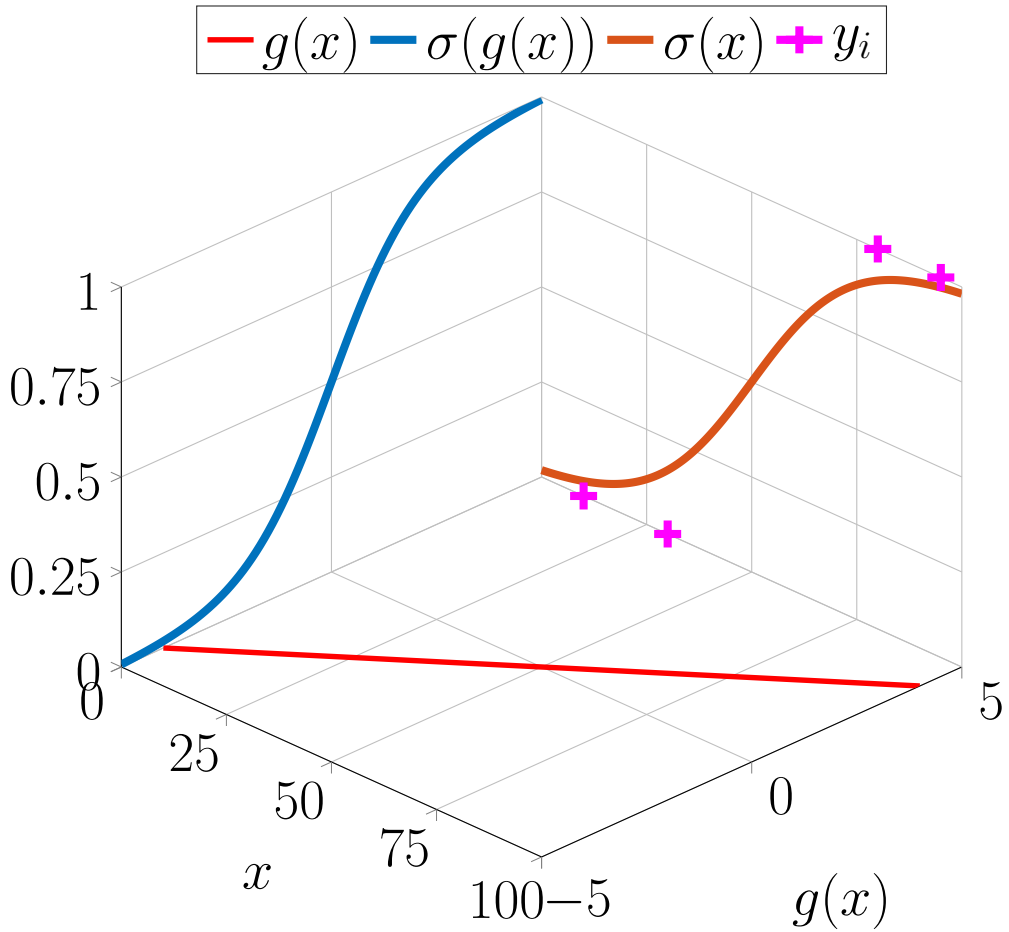
From linear to logistic regression

$$\underbrace{\mathbf{x} \in \mathbb{R}^{\mathtt{X}}}_{\text{covariates}} \rightarrow \underbrace{g(\mathbf{x}) = \mathbf{Xb} \in \mathbb{R}}_{\text{hidden/latent variable}} \rightarrow \underbrace{\sigma(g(\mathbf{x})) \in (0,1)}_{\Pr(Y=y|\mathbf{x})}$$

$$\sigma(z) = \frac{1}{1+\exp(-z)}$$

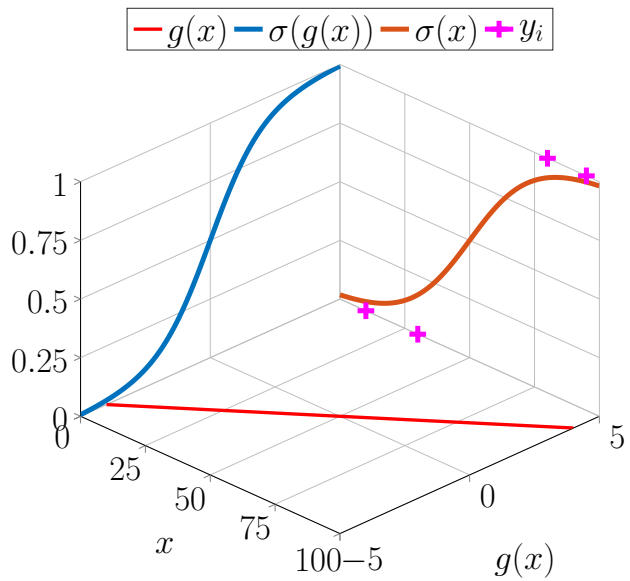


Example - Linear regression & sigmoid [.m]



[CIV_ML/Logistic_regression_exemple.m]

Likelihood



Marginal likelihoods

For $x_i : y_i = +1 \rightarrow \mathbf{X}_{(+1)}$

$$\Pr(Y = +1|x_i) = \sigma(\mathbf{X}_{(+1)}\mathbf{b}) \equiv \sigma(x_i)$$

For $x_i : y_i = -1 \rightarrow \mathbf{X}_{(-1)}$

$$\Pr(Y = -1|x_i) = 1 - \sigma(\mathbf{X}_{(-1)}\mathbf{b})$$

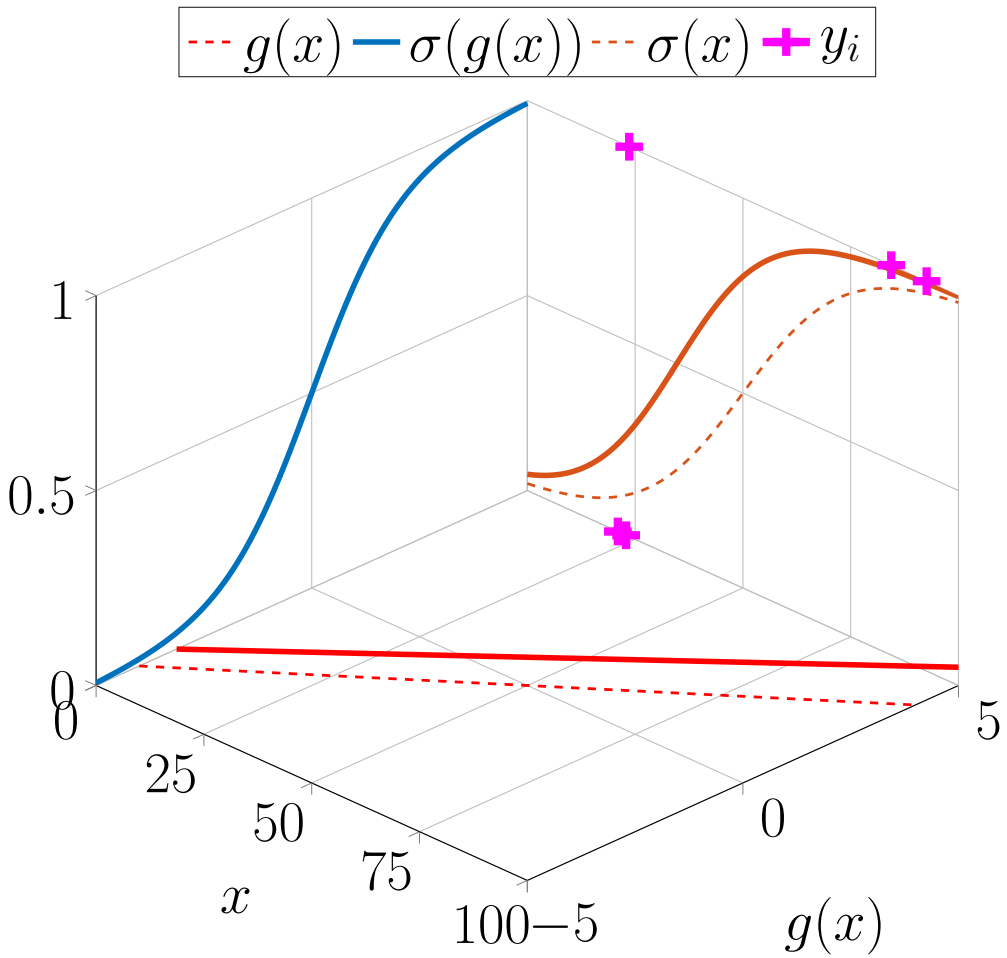
Joint Log-likelihood

$$\ln p(\mathcal{D}|\mathbf{b}) = \sum \ln(\sigma(\mathbf{X}_{(+1)}\mathbf{b})) + \sum \ln(1 - \sigma(\mathbf{X}_{(-1)}\mathbf{b}))$$

Maximum Likelihood Estimation

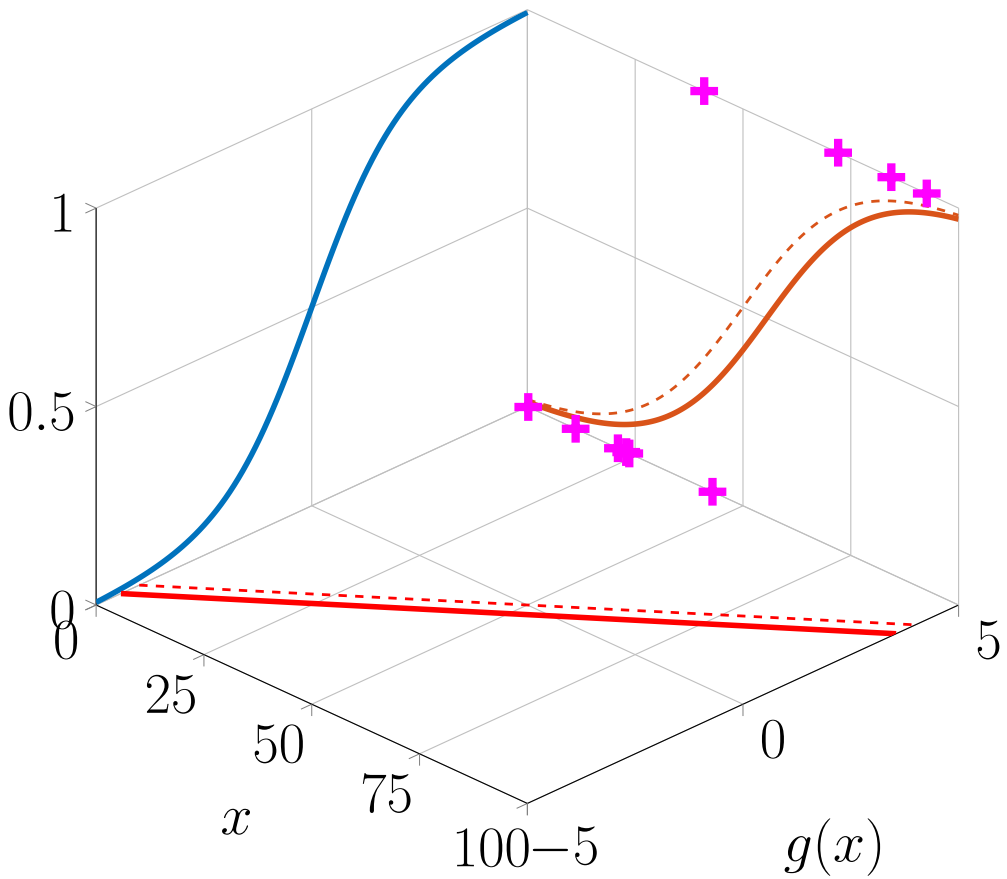
$$\mathbf{b}^* = \arg \max_{\mathbf{b}} \ln p(\mathcal{D}|\mathbf{b})$$

Example - Logistic regression [.m]



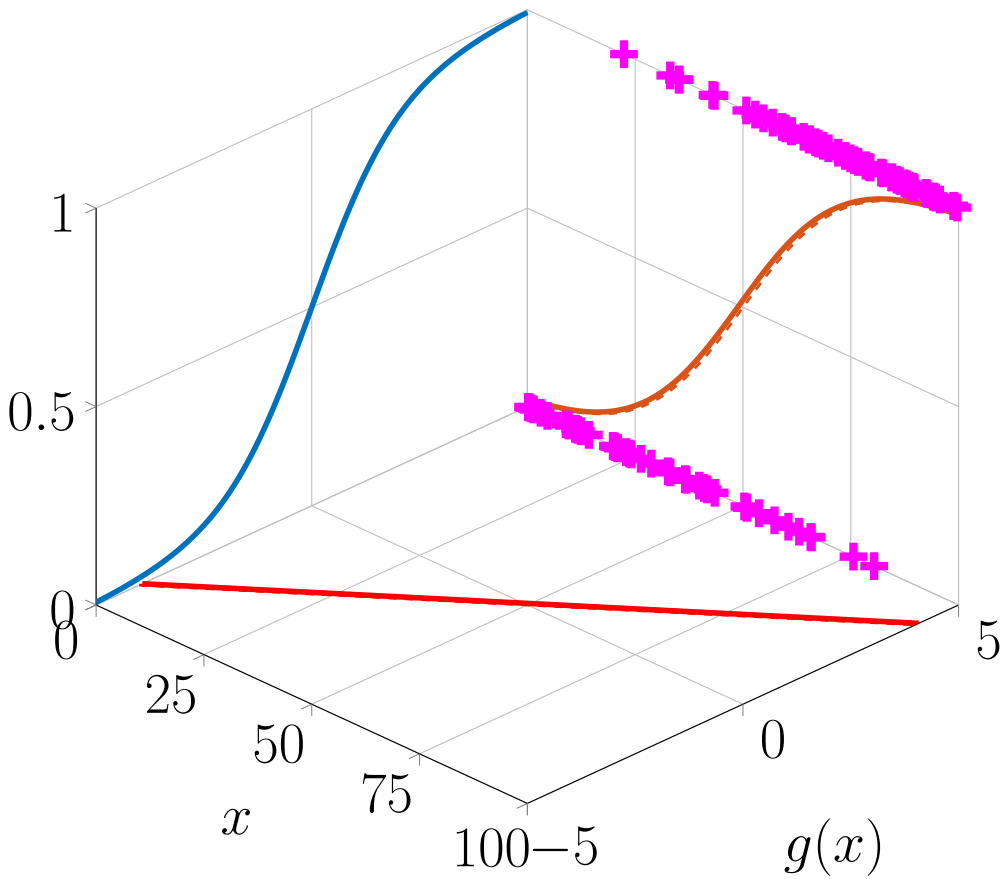
[CIV_ML/Logistic_regression_exemple_2.m]

Example - Logistic regression [.m]



[CIV_ML/Logistic_regression_exemple_2.m]

Example - Logistic regression [.m]



[CIV_ML/Logistic_regression_exemple_2.m]

Limitations

- ⚠ The performance depends on the capacity to **hand-pick the correct transformation functions**
  (Difficult when $\mathtt{X} > 1 \rightarrow$ **Cross-validation**)
- ⚠ No analytic formulation to estimate parameters
- ⚠ Only compatible with error-free direct observations
- ⚠ Not as powerful as modern methods

Why is it used then?

- ✓ Simple
- ✓ Interpretability of model parameters $\mathbf{b}$
  (Discrete choice models $\rightarrow$ transportation & economics)

Section Outline

GPC

4.1 GPR v.s. GPC

4.2 Updating a GP using exact observations

4.3 GPC formulation

Gaussian Process Regression summary

Given a **system response** so that

$$\underbrace{g_i}_{\text{observation}} = \overbrace{g(\mathbf{x}_i)}^{\text{reality}} \in \mathbb{R}, \quad \underbrace{\mathbf{x}_i = [x_1, x_2, \cdots, x_{\mathtt{X}}]_i^{\intercal}}_{\text{covariates}} \in \mathbb{R}^{\mathtt{X}}$$

Data: $\mathcal{D} = \{(\mathbf{x}_i, g_i), \forall i = 1 : \mathtt{D}\}$

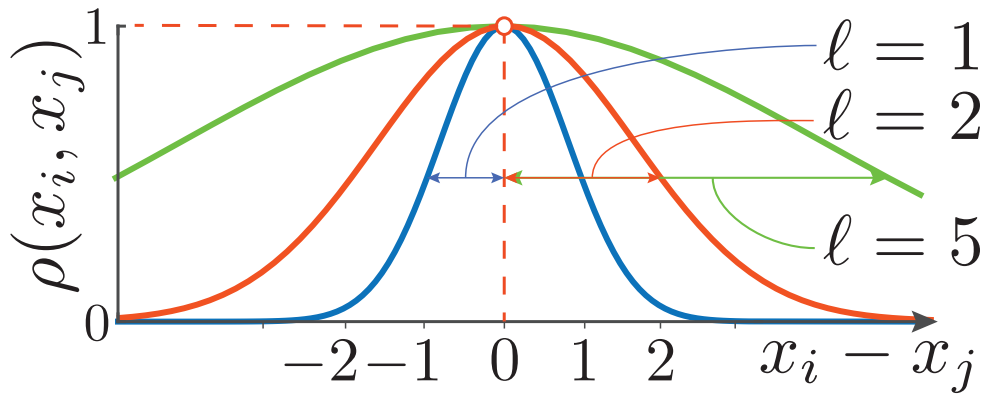
$$\mathbf{g} = [g_1, g_2, \cdots, g_{\mathtt{D}}]_{1\times \mathtt{D}}^{\intercal} \in \mathbb{R}^{\mathtt{D}}, \quad \mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\mathtt{X}}]_{\mathtt{D}\times \mathtt{X}}$$

Gaussian process: $g(\mathbf{x}) : \mathbf{G} \sim \mathcal{N}(g(\mathbf{x}); \mathbf{m}_{\mathbf{G}}, \mathbf{\Sigma}_{\mathbf{G}})$

Set of discrete Gaussian random variables for which the pairwise correlation between G_i and G_j is a function of the distance between attributes

$$[\mathbf{\Sigma}_{\mathbf{G}}]_{ij} = \rho(x_i, x_j)\sigma_{G_i}, \sigma_{G_j}, \quad \rho(x_i, x_j) = \text{fct}(|x_i - x_j|)$$

Correlation function



Updating a GP using exact observations

Given $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{g}_i),\ i = 1, \cdots \mathtt{D}\}$ a set of $\mathtt{D}$ observations and $\mathbf{x}_*$ a set of $\mathtt{X}_*$ covariates for which we want to predict

$$f(\mathbf{g}_*|\mathbf{x}_*, \mathcal{D})$$

Reminder: Gaussian conditionals are also gaussian

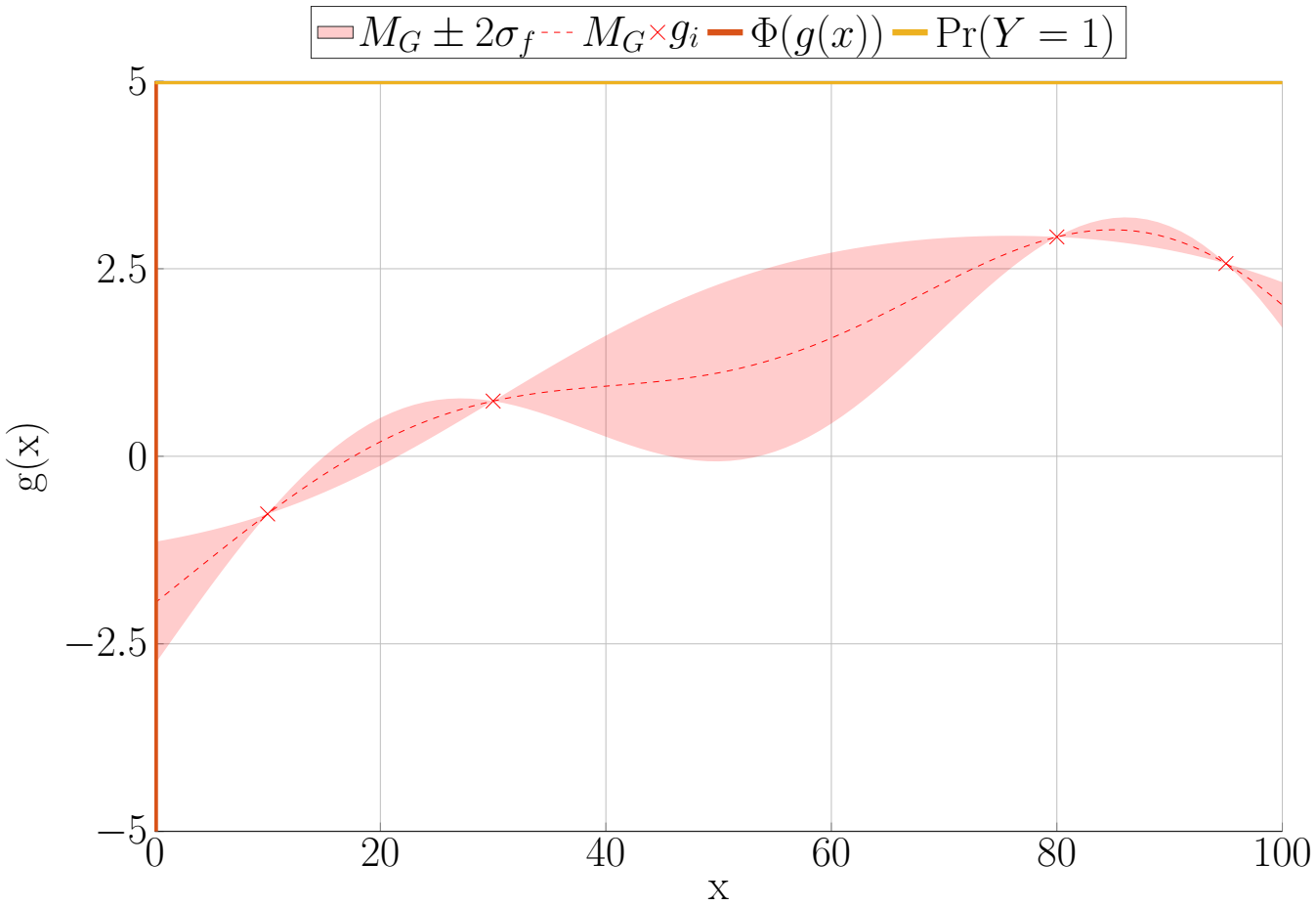
$$\underbrace{\left\{ \begin{array}{c} \mathbf{G} \\ \mathbf{G}_* \end{array} \right\}, \mathbf{m} = \left\{ \begin{array}{c} \mathbf{m}_\mathbf{G} \\ \hline \mathbf{m}_{\mathbf{G}_*} \end{array} \right\}, \mathbf{\Sigma} = \left[\begin{array}{c|c} \mathbf{\Sigma}_\mathbf{G} & \mathbf{\Sigma}_{\mathbf{G}_*} \\ \hline \mathbf{\Sigma}_{\mathbf{G}_*}^\mathsf{T} & \mathbf{\Sigma}_* \end{array} \right]}_{\text{Prior knowledge}}$$

$$[\mathbf{\Sigma}_\mathbf{G}]_{ij} = \rho(x_i, x_j)\sigma_G^2$$
$$[\mathbf{\Sigma}_{\mathbf{G}_*}]_{ij} = \rho(x_i, x_{*j})\sigma_G^2$$
$$[\mathbf{\Sigma}_*]_{ij} = \rho(x_{*i}, x_{*j})\sigma_G^2$$

$$f(\mathbf{g}_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(\mathbf{g}_*; \mathbf{m}_{*|\mathcal{D}}, \mathbf{\Sigma}_{*|\mathcal{D}})$$

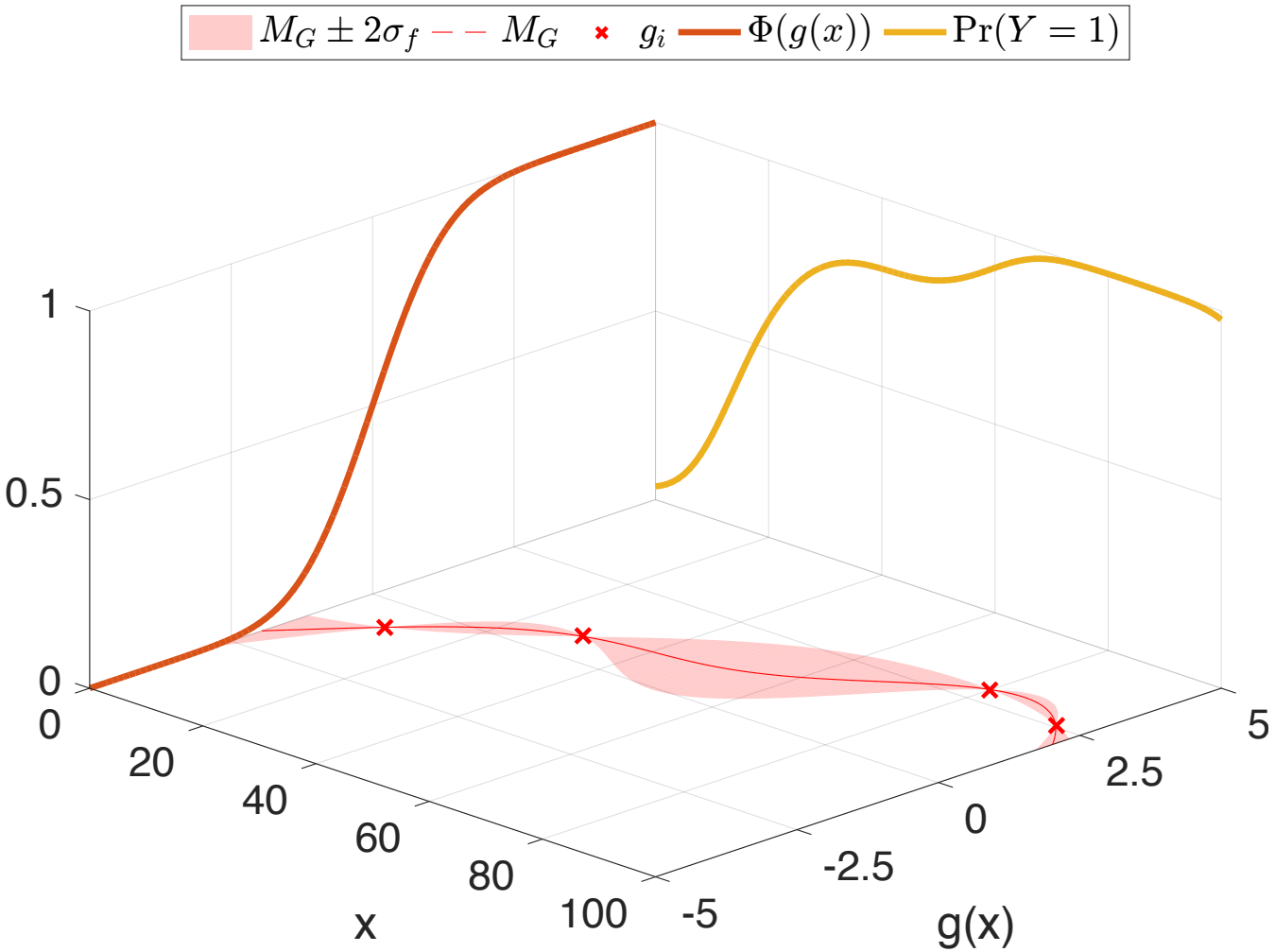
$$\underbrace{\begin{array}{l} \mathbf{m}_{*|\mathcal{D}} = \mathbf{m}_{\mathbf{G}_*} + \mathbf{\Sigma}_{\mathbf{G}_*}^\mathsf{T} \mathbf{\Sigma}_\mathbf{G}^{-1}(\mathbf{g} - \mathbf{m}_\mathbf{G}) \\ \mathbf{\Sigma}_{*|\mathcal{D}} = \mathbf{\Sigma}_* - \mathbf{\Sigma}_{\mathbf{G}_*}^\mathsf{T} \mathbf{\Sigma}_\mathbf{G}^{-1} \mathbf{\Sigma}_{\mathbf{G}_*} \end{array}}_{\text{Posterior knowledge}}$$

Example - GPR v.s. GPC [.m]



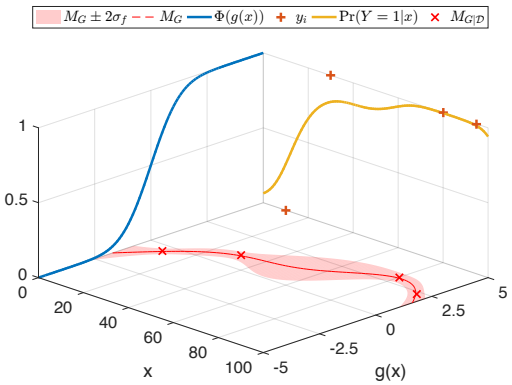
[CIV_ML/GPC_exemple_1.m]

Example - GPR v.s. GPC [.m]



[CIV_ML/GPC_exemple_1.m]

Example - GPR v.s. GPC [.m]



$$\Phi(\cdot) \triangleq \text{Normal CDF}$$

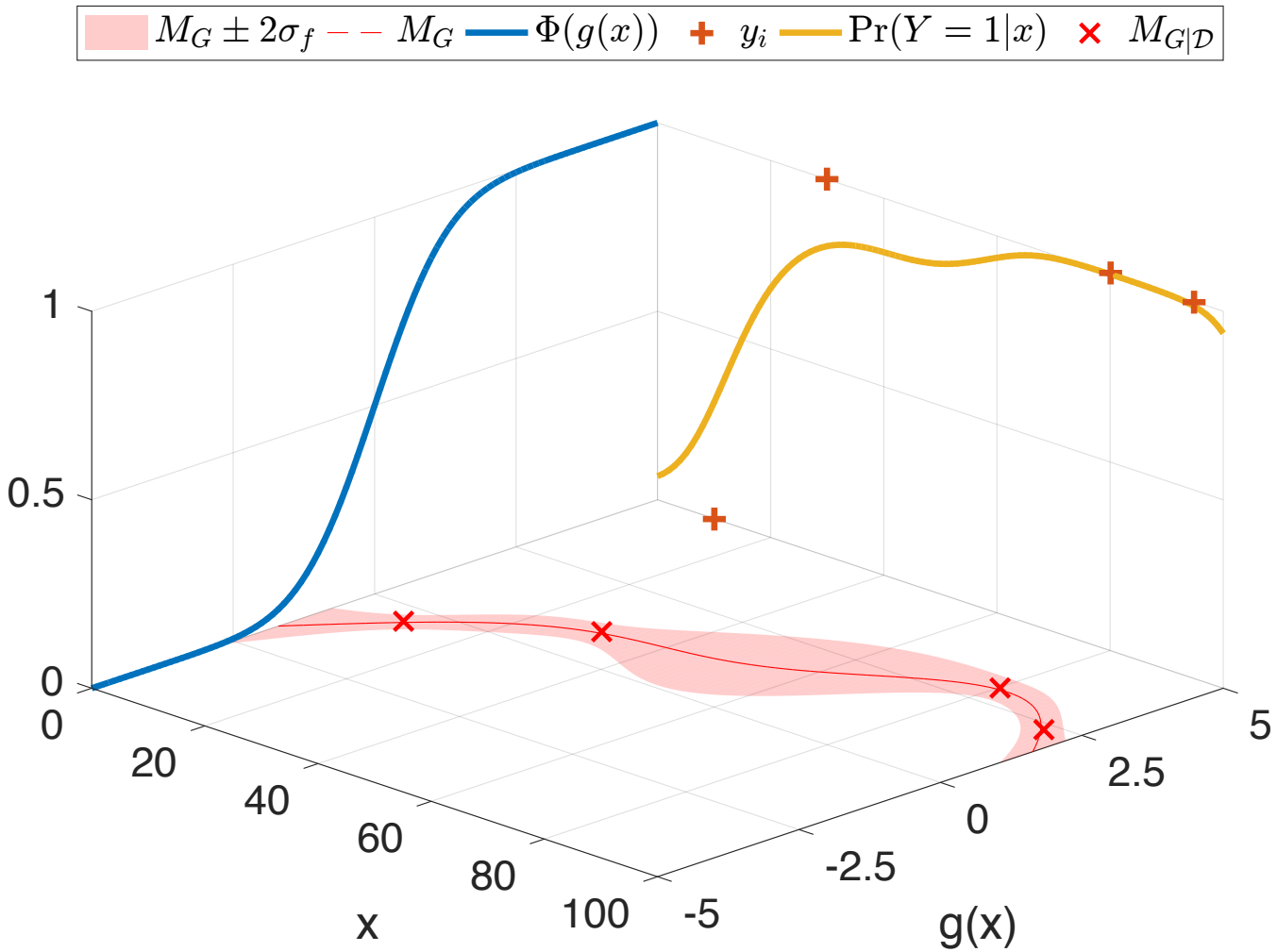
$$\begin{aligned} \Pr(Y=1|x_*,\mathcal{D}) &= \int \Phi(g_*) f(g_*|x_*,\mathcal{D}) dg_* \\ &= \Phi\left(\frac{m_{G_*|\mathbf{g}}}{\sqrt{1+\text{var}[G_*|\mathbf{g}]}}\right) \end{aligned}$$

In a classification setup, g is not observed
Only $y_i \in \{-1,1\}$ is observed.

For each $y_i \in \{-1,1\}$ we need to infer $\mathbf{g}_i$

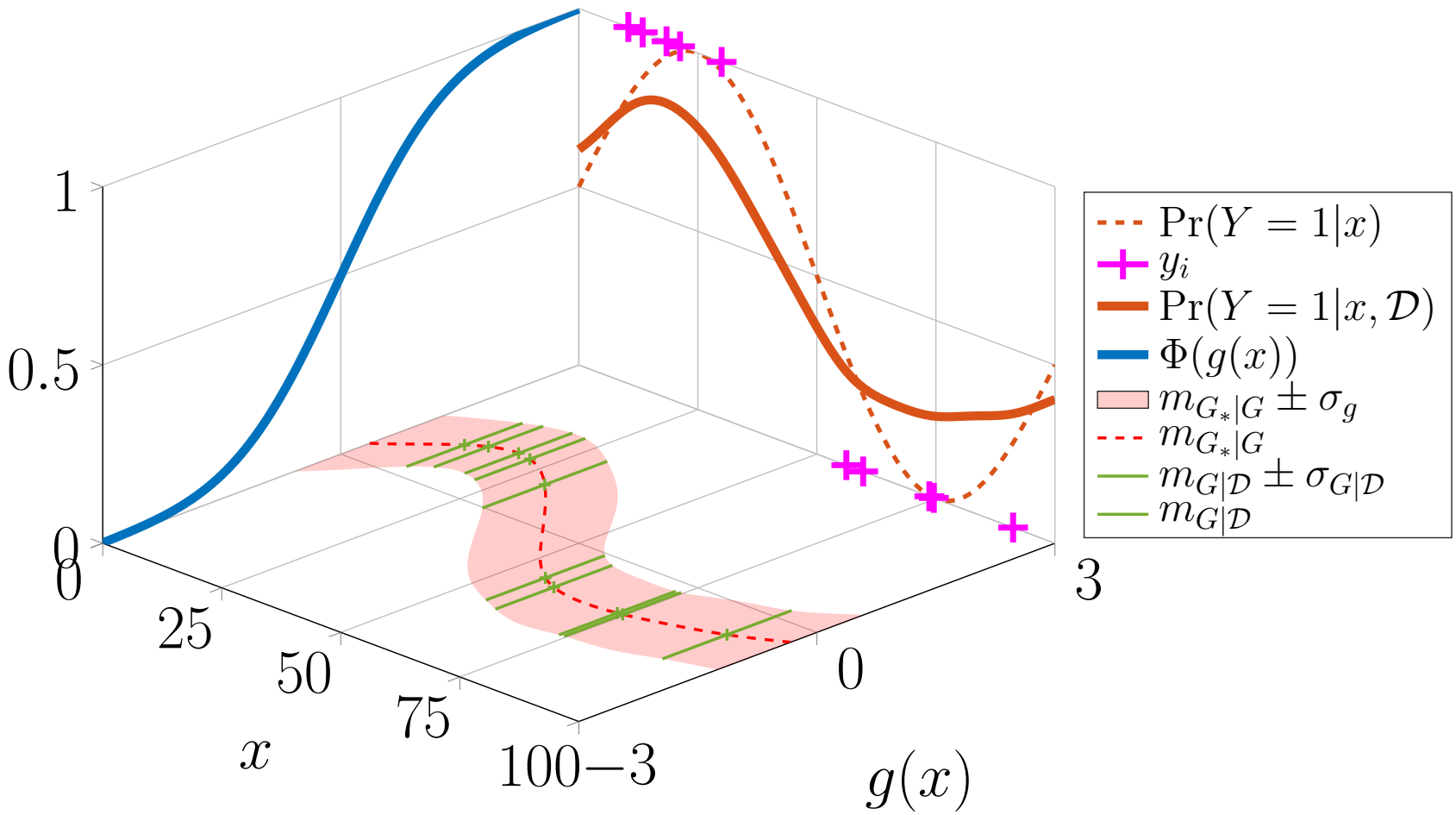
[CIV_ML/GPC_exemple_1.m]

Example - GPR v.s. GPC [.m]



[CIV_ML/GPC_exemple_2.m]

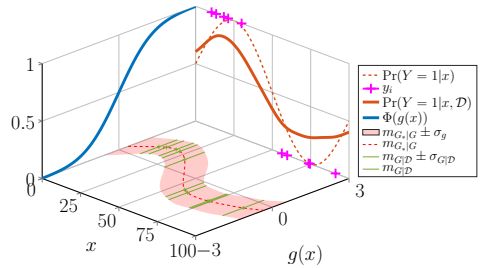
Example - GPC / GPML [.m]



[CIV_ML/GPC_gpml.m]

Gaussian Process Classification Formulation

$$\Pr(Y=1|x_*,\mathcal{D}) = \int \Phi(g_*)\cdot f(g_*|\mathbf{g},x_*,\mathcal{D})dg_*$$
$$= \Phi\left(\frac{\mathbb{E}[G_*|x_*,\mathbf{g},\mathcal{D}]}{\sqrt{1+\text{var}[G_*|x_*,\mathbf{g},\mathcal{D}]}}\right)$$



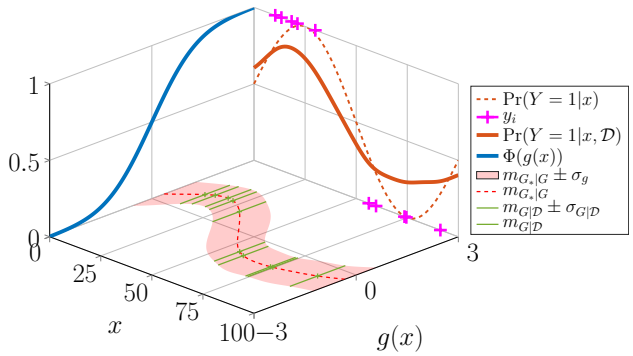
$$f(\mathbf{g}_*|\mathbf{g}) \equiv \mathcal{N}(\mathbf{g}_*;\boldsymbol{\mu}_{*|\mathbf{g},\mathcal{D}},\boldsymbol{\Sigma}_{*|\mathbf{g},\mathcal{D}}) \quad \text{(GPR)}$$

$$\boldsymbol{\mu}_{*|\mathbf{g},\mathcal{D}} = \boldsymbol{\Sigma}_{\mathbf{G}_*}^{\mathsf{T}}\boldsymbol{\Sigma}_{\mathbf{G}}^{-1}(\boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}}-\boldsymbol{\mu}_{\mathbf{G}}-\mathbf{0}) \quad \text{(conditional mean)}$$

$$\boldsymbol{\Sigma}_{*|\mathbf{g},\mathcal{D}} = \boldsymbol{\Sigma}_* - \boldsymbol{\Sigma}_{\mathbf{G}_*}^{\mathsf{T}}\boldsymbol{\Sigma}_{\mathbf{G}|\mathcal{D}}^{-1}\boldsymbol{\Sigma}_{\mathbf{G}_*} \quad \text{(conditional COV)}$$

$$f(\mathbf{g}|\mathcal{D}) = \frac{p(\mathcal{D}_y|\mathbf{g})\cdot f(\mathbf{g}|\mathcal{D}_x)}{p(\mathcal{D}_y|\mathcal{D}_x)} \quad \text{(posterior}$$
$$\propto p(\mathcal{D}_y|\mathbf{g})\cdot f(\mathbf{g}|\mathcal{D}_x) \quad \text{hidden covariates)}$$
$$\approx \mathcal{N}(\mathbf{g};\boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}},\boldsymbol{\Sigma}_{\mathbf{G}|\mathcal{D}}) \quad \text{(Laplace approx.)}$$

GPC - Hidden covariate estimation, $\mathbf{g}|\mathcal{D}$



$$p(\mathcal{D}_y|\mathbf{g}) = \prod_{i=1}^{D} p(y_i|g_i) = \prod_{i=1}^{D} \Phi(y_i \cdot g_i)$$

$$f(\mathbf{g}|\mathcal{D}) = \frac{p(\mathcal{D}_y|\mathbf{g}) \cdot f(\mathbf{g}|\mathcal{D}_x)}{p(\mathcal{D}_y|\mathcal{D}_x)} \propto f(\mathcal{D}_y|\mathbf{g}) \cdot f(\mathbf{g}|\mathcal{D}_x) \approx \mathcal{N}(\mathbf{g}; \boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}}, \boldsymbol{\Sigma}_{\mathbf{G}|\mathcal{D}})$$

$$\underbrace{\begin{aligned} p(y=+1|g) &= \Phi(g) \\ p(y=-1|g) &= 1-\Phi(g) \\ &= \Phi(-g) \end{aligned}}_{p(y|g)=\Phi(y\cdot g)}$$

Laplace approximation

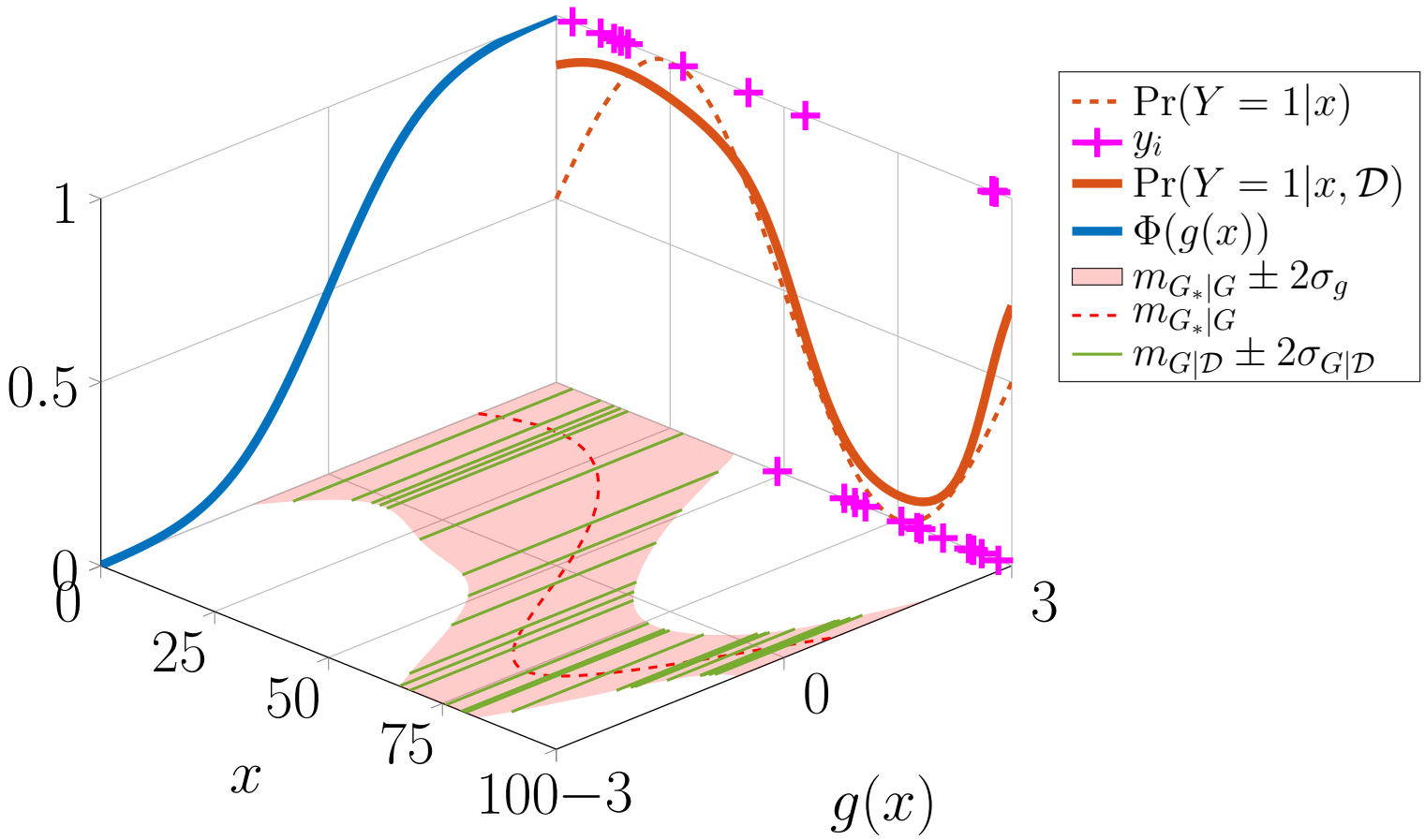
$$\boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}} = \arg\max_{\mathbf{g}} \ln f(\mathbf{g}|\mathcal{D}) = \boldsymbol{\Sigma}_{\mathbf{G}} \nabla \ln f(\mathcal{D}_y|\mathbf{g} = \boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}})$$

recursive estimation

start: $\boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}} = \mathbf{0}$

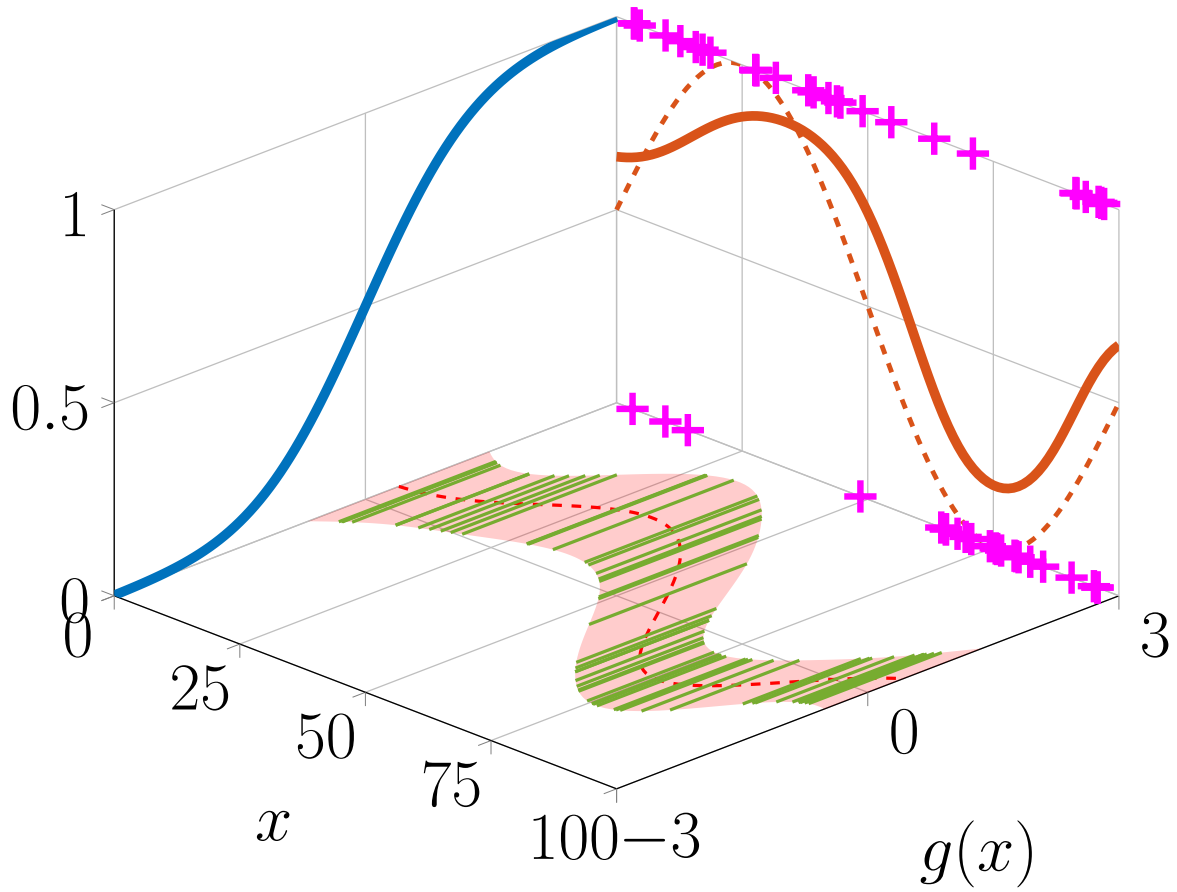
$$\boldsymbol{\Sigma}_{\mathbf{G}|\mathcal{D}} = (\boldsymbol{\Sigma}_{\mathbf{G}}^{-1} - \mathbf{H}[\ln f(\mathcal{D}_y|\mathbf{g} = \boldsymbol{\mu}_{\mathbf{G}|\mathcal{D}})])^{-1}$$

Example - GPC / GPML [.m]



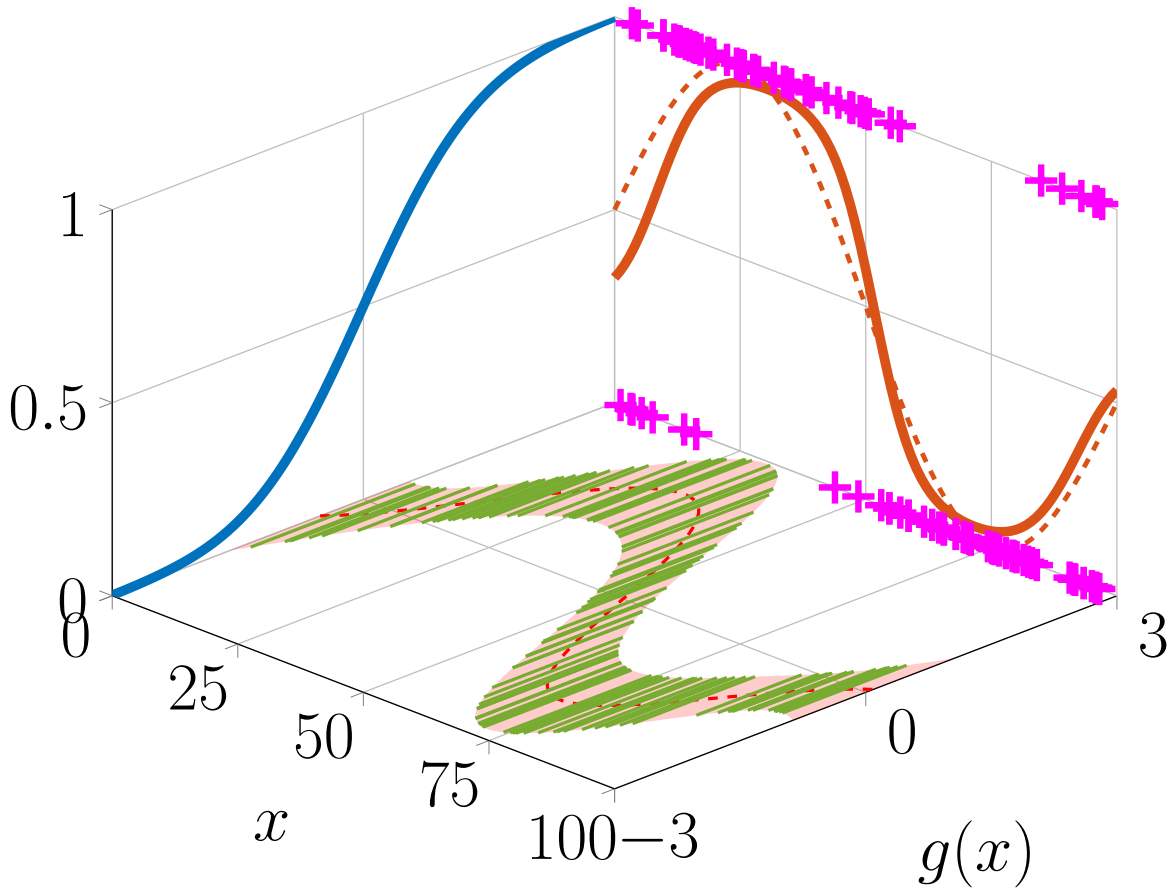
[CIV_ML/GPC_gpml.m]

Example - GPC / GPML [.m]



[CIV_ML/GPC_gpml.m]

Example - GPC / GPML [.m]



[CIV_ML/GPC_gpml.m]

Summary - Gaussian Process Classification

Strength

- Simple to set-up for any number of covariates **x**
- Allows interpolating and extrapolating (uncertainty estimates)

Limitations

- Only compatible with error-free direct observations
- When applicable, choose **regression** over **classification**, e.g. Soil contamination example
- The performance is limited for large datasets $\mathtt{D} > 10^5$

Section Outline

Neural Networks

5.1 Structure - From Logistic regression to NN
5.2 Structure - Bivariate classification
5.3 Structure - Multivariate classification
5.4 Deep Learning – CNN
5.5 Examples

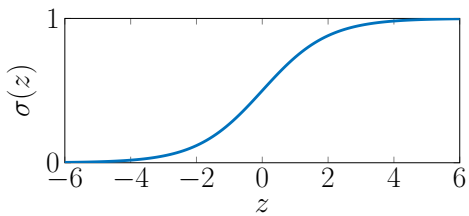
Introduction to the neural network structure

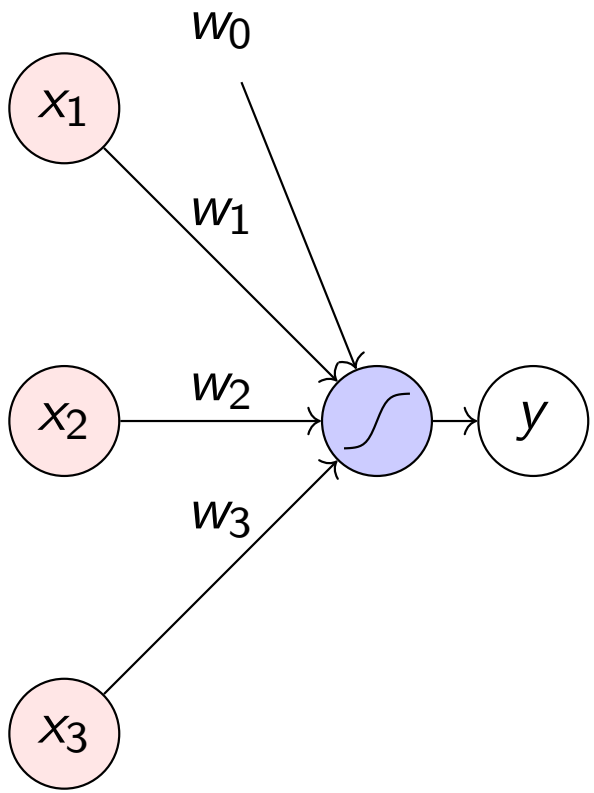
Data $\mathcal{D} = \{(x_i, y_i), \forall i = 1 : \mathtt{D}\}$

$$x_i \in \mathbb{R} : \begin{cases} \text{Covariate} \\ \text{attribute} \\ \text{regressor} \end{cases}$$

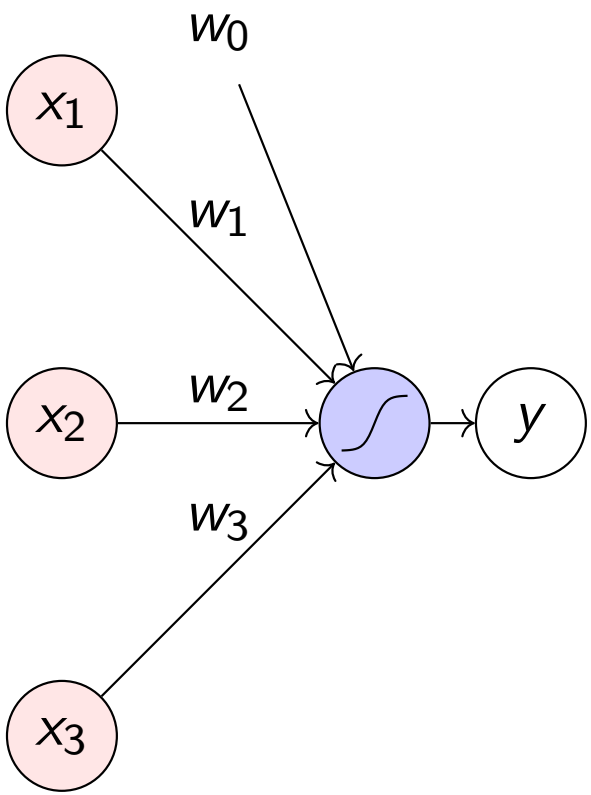
$y_i \in \{-1, 1\}$: Observation

$$\begin{aligned} Pr(Y = 1|\mathbf{x}) &= \sigma(\overbrace{w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3}^{z}) \\ &= \frac{1}{1 + \exp(-z)} \\ &= \end{aligned}$$

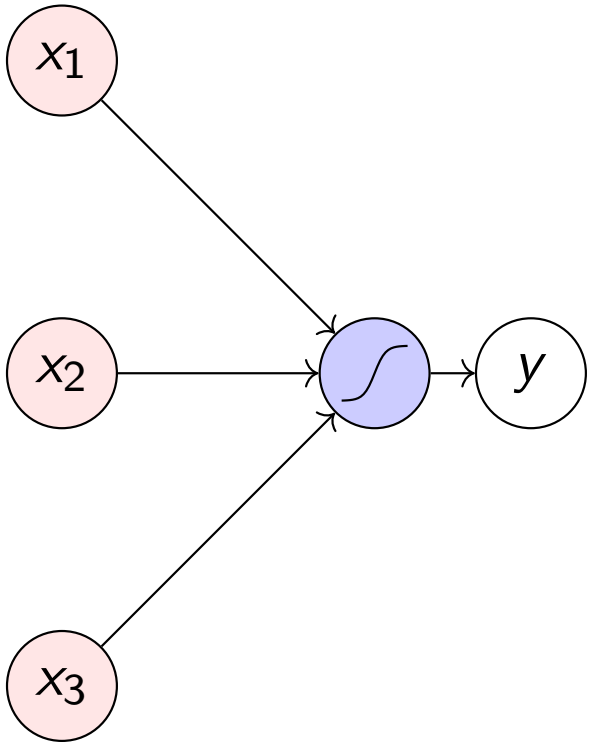




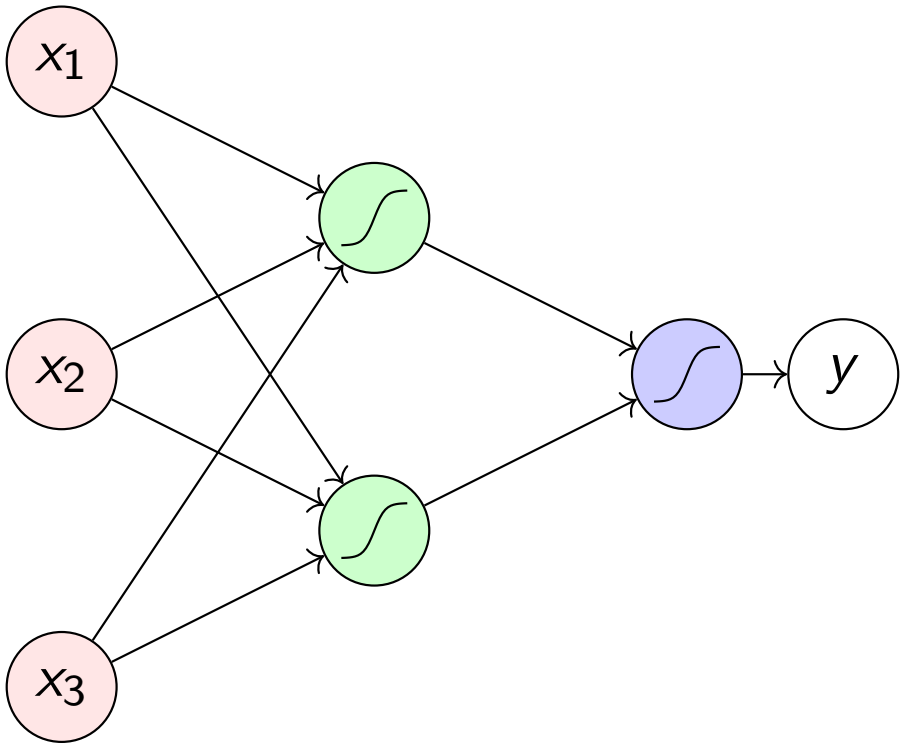
Neural network structure for binary classification



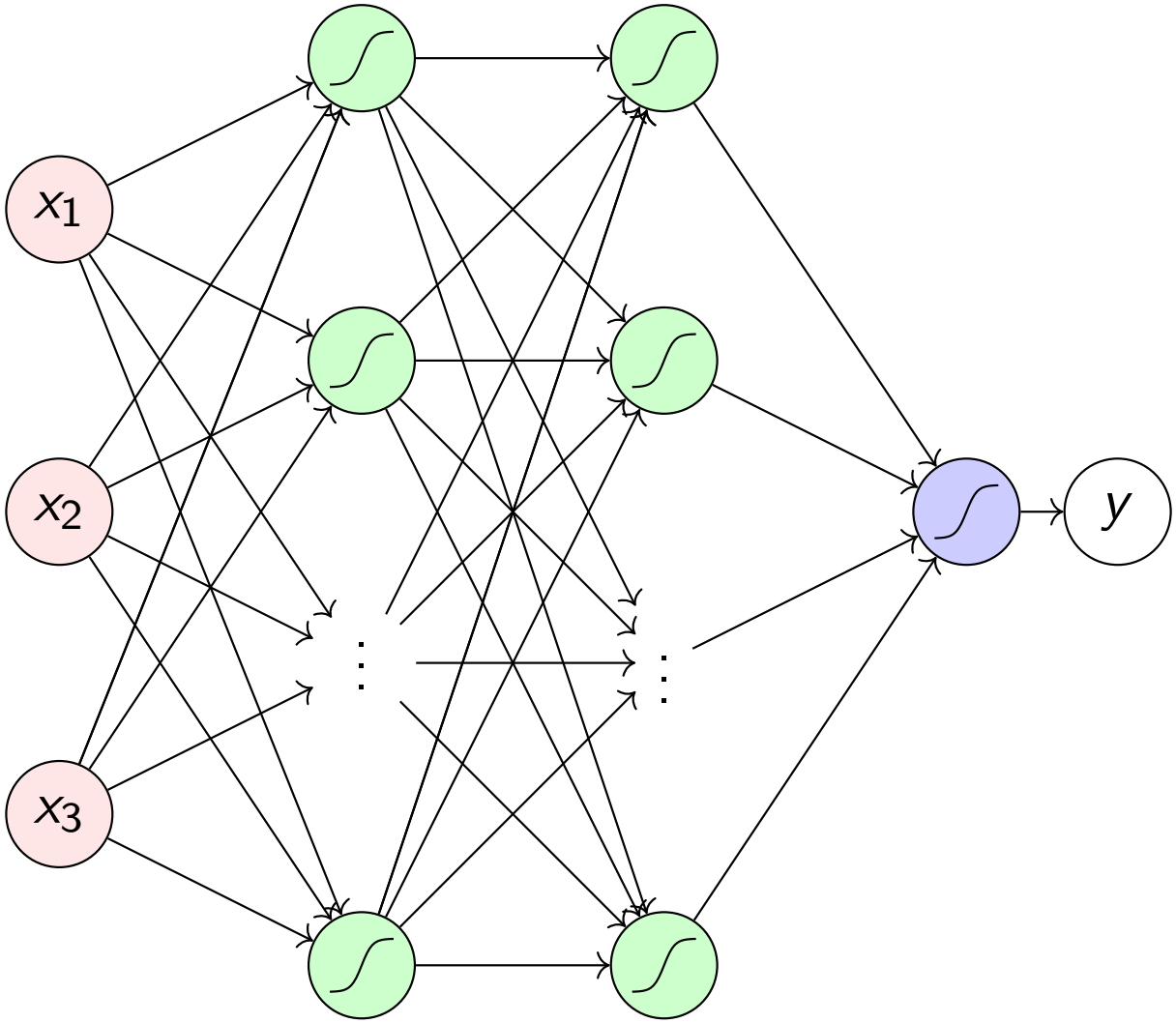
Neural network structure for binary classification



Neural network structure for binary classification

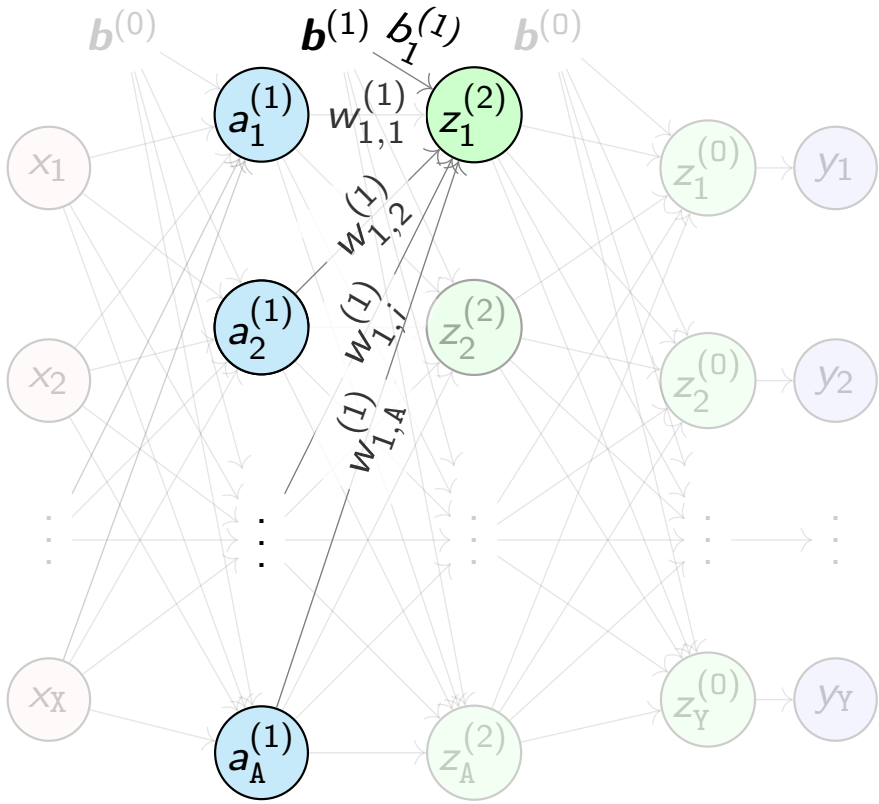


Neural network structure for binary classification



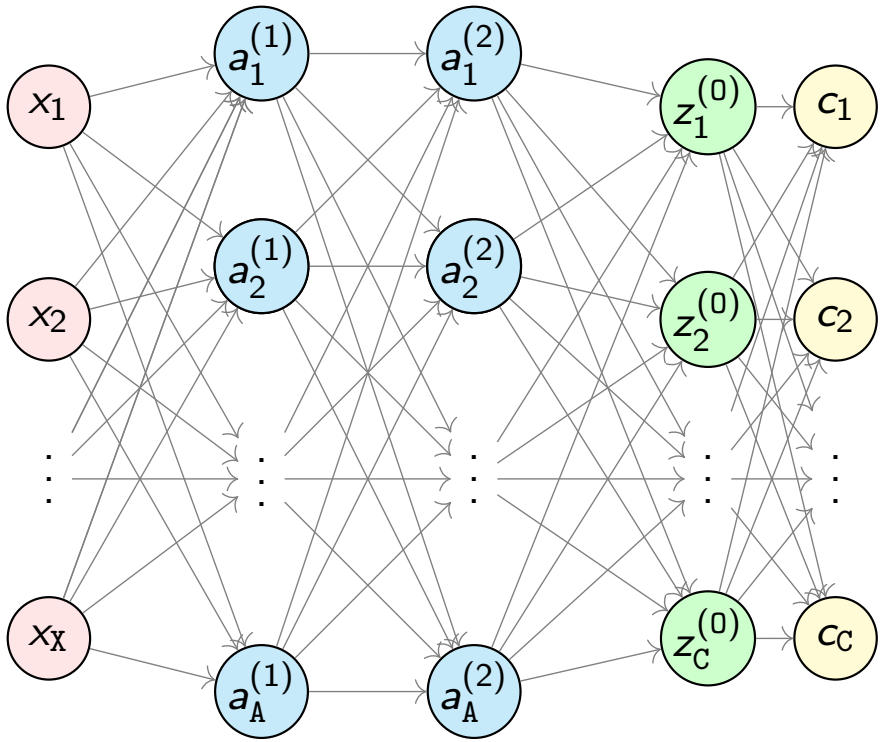
NN – Multivariate regression

- $\mathbf{x}$: Input layer / covariates
- $\mathbf{a}^{(l)}$: Activation units
  $a_i^{(j)} = \sigma(z_i^{(j)})$
- $\mathbf{z}^{(l)}$: Hidden units
  $z_i^{(2)} = \mathbf{w}^\intercal \mathbf{a}^{(1)} + b_i$
- $\mathbf{z}^{(0)}$: Output layer
- $\mathbf{y}$: Observations



NN – Multivariate classification

- $\mathbf{x}$: Input layer / covariates
- $\mathbf{a}^{(l)}$: Activation units
- $\mathbf{z}^{(l)}$: Hidden units
- $\mathbf{z}^{(0)}$: Output layer, $\mathbf{z}^{(0)} \in \mathbb{R}^{\mathrm{C}}$
- $\mathbf{c}$: Classes, $\mathbf{c} \in \{0, 1\}^{\mathrm{C}}$

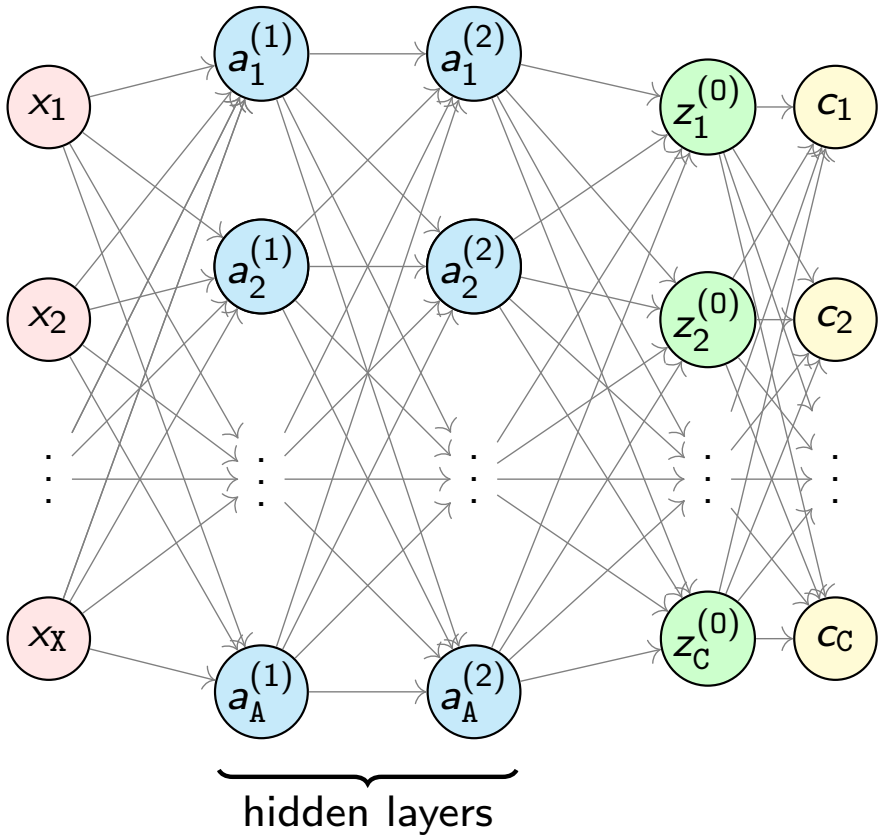


$$\Pr(c_i) = \text{softmax}(\boldsymbol{z}^{(0)}, i) = \frac{\exp(z_i^{(0)})}{\sum_{j=1}^{\mathrm{C}} \exp(z_j^{(0)})}$$

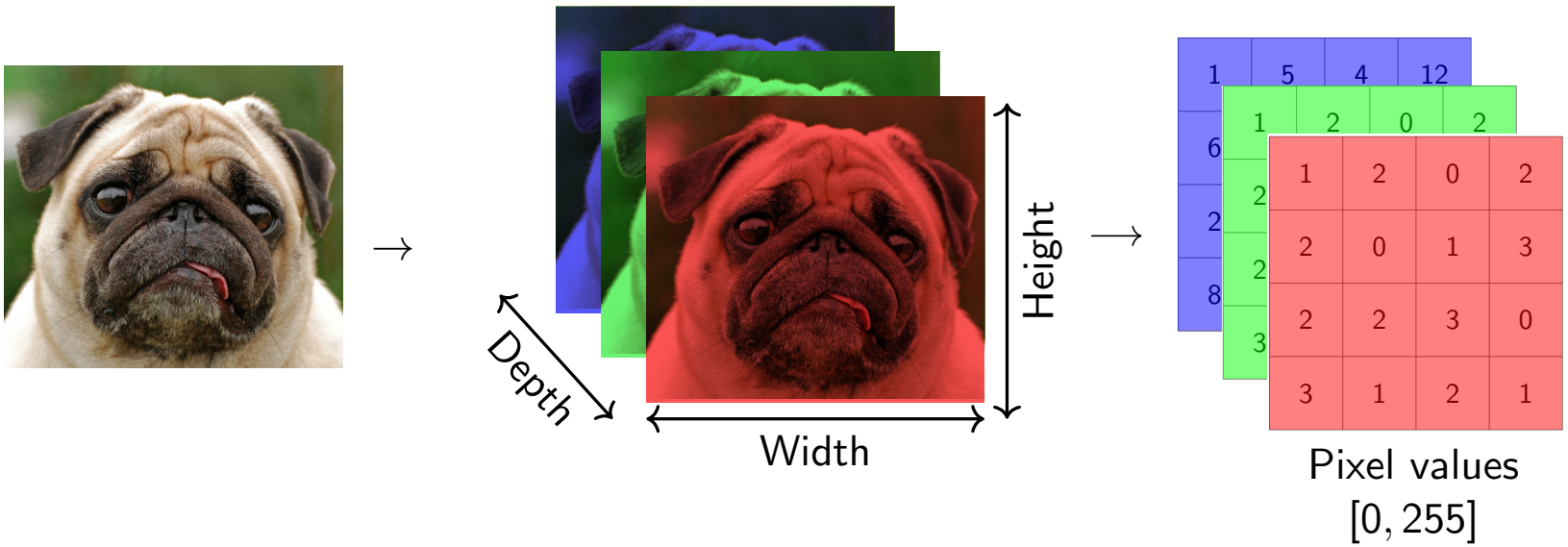
Deep Learning

Deep learning refers to a neural network which has a deep structure

i.e., a large number of **hidden layers**

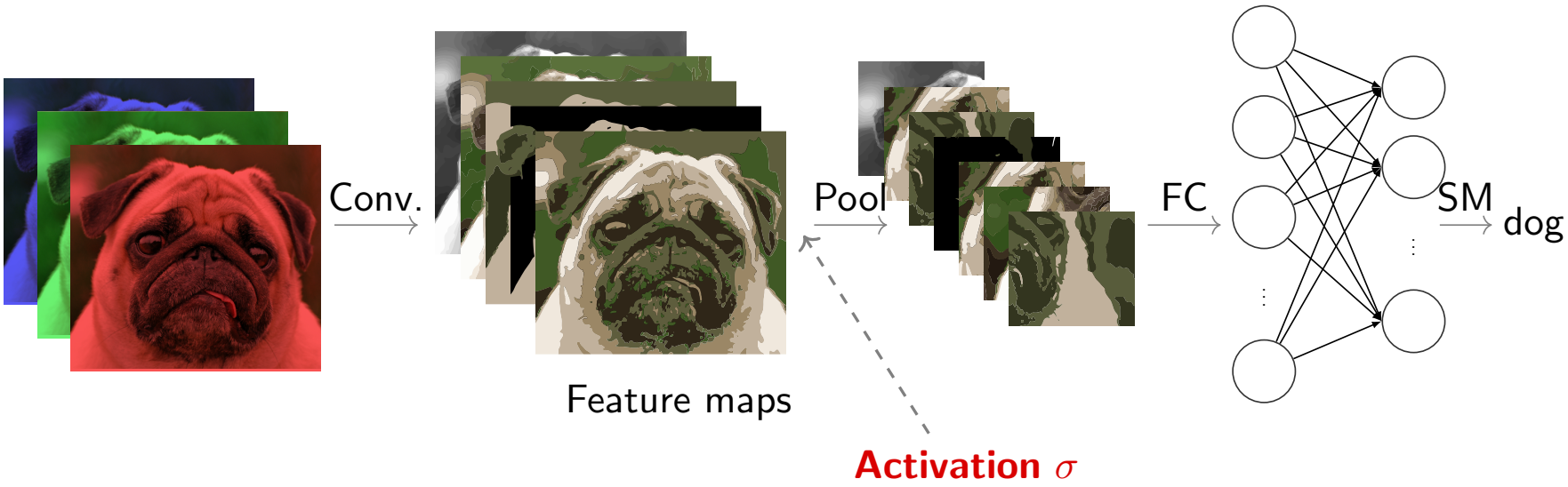


Convolutional Neural Networks (CNNs)



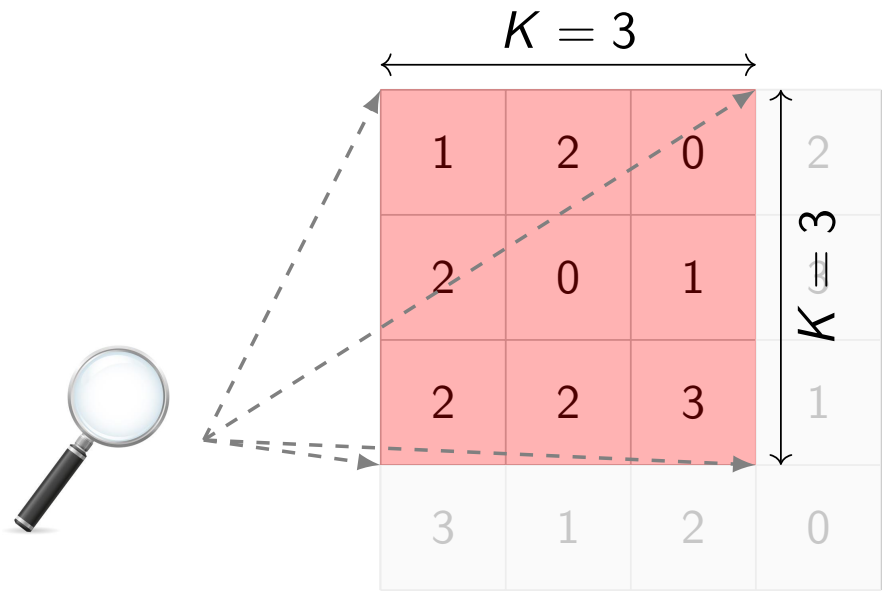
[L.H. Nguyen, petMD]

Convolutional Neural Networks (CNNs)

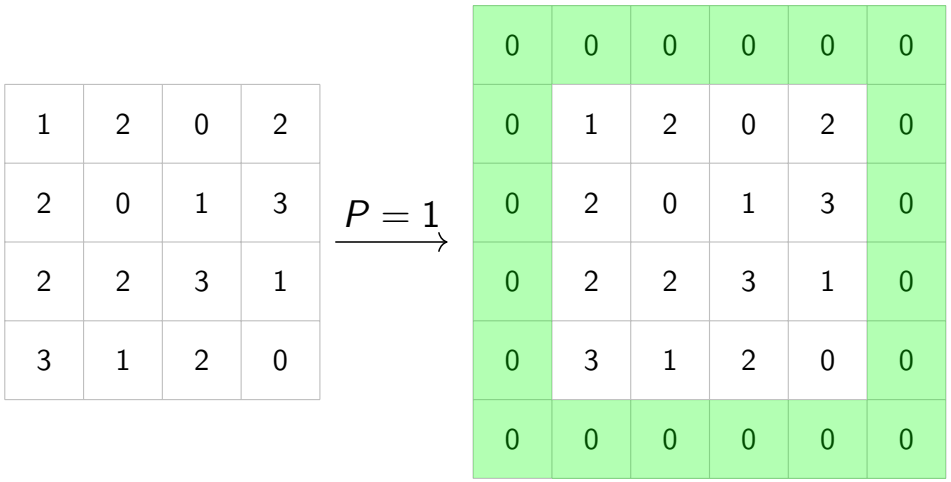


[petMD]

Kernel Size (K) & Stride (S)



Zero-padding (P)



$$W^{\mathrm{O}} = \frac{W^{\mathrm{I}} - K + 2P}{S} + 1$$

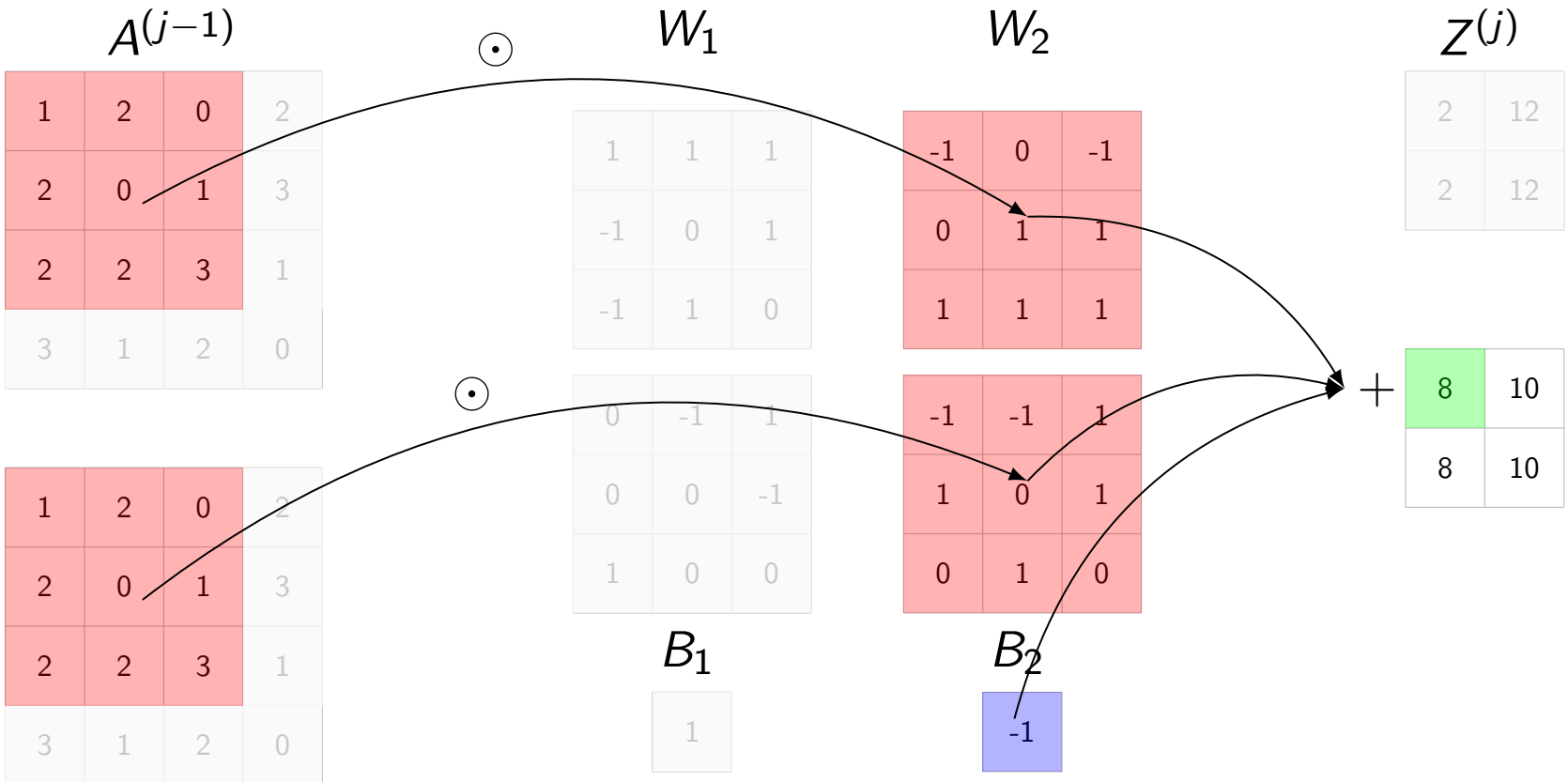
$$H^{\mathrm{O}} = \frac{H^{\mathrm{I}} - K + 2P}{S} + 1$$

Given $K = 3$, $S = 1$, $P = 1$

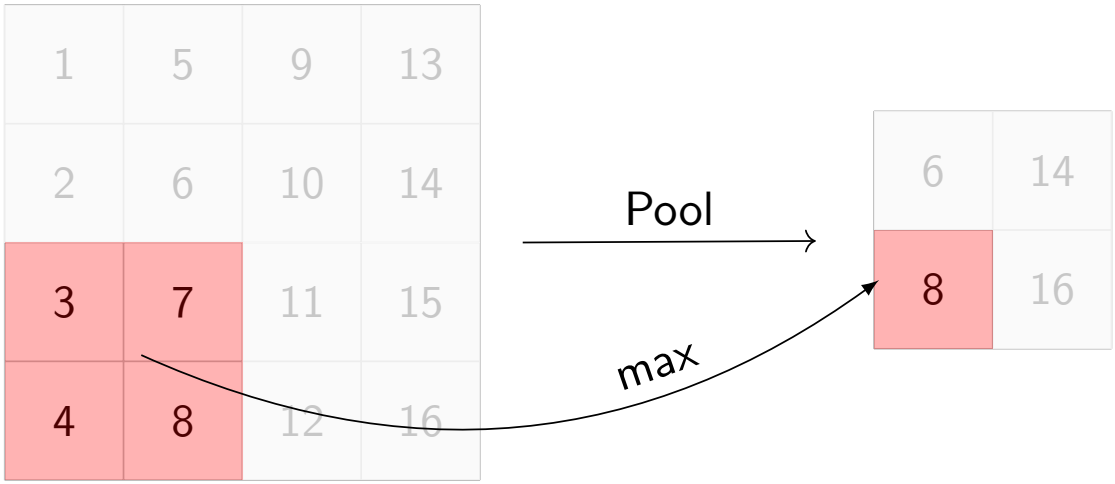
$$W^{\mathrm{O}} = \frac{4 - 3 + 2 \times 1}{1} + 1 = 4$$

$$H^{\mathrm{O}} = 4$$

Convolutional Layers – $\{K = 3, S = 1, P = 0\}$

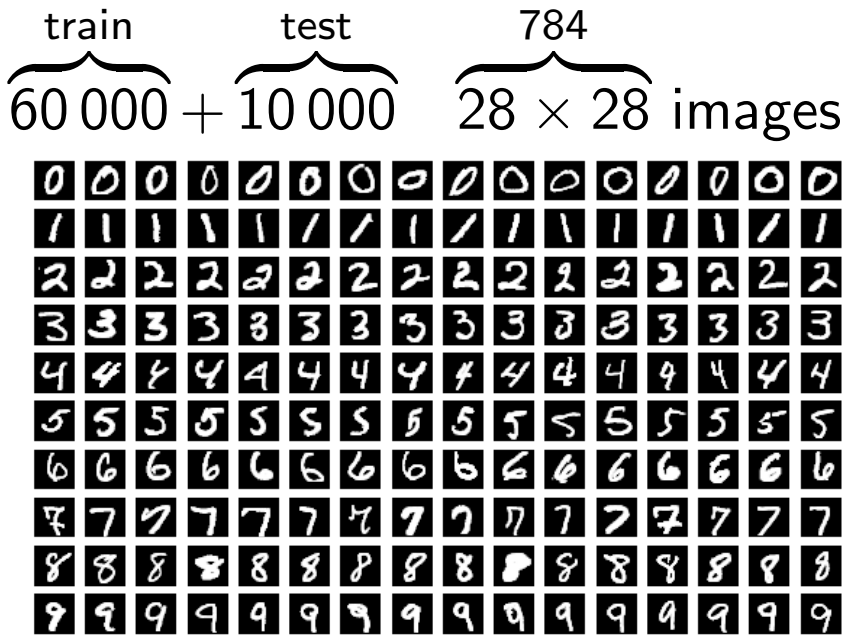


Pooling Layers – $\{K = 2, S = 2\}$



Max Pooling

MNIST



Network's Architecture

(Wan et al., 2013)

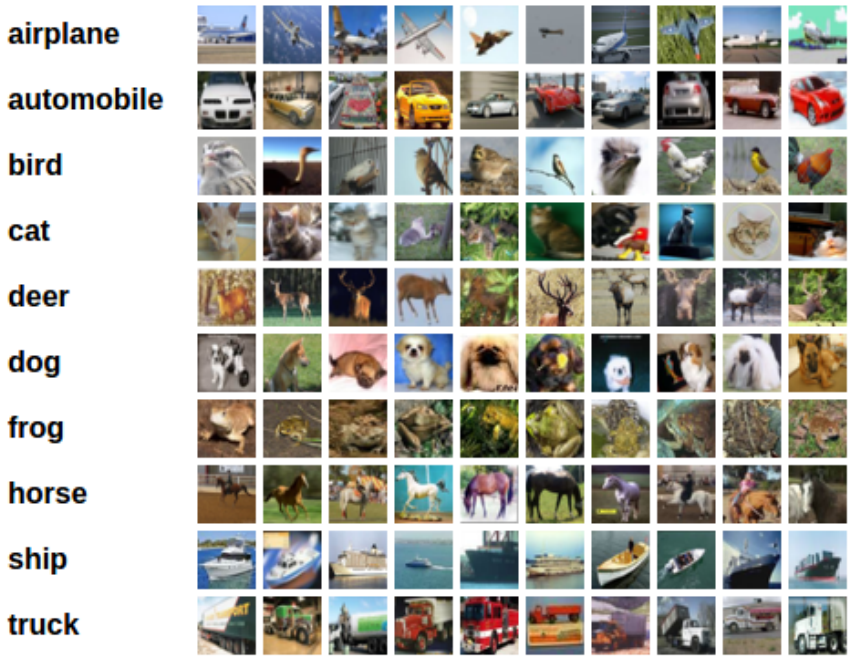
Layer	$D \times W \times H$	$K \times K$	P	S	σ
Input	$1 \times 28 \times 28$	-	-	-	-
Conv.	$32 \times 27 \times 27$	4×4	1	1	ReLU
Pooling	$32 \times 13 \times 13$	3×3	0	2	-
Conv.	$64 \times 9 \times 9$	5×5	0	1	ReLU
Pooling	$64 \times 4 \times 4$	3×3	0	2	-
FC	$150 \times 1 \times 1$	-	-	-	ReLU
Output	$10 \times 1 \times 1$	-	-	-	-

	Error Rate [%]	Hyperparameters	
	E $= e$	e	B
TAGI	**0.56**	13	1
Backprop	0.67	1000	128

[Nguyen and Goulet (2020)]

CIFAR10

$$\overbrace{50\,000}^{\text{train}} + \overbrace{10\,000}^{\text{test}} \quad \overbrace{3 \times 32 \times 32}^{3072} \text{ images}$$



Network's Architecture

(Wan et al., 2013)

Layer	$D \times W \times H$	$K \times K$	P	S	σ
Input	$3 \times 32 \times 32$	-	-	-	-
Conv.	$32 \times 32 \times 32$	5×5	2	1	ReLU
Pooling	$32 \times 16 \times 16$	3×3	1	2	-
Conv.	$32 \times 16 \times 16$	5×5	2	1	ReLU
Pooling	$32 \times 8 \times 8$	3×3	1	2	-
Conv.	$64 \times 8 \times 8$	5×5	2	1	ReLU
Pooling	$64 \times 4 \times 4$	3×3	1	2	-
FC	$64 \times 1 \times 1$	-	-	-	ReLU
Output	$10 \times 1 \times 1$	-	-	-	-

	Error Rate [%]	Hyperparameters	
	$\mathtt{E} = e$	e	B
TAGI	23.6	50	16
Backprop	23.5	150	128

[pytorch.com]

Summary - Neural Networks

Strength

- Neural networks is the **state-of-the-art** method
- Unequaled regression/classification accuracy

Limitations

- In order to outperform other methods (e.g. GPR/GPC) it requires many hidden units
  $\rightarrow$ large number of parameters to learn, $> 10^6$
- Learning a large number of parameters
  requires a large amount of data

Summary

Generative Classifier:

- Offers the most flexibility (e.g. indirect observations)
- Best suited for small datasets

Gaussian Process Classification:

- Quick & easy
- Allows interpolating and extrapolating (uncertainty estimates)
- Best suited for medium size datasets

Neural networks / Deep Learning:

- State-of-the-art method
- Best suited for large datasets

Temporary page!

LaTeX was unable to guess the total number of pages correctly. A
there was some unprocessed data that should have been added
the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page
will go away, because LaTeX now knows how many pages to expe
for this document.